Journal of Machine Learning Research 26 (2025) 1-57 Submitted 3/23; Revised 1/25; Published 6/25

# Randomly Projected Convex Clustering Model: Motivation, Realization, and Cluster Recovery Guarantees

**Ziwen Wang** ZWWANG@MATH.CUHK.EDU.HK
*Department of Mathematics*
*The Chinese University of Hong Kong*
*Hong Kong*

**Yancheng Yuan**[*] YANCHENG.YUAN@POLYU.EDU.HK
*Department of Applied Mathematics*
*The Hong Kong Polytechnic University*
*Hong Kong*

**Jiaming Ma** 22051002R@CONNECT.POLYU.HK
*Department of Applied Mathematics*
*The Hong Kong Polytechnic University*
*Hong Kong*

**Tieyong Zeng** ZENG@MATH.CUHK.EDU.HK
*Department of Mathematics*
*The Chinese University of Hong Kong*
*Hong Kong*

**Defeng Sun** DEFENG.SUN@POLYU.EDU.HK
*Department of Applied Mathematics*
*The Hong Kong Polytechnic University*
*Hong Kong*

**Editor:** Sanmi Koyejo

## Abstract

In this paper, we propose a randomly projected convex clustering model for clustering a collection of $n$ high dimensional data points in $\mathbb{R}^d$ with $K$ hidden clusters. Compared to the convex clustering model for clustering original data with dimension $d$, we prove that, under some mild conditions, the perfect recovery of the cluster membership assignments of the convex clustering model, if exists, can be preserved by the randomly projected convex clustering model with embedding dimension $m = O(\epsilon^{-2}\log(n))$, where $\epsilon > 0$ is some given parameter. We further prove that the embedding dimension can be improved to be $O(\epsilon^{-2}\log(K))$, which is independent of the number of data points. We also establish the recovery guarantees of our proposed model with uniform weights for clustering a mixture of spherical Gaussians. Extensive numerical results demonstrate the robustness and superior performance of the randomly projected convex clustering model. The numerical results will also demonstrate that the randomly projected convex clustering model can outperform other popular clustering models on the dimension-reduced data, including the randomly projected K-means model.

**Keywords:** convex clustering, Johnson-Lindenstrauss lemma, unsupervised learning.

[*] Corresponding author.

## 1. Introduction

Clustering is a fundamental and important problem in data science. Among many others, K-means is arguably the most popular algorithm. In practice, the K-means algorithm may suffer from the nonconvexity of the model and it is sensitive to the initialization. More critically, the K-means algorithm requires the number of clusters as a prior, which is not practical in many applications. Recently, researchers have proposed the convex clustering model, which aims to overcome the aforementioned challenges (Pelckmans et al., 2005; Hocking et al., 2011; Lindsten et al., 2011).

Given a collection of $n$ data points with $d$ features $A = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subseteq \mathbb{R}^d$, the general weighted convex clustering model (CCM) solves the following convex optimization problem

$$\min_{x_1,\ldots,x_n \in \mathbb{R}^d} \ \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}_i\|^2 + \gamma \sum_{i<j} w_{ij} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_q , \tag{CCM}$$

where $w_{ij} = w_{ji} \geq 0$ are given weights depending on the input data $A$, $\gamma > 0$ is a tuning parameter that controls the strength of the fusion penalty, and $\|\cdot\|_q$ is the vector $q$-norm ($q \geq 1$). In this paper, we focus on the convex clustering model with $q = 2$. We denote $\|\cdot\|$ as the vector 2-norm.

When applying the convex clustering model (CCM) for a given $\bar{\gamma}$, after obtaining the solution $\mathbf{x}_i^*(\bar{\gamma}),\ i \in [n]$ to the model (CCM), we will assign points $\mathbf{a}_i$ and $\mathbf{a}_j$ into the same cluster if $\mathbf{x}_i^*(\bar{\gamma}) = \mathbf{x}_j^*(\bar{\gamma})$. As one can realize, when $\gamma = 0$, we have $\mathbf{x}_i^*(0) = \mathbf{a}_i$, which implies we will have $n$ clusters if the input data points are distinct. As we increase the value of $\gamma$, some $\mathbf{x}_i^*(\gamma)$ will become identical due to the fusion penalty terms in the model. In practice, we will solve it for a sequence of values for the parameter $\gamma$, i.e., $0 \leq \gamma_1 < \gamma_2 < \cdots < \gamma_T < +\infty$, and obtain a clustering path of the data points. Importantly, Chi and Lange (2015) have proved that $\mathbf{x}^*(\gamma)$ is a continuous function of $\gamma$ for $\gamma \geq 0$. While the theoretical guarantees of the convex clustering model depend on the exact solution to (CCM) and an exact checking for $\mathbf{x}_i^*(\gamma) = \mathbf{x}_j^*(\gamma)$, only approximate solutions $\tilde{\mathbf{x}}_i(\gamma),\ i \in [n]$ can be obtained from iterative algorithms in general. Also, in practice, cluster assignments based on the inexact checking rule up to a given tolerance $\epsilon_{\text{clust}} > 0$ are widely adopted, i.e., points $\mathbf{a}_i$ and $\mathbf{a}_j$ are assigned into the same cluster if and only if $\|\tilde{\mathbf{x}}_i(\gamma) - \tilde{\mathbf{x}}_j(\gamma)\| < \epsilon_{\text{clust}}$. Interested readers can refer to (Jiang and Vavasis, 2021) for more detailed discussions.

A good choice of the weights $w_{ij}$ can enhance the performance of the model (CCM). A direct choice of the weights is setting $w_{ij} = 1$ for all $1 \leq i < j \leq n$, and the resulting model is usually called the convex clustering model with uniform weights. In practice, the following k-nearest neighbors-based weights are popular due to their robustness and computational efficiency:

$$w_{ij} = \begin{cases} \exp(-\phi \left\|\mathbf{a}_i - \mathbf{a}_j\right\|^2) & \text{if } (i,j) \in \mathcal{E}_A(k), \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where $\mathcal{E}_A(k) = \{(i,j) \mid \text{ if } \mathbf{a}_i \text{ (or } \mathbf{a}_j\text{) is in } \mathbf{a}_j\text{'s (or } \mathbf{a}_i\text{'s) k-nearest neighbors}, 1 \leq i \neq j \leq n\}$.

Extensive investigation has been conducted for the convex clustering model in recent years and impressive progress has been achieved from the perspectives of both the recovery properties and efficient numerical algorithms. From the theoretical understanding perspective, some deterministic and statistical cluster recovery guarantees have been established (Zhu et al., 2014; Tan and Witten, 2015; Panahi et al., 2017; Radchenko and Mukherjee, 2017; Chiquet et al., 2017; Chi and Steinerberger, 2019; Sun et al., 2021; Chi et al., 2020; Jiang et al., 2020; Dunlap and Mourrat, 2022). More specifically, under some mild conditions, there exists a nonempty interval of the tuning parameter $\gamma$ such that the convex clustering model can perfectly recover the cluster membership of the data (Panahi et al., 2017; Sun et al., 2021). From the perspective of optimization algorithms, impressive progress has been achieved in solving the convex clustering model with a large number of data points but with moderate feature dimensions (say with $d \leq 100$ in (CCM)). Along this direction, Chi and Lange (2015) adopted the alternating direction method of multipliers (ADMM) and proposed an alternating minimization algorithm (AMA). Later, Yuan et al. (2018) designed a semismooth Newton based augmented Lagrangian (SSNAL) method that can solve the convex clustering model efficiently with high accuracy. More recently, by taking advantage of the structured sparsity of the convex clustering model, Yuan et al. (2022) proposed dimension reduction techniques (in the sense of the number of data points) called adaptive sieving (AS) and enhanced adaptive sieving (EAS), which further accelerate SSNAL (and other algorithms). Consequently, the existing algorithms can be scalable with respect to the number of data points. However, it is still very challenging to solve the convex clustering model when the dimension of the data features is high (i.e., $d$ is large in (CCM)).

In this paper, we will design a dimension reduction technique for overcoming the computational challenges of the convex clustering model for clustering high dimensional data. Our approach is inspired by the Johnson-Lindenstrauss (JL) lemma (Johnson and Lindenstrauss, 1984) and the fact that the recovery guarantees of the convex clustering model mainly depend on the pairwise distances among the data points and centroids. In particular, we will propose a randomly projected (weighted) convex clustering model which clusters the data with a much smaller dimension obtained by applying a random projection mapping to the input data. Among other advantages, we want to mention that random projection is a computationally efficient approach to obtaining the embedded data. Importantly, we will prove that the randomly projected convex clustering model will preserve the recovery guarantees of the original convex clustering model. In other words, if there exists a nonempty interval of the parameter $\gamma$ such that the convex clustering model (CCM) perfectly recovers the cluster memberships of the input data, so will be the randomly projected model in high probability. This is a very interesting and inspiring result since we can obtain the clustering results of the original high dimensional data by solving a more tractable randomly projected convex clustering model with much smaller dimensions. Moreover, we will establish the cluster recovery guarantees for the randomly projected convex clustering model where the embedding dimension can be independent of the number of data points. Extensive numerical experiment results will be presented in this paper to justify the theoretical guarantees and to demonstrate the superior performance and robustness of the proposed model. To further demonstrate the superior performance of the randomly projected convex clustering model, we also compare its performance to  other popular clustering models on the embedding data.

We summarize the main contributions of this paper as follows:

1. We propose a randomly projected convex clustering model that is more computationally tractable than the convex clustering model (CCM).
2. We establish the recovery guarantees of the randomly projected convex clustering model under mild conditions. We further prove that the embedding dimension can be independent of the number of data points.
3. Additionally, we establish the recovery guarantees of the randomly projected convex clustering model with uniform weights for clustering a mixture of spherical Gaussians.
4. We conduct extensive numerical experiments to justify the established theoretical guarantees and demonstrate the superior performance of the proposed randomly projected convex clustering model.

The rest of the paper is organized as follows: In Section 2, we introduce some concepts and notation and then review some necessary preliminary results of the recovery guarantees of the convex clustering model. In Section 3, we will propose a randomly projected convex clustering model and prove its theoretical recovery guarantees. In Section 4, we will present recovery guarantees for clustering a mixture of spherical Gaussians using the randomly projected convex clustering model with uniform weights. We will then present the numerical results in Section 5. We will conclude the paper and include some discussion of future research directions in Section 6.

## 2. Convex Clustering Model

In this section, we introduce some preliminary results on the convex clustering model, which are necessary for the discussion of the rest of the paper. In this paper, we focus on the following problem setting.

**General problem setting**: Cluster a collection of $n$ given data points $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\} \subseteq \mathbb{R}^d$ with a hidden clustering partition $\mathcal{V} = \{V_1, V_2, \ldots, V_K\}$.

We define some notation in Table 1, which will be commonly used later in this paper. We summarize the definition of $w_i^{(\alpha)}$, $w^{(\alpha,\beta)}$, $\bar{w}^{(\alpha)}$, and $\mu_{ij}^{(\alpha)}$ in Table 1. The meanings of these quantities can be interpreted as follows:

1. $w_i^{(\alpha)}$ represents the coupling between point $\mathbf{a}_i$ and the $\alpha$-th cluster, and $w^{(\alpha,\beta)}$ represents the coupling between the $\alpha$-th cluster and the $\beta$-th cluster.
2. $\bar{w}^{(\alpha)}$ measures the total coupling between the $\alpha$-th cluster and all other $K-1$ clusters.
3. $\mu_{ij}^{(\alpha)}$ estimates the total difference in the couplings between two distinct points $\mathbf{a}_i$ and $\mathbf{a}_j$ in the $\alpha$-th cluster with all other $K-1$ clusters.

Following the settings in (Sun et al., 2021), we assume the following assumptions hold throughout this paper.

**Assumption 1** *In the general problem setting, the mean vector* $\mathbf{a}^{(0)}$ *and the centroids* $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(K)}$ *are all distinct.*

Table 1: Some commonly used notation. In this table, we assume by default that $1 \leq \alpha \neq \beta \leq K$.

| Notation | Definition |
|---|---|
| $I_\alpha$ | $\{i \mid \mathbf{a}_i \in V_\alpha\}$ |
| $n_\alpha$ | cardinality of $I_\alpha$ |
| $[m]$ for a given integer $m > 0$ | $[m] := \{1, 2, \ldots, m\}$ |
| $\mathbf{a}^{(\alpha)}$ | $\frac{1}{n_\alpha} \sum_{i \in I_\alpha} \mathbf{a}_i$ |
| $\mathbf{a}^{(0)}$ | $\frac{1}{n} \sum_{i=1}^{n} \mathbf{a}_i$ |
| $w^{(\alpha,\beta)}$ | $\sum_{i \in I_\alpha} \sum_{j \in I_\beta} w_{ij}$ |
| $\bar{w}^{(\alpha)}$ | $\frac{1}{n_\alpha} \sum_{1 \leq l \leq K, l \neq \alpha} w^{(\alpha,l)}$ |
| $w_i^{(\alpha)} \quad (i \in [n])$ | $\sum_{j \in I_\alpha} w_{ij}$ |
| $\mu_{ij}^{(\alpha)} \quad (i, j \in I_\alpha)$ | $\sum_{1 \leq l \leq K, l \neq \alpha} \left\lvert w_i^{(l)} - w_j^{(l)} \right\rvert$ |
| $C(n, k) \ (1 \leq k \leq n)$ | $\frac{n!}{k!(n-k)!}$ |

**Assumption 2** *The inequality $n \gg K$ holds, where $n$ is the number of data points and $K$ is the number of hidden clusters.*

**Assumption 3** *The specified weights $w_{ij}$ in the model (CCM) satisfy*

$$w_{ij} > 0 \quad \textit{and} \quad n_\alpha w_{ij} > \mu_{ij}^{(\alpha)}, \quad \forall i, j \in I_\alpha, 1 \leq \alpha \leq K. \tag{2}$$

The above assumptions are reasonable for the clustering problem. A quick comment is that Assumption 3 holds automatically for uniform weights. Now, we introduce some definitions which are necessary for the rest of the paper.

**Definition 1** *We say that a map $\psi : \mathbb{R}^d \to \mathbb{R}^{\bar{d}}$ perfectly recovers $\mathcal{V}$ on the data $A$ if $\psi(\mathbf{a}_i) = \psi(\mathbf{a}_j)$ is equivalent to $\mathbf{a}_i$ and $\mathbf{a}_j$ belonging to the same $V_\alpha$ for some $1 \leq \alpha \leq K$.*

**Definition 2** *We call a partition $\mathcal{W} = \{W_1, \ldots, W_L\}$ of $A$ a coarsening of $\mathcal{V}$ if there exists a partition $\{\alpha_1, \ldots, \alpha_L\}$ of $[K]$ such that $W_l = \bigcup_{i \in \alpha_l} V_i$ for all $1 \leq l \leq L$. We call $\mathcal{W}$ a non-trivial coarsening of $\mathcal{V}$ if $L > 1$.*

**Definition 3** *We define $\{\mathbf{x}_i^*(\gamma)\}_{i=1}^n$ as the optimal solution of the convex clustering model (CCM) at a given $\gamma \geq 0$, and define the map $\phi_\gamma(\mathbf{a}_i) = \mathbf{x}_i^*(\gamma)$ for all $i = 1, \ldots, n$.*

The following theorem (Sun et al., 2021) establishes the recovery guarantees of the convex clustering model.

**Theorem 1** (Sun et al., 2021, Theorem 5) *In the general problem setting, consider the model (CCM). Define*

$$\begin{gathered}\gamma_{\min} := \max_{1 \leq \alpha \leq K} \max_{i,j \in I_\alpha} \left\{ \frac{\|\mathbf{a}_i - \mathbf{a}_j\|}{n_\alpha w_{ij} - \mu_{ij}^{(\alpha)}} \right\}, \ \gamma_{\max} := \min_{1 \leq \alpha < \beta \leq K} \left\{ \frac{\|\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)}\|}{\bar{w}^{(\alpha)} + \bar{w}^{(\beta)}} \right\}, \\ \gamma_{\max 2} := \max_{1 \leq \alpha \leq K} \frac{\|\mathbf{a}^{(0)} - \mathbf{a}^{(\alpha)}\|}{\bar{w}^{(\alpha)}}, \ r := \frac{\gamma_{\max}}{\gamma_{\min}}, \ r_2 := \frac{\gamma_{\max 2}}{\gamma_{\min}}.\end{gathered} \tag{3}$$

*Under Assumption 1 and Assumption 3, we have*

*1. If $r > 1$ and $\gamma \in [\gamma_{\min}, \gamma_{\max})$, then the map $\phi_\gamma$ perfectly recovers $\mathcal{V}$.*

*2. If $r_2 > 1$ and $\gamma \in [\gamma_{\min}, \gamma_{\max 2})$, then the map $\phi_\gamma$ recovers a non-trivial coarsening of $\mathcal{V}$.*

Here, we include some remarks for Theorem 1.

1. On the one hand, the lower bound $\gamma_{\min}$ characterizes the maximum weighted distance between the data points in the same cluster. On the other hand, the upper bound $\gamma_{\max}$ characterizes the minimum weighted distance between different centroids. Thus, we can expect perfect recovery to be practically possible for the weighted convex clustering model if the upper bound is larger than the lower bound. Moreover, it indicates that the given data is difficult to cluster if $\gamma_{\min} > \gamma_{\max}$.

2. The theoretical values of $\gamma_{\min}$, $\gamma_{\max}$, and $\gamma_{\max 2}$ depend on the underlying partition of the given data, which cannot be calculated in advance. However, we want to emphasize that: 1) these values are well-defined for theoretical analysis; 2) we can generate a clustering path in practice by taking a sequence of $\gamma$ although we do not know the perfect recovery interval. It is a challenging open question to estimate the values of $\gamma_{\min}$, $\gamma_{\max}$, and $\gamma_{\max 2}$ empirically, which is regarded as a future research question of this paper.

## 3. A Randomly Projected Convex Clustering Model

The model (CCM) has promising recovery guarantees. However, solving the model can be computationally challenging, especially when the feature dimension $d$ is high. In this section, we will propose a randomly projected convex clustering model with much smaller feature dimensions. Moreover, we will prove that the recovery guarantees will be preserved with a high probability for the random projected convex clustering model. More specifically, for the given collection of data points $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\} \subseteq \mathbb{R}^d$ considered in the general problem setting, we solve the following projected convex clustering model

$$\min_{\hat{X} \in \mathbb{R}^{m \times n}} \; \frac{1}{2} \sum_{i=1}^{n} \|\hat{\mathbf{x}}_i - \Pi \mathbf{a}_i\|^2 + \gamma \sum_{i<j} w_{ij} \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|, \tag{RPCCM}$$

where $\Pi \in \mathbb{R}^{m \times d}$ is a given random matrix and $m < d$ is the embedding dimension. In this paper, we will choose $\Pi$ according to Johnson-Lindenstraus (DJL) lemma to preserve the distances between the concerned pairs of points. We call the corresponding model (RPCCM) a randomly projected convex clustering model. Our model (RPCCM) is essentially the model (CCM) applied to the embedded data $\{\Pi \mathbf{a}_i\}_{i=1}^n$. This allows us to utilize existing algorithms developed for the model (CCM), thereby enhancing the practical usefulness of the model (RPCCM).

### 3.1 Johnson-Lindenstrauss Lemma and the Random Projection

In this section, we introduce the Johnson-Lindenstrauss (JL) lemma (Johnson and Lindenstrauss, 1984), which is a key tool for this paper. Consider a collection of high-dimensional data points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{R}^d$, the JL lemma shows the existence of a mapping $f : \mathbb{R}^d \to \mathbb{R}^m$ such that for all points $\mathbf{x}_i \neq \mathbf{x}_j \in X$, $\|\mathbf{x}_i - \mathbf{x}_j\|$ are approximately maintained in a $m$ dimensional space within a distortion tolerance $\epsilon \in (0, 1)$. More surprisingly, the required embedding dimension $m = O(\epsilon^{-2} \log(n))$ is independent of $d$.

We define some notations in Table 2 for further convenience.

Table 2: Some additional notation for random matrices.

| Notation | Definition |
|---|---|
| $\epsilon$ | accuracy parameter in JL lemma, $\epsilon \in (0, 1)$ |
| $\delta$ | confidence parameter in JL lemma, $\delta \in (0, \frac{1}{2})$ |
| $\mathcal{D}_{\epsilon,\delta}$ | a DJL distribution over $\mathbb{R}^{m\times d}$ ($m$ will be specified) |
| $C$ | the absolute constant in the DJL lemma |
| $\Pi \sim \mathcal{D}_{\epsilon,\delta}$ | $\Pi \in \mathbb{R}^{m\times d}$ is randomly drawn from $\mathcal{D}_{\epsilon,\delta}$ |
| $R_{ij}$ | independent random variables with $\mathbb{E}[R_{ij}] = 0$,$\mathrm{Var}[R_{ij}] = 1$ and the same sub-Gaussian norm $\kappa$ |
| $R$ | $R \in \mathbb{R}^{m\times d}$ is a sub-Gaussian matrix with entries $R_{ij}$ $(m \leq d)$ |
| $\Pi = \frac{1}{\sqrt{m}} R$ | $\Pi \in \mathbb{R}^{m\times d}$ is randomly drawn from the distribution $\frac{1}{\sqrt{m}} R$ $(m \leq d)$ |
| $s_k(\Pi)$ | $k$-th largest singular value of $\Pi$ $(k \in [m])$ |
| $C_\kappa^2$ | $C_\kappa^2 > 0$ is a constant only depends on $\kappa$ |
| $\bar{S}(m, d, t)$ | $\frac{\sqrt{d}+C_\kappa^2 t}{\sqrt{m}} + C_\kappa^2$ $(t > 0)$ |
| $\underline{S}(m, d, t)$ | $\frac{\sqrt{d}-C_\kappa^2 t}{\sqrt{m}} - C_\kappa^2$ $(t > 0)$ |

**Lemma 1 (JL lemma)** (Johnson and Lindenstrauss, 1984, Lemma 1)) *For any given collection of $n$ data points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{R}^d$ and any $\epsilon \in (0, 1)$, there exists an $\epsilon$- isometry embedding $f : \mathbb{R}^d \to \mathbb{R}^m$ with $m = O\left(\min\{d, \epsilon^{-2} \log(n)\}\right)$. This implies $\forall$ $\mathbf{x}_i, \mathbf{x}_j \in X$,*

$$(1 - \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\| \leq \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\| \leq (1 + \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|. \tag{4}$$

We call a mapping $f$ satisfies (4) an $\epsilon$-JL Transform (or $\epsilon$-JLT in short) on $X$. The mapping $f$ can be found in randomized polynomial time (Dasgupta and Gupta, 2003). Moreover, if the mapping $f$ must be linear, then $m = \Omega\left(\min\{d, \epsilon^{-2} \log(n)\}\right)$ is optimal (Larsen and Nelson, 2016). The following Distributional Johnson-Lindenstrauss (DJL) lemma is useful.

**Lemma 2 (DJL lemma)** *For any $\epsilon \in (0, 1), \delta \in (0, 1/2)$ and an integer $d > 1$, there exists a distribution $\mathcal{D}_{\epsilon,\delta}$ over matrices $\Pi \in \mathbb{R}^{m\times d}$ for $m \geq C\epsilon^{-2} \log(1/\delta)$, where $C > 0$ is an absolute constant, such that for any $z \in \mathbb{R}^d$ with $\|z\| = 1$,*

$$\mathbb{P}_{\Pi\sim\mathcal{D}_{\epsilon,\delta}} [||\|\Pi z\| - 1| > \epsilon] < 2\delta. \tag{5}$$

*To be more specific, the distribution $\mathcal{D}_{\epsilon,\delta}$ with $m \geq C\epsilon^{-2} \log(1/\delta)$ satisfies:*
*1. $\mathbb{P}_{\Pi\sim\mathcal{D}_{\epsilon,\delta}} [\|\Pi z\| > 1 + \epsilon] < \delta$, for any $\epsilon > 0$.*
*2. $\mathbb{P}_{\Pi\sim\mathcal{D}_{\epsilon,\delta}} [\|\Pi z\| < 1 - \epsilon] < \delta$, for any $\epsilon \in (0, 1)$.*

We call a distribution $\mathcal{D}_{\epsilon,\delta}$ that satisfies (5) a DJL distribution, and denote $\Pi \sim \mathcal{D}_{\epsilon,\delta}$ if $\Pi$ is randomly sampled from a DJL distribution $\mathcal{D}_{\epsilon,\delta}$.

**Remark 1** *The DJL lemma can be generalized to a set $X \subseteq \mathbb{R}^d$ of $n$ data points by taking the union bound over $x \in X$. In particular, if we take $\delta = \frac{1}{n^p}$, where $p > 2$, then the probability for ensuring an $\epsilon$-isometry property of $\Pi \sim \mathcal{D}_{\epsilon,\delta}$ on $X$ is over $1 - 2C(n,2)\frac{1}{n^p} > 1 - \frac{1}{n^{p-2}}$.*

**Remark 2** *The absolute constant $C$ in the DJL lemma is independent of $d, n, \epsilon, \delta$. An empirical study (Venkatasubramanian and Wang, 2011) suggests that $m \geq \lceil \epsilon^{-2} \log(n_p) \rceil$ would be sufficient for $n_p$ data points. As a result, it suffices to set $m \geq \lceil 2\epsilon^{-2} \log(n) \rceil$ to ensure an $\epsilon$-isometry mapping on the given input data $A$ with high probability. We will follow this setting in experiments.*

The choices of $\mathcal{D}_{\epsilon,\delta}$ of the DJL lemma have been extensively explored, including the sub-Gaussians (Indyk and Motwani, 1998; Achlioptas, 2003; Matoušek, 2008), the Fast JL Transform (Ailon and Chazelle, 2009; Ailon and Liberty, 2009, 2013), and the Sparse JL Transform (Dasgupta et al., 2010; Kane and Nelson, 2010, 2014; Cohen et al., 2018). For simplicity, in this paper, we will follow (Matoušek, 2008, Theorem 3.1) and take the scaled sub-Gaussian matrix $\Pi = \frac{1}{\sqrt{m}} R \in \mathbb{R}^{m \times d}$ as the linear random projection, where $R_{ij}$ are independent random variables with zero mean and a uniform sub-Gaussian tail. Next, we include a useful lemma to estimate the singular values of the sub-Gaussian matrices, which is a direct consequence of Theorem 4.6.1 and Lemma 3.4.2 in (Vershynin, 2018).

**Lemma 3 (Two-sided bound on sub-Gaussian matrices)** *Let $\Pi = \frac{1}{\sqrt{m}} R \in \mathbb{R}^{m \times d}$ ($m \leq d$), where $R_{ij}$ are independent random variables with $\mathbb{E}[R_{ij}] = 0$,$\mathrm{Var}[R_{ij}] = 1$ and the sub-Gaussian norm $\kappa := \|R_{ij}\|_{\psi_2} := \inf\{s > 0 \mid \mathbb{E}[\exp(R_{ij}^2/s^2)] \leq 2\}$. Let $s_l(\Pi)$ be the $l$-th largest singular value of $\Pi$ for $l \in [m]$. For any $t \geq 0$, with probability over $1 - 2\exp(-t^2)$,*

$$\left| s_l(\Pi) - \sqrt{d/m} \right| \leq C_\kappa^2 (1 + \frac{t}{\sqrt{m}}),\ l \in [m], \tag{6}$$

*where $C_\kappa^2 > 0$ is an absolute constant that only depends on $\kappa$ and is independent of $d, n, m$. In other words, define*

$$\bar{S}(m,d,t) = \frac{\sqrt{d} + C_\kappa^2 t}{\sqrt{m}} + C_\kappa^2,\ \underline{S}(m,d,t) = \frac{\sqrt{d} - C_\kappa^2 t}{\sqrt{m}} - C_\kappa^2, \tag{7}$$

*then with probability at least $1 - 2\exp(-t^2)$,*

$$\underline{S}(m,d,t) \leq s_l(\Pi) \leq \bar{S}(m,d,t),\ l \in [m]. \tag{8}$$

If we further let $\Pi = \frac{1}{\sqrt{m}} G \in \mathbb{R}^{m \times d}$ be a scaled Gaussian matrix, i.e., $G_{ij}$ are i.i.d. standard normal variables, the following lemma provides more explicit bounds on the singular values of $\Pi$, which directly follows Theorem II.13 in (Davidson and Szarek, 2001) and Theorem 2.6 in (Rudelson and Vershynin, 2010).

**Lemma 4 (Two-sided bound on gaussian matrices)** *Let $\Pi = \frac{1}{\sqrt{m}} G \in \mathbb{R}^{m \times d}$ ($m \leq d$), where $G_{ij}$ are independent standard normal variables, then the two-side bounds $\bar{S}(m,d,t)$ and $\underline{S}(m,d,t)$ defined in (7) could be*

$$\bar{S}(m,d,t) = \frac{\sqrt{d} + t}{\sqrt{m}} + 1, \quad \underline{S}(m,d,t) = \frac{\sqrt{d} - t}{\sqrt{m}} - 1. \tag{9}$$

### 3.2 Cluster Recovery Guarantees of the Model (RPCCM) for the General Problem Setting

Next, we will establish the cluster recovery guarantees of the model (RPCCM) for the general problem setting. For later convenience, we introduce some necessary notation for the rest of the paper.

**Definition 4** *We define $\{\hat{\mathbf{x}}_i^*(\gamma)\}_{i=1}^n$ as the optimal solution of the model (RPCCM) with a randomly sampled projection matrix $\Pi \in \mathbb{R}^{m\times d}$ at a given $\gamma \geq 0$, and define the map $\hat{\phi}_\gamma(\mathbf{a}_i) = \hat{\mathbf{x}}_i^*(\gamma)$ for all $i = 1, ..., n$.*

**Definition 5** *In the general problem setting, consider the model (RPCCM) with some specified weights $w_{ij} = w_{ji} \geq 0$ $(1 \leq i \neq j \leq n)$ and a randomly sampled projection matrix $\Pi \in \mathbb{R}^{m\times d}$ (for some $m \geq 1$). Without explicitly mentioning the dependence on $\Pi$, we define*

$$\hat{\gamma}_{\min} := \max_{1\leq\alpha\leq K}\max_{i,j\in I_\alpha}\left\{\frac{\|\Pi(\mathbf{a}_i-\mathbf{a}_j)\|}{n_\alpha w_{ij}-\mu_{ij}^{(\alpha)}}\right\}, \quad \hat{\gamma}_{\max} := \min_{1\leq\alpha<\beta\leq K}\left\{\frac{\left\|\Pi(\mathbf{a}^{(\alpha)}-\mathbf{a}^{(\beta)})\right\|}{\bar{w}^{(\alpha)}+\bar{w}^{(\beta)}}\right\}. \tag{10}$$

It follows Theorem 1 that if $\hat{\gamma}_{\min} < \hat{\gamma}_{\max}$ for a specific $\Pi$, the model (RPCCM) performs the perfect cluster recovery in the interval of $\gamma \in [\hat{\gamma}_{\min}, \hat{\gamma}_{\max})$. This raises a natural question: Provided $\gamma_{\min} < \gamma_{\max}$, can we ensure that the condition $\hat{\gamma}_{\min} < \hat{\gamma}_{\max}$ is satisfied with high probability? A key observation is that the recovery guarantees of the convex clustering model (e.g., Theorem 1) mainly depend on the distances between data points within the same cluster (see $\gamma_{\min}$) and the distances between the centroids of different clusters (see $\gamma_{\max}$). Therefore, if these distances can be properly preserved with some sufficiently small distortions, the recovery guarantees can be inherited by the model (RPCCM).

Let $X_A = \{\mathbf{a}_i - \mathbf{a}_j \mid 1 \leq i < j \leq n\}$, $X_{\mathcal{V}(A)} = \cup_{\alpha=1}^K\{\mathbf{a}_i - \mathbf{a}_j \mid i,j \in I_\alpha, i \neq j\}$, and $X_{C(A)} = \left\{\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)} \mid 1 \leq \alpha < \beta \leq K\right\}$. The following proposition shows that a random projection by $\Pi \in \mathbb{R}^{m\times d}$ with $m = O(\epsilon^{-2}\log(n))$ can preserve all the norms in $X_{\mathcal{V}(A)}$ up to a multiplicative $(1+\epsilon)$ factor, while simultaneously preserving all the norms in $X_{C(A)}$ down to a multiplicative $(1-\epsilon_2)$ factor, where $\epsilon_2 < \epsilon$ will be specified in the proposition.

**Proposition 1** *In the general problem setting, let $p_1, p_2 > 2$, and assume that $C_{12} := \sqrt{\frac{\log(K)p_2}{\log(n)p_1}} < 1$. For any $\epsilon \in (0, C_{12}^{-1})$, let $\epsilon_2 = C_{12}\epsilon$. Let $\delta = \frac{1}{n^{p_1}}$ and $\Pi \sim D_{\epsilon,\delta} : \mathbb{R}^d \to \mathbb{R}^m$ with $m \geq p_1 C\epsilon^{-2}\log(n)$, then with probability over $1 - \frac{1}{2n^{p_1-2}} - \frac{1}{2K^{p_2-2}}$,*

$$\|\Pi(\mathbf{a}_i - \mathbf{a}_j)\| \;\leq (1+\epsilon)\|\mathbf{a}_i - \mathbf{a}_j\|,\ i,j \in I_\alpha, 1 \leq \alpha \leq K, \tag{11a}$$

$$\|\Pi(\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)})\| \geq (1-\epsilon_2)\|\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)}\|,\ 1 \leq \alpha \neq \beta \leq K. \tag{11b}$$

**Proof** On the one hand, for any $\epsilon \in (0, C_{12}^{-1})$ and $\delta > 0$, let $\Pi \sim \mathcal{D}_{\epsilon,\delta}$ with $m \geq C\epsilon^{-2}\log(1/\delta)$, the failure probability of (11a) is at most $|X_{\mathcal{V}(A)}|\delta$. Note that $|X_{\mathcal{V}(A)}| = \sum_{\alpha=1}^K C(n_\alpha, 2) < C(n, 2)$. If we take $\delta = \frac{1}{n^{p_1}}$ and $m \geq C\epsilon^{-2}\log(1/\delta) = p_1 C\epsilon^{-2}\log(n)$, where $p_1 > 2$, the failure probability of (11a) is at most

$$|X_{\mathcal{V}(A)}|\delta < C(n,2)\frac{1}{n^{p_1}} < \frac{1}{2n^{p_1-2}}.$$

On the other hand, let $\delta_2 = \frac{1}{K^{p_2}}$ and $\epsilon_2 = C_{12}\epsilon$, then $0 < \epsilon_2 < \min\{\epsilon, 1\}$, and we have $p_1 C\epsilon^{-2}\log(n) = p_2 C\epsilon_2^{-2}\log(K) = C\epsilon_2^{-2}\log(1/\delta_2)$. As a result, if we take $m \geq p_1 C\epsilon^{-2}\log(n)$, the probability that (11b) fails is at most

$$|X_{C(A)}|\frac{1}{K^{p_2}} = C(K,2)\frac{1}{K^{p_2}} < \frac{1}{2K^{p_2-2}}.$$

Taking a union bound, the probability that conditions (11) are satisfied is at least

$$1 - |X_{\mathcal{V}(A)}|\delta - |X_{C(A)}|\delta_2 > 1 - \frac{1}{2n^{p_1-2}} - \frac{1}{2K^{p_2-2}}.$$

■

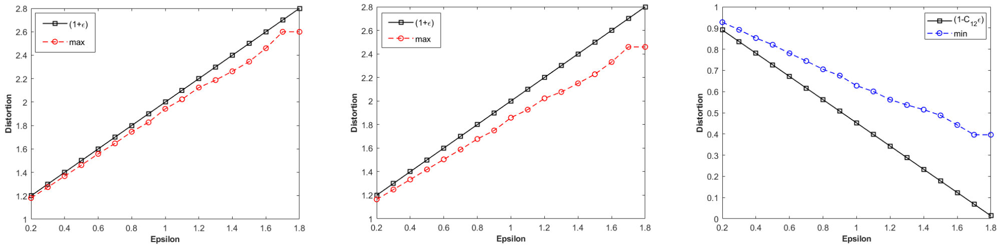
The intuition behind Proposition 1 is, for a given $m$, since $|X_{\mathcal{V}(A)}| \gg |X_{C(A)}|$, the distortion $\epsilon_2$ in (11b) could be much smaller compared to the distortion $\epsilon$ in (11a). We are motivated by the DJL lemma to estimate $\epsilon_2$ with $\epsilon_2 = C_{12}\epsilon$. Importantly, we restrict $\epsilon \in (0, C_{12}^{-1})$ to ensure $\epsilon_2 \in (0,1)$, which guarantees that the projected centroids $\left\{\Pi\mathbf{a}^{(1)}, \ldots, \Pi\mathbf{a}^{(K)}\right\}$ are distinct with high probability.

**Remark 3** *Indeed, under the same conditions in Proposition 1, the following stronger result holds:*

$$\|\Pi(\mathbf{a}_i - \mathbf{a}_j)\| \leq (1+\epsilon)\|\mathbf{a}_i - \mathbf{a}_j\|,\ 1 \leq i < j \leq n. \tag{12}$$

*This stronger result improves on (11a), allowing for a practical assessment of the sampled random projection matrix* $\Pi$*. More discussion will be found in Proposition 2.*

Before we present the theoretical recovery guarantees of (RPCCM), we numerically verify the theoretical bounds obtained in Proposition 1 using a simulated data $A_0 \in \mathbb{R}^{900\times 1024}$ with $K = 8$ clusters. On this sampled data $A_0$, we have that $C_{12} = \sqrt{\log(8)/\log(1024)} = \sqrt{3/10}$. We conduct numerical tests with $\epsilon \in [0.2 : 0.1 : 1.8] \subseteq (0, \frac{1}{C_{12}})$ and set $m = \lceil 2\epsilon^{-2}\log(n)\rceil$. For each pair of $(\epsilon, m)$, we randomly sampled 1000 random projection matrices $\Pi$ for verification. We calculate the average maximum distortion ranges in $X_{A_0}$ and $X_{\mathcal{V}(A_0)}$, as well as the average minimum distortion range in $X_{C(A_0)}$, respectively. The bounds are clearly illustrated by the numerical results presented in Figure 1.



(a) Distortion in $X_{A_0}$ (b) Distortion in $X_{\mathcal{V}(A_0)}$ (c) Distortion in $X_{C(A_0)}$

Figure 1: Verification on a simulated data $A_0$ with $d = 900$, $n = 1024$, $K = 8$.

The next theorem shows the theoretical recovery guarantees of the model (RPCCM).

**Theorem 2** *Consider the general problem setting and the models (CCM) and (RPCCM) with the same specified weights* $w_{ij} = w_{ji} \geq 0$*. Let* $p_1, p_2 > 2$*, and we assume that* $C_{12} := \sqrt{\frac{\log(K)p_2}{\log(n)p_1}} < 1$ *and* $\sqrt{\frac{p_2 C\log(K)}{d}} < 1$*, and thus* $\sqrt{\frac{p_1 C\log(n)}{d}} < \frac{1}{C_{12}}$*. Define*

$$\epsilon_{\min} = \sqrt{\frac{p_1 C\log(n)}{d}},\ \epsilon_{\sup} = \frac{r-1}{C_{12}r+1}, \tag{13}$$

*where* $r$ *is as defined in (3). If* $r > \frac{1+\epsilon_{\min}}{1-C_{12}\epsilon_{\min}}$*, then* $\epsilon_{\min} < \epsilon_{\sup}$*. For any* $(\epsilon, \gamma)$ *such that*

$$\epsilon \in (\epsilon_{\min}, \epsilon_{\sup}),\ \ \gamma \in [(1+\epsilon)\gamma_{\min}, (1-C_{12}\epsilon)\gamma_{\max})\,, \tag{14}$$

*let* $\delta = \frac{1}{n^{p_1}}$ *and* $\Pi \sim D_{\epsilon,\delta}$ *with* $m \in [p_1 C\epsilon^{-2}\log(n), d)$*, then with probability over* $1 - \frac{1}{2n^{p_1-2}} - \frac{1}{2K^{p_2-2}}$*, the map* $\hat{\phi}_\gamma$ *perfectly recovers* $\mathcal{V}$*.*

**Proof** It directly follows Proposition 1 that, with probability over $1 - \frac{1}{2n^{p_1-2}} - \frac{1}{2K^{p_2-2}}$, the following statements hold:

(i) The centroids $\left\{\Pi\mathbf{a}^{(1)}, \ldots, \Pi\mathbf{a}^{(K)}\right\}$ of the embedded data are distinct.

(ii) The parameters $\hat{\gamma}_{\min}$ and $\hat{\gamma}_{\max}$ defined in (10) satisfy the following inequalities:

$$\hat{\gamma}_{\min} \leq (1+\epsilon)\gamma_{\min}, \quad (1-C_{12}\epsilon)\gamma_{\max} \leq \hat{\gamma}_{\max}.$$

The above implies that

$$[(1+\epsilon)\gamma_{\min}, (1-C_{12}\epsilon)\gamma_{\max}) \subseteq [\hat{\gamma}_{\min}, \hat{\gamma}_{\max})\,. \tag{15}$$

Now, we prove the theorem. We claim here that it is sufficient to show: if $r > \frac{1+\epsilon_{\min}}{1-C_{12}\epsilon_{\min}}$, then $\epsilon_{\min} < \epsilon_{\sup}$, and for any $\epsilon \in (\epsilon_{\min}, \epsilon_{\sup})$, the interval $[(1+\epsilon)\gamma_{\min}, (1-C_{12}\epsilon)\gamma_{\max})$ is nonempty. In fact, if $[(1+\epsilon)\gamma_{\min}, (1-C_{12}\epsilon)\gamma_{\max})$ is nonempty, then by the first inclusion of (15), $[\hat{\gamma}_{\min}, \hat{\gamma}_{\max})$ is nonempty. Applying Theorem 1 to the embedded data $\Pi A$ implies that for any $\gamma \in [\hat{\gamma}_{\min}, \hat{\gamma}_{\max})$, the map $\hat{\phi}_{\hat{\gamma}}$ perfectly recovers $\mathcal{V}$.

On the one hand, we have

$$\begin{aligned} r > \tfrac{1+\epsilon_{\min}}{1-C_{12}\epsilon_{\min}} \quad &\Longrightarrow \quad (1-C_{12}\epsilon_{\min})r > 1+\epsilon_{\min} \\ &\Longrightarrow \quad \epsilon_{\min} < \tfrac{r-1}{C_{12}r+1} = \epsilon_{\sup}. \end{aligned}$$

This implies that the interval $(\epsilon_{\min}, \epsilon_{\sup})$ is nonempty.

On the other hand, we have

$$\begin{aligned} \epsilon < \epsilon_{\sup} \quad &\Longrightarrow \quad \epsilon < \tfrac{r-1}{C_{12}r+1} \\ &\Longrightarrow \quad \tfrac{1+\epsilon}{1-C_{12}\epsilon} < r \\ &\Longrightarrow \quad \tfrac{1+\epsilon}{1-C_{12}\epsilon} < \tfrac{\gamma_{\max}}{\gamma_{\min}} \\ &\Longrightarrow \quad (1+\epsilon)\gamma_{\min} < (1-C_{12}\epsilon)\gamma_{\max}. \end{aligned}$$

Thus, we have proved the theorem. ■

In practice, only pairwise distances between all input data points are checkable after a random projection matrix $\Pi$ is sampled (i.e., the condition (12), which covers the condition (11a)). The condition (11b) is uncheckable due to the lack of the true cluster membership of data $A$. The next corollary shows that the condition (11b) can be satisfied in high probability provided (12) holds.

**Proposition 2** *In the general problem setting, let $p_1, p_2 > 2$, and assume that $C_{12} := \sqrt{\frac{\log(K)p_2}{\log(n)p_1}} < 1$. For any $\epsilon \in (0, C_{12}^{-1})$, let $\epsilon_2 = C_{12}\epsilon$, then $\epsilon_2 < \min\{1, \epsilon\}$. Let $\delta = \frac{1}{n^{p_1}}$ and $\Pi \sim D_{\epsilon,\delta}$ with $m \geq p_1 C \epsilon^{-2} \log(n)$. Let $E_1$ be the event that $\Pi$ satisfies (12) and $E_2$ be the event that $\Pi$ satisfies (11b), respectively. Then, the conditional probability $\mathbb{P}[E_2 \mid E_1]$ satisfies*

$$\mathbb{P}[E_2 \mid E_1] > \left(1 - \frac{1}{2n^{p_1-2}} - \frac{1}{2K^{p_2-2}}\right) / \left(1 - \frac{1}{2n^{p_1-2}}\right). \tag{16}$$

**Proof** Direct calculation gives that

$$\begin{aligned}
\mathbb{P}[E_2 \mid E_1] &= 1 - \mathbb{P}[E_2^c \mid E_1] \\
&= 1 - \tfrac{\mathbb{P}[E_1 \bigcap E_2^c]}{\mathbb{P}[E_1]} \\
&\geq 1 - \tfrac{\mathbb{P}[E_2^c]}{\mathbb{P}[E_1]} \\
&\geq 1 - \tfrac{C(K,2)K^{-p_2}}{1 - C(n,2)n^{-p_1}} \\
&> 1 - \tfrac{\frac{1}{2K^{p_2-2}}}{1 - \frac{1}{2n^{p_1-2}}} \\
&= \left(1 - \tfrac{1}{2n^{p_1-2}} - \tfrac{1}{2K^{p_2-2}}\right) / \left(1 - \tfrac{1}{2n^{p_1-2}}\right).
\end{aligned}$$

■

The following corollary is useful in practice and its proof can be found in A.1.

**Corollary 1** *Let $\Pi \sim D_{\epsilon,\delta}$ be as described in Theorem 2. If the random matrix $\Pi$ satisfies (12), then, under the same assumptions, the statements of Theorem 2 hold with probability at least $\left(1 - \frac{1}{2n^{p_1-2}} - \frac{1}{2K^{p_2-2}}\right) / \left(1 - \frac{1}{2n^{p_1-2}}\right)$.*

Here, we want to make some remarks on the obtained recovery guarantees of the model (RPCCM).

1. The embedding dimension $m$ only depends on $\epsilon$, $n$, and $p_1$, but it is independent of the data dimension $d$. Also, $m$ grows very slowly with respect to $n$ (in $O(\log(n))$).

2. We derive the lower bound $\epsilon_{\min}$ and the upper bound $\epsilon_{\sup}$ of $\epsilon$ for perfect recovery of the model (RPCCM). In particular, the lower bound $\epsilon_{\min}$ can be very small for high dimensional data. The upper bound $\epsilon_{\sup}$ depends on the ratio of $\gamma_{\max}$ and $\gamma_{\min}$, and it is independent of the scale of the data.

3. The weights used in the models (CCM) and (RPCCM) in Theorem 2 are identical, which is necessary for the current proof of Theorem 2. In numerical experiments,

we will follow a popular practical setting of weights (1). Our numerical results show that the performance under this setting of weights is robust, and the time cost of weight construction is minimal compared to the run-time cost by solving the model along a sequence of values of $\gamma$ to generate a clustering path. It is interesting to explore theoretical guarantees using weights constructed from the embedded data. The potential changes of edges in the weight graph pose challenges for establishing theoretical guarantees, and a potential approach is to leverage the effectiveness of constructing an approximate k-nearest-neighbor graph using random projections. We would like to take this as a future research direction.

4. Dimension reduction based on the JL lemma has also been investigated for the K-means model (Cohen et al., 2015). However, for the K-means model, only the cost (the optimal objective function value of the K-means model) can be preserved up to a tolerance $\epsilon > 0$. Here, we prove that the perfect recovery guarantee of the model (CCM) can be inherited. A comparison of the empirical performance between the randomly projected K-means model and the model (RPCCM) can be found later in the numerical experiments.

The embedding dimension $m$ in Theorem 2 depends on $n$ of the order $O(\log(n))$. This arises as a consequence of the requirement to preserve pairwise distances for data points at the order of $n^2$ (see (11a)). Next, we will further show that $m$ can be $O(\log(K))$, which is independent of $n$. The key insights are that the $\epsilon$-JL transform is linear and we can upper-bound the distortion of the pairwise distances directly by its spectral norm (Lemma 3). The details of the recovery guarantees can be found in the next theorem.

**Theorem 3** *Consider the general problem setting and the models (CCM) and (RPCCM) with the same specified weights $w_{ij} = w_{ji} \geq 0$. We assume $p_2 > 2$ and $\sqrt{\frac{p_2 C \log(K)}{d}} < 1$. Define $C_0 = \frac{\sqrt{d}+2C_\kappa^2}{\sqrt{p_2 C\log(K)}}$ and*

$$\tilde{\epsilon}_{\min} = \sqrt{\tfrac{p_2 C \log(K)}{d}},\ \tilde{\epsilon}_{\sup} = \tfrac{r - C_\kappa^2}{C_0 + r}, \tag{17}$$

*where $r$ is as defined in (3). If $r > \frac{1+C_\kappa^2+\frac{2C_\kappa^2}{\sqrt{d}}}{1-\tilde{\epsilon}_{\min}}$, then $\tilde{\epsilon}_{\min} < \tilde{\epsilon}_{\sup}$. For any $(\epsilon, \gamma)$ such that*

$$\epsilon \in (\tilde{\epsilon}_{\min}, \tilde{\epsilon}_{\sup}),\ \gamma \in \big[\bar{S}(m,d,2)\gamma_{\min}, (1-\epsilon)\gamma_{\max}\big)\,, \tag{18}$$

*let $\delta = \frac{1}{K^{p_2}}$, and let $\Pi = \frac{1}{\sqrt{m}}R \in \mathbb{R}^{m\times d}$ and $\bar{S}(m,d,2)$ be as defined in Lemma 3 with $m \in \big[p_2 C\epsilon^{-2}\log(K), d\big)$, where $\epsilon \in (\sqrt{\frac{p_2 C\log(K)}{d}}, 1)$, then with probability over $1 - \frac{1}{2K^{p_2-2}} - 2\exp(-2^2)$, the map $\hat{\phi}_\gamma$ perfectly recovers $\mathcal{V}$.*

**Proof** With probability over $1 - \frac{1}{2K^{p-2}} - 2\exp(-2^2)$, we have that

$$\|\Pi(\mathbf{a}_i - \mathbf{a}_j)\| \ \leq \bar{S}(m,d,2)\|\mathbf{a}_i - \mathbf{a}_j\|,\ i,j \in I_\alpha, 1 \leq \alpha \leq K, \tag{19a}$$

$$\|\Pi(\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)})\| \geq (1-\epsilon)\|\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)}\|,\ 1 \leq \alpha \neq \beta \leq K, \tag{19b}$$

which implies that

(i) The centroids $\left\{\Pi\mathbf{a}^{(1)}, \dots, \Pi\mathbf{a}^{(K)}\right\}$ of the embedded data are distinct.

(ii) $\hat{\gamma}_{\min}$ and $\hat{\gamma}_{\max}$ as defined in (10) satisfy the following inequalities:

$$\hat{\gamma}_{\min} \leq s_1(\Pi)\gamma_{\min} \leq \bar{S}(m,d,2)\gamma_{\min}, \quad (1-\epsilon)\gamma_{\max} \leq \hat{\gamma}_{\max}.$$

The above implies that

$$\left[\bar{S}(m,d,2)\gamma_{\min}, (1-\epsilon)\gamma_{\max}\right) \subseteq \left[\hat{\gamma}_{\min}, \hat{\gamma}_{\max}\right). \tag{20}$$

Now, we prove the theorem. We claim here that it is sufficient to show: if $r > \frac{1+C_\kappa^2+\frac{2C_\kappa^2}{\sqrt{d}}}{1-\tilde{\epsilon}_{\min}}$, then $\tilde{\epsilon}_{\min} < \tilde{\epsilon}_{\sup}$, and for any $\epsilon \in (\tilde{\epsilon}_{\min}, \tilde{\epsilon}_{\sup})$, the interval $\left[\bar{S}(m,d,2)\gamma_{\min}, (1-\epsilon)\gamma_{\max}\right)$ is nonempty. In fact, if $\left[\bar{S}(m,d,2)\gamma_{\min}, (1-\epsilon)\gamma_{\max}\right)$ is nonempty, then by the first inclusion of (20), $[\hat{\gamma}_{\min}, \hat{\gamma}_{\max})$ is nonempty. Applying Theorem 1 to the embedded data $\Pi A$ implies that for any $\gamma \in [\hat{\gamma}_{\min}, \hat{\gamma}_{\max})$, the map $\hat{\phi}_\gamma$ perfectly recovers $\mathcal{V}$.

On the one hand, by definition of $\tilde{\epsilon}_{\min}$ and $C_0$, we have

$$\frac{1}{\sqrt{d}} = \frac{\tilde{\epsilon}_{\min}}{\sqrt{p_2 C \log(K)}}, \; C_0 = \tilde{\epsilon}_{\min}^{-1} + \frac{2C_\kappa^2}{\sqrt{p_2 C \log(K)}}. \tag{21}$$

As a result,

$$\begin{array}{rll} r > \frac{1+C_\kappa^2+\frac{2C_\kappa^2}{\sqrt{d}}}{1-\tilde{\epsilon}_{\min}} & \Longrightarrow & r > \frac{C_\kappa^2+\tilde{\epsilon}_{\min}\tilde{\epsilon}_{\min}^{-1}+\frac{\tilde{\epsilon}_{\min}2C_\kappa^2}{\sqrt{p_2 C\log(K)}}}{\sqrt{1-\tilde{\epsilon}_{\min}}} \\ & \Longrightarrow & r > \frac{C_\kappa^2+\tilde{\epsilon}_{\min}\left(\tilde{\epsilon}_{\min}^{-1}+\frac{2C_\kappa^2}{\sqrt{p_2 C\log(K)}}\right)}{1-\tilde{\epsilon}_{\min}} \\ & \Longrightarrow & r > \frac{C_\kappa^2+\tilde{\epsilon}_{\min}C_0}{1-\tilde{\epsilon}_{\min}} \\ & \Longrightarrow & C_\kappa^2 + \tilde{\epsilon}_{\min}C_0 < 1 - \tilde{\epsilon}_{\min}r \\ & \Longrightarrow & \tilde{\epsilon}_{\min} < \frac{r-C_\kappa^2}{C_0+r} = \tilde{\epsilon}_{\sup}. \end{array}$$

On the other hand, we have

$$\bar{S}(m,d,2) = \frac{\sqrt{d}+2C_\kappa^2}{\sqrt{m}} + C_\kappa^2 = \frac{\sqrt{d}+2C_\kappa^2}{\sqrt{p_2 C \log(K)}}\epsilon + C_\kappa^2 = C_0\epsilon + C_\kappa^2. \tag{22}$$

As a result,

$$\begin{array}{rll} \epsilon \in (\tilde{\epsilon}_{\min}, \tilde{\epsilon}_{\sup}) & \Longrightarrow & C_0\epsilon + C_\kappa^2 < (1-\epsilon)r \\ & \Longrightarrow & \bar{S}(m,d,2)\gamma_{\min} < (1-\epsilon)\gamma_{\max}. \end{array}$$

Thus, we have proved the theorem. ■

Here, we want to make some remarks on the obtained results.

1. The embedding dimension in Theorem 3 is independent of the number of data points $n$, which is important for clustering an extremely large number of data points.

2. The results of this theorem and Theorem 2 further demonstrate that the ratio $r = \gamma_{\max}/\gamma_{\min}$ is a data scale-invariant measure to characterize the difficulty of clustering a given collection of data. Since the embedding dimension of the JL lemma depends on $\mathrm{O}(\epsilon^{-2})$, the value $\tilde{\epsilon}_{\min}$ (and $\epsilon_{\min}$) can be interpreted as the lowest possible dimension reduction ratio obtained by the JL lemma. It has been known that the JL lemma is optimal if the $\epsilon$-isometry mapping is linear, thus, the condition $r > \frac{1+C_\kappa^2+\frac{2C_\kappa^2}{\sqrt{d}}}{1-\tilde{\epsilon}_{\min}}$ in Theorem 3 (and $r > \frac{1+\epsilon_{\min}}{1-C_{12}\epsilon_{\min}}$ in Theorem 2) shows that the dimension reduction results obtained in this paper are intrinsically depending on the difficulties of clustering the input data.

3. The choice of $t = 2$ in $S(m, d, t)$ in Theorem 3 is made for the sake of simplicity in the analysis. In particular, by taking $t = 2$, with probability over $1-2\exp\left(-2^2\right) \approx 0.9632$, the largest singular value $s_1(\Pi)$ is upper bounded by $\bar{S}(m, d, 2) = \sqrt{\frac{d}{m}} + C_\kappa^2(1 + \frac{2}{\sqrt{m}})$.

4. For the K-means model, Cohen et al. (2015) proved that the cost can be preserved up to a $(9 + \epsilon)$ approximation bounds if the embedding dimension $m = O(\epsilon^{-2}\log(K))$. This bound has been improved to $(1+\epsilon)$ provided $m = O(\epsilon^{-2}\log(K/\epsilon))$ (Makarychev et al., 2023). However, it is still unknown whether the randomly projected K-means model can preserve the cluster membership assignments or not.

## 4. Cluster Recovery Guarantees of the Model (RPCCM) with Uniform Weights for Recovering a Mixture of Spherical Gaussians

In this section, we shift our interests to the problem setting of clustering data points generated from a mixture of spherical Gaussians using the model (RPCCM) with uniform weights. We mainly follow the problem settings in (Jiang et al., 2020).

**Mixtures of spherical Gaussians (MSG) problem setting**: Cluster a collection of $n$ data points $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ that are sampled from a mixture of $K$ spherical Gaussian distributions $\mathcal{N}_d(\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1^2\mathbf{I}_d), \ldots, \mathcal{N}_d(\boldsymbol{\mu}_K, \boldsymbol{\sigma}_K^2\mathbf{I}_d)$ with positive probabilities $\mathbf{p}_1, \ldots, \mathbf{p}_K$ summing to 1. Each observation $\mathbf{a}_i$ is independently sampled from exactly one of the $K$ Gaussians, which is selected at random with probability $\mathbf{p}_\alpha$. The ground-truth index set associated with the $\alpha$-th Gaussian is $I_\alpha = \{i \in [n] \mid \mathbf{a}_i \text{ is sampled from } \mathcal{N}_d(\boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha^2\mathbf{I}_d)\}$.

We assume the following assumptions, which are reasonable for the MSG setting.

**Assumption 4** *The means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^d$ are all distinct.*

**Assumption 5** *The number of clusters $K$ satisfies $K \le \sqrt{n}$.*

The MSG setting is very challenging although we assume the covariance matrices are $\sigma_\alpha^2 I$ $(1 \le \alpha \le K)$ for simplicity. As pointed out in (Jiang et al., 2020), when $n$ is sufficiently large, some samples associated with one Gaussian can be placed arbitrarily near the mean of another Gaussian. Consequently, it's too restrictive to expect a perfect recovery of the Gaussians and a natural idea is to consider the recovery over the following index sets

$$\tilde{I}_\alpha(d, \theta) := \{i \in I_\alpha \mid \|\mathbf{a}_i - \boldsymbol{\mu}_\alpha\| \le \theta\boldsymbol{\sigma}_\alpha\}, \ \alpha \in [K], \tag{23}$$

which contains indexes of points associated with $\mathcal{N}_d(\boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha^2 \mathbf{I}_d)$ that are within a constant positive distance $\theta\boldsymbol{\sigma}_\alpha$ from $\boldsymbol{\mu}_\alpha$. The following proposition is useful and its proof can be found in A.2.

**Proposition 3** *In the MSG problem setting, let $\theta > 0$ be given, and let $\tilde{I}_\alpha(d,\theta)$ be as defined in (23). Let $F(d,\theta)$ denote the cumulative density function of the d-dimensional Chi distribution. For any $\alpha \in [K]$:*

1. $\|\mathbf{a}_i - \mathbf{a}_j\| \leq 2\theta\boldsymbol{\sigma}_\alpha$, *for any* $i, j \in \tilde{I}_\alpha(d,\theta)$.

2. $\mathbb{E}\left[|\tilde{I}_\alpha(d,\theta)|\right] = F(\theta,d)\mathbf{p}_\alpha n$. *For any* $\eta > 0$,

$$\mathbb{P}\left[|\tilde{I}_\alpha(d,\theta)| \geq (F(\theta,d) - \eta)\,\mathbf{p}_\alpha n\right] \geq 1 - \exp\left(-2\mathbf{p}_\alpha^2 \eta^2 n\right). \tag{24}$$

3. *For any nonzero* $h \in \mathbb{R}^d$ *and any* $1 \leq \tilde{n} \leq n$,

$$\mathbb{P}\left[\exists i \in \tilde{I}_\alpha(d,\theta) \text{ s.t. } \left(\mathbf{a}_i - \boldsymbol{\mu}_\alpha\right)^\top h \geq 0 \mid |\tilde{I}_\alpha(d,\theta)| \geq \tilde{n}\right] \geq 1 - 2^{-\tilde{n}}. \tag{25}$$

*Particularly, for any* $\eta > 0$,

$$\begin{aligned}&\mathbb{P}\left[|\tilde{I}_\alpha(d,\theta)| \geq (F(\theta,d) - \eta)\,\mathbf{p}_\alpha n, \text{ and } \exists i \in \tilde{I}_\alpha(d,\theta) \text{ s.t. } \left(\mathbf{a}_i - \boldsymbol{\mu}_\alpha\right)^\top h \geq 0\right] \\ &\geq \left(1 - 2^{-(F(\theta,d)-\eta)\mathbf{p}_\alpha n}\right)\left(1 - \exp\left(-2\mathbf{p}_\alpha^2\eta^2 n\right)\right).\end{aligned} \tag{26}$$

### 4.1 Cluster Recovery Guarantees of the Model (CCM) with Uniform Weights for the MSG Problem Setting

In this section, we introduce the recovery guarantees of the model (CCM) with uniform weights for the MSG problem setting established in (Jiang et al., 2020). For later convenience, we first introduce some necessary notation.

**Definition 6** *We say that a map $\psi : \mathbb{R}^d \to \mathbb{R}^{\bar{d}}$ correctly recovers points indexed by $L > 1$ disjoint index sets $\tilde{I}_1, \ldots, \tilde{I}_L$, $\cup_{l=1}^L \tilde{I}_l \subseteq \{1, ..., n\}$, on the data $A$ if for any $1 \leq \alpha \neq \beta \leq K$, $\psi(\mathbf{a}_i) = \psi(\mathbf{a}_j)$ for any $i, j \in \tilde{I}_\alpha$, and $\psi(\mathbf{a}_i) \neq \psi(\mathbf{a}_{i'})$ for any $i \in \tilde{I}_\alpha, i' \in \tilde{I}_\beta$.*

For later convenience in our theoretical analysis, we introduce the following corollary, which is a direct consequence of (Jiang et al., 2020, Theorem 6). It states that under mild conditions, points indexed by $\tilde{I}_\alpha(d,\theta), \alpha \in [K]$ can be correctly labeled using the model (CCM) with uniform weights with high probability.

**Corollary 2** *In the MSG problem setting, let $\theta > 0$ and $\eta > 0$ be arbitrary. Consider the model (CCM) with uniform weights. Define*

$$\gamma_{\min}^G := \max_{1\leq\alpha\leq K} \tfrac{2\theta\boldsymbol{\sigma}_\alpha}{(F(\theta,d)-\eta)\boldsymbol{p}_\alpha n}, \gamma_{\max}^G := \min_{1\leq\alpha<\beta\leq K} \tfrac{\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\|}{2(n-1)},\ r^G := \gamma_{\max}^G / \gamma_{\min}^G. \tag{27}$$

*Let $P_G = -K + \sum_{\alpha\in[K]}\left(1 - \exp\left(-2\mathbf{p}_\alpha^2\eta^2 n\right)\right)\left(1 - (K-1)2^{-(F(\theta,d)-\eta)\mathbf{p}_\alpha n}\right)$. If $r^G > 1$, with probability over $1 - P_G$, for any $\gamma \in \left[\gamma_{\min}^G, \gamma_{\max}^G\right)$, the map $\hat{\phi}_\gamma$ correctly recovers points indexed by $\tilde{I}_\alpha(d,\theta), \alpha \in [K]$.*

The next lemma will be useful.

**Lemma 5** (Jiang et al., 2023, Theorem 6) *In the general problem setting, let $C_\alpha \subseteq I_\alpha$, $\alpha \in [K]$. Consider the model (CCM) with uniform weights. For any $1 \leq \alpha \neq \beta \leq K$,*

1. *The points indexed by $C_\alpha$ are assigned in the same cluster provided*

$$\gamma \geq \max_{i,j \in C_\alpha} \frac{\|\mathbf{a}_i - \mathbf{a}_j\|}{|C_\alpha|}. \tag{28}$$

2. *Furthermore, if*

$$\max_{l \in \{\alpha,\beta\}} \max_{i,j \in C_l} \frac{\|\mathbf{a}_i - \mathbf{a}_j\|}{|C_l|} < \max_{\substack{i \in C_\alpha \\ i' \in C_\beta}} \frac{\|\mathbf{a}_i - \mathbf{a}_{i'}\|}{2(n-1)}, \tag{29}$$

*then for any $\gamma$ such that*

$$\max_{l \in \{\alpha,\beta\}} \max_{i,j \in C_l} \frac{\|\mathbf{a}_i - \mathbf{a}_j\|}{|C_l|} \leq \gamma < \max_{\substack{i \in C_\alpha \\ i' \in C_\beta}} \frac{\|\mathbf{a}_i - \mathbf{a}_{i'}\|}{2(n-1)}, \tag{30}$$

*the map $\hat{\phi}_\gamma$ correctly recovers points indexed by $C_\alpha$ and $C_\beta$.*

### 4.2 Cluster Recovery Guarantees of the Model (RPCCM) with Uniform Weights for the MSG Problem Setting

In this section, we will establish the recovery guarantees of the model (RPCCM) with uniform weights for the MSG problem setting. We introduce some necessary notation first.

**Definition 7** *In the MSG problem setting, let $\Pi \in \mathbb{R}^{m \times d}$ (for some $m \geq 1$) be a randomly sampled projection matrix. Consider the model (RPCCM) with uniform weights. Without explicitly mentioning the dependence on $\Pi$, we define*

$$\begin{aligned} \hat{\gamma}^G_{\min} &:= \max_{\alpha \in [K]} \max_{i,j \in \tilde{I}_\alpha(d,\theta)} \frac{\|\Pi(\mathbf{a}_i - \boldsymbol{\mu}_\alpha)\| + \|\Pi(\mathbf{a}_j - \boldsymbol{\mu}_\alpha)\|}{(F(\theta,d) - \eta)\, \boldsymbol{p}_\alpha n}, \\ \hat{\gamma}^G_{\max} &:= \min_{1 \leq \alpha < \beta \leq K} \frac{\left\|\Pi\left(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right)\right\|}{2(n-1)}. \end{aligned} \tag{31}$$

The following proposition is useful. Its proof can be found in A.3.

**Proposition 4** *In the MSG problem setting, let $\theta > 0$, and let $\tilde{I}_\alpha(d,\theta)$ be as defined in (23). Let $\Pi \in \mathbb{R}^{m \times d}$ (for some $m \geq 1$) be a randomly sampled projection matrix. Assume that $\Pi\boldsymbol{\mu}_1, \ldots, \Pi\boldsymbol{\mu}_K \in \mathbb{R}^m$ are all distinct. For any $\eta > 0$, we have that*

$$\hat{\gamma}^G_{\min} \geq \max_{\alpha \in [K]} \max_{i,j \in \tilde{I}_\alpha(d,\theta)} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_j)\|}{|\tilde{I}_\alpha(d,\theta)|},\ \hat{\gamma}^G_{\max} \leq \min_{1 \leq \alpha < \beta \leq K} \max_{\substack{i \in \tilde{I}_\alpha(d,\theta) \\ i' \in \tilde{I}_\beta(d,\theta)}} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_{i'})\|}{2(n-1)}, \tag{32}$$

*with probability over $1 - P_G$, where*

$$P_G := -K + \sum_{\alpha \in [K]} \left(1 - \exp\left(-2\mathbf{p}_\alpha^2 \eta^2 n\right)\right) \left(1 - (K-1) 2^{-(F(\theta,d) - \eta)\mathbf{p}_\alpha n}\right). \tag{33}$$

The following proposition is a consequence of Lemma 5 and Proposition 4. Its proof can be found in A.4.

**Proposition 5** *In the MSG problem setting, let $\theta > 0$, and let $\tilde{I}_\alpha(d, \theta)$ be as defined in (23). Let $\eta > 0$ be arbitrary. Consider (RPCCM) with uniform weights. Let $\Pi \in \mathbb{R}^{m\times d}$ (for some $m \geq 1$) be a randomly sampled projection matrix. Assume that $\Pi\boldsymbol{\mu}_1, \dots, \Pi\boldsymbol{\mu}_K \in \mathbb{R}^m$ are all distinct. Let $P_G$ be as defined in (33). If $\hat{\gamma}^G_{\min} < \hat{\gamma}^G_{\max}$, then with probability over $1 - P_G$, for any $\gamma \in \left[\hat{\gamma}^G_{\min}, \hat{\gamma}^G_{\max}\right)$, the map $\hat{\phi}_\gamma$ correctly recovers points indexed by $\tilde{I}_\alpha(d, \theta), \alpha \in [K]$.*

It follows Proposition 5 that if $\hat{\gamma}^G_{\min} < \hat{\gamma}^G_{\max}$ for a specific $\Pi$, the model (RPCCM) with uniform weights can correctly recover points indexed by $\tilde{I}_\alpha(d, \theta)$ for all $\alpha \in [K]$, provided $\gamma \in \left[\hat{\gamma}^G_{\min}, \hat{\gamma}^G_{\max}\right)$. The definitions of $\hat{\gamma}^G_{\min}$ and $\hat{\gamma}^G_{\max}$ inspire us to adapt our analysis on the model (RPCCM) for the general problem setting to the MSG problem setting. Let

$$X_G := \left\{\mathbf{a}_i - \boldsymbol{\mu}_\alpha \mid i \in \tilde{I}_\alpha(\theta, d), \alpha \in [K]\right\},\ X_{\boldsymbol{\mu}} := \left\{\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta \mid 1 \leq \alpha < \beta \leq K\right\}.$$

The following proposition shows that a random projection $\Pi \in \mathbb{R}^{m\times d}$ with $m = O(\epsilon^{-2}\log(n))$ can preserve all the norms in $X_G$ up to a multiplicative $(1 + \epsilon)$ factor, while simultaneously preserving all the norms in $X_{\boldsymbol{\mu}}$ down to a multiplicative $(1 - \epsilon_2)$ factor, where $\epsilon_2 < \epsilon$ will be specified in the proposition. The proof of Proposition 6 can be found in A.5.

**Proposition 6** *In the MSG problem setting, let $p_1 > 1, p_2 > 2$, and assume that $C_{12} := \sqrt{\frac{\log(K)p_2}{\log(n)p_1}} < 1$. For any $\epsilon \in (0, 1/C_{12})$, let $\epsilon_2 = C_{12}\epsilon$. Let $\delta = \frac{1}{n^{p_1}}$ and $\Pi \sim D_{\epsilon,\delta}$ with $m \geq p_1 C\epsilon^{-2}\log(n)$, then with probability over $1 - \frac{1}{n^{p_1-1}} - \frac{1}{2K^{p_2-2}}$,*

$$\|\Pi(\mathbf{a}_i - \boldsymbol{\mu}_\alpha)\| \leq (1 + \epsilon)\|\mathbf{a}_i - \boldsymbol{\mu}_\alpha\|,\ i \in \tilde{I}_\alpha(\theta, d), 1 \leq \alpha \leq K, \tag{34a}$$
$$\|\Pi(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta)\| \geq (1 - \epsilon_2)\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\|,\ 1 \leq \alpha \neq \beta \leq K. \tag{34b}$$

**Remark 4** *Proposition 1 and Proposition 6 differ due to the substitution of (11) with (34). In the general problem setting, our focus is on preserving distances within clusters and between cluster centroids. However, in the MSG problem setting, the emphasis is on preserving distances between points and their associated Gaussian means, as well as distances between Gaussian means, which reduces the number of distances to be preserved from the order of $n^2$ to $n$.*

The next theorem shows the theoretical recovery guarantee of the model (RPCCM) with uniform weights for the MSG problem setting. We include its proof in A.6.

**Theorem 4** *In the MSG problem setting, let $\theta > 0$, and let $\tilde{I}_\alpha(d, \theta)$ be as defined in (23). Let $\eta > 0$ be arbitrary. Consider the model (RPCCM) with uniform weights. Let $p_1 > 1, p_2 > 2$, and we assume that $C_{12} := \sqrt{\frac{\log(K)p_2}{\log(n)p_1}} < 1$ and $\sqrt{\frac{p_2 C\log(K)}{d}} < 1$, and thus $\sqrt{\frac{p_1 C\log(n)}{d}} < \frac{1}{C_{12}}$. Define*

$$\epsilon^G_{\min} = \sqrt{\frac{p_1 C\log(n)}{d}},\ \epsilon^G_{\sup} = \frac{r^G - 1}{C_{12}r^G + 1}, \tag{35}$$

*where $r^G$ is as defined in (27). Let $P_G$ be as defined in (33). If $r^G > \frac{1+\epsilon^G_{\min}}{1-C_{12}\epsilon^G_{\min}}$, then $\epsilon^G_{\min} < \epsilon^G_{\sup}$. For any $(\epsilon, \gamma)$ such that*

$$\epsilon \in (\epsilon^G_{\min}, \epsilon^G_{\sup}), \;\; \gamma \in \left[(1+\epsilon)\gamma^G_{\min}, (1-C_{12}\epsilon)\gamma^G_{\max}\right), \tag{36}$$

*let $\delta = \frac{1}{n^{p_1}}$ and $\Pi \sim D_{\epsilon,\delta}$ with $m \in [p_1 C\epsilon^{-2}\log(n), d)$, then with probability over $1 - \frac{1}{n^{p_1-1}} - \frac{1}{2K^{p_2-2}} - P_G$, the map $\hat{\phi}_\gamma$ correctly recovers points indexed by $\tilde{I}_\alpha(d,\theta), \alpha \in [K]$.*

The embedding dimension $m$ in Theorem 4 depends on $n$ of the order $O(\log(n))$. Next, we will further show that $m$ can be $O(\log(K))$, which is independent of $n$. We include its proof in A.7.

**Theorem 5** *In the MSG problem setting, let $\theta > 0$, and let $\tilde{I}_\alpha(d,\theta)$ be as defined in (23). Let $\eta > 0$ be arbitrary. Consider the model (RPCCM) with uniform weights. We assume $\sqrt{\frac{p_2 C\log(K)}{d}} < 1$, where $p_2 > 2$. Define $C_0 = \frac{\sqrt{d}+2C_\kappa^2}{\sqrt{p_2 C\log(K)}}$ and*

$$\tilde{\epsilon}^G_{\sup} = \frac{r^G - C_\kappa^2}{C_0 + r^G}, \; \tilde{\epsilon}^G_{\min} = \sqrt{\frac{p_2 C\log(K)}{d}}, \tag{37}$$

*where $r^G$ is as defined in (27). Let $P_G$ be as defined in (33). If $r^G > \frac{1+C_\kappa^2+\frac{2C_\kappa^2}{\sqrt{d}}}{1-\tilde{\epsilon}^G_{\min}}$, then $\tilde{\epsilon}^G_{\min} < \tilde{\epsilon}^G_{\sup}$. For any $(\epsilon, \gamma)$ such that*

$$\epsilon \in (\tilde{\epsilon}^G_{\min}, \tilde{\epsilon}^G_{\sup}), \;\; \gamma \in \left[\bar{S}(m,d,2)\gamma^G_{\min}, (1-\epsilon)\gamma^G_{\max}\right), \tag{38}$$

*let $\delta = \frac{1}{K^{p_2}}$, and let $\Pi = \frac{1}{\sqrt{m}}R \in \mathbb{R}^{m\times d}$ and $\bar{S}(m,d,2)$ be as defined in Lemma 3 with $m \in \left[p_2 C\epsilon^{-2}\log(K), d\right)$, then with probability over $1 - \frac{1}{2K^{p_2-2}} - 2\exp(-2^2) - P_G$, the map $\hat{\phi}_\gamma$ correctly recovers points indexed by $\tilde{I}_\alpha(d,\theta), \alpha \in [K]$.*

**Remark 5** *To state a simpler formulation for $r^G$, following (Jiang et al., 2020), we can fix some values. For example, we can take $\theta = \theta_d := 2\sqrt{d}$ and let $c_d = F(\theta_d, d)$. It follows Theorem 3.1.1 in (Vershynin, 2018) that $c_d \geq 1 - 2\exp(-cd)$, where $c > 0$ is a constant that does not depend on d, which implies that $c_d \to 1$ exponentially as d increases. In this case, we have that*

$$r^G = \frac{\min_{1\leq\alpha<\beta\leq K}\left\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right\|}{\max_{1\leq\alpha\leq K}\boldsymbol{\sigma}_\alpha/\mathbf{p}_\alpha}\frac{(c_d-\eta)\,n}{8\sqrt{d}(n-1)}. \tag{39}$$

*If we further let $\mathbf{p}_{\min} = \min_{\alpha\in[K]}\mathbf{p}_\alpha$, $\boldsymbol{\sigma}_{\max} = \max_{\alpha\in[K]}\boldsymbol{\sigma}_\alpha$, and $\eta = c_d/2$, then we have*

$$r^G > \frac{(c_d\boldsymbol{p}_{\min})\min_{1\leq\alpha<\beta\leq K}\left\|\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right\|}{16\sqrt{d}\boldsymbol{\sigma}_{\max}}. \tag{40}$$

**Remark 6** *Several approaches (Dasgupta, 1999; Vempala and Wang, 2004; Achlioptas and McSherry, 2005; Dasgupta and Schulman, 2007) have been investigated for recovering or learning a mixture of spherical Gaussians. In particular, a spectral projection method (Vempala and Wang, 2004) demonstrated the capacity to maintain the separation between Gaussian means while reducing the radii of Gaussians, facilitating easier Gaussian separation* *compared to random projection approaches. Here, we want to emphasize that the primary focus of this section is to establish theoretical guarantees for the model (RPCCM) in the MSG problem setting. To this end, we demonstrate that under mild conditions, if the model (CCM) can correctly recover the Gaussians, the model (RPCCM) can achieve the same performance. As a consequence, the computational challenge of the model (CCM) for clustering high-dimensional data can be alleviated by considering the model (RPCCM) instead.*

## 5. Numerical Experiments

In this section, we will present extensive numerical results to demonstrate the robust performance of the model (RPCCM). We will first verify the theoretical recovery guarantees as established in this paper. Then, we will conduct numerical experiments to show its superior practical performance compared to several other popular clustering algorithms. All our computational results were obtained using MATLAB on a Windows laptop (Intel Core i7-8750H @ 2.20GHz RAM 32GB). Before we present the detailed results, we first describe the settings in our numerical experiments.

**Datasets:** We perform tests on seven simulated datasets and five real datasets with varying $d$, $n$, and $K$, whose details will be introduced later. We denote the tested dataset as a $d$ by $n$ matrix $A \in \mathbb{R}^{d\times n}$. We normalize the dataset by dividing the maximum Euclidean distance between all pairs of data points by default.

**Construction of the $\epsilon$-JL Transform**: We construct the random projection matrix $\Pi \in \mathbb{R}^{m\times d}$ with a specified embedding dimension $m$ as

$$\Pi = \frac{1}{\sqrt{m}} G \in \mathbb{R}^{m\times d}, \tag{41}$$

where $G_{ij}$ are independently sampled from the standard normal distribution. We obtain the embedded data $\Pi A \in \mathbb{R}^{m\times d}$ by applying the projection matrix $\Pi$ to the data $A$.

**Construction of weights:** In this paper, we set identical weights for both (RPCCM) and (CCM). We construct weights based on data $A$ as

$$w_{ij} = \begin{cases} \exp(-\left\|\mathbf{a}_i - \mathbf{a}_j\right\|^2/2) & \text{if } (i,j) \in \mathcal{E}_A, \\ 0 & \text{otherwise}, \end{cases} \tag{42}$$

where $\mathcal{E}_A$ will be specified later in the experiments.

**Clustering implementation:** We will solve (CCM) and (RPCCM) with the specified weights $w_{ij}$ for a sequence of $\gamma$ and generate clustering paths. For simplicity, we choose the same sequence of $\gamma$ (i.e. $\infty > \gamma_1 > \gamma_2 > \cdots > \gamma_T > 0$) for both models. The particular values of $\gamma$ will be specified in the experiments. Note that (RPCCM) is essentially (CCM) applied to the embedded data $\Pi A$. Therefore, the same algorithms can be applied to solve both models.

1. **Algorithm:** We employ the semismooth Newton-based augmented Lagrangian method with an adaptive sieving technique (AS-SSNAL) as our optimization solver (Yuan et al., 2022), which is arguably the most efficient algorithm for solving the model

(CCM). We adopt the stopping criterion in (Yuan et al., 2022) with a tolerance $\epsilon_{\mathrm{tol}} = 10^{-7}$. For more details of the algorithm, one can refer to (Yuan et al., 2022) and the references therein. We include a computational complexity analysis of AS-SSNAL for solving (CCM) and (RPCCM) in Appendix A.8.

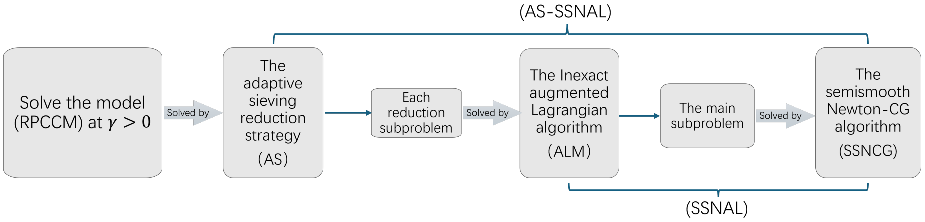

(AS-SSNAL)
Solve the model (RPCCM) at $\gamma > 0$
Solved by
The adaptive sieving reduction strategy (AS)
Each reduction subproblem
Solved by
The Inexact augmented Lagrangian algorithm (ALM)
The main subproblem
Solved by
The semismooth Newton-CG algorithm (SSNCG)
(SSNAL)

Figure 2: The overall structure of AS-SSNAL for solving the model (RPCCM).

2. **Clustering path:** We obtain a clustering path based on the inexact checking rule up to the tolerance $\epsilon_{\mathrm{clust}} = 10^{-5}$.

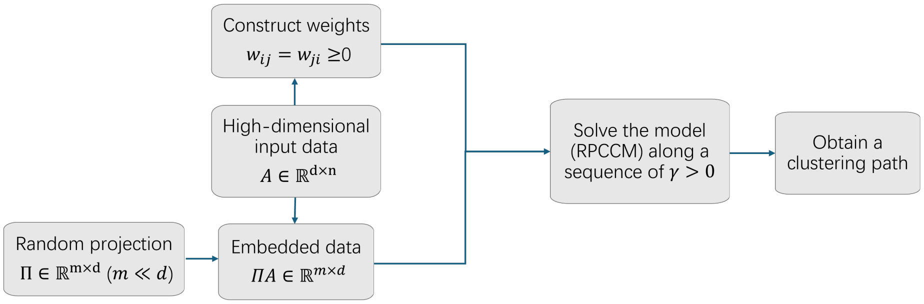

Construct weights $w_{ij} = w_{ji} \geq 0$
High-dimensional input data $A \in \mathbb{R}^{d\times n}$
Random projection $\Pi \in \mathbb{R}^{m\times d}$ $(m \ll d)$
Embedded data $\Pi A \in \mathbb{R}^{m\times d}$
Solve the model (RPCCM) along a sequence of $\gamma > 0$
Obtain a clustering path

Figure 3: Procedures for implementing the model (RPCCM).

**Evaluation criteria for clustering results:** We adopt the commonly used Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) and Adjusted Mutual Information (AMI) (Vinh et al., 2009, 2010) to quantitatively measure the quality of the obtained clustering results. Particularly, ARI=1 (AMI=1) indicates a perfect cluster recovery.

We organize our numerical experiment results as follows: In Section 5.1, we verify the theoretical guarantees of the model (RPCCM) as stated in Theorem 2 and Theorem 3. We further verify the recovery guarantees for the MSG problem setting as stated in Theorem 4 and Theorem 5 in A.10. In Section 5.2, we test the practical performance of the model (RPCCM) on simulated and real data and compare its performance with other popular clustering algorithms on the dimension-reduced data. We further test the performance of the model (RPCCM) with weights constructed from the embedded data.

### 5.1 Numerical Verification for the Model (RPCCM)

In this section, we verify the theoretical recovery guarantees in Theorem 2 and Theorem 3, respectively. We conduct tests on seven simulated datasets denoted as $A_1, \cdots, A_7$.

Each dataset is generated from a mixture of $K$ spherical Gaussians $\mathcal{N}(\mathbf{e}_k, 0.01\mathbf{I}_{900})$ with probability $p_k = \frac{1}{K}$ for all $k = 1, \ldots, K$, where $\mathbf{e}_k \in \mathbb{R}^{900}$ is the $k$-th column of the identity matrix $\mathbf{I}_{900}$. Detailed information is listed in Table 3. For each $A_i$, we denote $X_{A_i} = \{\mathbf{a}_i - \mathbf{a}_j \mid 1 \leq i < j \leq n\}$, $X_{\mathcal{C}(A_i)} = \{\mathbf{a}^{(\alpha)} - \mathbf{a}^{(\beta)} \mid 1 \leq \alpha < \beta \leq K\}$, and $X_{\mathcal{V}(A_i)} = \cup_{\alpha=1}^{K} \{\mathbf{a}_i - \mathbf{a}_j \mid i, j \in I_\alpha, i \neq j\}$.

According to Assumption 3, for theoretical verification purposes, we set weights $w_{ij}$ as (42) with $\mathcal{E}_{A_i}(k_i, \mathcal{V}(A_i))$, where $k_i \geq 1$ and

$$\mathcal{E}_{A_i}(k_i, \mathcal{V}(A_i)) := \mathcal{E}_{A_i}(k_i) \cup_{\alpha=1}^{K} \{(i,j) \mid i, j \in I_\alpha, i \neq j\}. \tag{43}$$

We summarize the details of $k_i$ in Table 3. With the chosen weights, we can compute the theoretical range $[\gamma_{\min}, \gamma_{\max})$ and the ratio $r$ defined in (3). The values summarized in Table 3 imply that the model (CCM) can perform a perfect cluster recovery on each dataset provided $\gamma \in [\gamma_{\min}, \gamma_{\max})$. We can observe that the ratios $r$ for these seven datasets are large, which further indicates the feasibility of the perfect recovery of the model (RPCCM) with a suitable embedding dimension $m = \lceil C\epsilon^{-2} \log(n) \rceil$ (or $\tilde{m} = \lceil C\epsilon^{-2} \log(K) \rceil$). In this section, we set $C = 2$ by default (see Remark 2 for discussion). We conduct the following numerical experiments for a set of pairs $(\epsilon, m)$:

Table 3: Data information, selected weights, the theoretical range $[\gamma_{\min}, \gamma_{\max})$ and the ratio $r$ for data $A_1$ to $A_7$ using selected weights. $A_1$, $A_2$, $A_3$, and $A_4$ are generated from the same distribution with $K = 8$; $A_1$, $A_5$, $A_6$, and $A_7$ have the same data size $n = 1024$.

| **Data** | $d$ | $n$ | $K$ | $w_{ij}$ | $[\gamma_{\min}, \gamma_{\max})$ | $r$ |
|---|---|---|---|---|---|---|
| $A_4$ | 900 | 128 | 8 | $\mathcal{E}_{A_4}(2, \mathcal{V}(A_4))$ | [0.0061, 2.7329) | 29.9326 |
| $A_2$ | 900 | 256 | 8 | $\mathcal{E}_{A_2}(4, \mathcal{V}(A_2))$ | [0.0409, 1.4287) | 38.3856 |
| $A_3$ | 900 | 512 | 8 | $\mathcal{E}_{A_3}(8, \mathcal{V}(A_3))$ | [0.0229, 0.7330) | 31.9783 |
| $A_1$ | 900 | 1024 | 8 | $\mathcal{E}_{A_1}(16, \mathcal{V}(A_1))$ | [0.0130, 0.4008) | 30.8786 |
| $A_5$ | 900 | 1024 | 4 | $\mathcal{E}_{A_5}(48, \mathcal{V}(A_5))$ | [0.0056, 0.2445) | 43.4597 |
| $A_6$ | 900 | 1024 | 16 | $\mathcal{E}_{A_6}(4, \mathcal{V}(A_6))$ | [0.0277, 0.7843) | 28.3464 |
| $A_7$ | 900 | 1024 | 32 | $\mathcal{E}_{A_7}(2, \mathcal{V}(A_7))$ | [0.0467, 1.2207) | 26.1140 |

1. Verify the robustness of random projections in preserving the pairwise distances of the data points under the given distortion tolerance. 1000 projection matrices will be randomly sampled for verification.

2. Verify the recovery guarantees in Theorem 2 and Theorem 3. We will randomly sample 10 projection matrices for verification. To achieve this goal, we first compute $\hat{\gamma}_{\min}$ and $\hat{\gamma}_{\max}$ using (10) for each projection matrix. The average value of $\hat{\gamma}_{\min}$ and $\hat{\gamma}_{\max}$ over the 10 projections will be also computed. Then, we test the probability that (15) in Theorem 2 (or (20) in Theorem 3) is satisfied. Furthermore, we test the performance of the model (RPCCM) at some specific $\gamma$ in the valid perfect recovery interval. The running time for solving models (CCM) and (RPCCM) will also be presented.

3. The qualities of the clustering paths generated along the sequence of $\gamma \in [3 : -0.01 : 0.01]$ (following the Matlab notation) by models (CCM) and (RPCCM) will be evaluated.

### 5.1.1 Numerical Verification for Theorem 2 on the Model (RPCCM)

We verify the recovery guarantees of Theorem 2 on the dataset $A_1 \in \mathbb{R}^{900\times 1024}$. From Table 3, we can obtain

$$\gamma_{\min} = 0.0130, \quad \gamma_{\max} = 0.4008, \quad C_{12} = \sqrt{3/10}, \quad r = 30.8786.$$

The values $\epsilon_{\min}$ and $\epsilon_{\sup}$ in Theorem 2 are then given by

$$\epsilon_{\min} = 0.1241, \quad \epsilon_{\sup} = \sqrt{10/3},$$

which implies that for any $\epsilon \in [0.1241, \sqrt{10/3}), \gamma \in [(1+\epsilon)0.0130, (1-\sqrt{3/10}\epsilon)0.4008)$, the model (RPCCM) with $m = \lceil 2\epsilon^{-2}\log(1024)\rceil$ can perform a perfect recovery on $A_1$ with high probability. Inspired by the valid interval of $\epsilon$, we choose $\epsilon \in \{0.2, 0.4, 0.6, 0.8, 0.99\}$ for verification. The corresponding embedding dimensions are $m \in \{347, 87, 39, 22, 15\}$. We summarize the numerical results in Table 4, Table 5, Table 6, and Figure 4, respectively. From the results, we can observe that:

1. The random projections with $m = \lceil 2\epsilon^{-2}\log(1024)\rceil$ can robustly preserve norms in the sets $X_{A_1}$, $X_{\mathcal{V}(A_1)}$, and $X_{\mathcal{C}(A_1)}$. Details can be found in Table 4.
2. The model (RPCCM) can robustly perform the perfect cluster recovery on data $A_1$ in the range $[(1+\epsilon)\gamma_{\min}, (1-C_{12}\epsilon)\gamma_{\max})$ in Theorem 2. Details can be found in Table 5 and Figure 4.
3. To further check the clustering performance and compare the running time, we consider the model (RPCCM) at the value of $\gamma = 0.15$ in the valid perfect recovery interval on data $A_1$. The perfect cluster recovery has been observed and the running time for solving (RPCCM) is over 50 times faster than it for solving the model (CCM) at the same $\gamma$. Details can be found in Table 6. It is worthwhile mentioning that, although we compute the weights $w_{ij}$ on the original data, the time for constructing the weights is affordable, compared to the running time for solving the model.
4. We also compare the running time for solving the models on the solution path with $\gamma \in [3 : -0.01 : 0.01]$, which further demonstrate the computational efficiency of (RPCCM). Details can be found in Figure 4(d).

### 5.1.2 Numerical Verification for Theorem 3 on the Model (RPCCM)

In this section, we aim to verify the recovery guarantees in Theorem 3. As mentioned above, we set $\tilde{m} = \lceil 2\epsilon^{-2}\log(K)\rceil$ and test on data $A_1$ to $A_7$ with some suitable $\epsilon \in (0,1)$, which will be specified later.

Let's take data $A_1$ as an example. Using the selected weights in Table 3, we have $\gamma_{\min} = 0.0130$, $\gamma_{\max} = 0.4008$, and $r = 30.8786$. For any $\Pi \in \mathbb{R}^{\tilde{m}\times 900}$ ($\tilde{m} < 900$) defined as (41), the largest singular value $s_1(\Pi)$ is bounded by

$$s_1(\Pi) \le \bar{S}(\tilde{m}, 900, 2) = \frac{30+2}{\sqrt{\tilde{m}}} + 1 \tag{44}$$

Table 4: Performance of random projections on data $A_1$. In the table, $p_{X_{A_1}}$ and $p_{X_{\mathcal{V}(A_1)}}$ represent the probability that norms in $X_{A_1}$ and $X_{\mathcal{V}(A_1)}$ are respectively preserved up to a $(1+\epsilon)$ factor, and $p_{X_{\mathcal{C}(A_1)}}$ represents the probability that norms in $X_{\mathcal{C}(A_1)}$ are preserved down to a $(1-C_{12}\epsilon)$ factor.

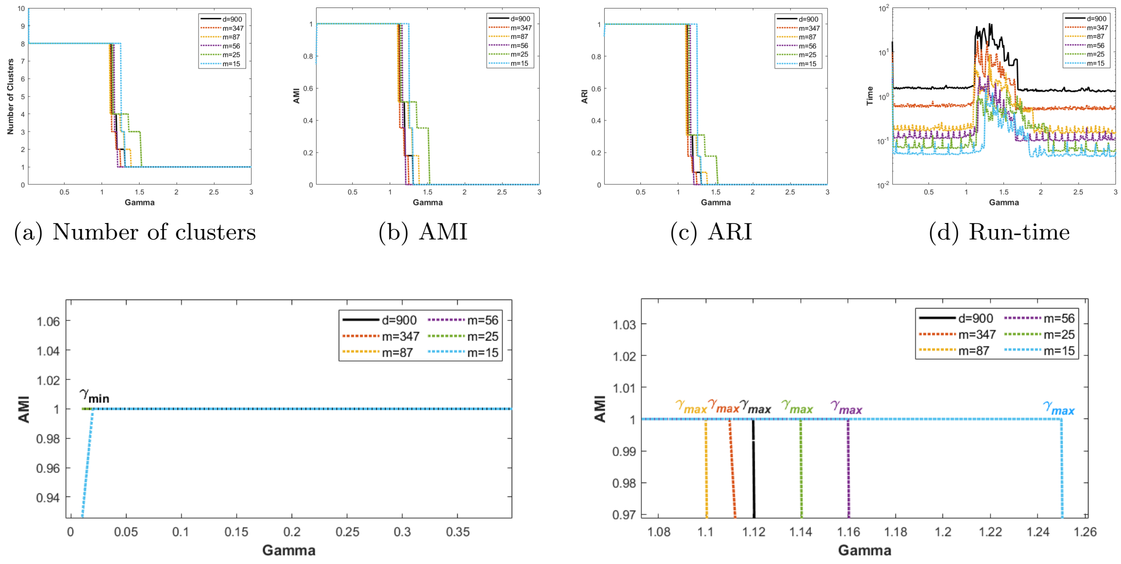
| **Dimension (distortion)** | $p_{X_{A_1}}$ | $p_{X_{\mathcal{V}(A_1)}}$ | $p_{X_{\mathcal{C}(A_1)}}$ |
|---|---|---|---|
| $m = 347$ $(\epsilon = 0.20)$ | 935/1000 | 990/1000 | 959/1000 |
| $m = 87$ $(\epsilon = 0.40)$ | 903/1000 | 992/1000 | 956/1000 |
| $m = 56$ $(\epsilon = 0.60)$ | 863/1000 | 981/1000 | 970/1000 |
| $m = 25$ $(\epsilon = 0.80)$ | 848/1000 | 975/1000 | 957/1000 |
| $m = 15$ $(\epsilon = 0.99)$ | 894/1000 | 991/1000 | 974/1000 |

Table 5: Estimated valid intervals for perfect recovery of the model (RPCCM) on data $A_1$. In the table, $p_\gamma$ denotes the probability that (15) in Theorem 2 is satisfied. The values of $\hat{\gamma}_{\min}$ and $\hat{\gamma}_{\max}$ are averaged over 10 projections.

| **Dimension (distortion)** | $[(1+\epsilon)\gamma_{\min}, (1-C_{12}\epsilon)\gamma_{\max})$ | $p_\gamma$ | $[\hat{\gamma}_{\min}, \hat{\gamma}_{\max})$ |
|---|---|---|---|
| $m = 347$ $(\epsilon = 0.20)$ | [0.0156, 0.3569) | 10/10 | [0.0133, 0.3924) |
| $m = 87$ $(\epsilon = 0.40)$ | [0.0182, 0.3130) | 10/10 | [0.0145, 0.3792) |
| $m = 56$ $(\epsilon = 0.60)$ | [0.0208, 0.2691) | 10/10 | [0.0151, 0.3651) |
| $m = 25$ $(\epsilon = 0.80)$ | [0.0234, 0.2252) | 10/10 | [0.0171, 0.3574) |
| $m = 15$ $(\epsilon = 0.99)$ | [0.0258, 0.1834) | 10/10 | [0.0182, 0.3337) |

Table 6: Performance of models (CCM) and (RPCCM) at $\gamma = 0.15$ on data $A_1$. In the table, $T_{w_{ij}}$ denotes the time for constructing the weights, $T_\Pi$ denotes the time for obtaining the embedded data, and $T_\gamma$ denotes the run-time for solving models at a specific $\gamma$.

| **Dimension** | $\gamma$ | **Perfect Recovery at** $\gamma$ | $T_{w_{ij}}$ | $T_\Pi$ | $T_\gamma$ |
|---|---|---|---|---|---|
| $d = 900$ | 0.15 | 1/1 | 0.2120 | / | 21.9280 |
| $m = 347$ | | 10/10 | | 0.0085 | 7.6150 |
| $m = 87$ | | 10/10 | | 0.0040 | 2.0720 |
| $m = 56$ | | 10/10 | | 0.0034 | 1.4750 |
| $m = 25$ | | 10/10 | | 0.0026 | 0.6410 |
| $m = 15$ | | 10/10 | | 0.0021 | 0.4310 |



(a) Number of clusters

(b) AMI

(c) ARI

(d) Run-time

(e) Practical lower bound for perfect recovery

(f) Practical upper bound for perfect recovery

Figure 4: Performance of the clustering paths generated by models (CCM) and (RPCCM) for data $A_1$ along $\gamma \in [3 : -0.01 : 0.01]$. Notably, the perfect recovery is indicated by ARI=1 (AMI=1). It is possible to recover the “correct” number of clusters while incorrectly partitioning the data.

with a probability over $1 - 2\exp(-2^2) = 0.9634$. Based on the above information, the values $\tilde{\epsilon}_{\min}$ and $\tilde{\epsilon}_{\sup}$ in Theorem 3 are determined as follows:

$$\tilde{\epsilon}_{\min} = 0.0680, \quad \tilde{\epsilon}_{\max} = 0.6416.$$

This implies that for any $\epsilon \in [0.0680, 0.6416)$, and $\gamma \in [\bar{S}(\tilde{m}, 900, 2)\gamma_{\min}, (1-\epsilon)\gamma_{\max})$, the model (RPCCM) with $\tilde{m} = \lceil 2\epsilon^{-2}\log(8)\rceil$ can achieve a perfect cluster recovery on data $A_1$ with high probability. For example, if one takes $\epsilon = 0.6$, then the embedding dimension can be reduced from $d = 900$ to $\tilde{m} = 12$ for data $A_1$.

The valid intervals of $[\tilde{\epsilon}_{\min}, \tilde{\epsilon}_{\max})$ for the model (RPCCM) on data $A_1$ to $A_7$ using our chosen weights are summarized in Table 7. Based on these valid intervals, we set $\epsilon = 0.60$ for all seven datasets. Recall that data $A_1$, $A_2$, $A_3$, and $A_4$ have $K = 8$ clusters, while data $A_5$, $A_6$, and $A_7$ have $K = 4, 12, 16$ clusters, respectively. In other words, we test the model (RPCCM) with an identical embedding dimension $\tilde{m} = 12$ on data $A_1$ to $A_4$, and test with $\tilde{m} = 8, 16, 20$ on data $A_5$ to $A_7$, respectively.

We summarize the numerical results in Table 8, Table 9, Table 10, and Figure 5, respectively. From the results, we can observe:

1. The numerical results in Table 8 demonstrate that random projections with $\tilde{m} = \lceil 2\epsilon^{-2}\log(8)\rceil$ are robust in preserving the norms in $X_{\mathcal{C}(A_1)}$ and bounding the spectral norm of the random projection.

2. The numerical results in Table 9 and Figure 5 show that the model (RPCCM) can robustly perform a perfect cluster recovery for data $A_1$ to $A_7$ in the range $\big[\bar{S}(\tilde{m}, d, 2)\gamma_{\min}, (1-\epsilon)\gamma_{\max}\big)$ in Theorem 3.

3. Similar to the $O(\epsilon^{-2}\log(n))$ case presented in the previous subsection, the model (RPCCM) can perform perfect cluster recovery at some specific value of $\gamma$ in the valid perfect recovery interval for datasets $A_1$ to $A_7$, and the computational cost is effectively reduced. Details can be found in Table 10.

4. The running time for solving the models on the solution path with $\gamma \in [3 : -0.01 : 0.01]$ further demonstrates the computational efficiency of (RPCCM). Details can be found in Figure 5.

Furthermore, numerical experiments in this section demonstrate that the embedding dimension of the model (RPCCM) can be $O(\epsilon^{-2}\log(K))$, based on the following observations:

1. For data $A_1$ to $A_4$, where the number of clusters is fixed as $K = 8$ but the number of data points varies ($n = 1024, 256, 512, 128$), the model (RPCCM) with the same value of $\tilde{m} = 12$ ($\tilde{m} = \lceil 2\epsilon^{-2}\log(K)\rceil$ with $\epsilon = 0.60$) achieves favorable performance. This finding suggests that the embedding dimension can be independent of $n$.

2. For data $A_5$, $A_1$, $A_6$, and $A_7$, where the number of clusters increases ($K = 4, 8, 16, 32$) while the size is fixed as $n = 1024$, the model (RPCCM) with increasing embedding dimensions $\tilde{m} = 8, 12, 16, 20$ ($\tilde{m} = \lceil 2\epsilon^{-2}\log(K)\rceil$ with $\epsilon = 0.60$) is effective. Thus, the result aligns with the theoretical claim of a $O(\log(K))$ embedding dimension.

Table 7: Valid ranges $[\tilde{\epsilon}_{\min}, \tilde{\epsilon}_{\max})$ for the model (RPCCM) on data $A_1$ to $A_7$.

| **Data** | $[\gamma_{\min}, \gamma_{\max})$ | $r$ | $[\tilde{\epsilon}_{\min}, \tilde{\epsilon}_{\max})$ |
|---|---|---|---|
| $A_1$ | [0.0130, 0.4008) | 30.8786 | [0.0680, **0.6416**) |
| $A_2$ | [0.0409, 1.4287) | 38.3856 | [0.0680, **0.6913**) |
| $A_3$ | [0.0229, 0.7330) | 31.9783 | [0.0680, **0.6499**) |
| $A_4$ | [0.0061, 2.7329) | 29.9326 | [0.0680, **0.6342**) |
| $A_5$ | [0.0056, 0.2445) | 43.4597 | [0.0555, **0.6774**) |
| $A_6$ | [0.0277, 0.7843) | 28.3464 | [0.0785, **0.6521**) |
| $A_7$ | [0.0467, 1.2207) | 26.1140 | [0.0878, **0.6563**) |

Table 8: Performance of random projections on data $A_1$ to $A_7$. In the table, $p_{S_1}$ represents the probability that $s_1(\Pi) \leq \bar{S}(\tilde{m}, 900, 2)$, and $p_{X_{\mathcal{C}(A_i)}}$ represents the probability that norms in $X_{\mathcal{C}(A_i)}$ are preserved down to a $(1-\epsilon)$ factor for each dataset.

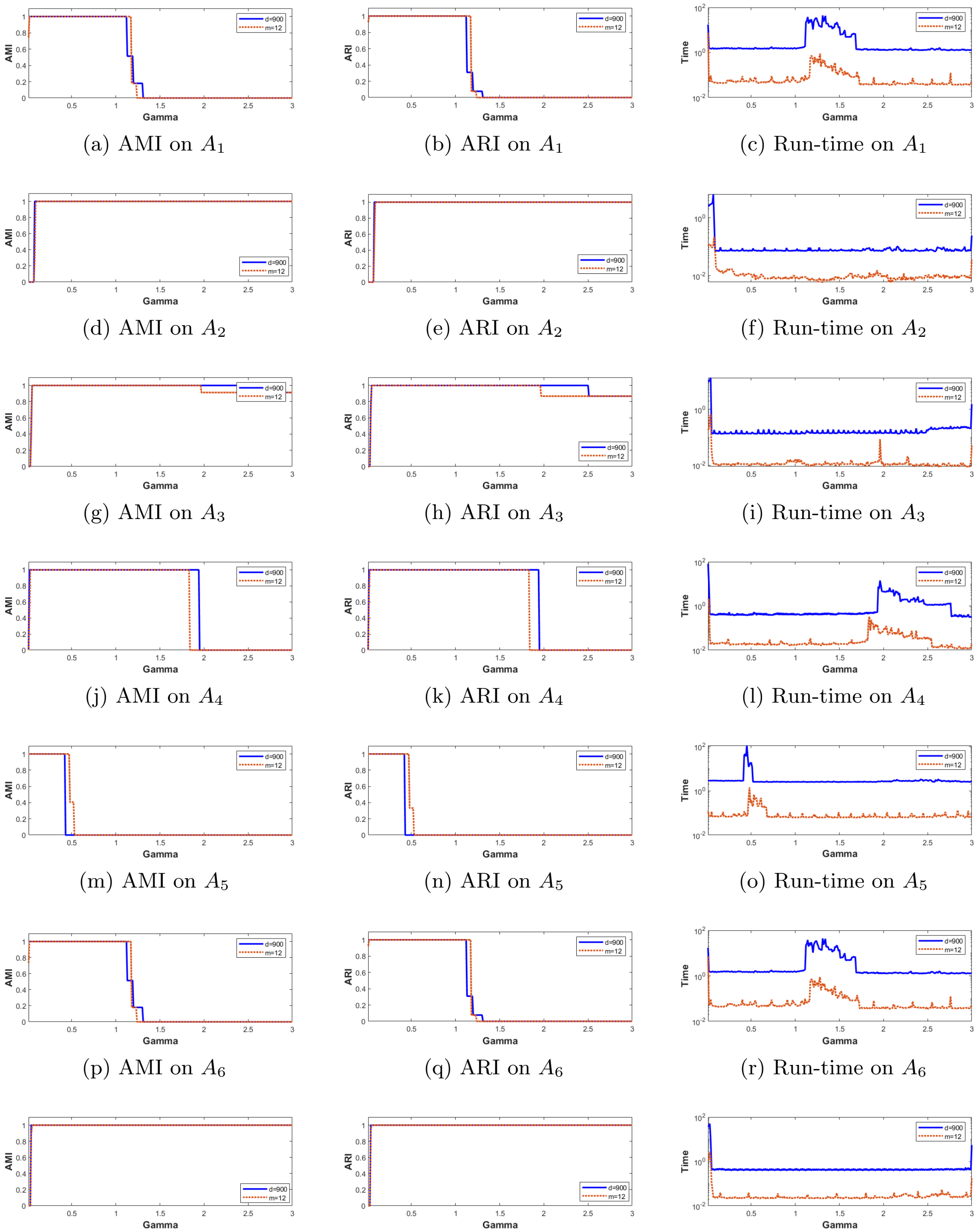
| **Data** | **Dimension (distortion)** | $p_{S_1}$ | $p_{X_{\mathcal{C}(A_i)}}$ |
|---|---|---|---|
| $A_1$ | $\tilde{m} = 12 \quad (\epsilon = 0.60)$ | 1000/1000 | 993/1000 |
| $A_2$ | | | 987/1000 |
| $A_3$ | | | 996/1000 |
| $A_4$ | | | 991/1000 |
| $A_5$ | $\tilde{m} = 8 \quad (\epsilon = 0.60)$ | 1000/1000 | 954/1000 |
| $A_6$ | $\tilde{m} = 16 \quad (\epsilon = 0.60)$ | 1000/1000 | 999/1000 |
| $A_7$ | $\tilde{m} = 20 \quad (\epsilon = 0.60)$ | 1000/1000 | 1000/1000 |

Table 9: Estimated ranges for perfect recovery of the model (RPCCM) for data $A_1$ to $A_7$. In the table, $p_\gamma$ denotes the probability that (20) in Theorem 3 is satisfied. The values of $\hat{\gamma}_{\min}$ and $\hat{\gamma}_{\max}$ are averaged over 10 projections.

| **Data** | **Dimension (distortion)** | $\big[\bar{S}(\tilde{m}, 900, 2)\gamma_{\min}, (1-\epsilon)\gamma_{\max}\big)$ | $p_\gamma$ | $\big[\hat{\gamma}_{\min}, \hat{\gamma}_{\max}\big)$ |
|---|---|---|---|---|
| $A_1$ | $\tilde{m} = 12$ ($\epsilon = 0.60$) | [0.1458, 0.1603) | 10/10 | [0.0196, 0.3111) |
| $A_2$ | $\tilde{m} = 12$ ($\epsilon = 0.60$) | [0.5254, 0.7178) | 10/10 | [0.0727, 1.5240) |
| $A_3$ | $\tilde{m} = 12$ ($\epsilon = 0.60$) | [0.2576, 0.2932) | 10/10 | [0.0375, 0.5595) |
| $A_4$ | $\tilde{m} = 12$ ($\epsilon = 0.60$) | [1.0260, 1.0932) | 10/10 | [0.1345, 2.0504) |
| $A_5$ | $\tilde{m} = 8$ ($\epsilon = 0.60$) | [0.0749, 0.0978) | 10/10 | [0.0105, 0.1982) |
| $A_6$ | $\tilde{m} = 16$ ($\epsilon = 0.60$) | [0.2767, 0.3137) | 10/10 | [0.0363, 0.6186) |
| $A_7$ | $\tilde{m} = 20$ ($\epsilon = 0.60$) | [0.4280, 0.4883) | 10/10 | [0.0659, 0.9292) |

Table 10: Performance of models (CCM) and (RPCCM) on data $A_1$ to $A_7$. In the table, $T_{w_{ij}}$ denotes the time for constructing the weights, $T_\Pi$ denotes the time for obtaining the embedded data, and $T_\gamma$ denotes the run-time for solving models at a specific $\gamma$.

| **Data** | **Dimension** | $\gamma$ | **Perfect Recovery at** $\gamma$ | $T_{w_{ij}}$ | $T_\Pi$ | $T_\gamma$ |
|---|---|---|---|---|---|---|
| $A_1$ | $d = 900$ | 0.15 | 1/1 | 0.2110 | / | 21.9280 |
| | $\tilde{m} = 12$ | | 10/10 | | 0.0020 | 0.3610 |
| $A_2$ | $d = 900$ | 0.60 | 1/1 | 0.0200 | / | 1.5560 |
| | $\tilde{m} = 12$ | | 10/10 | | 0.0015 | 0.0410 |
| $A_3$ | $d = 900$ | 0.27 | 1/1 | 0.0510 | / | 5.2880 |
| | $\tilde{m} = 12$ | | 10/10 | | 0.0018 | 0.1230 |
| $A_4$ | $d = 900$ | 1.05 | 1/1 | 0.0110 | / | 0.2480 |
| | $\tilde{m} = 12$ | | 10/10 | | 0.0014 | 0.0290 |
| $A_5$ | $d = 900$ | 0.08 | 1/1 | 0.2320 | / | 42.1100 |
| | $\tilde{m} = 8$ | | 10/10 | | 0.0022 | 0.4740 |
| $A_6$ | $d = 900$ | 0.30 | 1/1 | 0.2150 | / | 10.1740 |
| | $\tilde{m} = 16$ | | 10/10 | | 0.0024 | 0.2520 |
| $A_7$ | $d = 900$ | 0.45 | 1/1 | 0.1990 | / | 5.4460 |
| | $\tilde{m} = 20$ | | 10/10 | | 0.0023 | 0.1730 |



(a) AMI on $A_1$

(b) ARI on $A_1$

(c) Run-time on $A_1$

(d) AMI on $A_2$

(e) ARI on $A_2$

(f) Run-time on $A_2$

(g) AMI on $A_3$

(h) ARI on $A_3$

(i) Run-time on $A_3$

(j) AMI on $A_4$

(k) ARI on $A_4$

(l) Run-time on $A_4$

(m) AMI on $A_5$

(n) ARI on $A_5$

(o) Run-time on $A_5$

(p) AMI on $A_6$

(q) ARI on $A_6$

(r) Run-time on $A_6$

(s) AMI on $A_7$

(t) ARI on $A_7$

(u) Run-time on $A_7$

Figure 5: Performance of the clustering paths generated by models (CCM) and (RPCCM) for data $A_1$ to $A_7$ along $\gamma \in [3 : -0.01 : 0.01]$.

### 5.2 Practical Performance of the Model (RPCCM)

In this section, we demonstrate the robust practical performance of the model (RPCCM). We first discuss the settings for the model (RPCCM), and then conduct numerical tests on both simulated and real datasets with varying $(d, n, K)$. Additionally, we provide comparisons with other popular clustering algorithms to further demonstrate the superior performance of the model (RPCCM). We further test the performance of the model (RPCCM) with weights constructed from the embedded data.

#### 5.2.1 Practical Settings for the Model (RPCCM)

We have verified the recovery guarantees of (RPCCM) with specified weights according to the assumptions of the theorems in the previous section. However, it is not practical for us to construct the weights as (43) since the true cluster assignments are unknown. In this section, our focus is to test the performance and robustness of the (RPCCM) model in practice. In particular, we adopt the following settings of the weights and embedding dimensions:

1. We adopt the popular choice of weights constructed by the Gaussian kernel (42) with a $k$-nearest neighbors graph with $k = 5$ (Chi and Lange, 2015; Yuan et al., 2018; Sun et al., 2021).
2. Instead of setting the embedding dimension $m$ based on $O(\epsilon^{-2}\log(n))$ or $O(\epsilon^{-2}\log(K))$ as in the last section, we directly test different scales of embedding dimensions $m \in \{10, 20, 50, 100, 200\}$[1] to demonstrate the robustness of the model.

#### 5.2.2 Practical Performance of the Model (RPCCM) on Simulated Data

First, we evaluate the practical performance of the model (RPCCM) by conducting tests on simulated data. Since we will also conduct extensive experiments on real datasets, we only test on the data $A_1$ constructed in the last section for simplicity.

Following the settings in Section 5.2.1 for model weights and embedding dimensions, we generate clustering paths for models (CCM) and (RPCCM) on $A_1$ with $\gamma \in [20 : -0.1 : 0.1]$. The results are summarized in Figure 7 and Table 11. The results demonstrate that the perfect cluster assignments of $A_1$ are contained in the clustering paths for both models in a notable interval of the parameter $\gamma$. Furthermore, as the embedding dimension decreases, the computational cost can be significantly reduced for (RPCCM) while maintaining satisfactory clustering performances.

#### 5.2.3 Practical Performance of the Model (RPCCM) on real Data

Next, we test the practical performance of the model (RPCCM) on real datasets.

**Datasets:** We conduct tests on the following real datasets: LIBRAS and LIBRAS-6 (Dias et al., 2009a), COIL-20 (Nene et al., 1996), LUNG (Lee et al., 2010), and MNIST (LeCun et al., 1998). We summarize the key information of the datasets in Table 12 and include visualization of the datasets in Figure 6.

[1] For LIBRAS and LIBRAS-6, since the original dimension is 90, we simply test $m \in \{10, 20, 50\}$.

Table 11: Practical performance of models (CCM) and (RPCCM) on data $A_1$ at $\gamma = 1$. The results are averaged over 10 random projections. We use the format "mean ± standard deviation" to report the results of ARI and AMI.

| Model | | ARI | AMI | Time |
|---|---|---|---|---|
| (RP) CCM | $d = 900$ | 1.0000 ± 0.0000 | 1.0000 ± 0.0000 | 2.3540 |
| | $m = 200$ | 1.0000 ± 0.0000 | 1.0000 ± 0.0000 | 0.5455 |
| | $m = 100$ | 1.0000 ± 0.0000 | 1.0000 ± 0.0000 | 0.3091 |
| | $m = 50$ | 1.0000 ± 0.0000 | 1.0000 ± 0.0000 | 0.1874 |
| | $m = 20$ | 1.0000 ± 0.0000 | 1.0000 ± 0.0000 | 0.1031 |
| | $m = 10$ | 1.0000 ± 0.0000 | 1.0000 ± 0.0000 | 0.0810 |

Table 12: Key information of the real datasets.

| Data | Type | $d$ | $n$ | $K$ |
|---|---|---|---|---|
| LIBRAS-6 | time series | 90 | 144 | 6 |
| LIBRAS | time series | 90 | 360 | 15 |
| MNIST | image | 784 | 10000 | 10 |
| COIL-20 | image | 1024 | 1440 | 20 |
| LUNG | gene | 12625 | 56 | 4 |

**Implementation of models (CCM) and (RPCCM):** Using the specified weights and embedding dimensions in Section 5.2.1, we solve the models (CCM) and (RPCCM) along a sequence of values of $\gamma$ as follows.

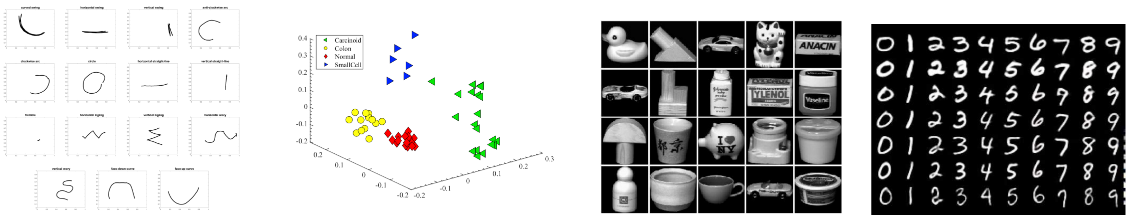
1. For LUNG, we set $\gamma \in [6 : -0.01 : 0.1]$.

2. For COIL-20, we set $\gamma \in [3 : -0.01 : 0.2]$.

3. For MNIST, we set $\gamma \in [2 : -0.01 : 0.5]$.

4. For LIBRAS, we set $\gamma \in [3 : -0.01 : 0.1]$.

5. For LIBRAS-6, we set $\gamma \in [5 : -0.01 : 0.1]$.



(a) LIBRAS (b) LUNG (c) COIL-20 (d) MNIST

Figure 6: Visualization of the real datasets.

**Baselines:** We compare the performance of models (CCM) and (RPCCM) with the following popular clustering algorithms: K-means++ (KM++) (Vassilvitskii and Arthur, 2006), spectral clustering (SC) (Ng et al., 2001; Von Luxburg, 2007), hierarchical complete linkage (CLINK) (Defays, 1977; Manning and Schutze, 1999), hierarchical density-based spatial clustering of applications with noise (HDB) (Campello et al., 2013, 2015), and mean shift (MS) (Fukunaga and Hostetler, 1975; Comaniciu and Meer, 2002). We also compare with their corresponding randomly projected versions applied to the embedded data (say, RP KM++, RP SC, RP CLINK, RP HDB, and RP MS). For comparison fairness, we choose their open-source implementations in MATLAB, and we tune their parameters as follows.

1. (RP) KM++: We use the K-means[2] function with 'Number of clusters'=$K$, where $K$ is the ground-truth number of clusters. We set 'MaxIter'=1000 and 'Replicates'=30.
2. (RP) SC: We use the spectralcluster[3] function with 'Number of clusters'=$K$. We set 'NumNeighbors'=5 and tune 'KernelScale' $\in \{2^{-5}, 2^{-4.5}, \ldots, 2^{4.5}, 2^{5}\}$ to construct the affinity similarity matrix.
3. (RP) CLINK: We use the clusterdata[4] function. We set 'Linkage'='complete' and 'Maxclust'=$K$.
4. (RP) HDB: We use the HDBSCAN[5] function. We set 'Minclustsize'='Minpts' following (Campello et al., 2015) for simplicity and turn 'Minpts' $\in \{2, ..., 100\}$.
5. (RP) MS: We use the MeanShiftCluster[6] function. We set 'BandWidth' $\in \{0.01, ..., 0.99\}$.

**Evaluation criteria:** We use ARI and AMI as the clustering evaluation criteria. For comparison fairness, we run each model 10 times on each input data and report their average performance. For randomly projected (RP) models, we generate 10 independently sampled random projections $\Pi$ and take the embedded data $\Pi A$ as input data. We also present the computational time of each model.

**Results of models (CCM) and (RPCCM):** Table 13 provides a summary of the practical performance of models (CCM) and (RPCCM) on real data. The table includes the AMI and ARI achieved by (CCM), along with the best values of AMI and ARI obtained by (RPCCM) across 10 samples. It also includes the computational time of each model. The following conclusions can be drawn based on the information presented in Table 13:

1. The performance of (RPCCM) is comparable with (CCM) in terms of ARI and AMI, even with an embedding dimension $m = 10$, demonstrating the robustness of (RPCCM).
2. The computational time for solving (RPCCM) has been substantially reduced compared to the running time for solving (CCM).

[2] https://www.mathworks.com/help/stats/k-means-clustering.html
[3] https://ww2.mathworks.cn/help/stats/spectralcluster.html
[4] https://ww2.mathworks.cn/help/stats/clusterdata.html
[5] https://ww2.mathworks.cn/matlabcentral/fileexchange/64864-jorsorokin-HDB
[6] https://ww2.mathworks.cn/matlabcentral/fileexchange/10161-mean-shift-clustering

Table 13: Clustering performance of models (CCM) and (RPCCM). The results are averaged over 10 random projections. We use the format "mean ± standard deviation" to report the results of ARI and AMI.

| **Data** | **Dimension** | **ARI** | **AMI** | **Time** |
|---|---|---|---|---|
| MNIST | $d = 784$ | 0.6468 ± 0.0000 | 0.7054 ± 0.0000 | 1240.4120 |
| | $m = 200$ | 0.6125 ± 0.0350 | 0.7134 ± 0.0249 | 180.3215 |
| | $m = 100$ | 0.6118 ± 0.0374 | 0.7265 ± 0.0315 | 133.4112 |
| | $m = 50$ | 0.6207 ± 0.0399 | 0.7102 ± 0.0232 | 59.2690 |
| | $m = 20$ | 0.6116 ± 0.0519 | 0.7152 ± 0.0416 | 39.5231 |
| | $m = 10$ | 0.6042 ± 0.0547 | 0.7305 ± 0.0242 | 22.2365 |
| COIL-20 | $d = 1024$ | 0.8136 ± 0.0000 | 0.9165 ± 0.0000 | 36.1010 |
| | $m = 200$ | 0.8123 ± 0.0031 | 0.9164 ± 0.0008 | 7.1021 |
| | $m = 100$ | 0.8119 ± 0.0035 | 0.9152 ± 0.0031 | 3.5995 |
| | $m = 50$ | 0.8106 ± 0.0053 | 0.9144 ± 0.0042 | 2.2270 |
| | $m = 20$ | 0.8147 ± 0.0025 | 0.9166 ± 0.0022 | 1.3453 |
| | $m = 10$ | 0.8112 ± 0.0073 | 0.9133 ± 0.0058 | 0.8810 |
| LUNG | $d = 12625$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 3.9890 |
| | $m = 200$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.1521 |
| | $m = 100$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.1115 |
| | $m = 50$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.0863 |
| | $m = 20$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.0693 |
| | $m = 10$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.0588 |
| LIBRAS | $d = 90$ | 0.3767 ± 0.0000 | 0.6119 ± 0.0000 | 0.3550 |
| | $m = 50$ | 0.3817 ± 0.0088 | 0.6158 ± 0.0046 | 0.2781 |
| | $m = 20$ | 0.3740 ± 0.0054 | 0.6125 ± 0.0049 | 0.2510 |
| | $m = 10$ | 0.3651 ± 0.0095 | 0.6093 ± 0.0055 | 0.1460 |
| LIBRAS-6 | $d = 90$ | 0.7674± 0.0000 | 0.8065± 0.0000 | 0.1120 |
| | $m = 50$ | 0.7674 ± 0.0000 | 0.8065 ± 0.0000 | 0.0715 |
| | $m = 20$ | 0.7674 ± 0.0000 | 0.8065 ± 0.0000 | 0.0710 |
| | $m = 10$ | 0.7505 ± 0.0218 | 0.8081 ± 0.0020 | 0.0650 |

To better illustrate the performance of models (CCM) and (RPCCM), for each dataset, we visualize the clustering paths generated by the two models. We also present the performance of the models in terms of ARI and AMI along the clustering paths. The computational time for solving each problem on the path is also presented. We discuss the results on each dataset as follows:

1. MNIST: (CCM) achieves an AMI value of 0.6468 and an ARI value of 0.7054. It can be observed from the results in Figure 8 that the performance of (RPCCM) is comparable to the performance of (CCM) on the clustering paths.

2. COIL-20: (CCM) achieves an AMI value of 0.8136 and an ARI value of 0.9165. Observed from the results from Figure 9, the strong performance of (CCM) is robustly preserved by (RPCCM).

3. LUNG: (CCM) achieves an AMI value of 0.9586 and an ARI value of 0.9426. As observed from Figure 10, (CCM) mislabels only one data point from the Carcinoid cluster. A possible reason is that this wrongly clustered data point is closer to the SmallCell cluster despite being labeled as belonging to the Carcinoid cluster. The performance of (RPCCM) on data LUNG remains robust, as depicted in Figure 10.

4. LIBRAS: (CCM) achieves an AMI value of 0.6119 and an ARI value of 0.3767. A possible reason for the relatively low ARI and AMI is that some clusters in LIBRAS exhibit similar representations (Dias et al., 2009b), which can be observed from Figure

11. The model (RPCCM) is stable in preserving the performance of (CCM). It is worthwhile mentioning that, although these values are relatively low, as shown in Appendix A.9, the model (CCM) is comparable to the spectral clustering model and substantially outperforms other baseline models.

5. LIBRAS-6: (CCM) achieves an AMI value of 0.7674 and an ARI value of 0.8065, which are much higher compared to the values obtained by (CCM) on LIBRAS, because the six selected classes in LIBRAS-6 are more distinguishable. As depicted in Figure 12, (CCM) could successfully classify VSwing, ACarc, Carc, and VWavy, while there are still some overlaps in HLine and HWavy. An explanation is that HLine and HWavy are both horizontal movements, and some samples in the two classes are similar. Moreover, (RPCCM) maintains the performance of (CCM).

In summary, the numerical results presented in this section demonstrate the reliable and efficient clustering performance of the model (RPCCM) compared to the model (CCM), making it valuable for real applications.

**Comparisons with baselines:** We further conduct experiments to compare the clustering performance of the models (CCM) and (RPCCM) with the selected baseline algorithms. The detailed results can be found in the Appendix A.9. The results demonstrate that the performance of other algorithms becomes unreliable as the value of the embedding dimension $m$ decreases. In contrast, the model (RPCCM) exhibits robustness in its performance across different values of $m$. This highlights the superiority of the model (RPCCM) over the baseline algorithms when applied to embedded data with lower dimensions.

#### 5.2.4 Comparison of the Model (RPCCM) with Different Weights: Original Data Weights v.s. Embedded Data Weights

In this section, we further test the performance of the model (RPCCM) with weights constructed from the embedded data. We will then compare the results obtained with those of the model (RPCCM) with weights $w_{ij}$[7] as discussed in Section 5.2.3. In particular, we conduct tests on the high-dimensional data LUNG with $d = 12625$.

Let $B$ denote the embedded data after normalization[8]. We construct weights $\hat{w}_{ij}$ from data $B$ as

$$\hat{w}_{ij} = \begin{cases} \exp(-\left\| B_{:,i} - B_{:,j} \right\|^2 /2) & \text{if } (i,j) \in \mathcal{E}_B(5), \\ 0 & \text{otherwise.} \end{cases} \tag{45}$$

To ensure fair comparisons, we used the identical random projection matrices, as utilized in the tests of (RPCCM) with weights $w_{ij}$ in Section 5.2.3, to assess the performance of (RPCCM) with weights $\hat{w}_{ij}$. The results, summarized in Table 14, reveal the following observations:

1. (RPCCM) with weights $\hat{w}_{ij}$ could still achieve remarkable clustering performance with a modest embedding dimension $m$, e.g., $m \in \{200, 100, 50\}$. However, as $m$ decreases,

[7] Note that $w_{ij}$ are constructed as (42) with a 5-nearest neighbors graph of original data.

[8] Note that normalization in this context refers to dividing the dataset by the maximum Euclidean distance between all pairs of data points.

the model's performance degrades. In contrast, the performance of (RPCCM) with weights $w_{ij}$ remains robust, even for an embedding dimension as low as $m = 10$.

2. The time required for constructing weights $w_{ij}$ is minimal compared to the runtime for solving the model (RPCCM) using these weights. Conversely, although the time for constructing weights $\hat{w}_{ij}$ is significantly reduced, solving the model (RPCCM) with these weights could take more time compared to using the weights $w_{ij}$, especially in the case that $m$ is very low.

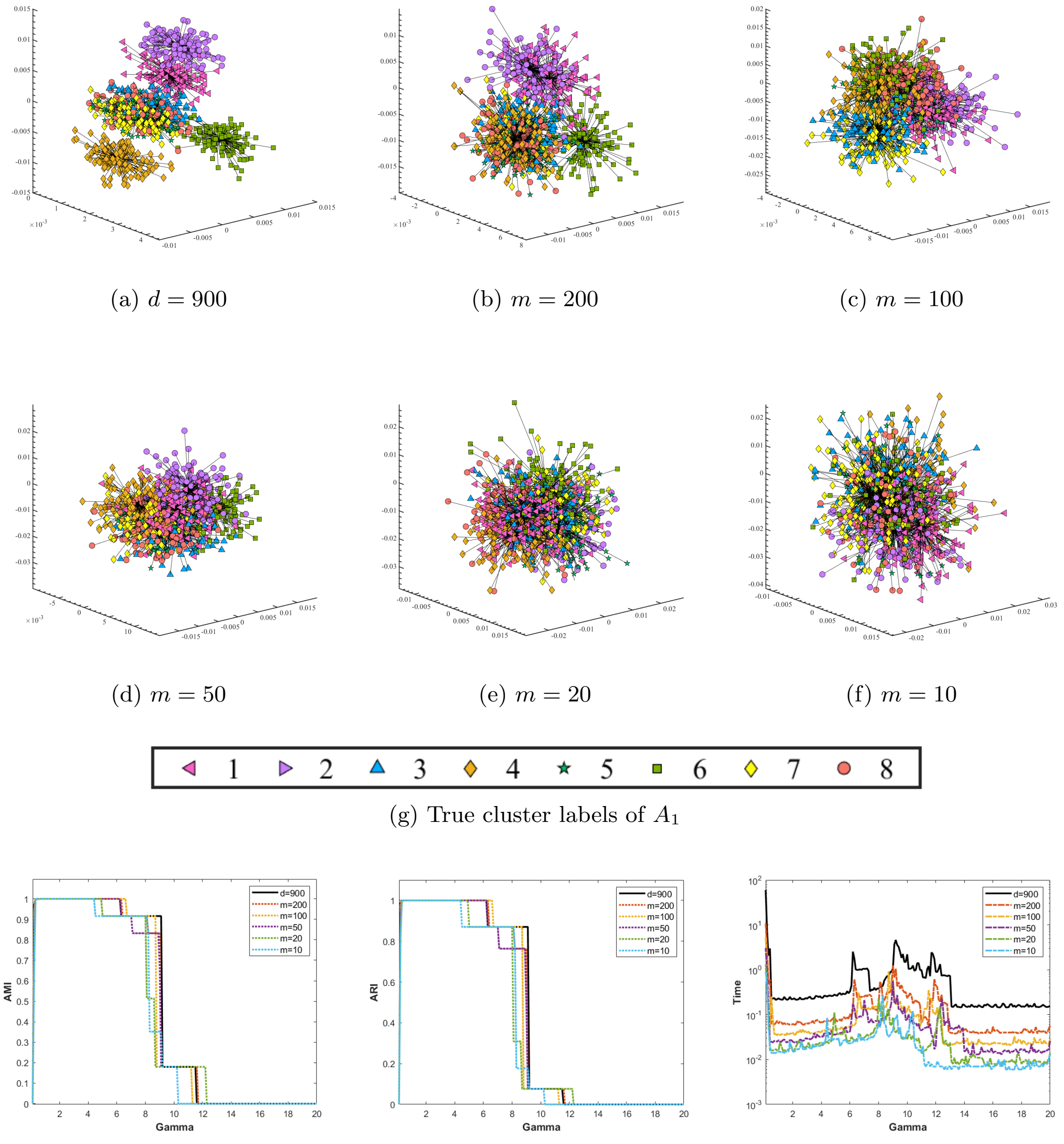
One possible explanation for these findings is that as the embedding dimension $m$ decreases, the distortions in pairwise distances become more noticeable. This leads to significant changes in the clustering structure of the embedded data. Consequently, the model (RPCCM) with weights $\hat{w}_{ij}$ becomes less stable as $m$ decreases. On the other hand, the model (RPCCM) with weights $w_{ij}$ effectively utilizes the clustering structure inherent in the original data points. This allows it to maintain good performance even with a low embedding dimension.

Table 14: Comparison of the model (RPCCM) with weights $w_{ij}$ and $\hat{w}_{ij}$ on data LUNG. The results are averaged over 10 random projections. In the table, $T_w$ denotes the time for constructing the weights, $T_\Pi$ denotes the time for obtaining the embedded data, and $T_\gamma$ denotes the run-time for solving each model at a specific $\gamma$, where the best values of AMI and ARI are achieved. We use the format "mean ± standard deviation" to report the results of ARI and AMI.

| **Model** | | **ARI** | **AMI** | **Time** | $T_w$ | $T_\Pi$ | $T_\gamma$ |
|---|---|---|---|---|---|---|---|
| CCM ($w_{ij}$) | $d = 12625$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 3.9890 | 0.0140 | / | 3.9750 |
| RPCCM ($w_{ij}$) | $m = 200$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.1521 | 0.0140 | 0.0256 | 0.1125 |
| | $m = 100$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.1115 | 0.0140 | 0.0124 | 0.0851 |
| | $m = 50$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.0863 | 0.0140 | 0.0063 | 0.0660 |
| | $m = 20$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.0693 | 0.0140 | 0.0033 | 0.0520 |
| | $m = 10$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.0588 | 0.0140 | 0.0018 | 0.0430 |
| RPCCM ($\hat{w}_{ij}$) | $m = 200$ | 0.9471 ± 0.0555 | 0.9262 ± 0.0707 | 0.2319 | 0.0010 | 0.0256 | 0.2053 |
| | $m = 100$ | 0.9346 ± 0.0670 | 0.9083 ± 0.0750 | 0.1322 | 0.0010 | 0.0124 | 0.1188 |
| | $m = 50$ | 0.9342 ± 0.0658 | 0.8952 ± 0.0819 | 0.1402 | 0.0010 | 0.0063 | 0.1329 |
| | $m = 20$ | 0.8340 ± 0.1025 | 0.7829 ± 0.0911 | 0.0893 | 0.0010 | 0.0033 | 0.0850 |
| | $m = 10$ | 0.7239 ± 0.1337 | 0.6728 ± 0.1239 | 0.1176 | 0.0010 | 0.0018 | 0.1148 |

## 6. Conclusion and Future Works

In this paper, we proposed a randomly projected convex clustering model (RPCCM) for clustering high dimensional data. We proved that, under some mild conditions, the perfect recovery of the cluster membership assignments of the convex clustering model on the original data, if exists, can be preserved by the model (RPCCM) with a much smaller embedding dimension. In particular, we proved that the embedding dimension can be $m = O(\epsilon^{-2}\log(n))$, where $n$ is the number of data points and $\epsilon > 0$ is some given tolerance. We further proved that the embedding dimension can be $m = O(\epsilon^{-2}\log(K))$, where $K$ is the number of hidden clusters, which is independent of the number of data points. Furthermore, we also established the recovery guarantees of our proposed model



(a) $d = 900$

(b) $m = 200$

(c) $m = 100$

(d) $m = 50$

(e) $m = 20$

(f) $m = 10$

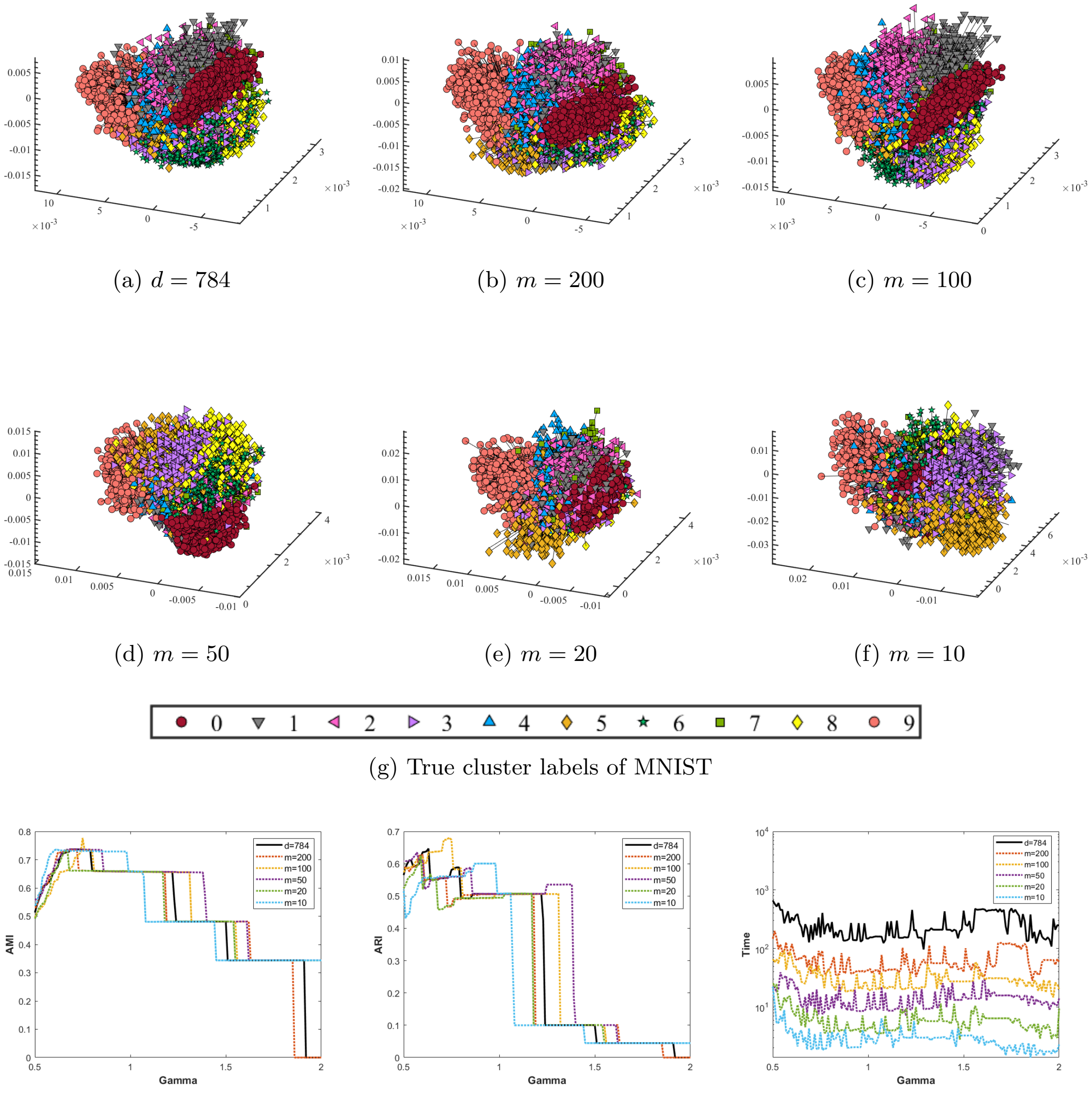
(g) True cluster labels of $A_1$

(h) AMI

(i) ARI

(j) Run-time

Figure 7: Visualization and performance of the clustering paths on data $A_1$.



(a) $d = 784$

(b) $m = 200$

(c) $m = 100$

(d) $m = 50$

(e) $m = 20$

(f) $m = 10$

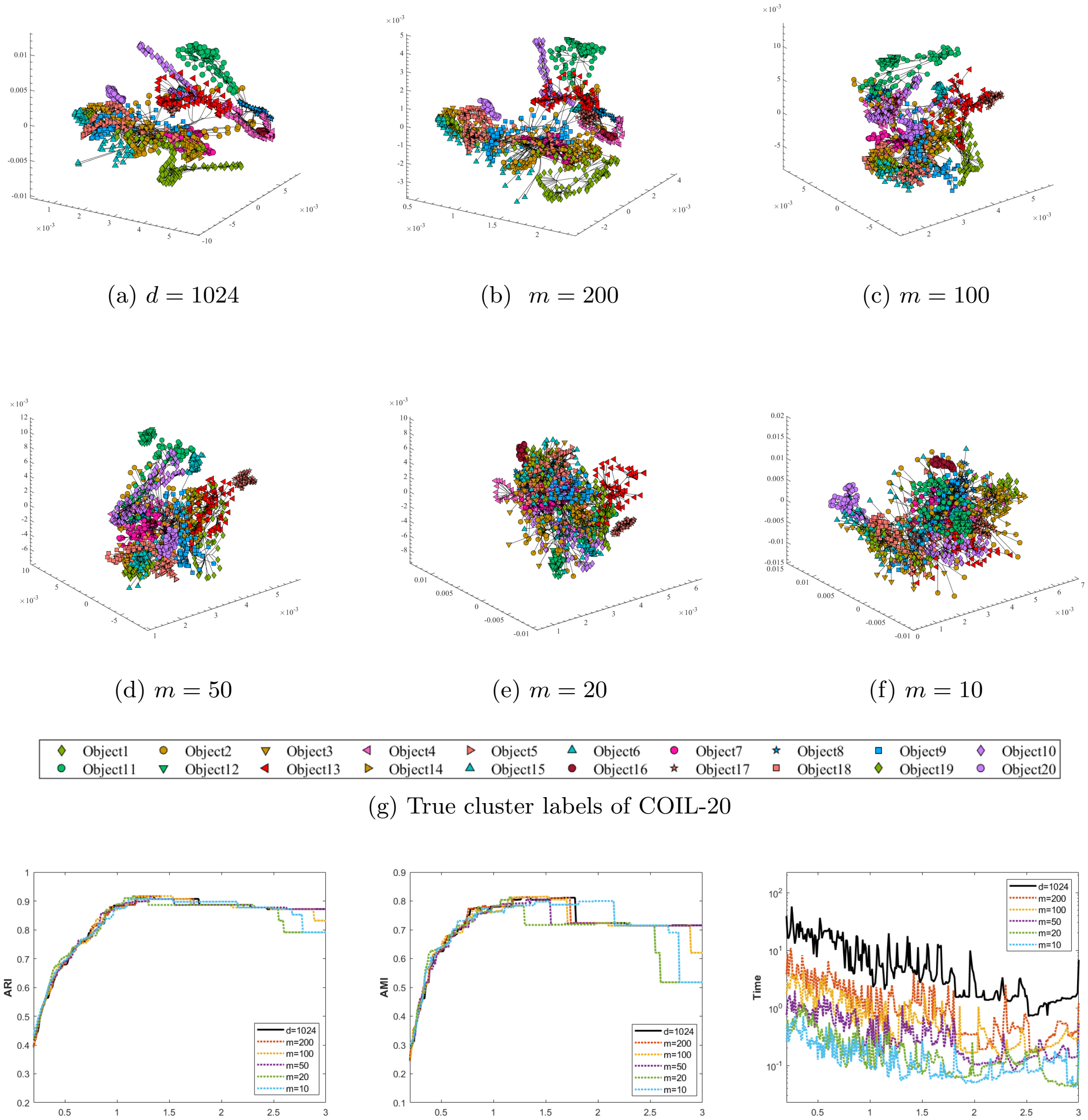
(g) True cluster labels of MNIST

(h) AMI

(i) ARI

(j) Run-time

Figure 8: Visualization and performance of the clustering paths on data MNIST.



(a) $d = 1024$

(b) $m = 200$

(c) $m = 100$

(d) $m = 50$

(e) $m = 20$

(f) $m = 10$

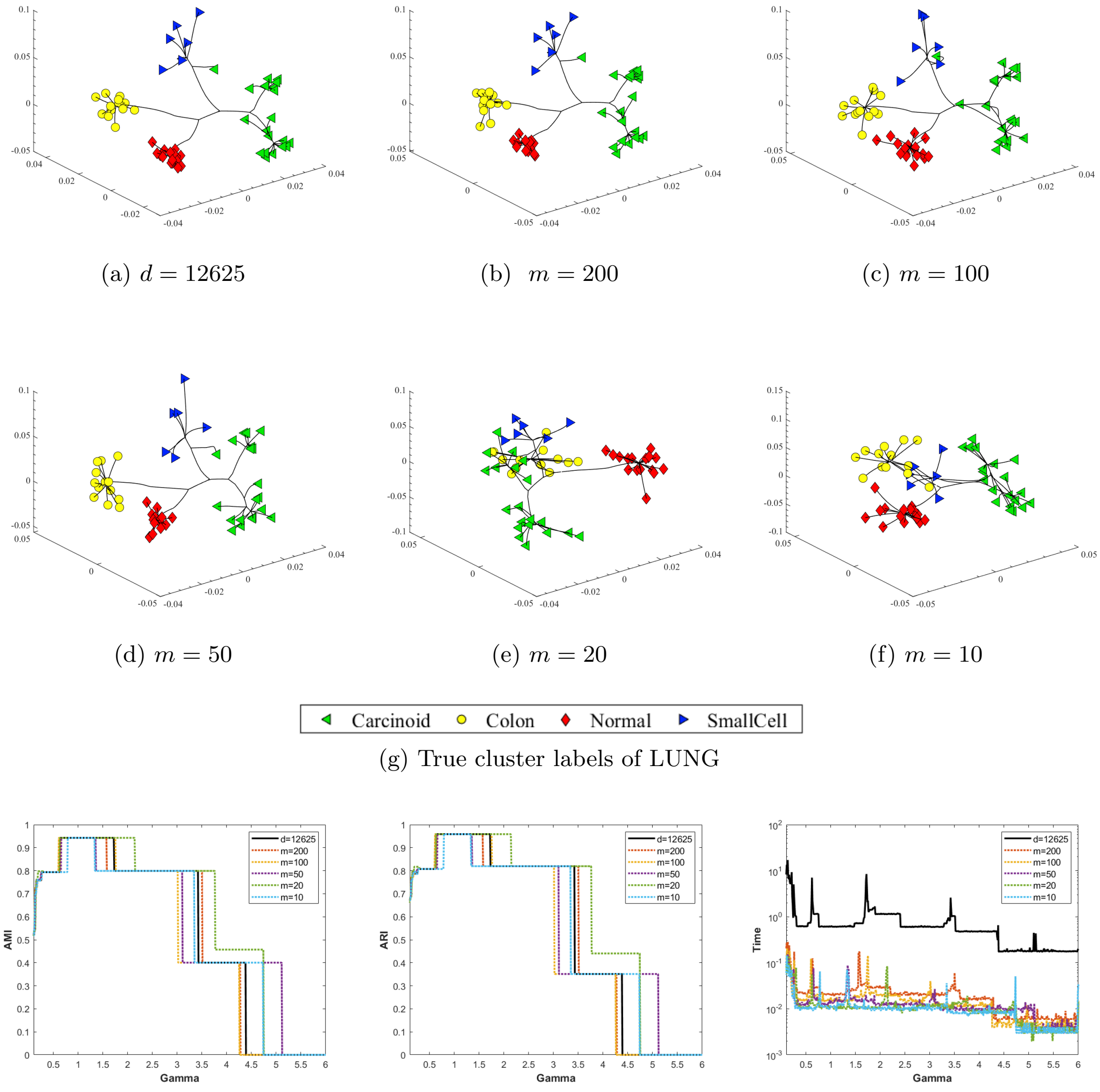
(g) True cluster labels of COIL-20

(h) AMI

(i) ARI

(j) Run-time

Figure 9: Visualization and performance of the clustering paths on data COIL-20.



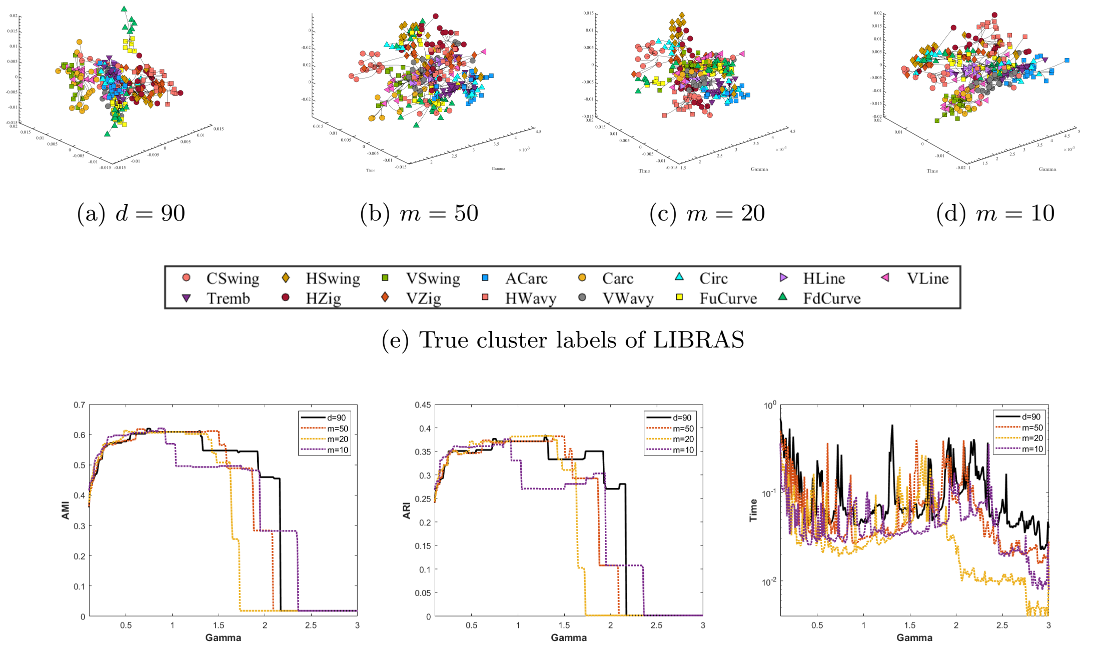
(a) $d = 12625$ (b) $m = 200$ (c) $m = 100$

(d) $m = 50$ (e) $m = 20$ (f) $m = 10$

(g) True cluster labels of LUNG

(h) AMI (i) ARI (j) Run-time

Figure 10: Visualization and performance of the clustering paths on data LUNG.

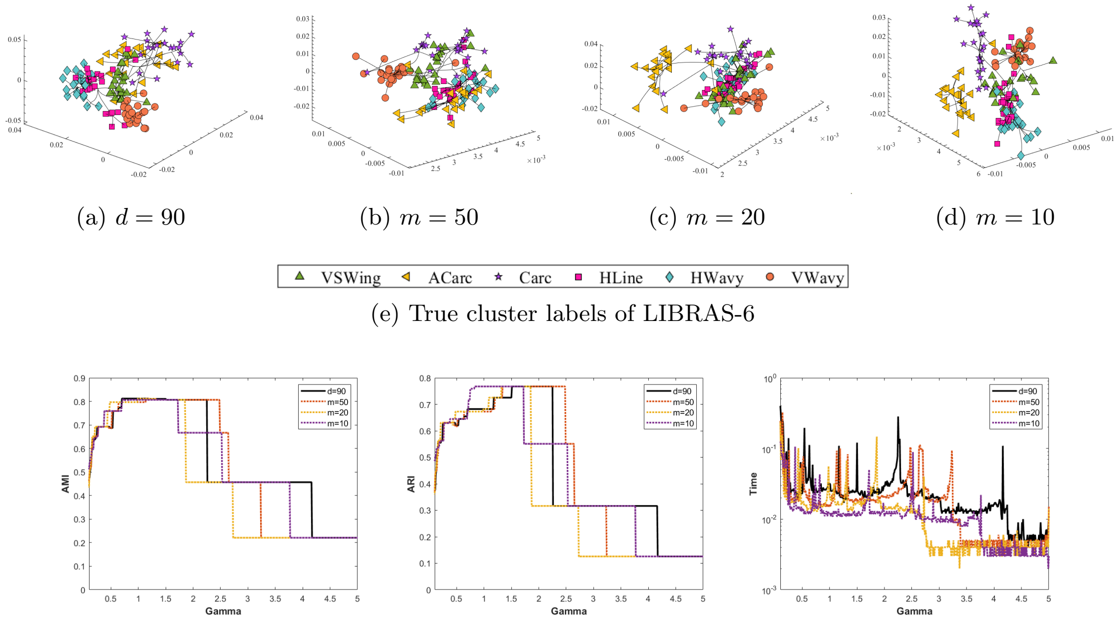


(a) $d = 90$ (b) $m = 50$ (c) $m = 20$ (d) $m = 10$

(e) True cluster labels of LIBRAS

(f) AMI (g) ARI (h) Run-time

Figure 11: Visualization and performance of the clustering paths on data LIBRAS.



(a) $d = 90$ (b) $m = 50$ (c) $m = 20$ (d) $m = 10$

(e) True cluster labels of LIBRAS-6

(f) AMI (g) ARI (h) Run-time

Figure 12: Visualization and performance of the clustering paths on data LIBRAS-6.

with uniform weights for clustering a mixture of spherical Gaussians. Extensive numerical experiment results were presented in this paper to demonstrate the robustness and superior performance of the model (RPCCM). The numerical results presented in this paper also demonstrated that the model (RPCCM) can outperform other popular clustering models on the dimension-reduced data in practice.

It is worthwhile pointing out that the practical performance of the models (CCM) and (RPCCM) depends on the quality of the input data features. We regard it as a future research direction to investigate a new technique that can do dimension reduction and feature representation learning simultaneously. In particular, it is interesting to explore whether other dimensionality reduction methods, such as spectral projection, could enhance the clustering recovery performance of the model (CCM) under some specific problem settings, e.g., the MSG problem setting. Moreover, the choice of the weights $w_{ij}$ is a key to the success of the models (CCM) and (RPCCM). We will further investigate adaptive weights to improve the practical performance and robustness of the models in the future research. Another key challenge for practical implementations of the models (CCM) and (RPCCM) is the tuning of the parameter $\gamma$. We regard developing robust tuning strategies for $\gamma$ as a future research direction.

## Acknowledgments and Disclosure of Funding

The research of Yancheng Yuan is supported in part by The Hong Kong Polytechnic University under grant P0038284. The research of Defeng Sun is supported in part by the Hong Kong Research Grant Council under grant 15304721.

## A. Appendix

### A.1 Proof of Corollary 1

**Proof** Note that the statements of Theorem 2 hold provided events $E_1$ and $E_2$ are satisfied, where $E_1$ is the event that $\Pi$ satisfies (12) and $E_2$ is the event that $\Pi$ satisfies (11b), respectively. Also note that Proposition 1 implies that

$$\mathbb{P}\left[E_2 \mid E_1\right] \geq 1 - \frac{1}{2K^{p_2-2}}\left(\frac{1}{1-\frac{1}{2n^{p_1-2}}}\right). \tag{46}$$

As a result, under the assumptions in Theorem 2, if we assume that the random projection $\Pi$ satisfies (12), the statements of Theorem 2 hold with probability at least

$$1 - \frac{1}{2K^{p_2-2}}\left(\frac{1}{1-\frac{1}{2n^{p_1-2}}}\right).$$

■

### A.2 Proof of Proposition 3

**Proof** Suppose $1 \leq \alpha \leq K$.

1. $\|\mathbf{a}_i - \mathbf{a}_j\| \leq \|\mathbf{a}_i - \boldsymbol{\mu}_\alpha\| + \|\mathbf{a}_j - \boldsymbol{\mu}_\alpha\| \leq 2\theta\boldsymbol{\sigma}_\alpha$, for any $i, j \in \tilde{I}_\alpha(d,\theta)$.

2. By definition of $\tilde{I}_\alpha(d,\theta)$, for any $i \in [n]$, we have

$$\mathbb{P}\left[i \in \tilde{I}_\alpha(d,\theta)\right] = \mathbb{P}\left[\|\mathbf{a}_i - \boldsymbol{\mu}_\alpha\| \leq \theta\boldsymbol{\sigma}_\alpha \mid i \in I_\alpha\right]\mathbb{P}\left[i \in I_\alpha\right] = F(\theta, d)\mathbf{p}_\alpha.$$

Therefore, $\mathbb{E}\left[|\tilde{I}_\alpha(d,\theta)|\right] = F(\theta,d)\mathbf{p}_\alpha n$. For any $\eta > 0$, it follows Hoeffding's inequality (Hoeffding, 1963) for the binomial distribution that (24) holds.

3. For any nonzero $h \in \mathbb{R}^d$, we claim that

$$\begin{aligned}
&\mathbb{P}_{x\sim\mathcal{N}(\boldsymbol{\mu}_\alpha,\boldsymbol{\sigma}_\alpha^2\mathbf{I}_d)}\left[(x-\boldsymbol{\mu}_\alpha)^\top h \leq 0 \mid \|x-\boldsymbol{\mu}_\alpha\| \leq \theta\boldsymbol{\sigma}_\alpha\right]\\
=&\mathbb{P}_{x\sim\mathcal{N}(\boldsymbol{\mu}_\alpha,\boldsymbol{\sigma}_\alpha^2\mathbf{I}_d)}\left[(x-\boldsymbol{\mu}_\alpha)^\top h \geq 0 \mid \|x-\boldsymbol{\mu}_\alpha\| \leq \theta\boldsymbol{\sigma}_\alpha\right] \qquad (47)\\
=&1/2.
\end{aligned}$$

In fact, let $y := (x-\boldsymbol{\mu}_\alpha)^\top h$, then $y \sim \mathcal{N}(0, \boldsymbol{\sigma}_\alpha^2\|h\|^2)$ provided $x \sim \mathcal{N}(\boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha^2\mathbf{I}_d)$. Moreover, let $S := \left\{y = (x-\boldsymbol{\mu}_\alpha)^\top h \in \mathbb{R} \mid \|x-\boldsymbol{\mu}_\alpha\| \leq \theta\boldsymbol{\sigma}_\alpha\right\}$, then $S$ is symmetric w.r.t. 0. Thus, (47) holds by the fact that the density function of a Gaussian distribution is symmetric around its mean.

Moreover, by the definition of $\tilde{I}_\alpha(d,\theta)$, we have that

$$\mathbb{P}\left[(\mathbf{a}_i-\boldsymbol{\mu}_\alpha)^\top h \leq 0 \mid i \in \tilde{I}_\alpha(d,\theta)\right] = \mathbb{P}\left[(\mathbf{a}_i-\boldsymbol{\mu}_\alpha)^\top h \geq 0 \mid i \in \tilde{I}_\alpha(d,\theta)\right] = 1/2.$$

Taking the union bound over $i \in \tilde{I}_\alpha(d,\theta)$, we have that for any $n \geq \tilde{n} \geq 1$,

$$\mathbb{P}\left[\exists i \in \tilde{I}_\alpha(d,\theta) \text{ s.t. } (\mathbf{a}_i-\boldsymbol{\mu}_\alpha)^\top h \geq 0 \mid |\tilde{I}_\alpha(d,\theta)| \geq \tilde{n}\right] \geq 1 - 2^{-\tilde{n}}.$$

Particularly, for any $\eta > 0$, let $\tilde{n}_\alpha := (F(\theta,d)-\eta)\,\mathbf{p}_\alpha n$, the Bayes' Theorem tells that

$$\begin{aligned}
&\mathbb{P}\left[|\tilde{I}_\alpha(d,\theta)| \geq \tilde{n}_\alpha, \text{ and } \exists i \in \tilde{I}_\alpha(d,\theta) \text{ s.t. } (\mathbf{a}_i-\boldsymbol{\mu}_\alpha)^\top h \geq 0\right]\\
\geq&\mathbb{P}\left[\exists i \in \tilde{I}_\alpha(d,\theta) \text{ s.t. } (\mathbf{a}_i-\boldsymbol{\mu}_\alpha)^\top h \geq 0 \mid |\tilde{I}_\alpha(d,\theta)| \geq \tilde{n}_\alpha\right]\mathbb{P}\left[|\tilde{I}_\alpha(d,\theta)| \geq \tilde{n}_\alpha\right] \qquad (48)\\
\geq&\left(1 - 2^{-(F(\theta,d)-\eta)\mathbf{p}_\alpha n}\right)\left(1-\exp\left(-2\mathbf{p}_\alpha^2\eta^2 n\right)\right).
\end{aligned}$$

■

### A.3 Proof of Proposition 4

**Proof** Let $\tilde{n}_\alpha = (F(\theta,d)-\eta)\,\mathbf{p}_\alpha n, \alpha \in [K]$. We claim here that it suffices to show the following statement: For any given $\alpha \in [K]$, with probability over

$$\left(1-(K-1)\,2^{-\tilde{n}_\alpha}\right)\left(1-\exp\left(-2\mathbf{p}_\alpha^2\eta^2 n\right)\right),$$

we have $|\tilde{I}_\alpha(d,\theta)| \geq \tilde{n}_\alpha$, and for any $\beta \neq \alpha$, there exists $i \in \tilde{I}_\alpha(d,\theta)$ such that

$$(\mathbf{a}_i - \boldsymbol{\mu}_\alpha)^\top \Pi^\top \Pi \left(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right) \geq 0.$$

In fact, if the statement being claimed is true, then by taking the union bound over $\alpha \in [K]$, we have that with probability over $1 - P_G$, for any $1 \leq \alpha \neq \beta \leq K$,

$$|\tilde{I}_\alpha(d,\theta)| \geq \tilde{n}_\alpha, \tag{49}$$

and there exists $i \in \tilde{I}_\alpha(d,\theta), i' \in \tilde{I}_\beta(d,\theta)$ such that

$$\begin{aligned}(\mathbf{a}_i - \boldsymbol{\mu}_\alpha)^\top \Pi^\top \Pi \left(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right) \geq 0,\\ \left(\mathbf{a}_{i'} - \boldsymbol{\mu}_\beta\right)^\top \Pi^\top \Pi \left(\boldsymbol{\mu}_\beta - \boldsymbol{\mu}_\alpha\right) \geq 0,\end{aligned}$$

which further implies that

$$\begin{aligned}\|\Pi\left(\mathbf{a}_i - \mathbf{a}_{i'}\right)\|^2 &= \left\|\Pi\left(\mathbf{a}_i - \boldsymbol{\mu}_\alpha - \mathbf{a}_{i'} + \boldsymbol{\mu}_\beta + \boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right)\right\|^2\\ &= \left\|\Pi\left(\mathbf{a}_i - \boldsymbol{\mu}_\alpha - \mathbf{a}_{i'} + \boldsymbol{\mu}_\beta\right)\right\|^2 + 2\left(\mathbf{a}_i - \boldsymbol{\mu}_\alpha\right)^\top \Pi^\top \Pi \left(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right)\\ &\quad + 2\left(\mathbf{a}_{i'} - \boldsymbol{\mu}_\beta\right)^\top \Pi^\top \Pi \left(\boldsymbol{\mu}_\beta - \boldsymbol{\mu}_\alpha\right) + \left\|\Pi\left(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right)\right\|^2\\ &\geq \left\|\Pi\left(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right)\right\|^2.\end{aligned} \tag{50}$$

Combining (49), (50), and the triangular inequality, we have that for any $1 \leq \alpha \neq \beta \leq K$,

$$\begin{aligned}\max_{i,j \in \tilde{I}_\alpha(d,\theta)} \frac{\|\Pi(\mathbf{a}_i - \boldsymbol{\mu}_\alpha)\| + \|\Pi(\mathbf{a}_j - \boldsymbol{\mu}_\alpha)\|}{\left(F(\theta,d) - \eta\right)\boldsymbol{p}_\alpha n} &\geq \max_{i,j \in \tilde{I}_\alpha(d,\theta)} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_j)\|}{|\tilde{I}_\alpha(d,\theta)|},\\ \max_{\substack{i \in \tilde{I}_\alpha(d,\theta)\\ i' \in \tilde{I}_\beta(d,\theta)}} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_{i'})\|}{2(n-1)} &\geq \frac{\left\|\Pi\left(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right)\right\|}{2(n-1)},\end{aligned} \tag{51}$$

which implies (32).

Now, we prove the statement being claimed. Fix any given $\alpha \in [K]$. For any $\beta \in [K], \beta \neq \alpha$, let $h_{\alpha,\beta} := \Pi^\top \Pi \left(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right) \in \mathbb{R}^d$, and let $E_{\alpha,\beta}$ denote the event that $\exists i \in \tilde{I}_\alpha(d,\theta)$ s.t. $\left(\mathbf{a}_i - \boldsymbol{\mu}_\alpha\right)^\top h_{\alpha,\beta} \geq 0$. Under the assumption that $\Pi\boldsymbol{\mu}_1, \ldots, \Pi\boldsymbol{\mu}_K \in \mathbb{R}^m$ are all distinct, we have $\|\Pi(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta)\|^2 = (\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta)^\top \Pi^\top \Pi (\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta) > 0$, which implies that $h_{\alpha,\beta}$ is nonzero. Therefore, by Proposition (3), we have that

$$\mathbb{P}\left[E_{\alpha,\beta} \mid |\tilde{I}_\alpha(d,\theta)| \geq \tilde{n}_\alpha\right] \geq 1 - 2^{-\tilde{n}_\alpha}. \tag{52}$$

Taking the union bound over $\beta \in [K], \beta \neq \alpha$, we have that

$$\mathbb{P}\left[\cap_{\beta \in [K], \beta \neq \alpha} E_{\alpha,\beta} \mid |\tilde{I}_\alpha(d,\theta)| \geq \tilde{n}_\alpha\right] \geq 1 - (K-1)2^{-\tilde{n}_\alpha}. \tag{53}$$

Thus, it follows the Bayes' Theorem that with probability over

$$\left(1 - (K-1)\,2^{-\tilde{n}_\alpha}\right)\left(1 - \exp\left(-2\mathbf{p}_\alpha^2 \eta^2 n\right)\right),$$

$|\tilde{I}_\alpha(d,\theta)| \geq \tilde{n}_\alpha$, and for any $\beta \neq \alpha$, there exists $i \in \tilde{I}_\alpha(d,\theta)$ such that

$$(\mathbf{a}_i - \boldsymbol{\mu}_\alpha)^\top \Pi^\top \Pi \left(\boldsymbol{\mu}_\alpha - \boldsymbol{\mu}_\beta\right) \geq 0.$$

■

### A.4 Proof of Proposition 5

**Proof** If $\hat{\gamma}^G_{\min} < \hat{\gamma}^G_{\max}$, by Proposition (4), with probability over $1 - P_G$, we have that

$$\max_{\alpha\in[K]} \max_{i,j\in\tilde{I}_\alpha(d,\theta)} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_j)\|}{|\tilde{I}_\alpha(d,\theta)|} \leq \hat{\gamma}^G_{\min} < \hat{\gamma}^G_{\max} \leq \min_{1\leq\alpha<\beta\leq K} \max_{\substack{i\in\tilde{I}_\alpha(d,\theta)\\ i'\in\tilde{I}_\beta(d,\theta)}} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_{i'})\|}{2(n-1)}, \tag{54}$$

which implies that for any $\gamma \in \left[\hat{\gamma}^G_{\min}, \hat{\gamma}^G_{\max}\right)$, for any $1 \leq \alpha < \beta \leq K$,

$$\max_{l\in\{\alpha,\beta\}} \max_{i,j\in\tilde{I}_l(d,\theta)} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_j)\|}{|\tilde{I}_l(d,\theta)|} \leq \gamma < \max_{\substack{i\in\tilde{I}_\alpha(d,\theta)\\ i'\in\tilde{I}_\beta(d,\theta)}} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_{i'})\|}{2(n-1)}.$$

Applying Lemma 5 to the embedded data $\Pi A$ implies that points indexed by $\tilde{I}_\alpha(d,\theta), \alpha \in [K]$ are correctly labeled. ■

### A.5 Proof of Proposition 6

**Proof** On the one hand, for any $\epsilon \in (0, 1/C_{12})$ and $\delta > 0$, let $\Pi \sim \mathcal{D}_{\epsilon,\delta}$ with $m \geq C\epsilon^{-2}\log(1/\delta)$, the failure probability of (34a) is at most $|X_G|\delta$. Note that

$$|X_G| = \sum_{\alpha=1}^{K} |\tilde{I}_\alpha(\theta, d)| \leq n.$$

If we take $\delta = \frac{1}{n^{p_1}}$ and $m \geq C\epsilon^{-2}\log(1/\delta) = p_1 C\epsilon^{-2}\log(n)$, where $p_1 > 1$, the failure probability of (34a) is at most

$$|X_G|\delta \leq \frac{1}{n^{p_1-1}}.$$

On the other hand, let $\delta_2 = \frac{1}{K^{p_2}}$ and $\epsilon_2 = C_{12}\epsilon$, where $p_2 > 2$, then $0 < \epsilon_2 < \min\{\epsilon, 1\}$, and we have $p_1 C\epsilon^{-2}\log(n) = p_2 C\epsilon_2^{-2}\log(K) = C\epsilon_2^{-2}\log(1/\delta_2)$. As a result, if we take $m \geq p_1 C\epsilon^{-2}\log(n)$, the probability that (34b) fails is at most

$$|X_{\boldsymbol{\mu}}|\frac{1}{K^{p_2}} = C(K,2)\frac{1}{K^{p_2}} < \frac{1}{2K^{p_2-2}}.$$

Taking a union bound, the probability that conditions (34) are satisfied is at least

$$1 - |X_G|\delta - |X_{\boldsymbol{\mu}}|\delta_2 > 1 - \frac{1}{n^{p_1-1}} - \frac{1}{2K^{p_2-2}}.$$

■

### A.6 Proof of Theorem 4

**Proof** It follows Proposition 5 and Proposition 6 that, with probability over $1 - \frac{1}{n^{p_1-1}} - \frac{1}{2K^{p_2-2}} - P_G$, the following statements hold:

(i) The means $\{\Pi\boldsymbol{\mu}_1, \dots, \Pi\boldsymbol{\mu}_K\}$ of the embedded data are distinct.

(ii) $\hat{\gamma}^G_{\min}, \hat{\gamma}^G_{\max}$ defined in (31) and $\gamma^G_{\min}, \gamma^G_{\max}$ defined in (27) satisfy

$$\begin{aligned} \max_{\alpha\in[K]} \max_{i,j\in\tilde{I}_\alpha(d,\theta)} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_j)\|}{|\tilde{I}_\alpha(d,\theta)|} \leq \hat{\gamma}^G_{\min} \leq (1+\epsilon)\gamma^G_{\min}, \\ (1 - C_{12}\epsilon)\gamma^G_{\max} \leq \hat{\gamma}^G_{\max} \leq \min_{1\leq\alpha<\beta\leq K} \max_{\substack{i\in\tilde{I}_\alpha(d,\theta)\\ i'\in\tilde{I}_\beta(d,\theta)}} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_{i'})\|}{2(n-1)}. \end{aligned} \tag{55}$$

Now, we prove the theorem. We claim here that it is sufficient to show: if $r^G > \frac{1+\epsilon^G_{\min}}{1-C_{12}\epsilon^G_{\min}}$, then $\epsilon^G_{\min} < \epsilon^G_{\sup}$, and for any $\epsilon \in (\epsilon^G_{\min}, \epsilon^G_{\sup})$, the interval $\big[(1+\epsilon)\gamma^G_{\min}, (1-C_{12}\epsilon)\gamma^G_{\max}\big)$ is nonempty. In fact, if $\big[(1+\epsilon)\gamma^G_{\min}, (1-C_{12}\epsilon)\gamma^G_{\max}\big)$ is nonempty, then by (55), $\big[\hat{\gamma}^G_{\min}, \hat{\gamma}^G_{\max}\big)$ is nonempty. It follows Proposition 5 that for any $\hat{\gamma} \in \big[\hat{\gamma}^G_{\min}, \hat{\gamma}^G_{\max}\big)$, the points indexed by $\tilde{I}_\alpha(d,\theta), \alpha \in [K]$ are correctly labeled by the model.

On the one hand, we have

$$\begin{aligned} r^G > \tfrac{1+\epsilon_{\min}}{1-C_{12}\epsilon_{\min}} \quad &\Longrightarrow \quad (1 - C_{12}\epsilon^G_{\min})r^G > 1 + \epsilon^G_{\min} \\ &\Longrightarrow \quad \epsilon^G_{\min} < \tfrac{r^G - 1}{C_{12}r^G + 1} = \epsilon^G_{\sup}. \end{aligned}$$

This implies that the interval $(\epsilon^G_{\min}, \epsilon^G_{\sup})$ is nonempty.

On the other hand, we have

$$\begin{aligned} \epsilon < \epsilon^G_{\sup} \quad &\Longrightarrow \quad \epsilon < \tfrac{r^G-1}{C_{12}r^G+1} \\ &\Longrightarrow \quad \tfrac{1+\epsilon}{1-C_{12}\epsilon} < r^G \\ &\Longrightarrow \quad \tfrac{1+\epsilon}{1-C_{12}\epsilon} < \tfrac{\gamma^G_{\max}}{\gamma^G_{\min}} \\ &\Longrightarrow \quad (1+\epsilon)\gamma^G_{\min} < (1-C_{12}\epsilon)\gamma^G_{\max}. \end{aligned}$$

Thus, we have proved the theorem. ■

### A.7 Proof of Theorem 5

**Proof** With probability over $1 - \frac{1}{n^{p_1-1}} - \frac{1}{2K^{p_2-2}} - P_G$, we have:

(i) The means $\{\Pi\boldsymbol{\mu}_1, \dots, \Pi\boldsymbol{\mu}_K\}$ of the embedded data are distinct.

(ii) $\hat{\gamma}^G_{\min}, \hat{\gamma}^G_{\max}$ defined in (31) and $\gamma^G_{\min}, \gamma^G_{\max}$ defined in (27) satisfy

$$\begin{aligned} \max_{\alpha\in[K]} \max_{i,j\in\tilde{I}_\alpha(d,\theta)} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_j)\|}{|\tilde{I}_\alpha(d,\theta)|} \leq \hat{\gamma}^G_{\min} \leq \bar{S}(m,d,2)\gamma^G_{\min}, \\ (1 - \epsilon)\gamma^G_{\max} \leq \hat{\gamma}^G_{\max} \leq \min_{1\leq\alpha<\beta\leq K} \max_{\substack{i\in\tilde{I}_\alpha(d,\theta)\\ i'\in\tilde{I}_\beta(d,\theta)}} \frac{\|\Pi(\mathbf{a}_i - \mathbf{a}_{i'})\|}{2(n-1)}. \end{aligned} \tag{56}$$

Now, we prove the theorem. We claim here that it is sufficient to show: if $r^G > \frac{1+C_\kappa^2+\frac{2C_\kappa^2}{\sqrt{d}}}{1-\tilde{\epsilon}_{\text{min}}^G}$, then $\tilde{\epsilon}_{\text{min}}^G < \tilde{\epsilon}_{\text{sup}}^G$, and for any $\epsilon \in (\tilde{\epsilon}_{\text{min}}^G, \tilde{\epsilon}_{\text{sup}}^G)$, the interval $\big[\bar{S}(m,d,2)\gamma_{\text{min}}^G, (1-\epsilon)\gamma_{\text{max}}^G\big)$ is nonempty. In fact, if $\big[\bar{S}(m,d,2)\gamma_{\text{min}}^G, (1-\epsilon)\gamma_{\text{max}}^G\big)$ is nonempty, then by (56), $\big[\hat{\gamma}_{\text{min}}^G, \hat{\gamma}_{\text{max}}^G\big)$ is nonempty. It follows Proposition 5 that for any $\hat{\gamma} \in \big[\hat{\gamma}_{\text{min}}^G, \hat{\gamma}_{\text{max}}^G\big)$, points indexed by $\tilde{I}_\alpha(d,\theta), \alpha \in [K]$ are correctly labeled by the model.

On the one hand, by definition of $\tilde{\epsilon}_{\text{min}}^G$ and $C_0$, we have

$$\frac{1}{\sqrt{d}} = \frac{\tilde{\epsilon}_{\text{min}}^G}{\sqrt{p_2 C \log(K)}},\ C_0 = 1/\tilde{\epsilon}_{\text{min}}^G + \frac{2C_\kappa^2}{\sqrt{p_2 C \log(K)}}. \tag{57}$$

As a result,

$$\begin{aligned}
r^G > \tfrac{1+C_\kappa^2+\frac{2C_\kappa^2}{\sqrt{d}}}{1-\tilde{\epsilon}_{\text{min}}^G} \quad &\Longrightarrow \quad r^G > \tfrac{C_\kappa^2+1+\frac{\tilde{\epsilon}_{\text{min}}^G 2C_\kappa^2}{\sqrt{p_2 C \log(K)}}}{\sqrt{1-\tilde{\epsilon}_{\text{min}}^G}} \\
&\Longrightarrow \quad r^G > \tfrac{C_\kappa^2+\tilde{\epsilon}_{\text{min}}^G\left(1/\tilde{\epsilon}_{\text{min}}^G+\frac{2C_\kappa^2}{\sqrt{p_2 C \log(K)}}\right)}{1-\tilde{\epsilon}_{\text{min}}^G} \\
&\Longrightarrow \quad r^G > \tfrac{C_\kappa^2+\tilde{\epsilon}_{\text{min}}^G C_0}{1-\tilde{\epsilon}_{\text{min}}^G} \\
&\Longrightarrow \quad C_\kappa^2 + \tilde{\epsilon}_{\text{min}} C_0 < 1 - \tilde{\epsilon}_{\text{min}}^G r^G \\
&\Longrightarrow \quad \tilde{\epsilon}_{\text{min}}^G < \tfrac{r^G - C_\kappa^2}{C_0 + r^G} = \tilde{\epsilon}_{\text{sup}}^G.
\end{aligned}$$

On the other hand, we have

$$\bar{S}(m,d,2) = \frac{\sqrt{d}+2C_\kappa^2}{\sqrt{m}} + C_\kappa^2 = \frac{\sqrt{d}+2C_\kappa^2}{\sqrt{p_2 C \log(K)}}\epsilon + C_\kappa^2 = C_0\epsilon + C_\kappa^2. \tag{58}$$

As a result,

$$\begin{aligned}
\epsilon \in (\tilde{\epsilon}_{\text{min}}^G, \tilde{\epsilon}_{\text{sup}}^G) \quad &\Longrightarrow \quad C_0\epsilon + C_\kappa^2 < (1-\epsilon) r^G \\
&\Longrightarrow \quad \bar{S}(m,d,2)\gamma_{\text{min}}^G < (1-\epsilon)\gamma_{\text{max}}^G.
\end{aligned}$$

Thus, we have proved the theorem. ■

### A.8 Computational Complexity Analysis for Implementing (RPCCM)

In this section, we will discuss the computational complexity for the implementation of the model (RPCCM). Let $A \in \mathbb{R}^{d\times n}$ the input data and $k$ be a given positive integer (e.g., $k = 5$). From the paradigm Figure 3, we can summarize the main steps in our implementations as follows:

Step 1 : Construct Gaussian kernel weights $w_{ij}$ according to (1).

Step 2 : Sample a random projection matrix $\Pi \in \mathbb{R}^{m\times d}$ with a given embedding dimension $m < d$ and obtain the embedded data $\Pi A \in \mathbb{R}^{m\times n}$.

Step 3 : Apply the AS-SSNAL algorithm to solve the model (RPCCM) along a sequence of the given regularization parameters $\infty > \gamma_1 > \gamma_2 > \cdots > \gamma_T > 0$ and obtain the clustering path.

For simplicity, we focus on discussing the computational complexity for a fixed $\gamma > 0$. The computational complexities for Step 1 and Step 2 can be easily calculated, which are summarized in Table 15.

Table 15: Computational complexity for constructing weights and embedded data.

| | Step 1 | | Step 2 | |
|---|---|---|---|---|
| | Construct $\mathcal{E}(k)$ | Compute $w_{ij}$ | Construct $\Pi$ | Construct $\Pi A$ |
| Complexity | $O(n^2d + n^2 \min\{\log(n), k\})$ | $O(dnk)$ | $O(dm)$ | $O(nmd)$ |

Now, we discuss the computational complexity for Step 3 in detail. Although it is difficult to obtain the overall computational complexity bounds of the AS-SSNAL algorithm, we will discuss the computational complexity for each of its main steps. From an illustrative paradigm in Figure 2, we can summarize the main computational components of the AS-SSNAL algorithm for solving (RPCCM) as follows:

Step 3.1 Apply the adaptive sieving (AS) strategy developed in (Yuan et al., 2022) to reduce the dimension in terms of $n$. This is motivated by the fact that the solution $\{x_i^*\}$ corresponding to the same identified cluster by the model (CCM) will be identical. In short, with the AS technique, we can obtain a solution to (RPCCM) by solving a sequence of reduced subproblems.

Step 3.2 Apply the augmented Lagrangian method (ALM) to solve the reduced subproblems generated by the AS technique.

Step 3.3 Apply the semismooth Newton (SSN) method to solve the subproblems of ALM.

Step 3.4 Apply the conjugate gradient (CG) method to solve the linear systems and obtain the Newton directions.

First of all, according to (Yuan et al., 2022, Theorem 2.6), the AS technique is guaranteed to converge in a finite number of iterations. As reported in (Yuan et al., 2022), for solving the convex clustering problem (CCM), it typically requires less than 3 AS iterations to obtain a solution to the model (CCM). The key step for constructing the reduced subproblem is to identify the connected components of the subgraph generated by $n$ nodes and some given edge set $\bar{\mathcal{E}} \subseteq \mathcal{E}(k)$ (Yuan et al., 2022, Section 3.1). The computational complexity for this step is bound by $O(nk)$. The AS technique will reduce $n$ to $\bar{n}$, where $\bar{n}$ is the number of connected components. Here, an isolated node will also be regarded as a connected component. Therefore, the computational complexity of SSNAL for solving the reduced subproblems of (RPCCM) can also be obtained after we analyze the SSNAL algorithm for solving (RPCCM).

Now, we move on to discuss the convergence rate for the ALM algorithm in Step 3.2. It has been known that the (inexact) ALM enjoys an asymptotically superlinear convergence rate for solving convex programming problems under mild error-bound conditions

(Rockafellar, 1976a,b; Cui et al., 2019). Due to the close connection between the proximal point algorithm (PPA) and the ALM (Rockafellar, 1976b), it follows from the results for PPA in (Güler, 1991) that the convergence rate in terms of objective function value for the exact ALM for solving (RPCCM) is at least $O(1/k)$. On the other hand, since (RPCCM) is strongly convex, it follows from (Xu, 2021) that it requires $O(\tau^{-1/2}|\log(\tau)|)$ to obtain an $\tau$-solution to (RPCCM) in terms of objective function value. In practice, we can usually obtain a solution to (RPCCM) to the required accuracy within 30 iterations of inexact ALM in our numerical experiments.

For Step 3.3, the locally superlinear (quadratic) convergence rate of the SSN method for solving the subproblems of ALM has been shown in (Sun et al., 2021) for (CCM). However, as far as we know, the iteration complexity of the SSN method remains open. We plan to investigate this challenging research question in the future. Regarding the numerical performance of the SSN method, we have observed that it usually terminates within 3 iterations for most of the examples we tested for (RPCCM).

Lastly, we discuss the computational complexity of the CG method for obtaining the Newton direction in the SSN method. It is well known that the convergence rate of the CG method depends critically on the condition number of the coefficient matrix in the Newton system. Fortunately, as discussed in (Sun et al., 2021, Section 5.4), the condition number $\rho \leq (1 + \sigma\lambda_{\max}(L_{\mathcal{E}(k)}))$, where $\sigma$ is the penalty parameter in the augmented Lagrangian function and $\lambda_{\max}(L_{\mathcal{E}(k)})$ is the largest eigenvalue of the Laplacian matrix of the graph by $\mathcal{E}(k)$. Moreover, it follows from (Sun et al., 2021) that the per-iteration computational cost for CG is $O(m|\mathcal{E}(k)|)$.

### A.9 Comparisons with Baseline Algorithms

Here, we include the numerical results of the clustering algorithms on the real datasets.

Table 16: Comparisons with baselines on data LIBRAS. The results are averaged over 10 random projections. We use the format "mean ± standard deviation" to report the results of ARI and AMI.

| **Model** | | **ARI** | **AMI** | **Time** |
|---|---|---|---|---|
| (RP) CCM | $d = 90$ | 0.3767 ± 0.0000 | 0.6119 ± 0.0000 | 0.3550 |
| | $m = 50$ | 0.3817 ± 0.0088 | 0.6158 ± 0.0046 | 0.2781 |
| | $m = 20$ | 0.3740 ± 0.0054 | 0.6125 ± 0.0049 | 0.2510 |
| | $m = 10$ | 0.3651 ± 0.0095 | 0.6093 ± 0.0055 | 0.1460 |
| (RP) KM++ | $d = 90$ | 0.3049 ± 0.0130 | 0.5259 ± 0.0124 | 0.1159 |
| | $m = 50$ | 0.3054 ± 0.0221 | 0.5206 ± 0.0191 | 0.0770 |
| | $m = 20$ | 0.2680 ± 0.0274 | 0.4733 ± 0.0282 | 0.0755 |
| | $m = 10$ | 0.2286 ± 0.0218 | 0.4317 ± 0.0168 | 0.0774 |
| (RP) SC | $d = 90$ | 0.3956 ± 0.0000 | 0.6026 ± 0.0000 | 0.0318 |
| | $m = 50$ | 0.3790 ± 0.0173 | 0.5895 ± 0.0148 | 0.0383 |
| | $m = 20$ | 0.3414 ± 0.0222 | 0.5522 ± 0.0190 | 0.0382 |
| | $m = 10$ | 0.2906 ± 0.0222 | 0.4906 ± 0.0190 | 0.0365 |
| (RP) CLINK | $d = 90$ | 0.2275 ± 0.0000 | 0.4565 ± 0.0000 | 0.0039 |
| | $m = 50$ | 0.2518 ± 0.0237 | 0.4591 ± 0.0275 | 0.0022 |
| | $m = 20$ | 0.2253 ± 0.0244 | 0.4257 ± 0.0288 | 0.0019 |
| | $m = 10$ | 0.1875 ± 0.0256 | 0.3805 ± 0.0349 | 0.0017 |
| (RP) HDB | $d = 90$ | 0.0546 ± 0.0000 | 0.3237 ± 0.0000 | 0.0353 |
| | $m = 50$ | 0.0779 ± 0.0229 | 0.3565 ± 0.0358 | 0.0357 |
| | $m = 20$ | 0.0842 ± 0.0239 | 0.3470 ± 0.0408 | 0.0360 |
| | $m = 10$ | 0.0598 ± 0.0214 | 0.3055 ± 0.0454 | 0.0388 |
| (RP) MS | $d = 90$ | 0.3026 ± 0.0214 | 0.4996 ± 0.0454 | 0.0171 |
| | $m = 50$ | 0.2927 ± 0.0224 | 0.4940 ± 0.0329 | 0.0118 |
| | $m = 20$ | 0.2768 ± 0.0316 | 0.4598 ± 0.0405 | 0.0063 |
| | $m = 10$ | 0.1944 ± 0.0664 | 0.3955 ± 0.0955 | 0.0057 |

Table 17: Comparisons with baselines on data LIBRAS-6. The results are averaged over 10 random projections. We use the format "mean ± standard deviation" to report the results of ARI and AMI.

| **Model** | | **ARI** | **AMI** | **Time** |
|---|---|---|---|---|
| (RP) CCM | $d = 90$ | 0.7674 ± 0.0000 | 0.8065 ± 0.0000 | 0.1120 |
| | $m = 50$ | 0.7674 ± 0.0000 | 0.8065 ± 0.0000 | 0.0715 |
| | $m = 20$ | 0.7674 ± 0.0000 | 0.8065 ± 0.0000 | 0.0710 |
| | $m = 10$ | 0.7505 ± 0.0218 | 0.8081 ± 0.0020 | 0.0650 |
| (RP) KM++ | $d = 90$ | 0.5119 ± 0.0075 | 0.6162 ± 0.0102 | 0.0417 |
| | $m = 50$ | 0.5268 ± 0.0614 | 0.6281 ± 0.0518 | 0.0348 |
| | $m = 20$ | 0.4695 ± 0.0868 | 0.5795 ± 0.0784 | 0.0372 |
| | $m = 10$ | 0.4025 ± 0.1024 | 0.5140 ± 0.0969 | 0.0377 |
| (RP) SC | $d = 90$ | 0.7255 ± 0.0017 | 0.8057 ± 0.0011 | 0.0146 |
| | $m = 50$ | 0.6846 ± 0.0429 | 0.7729 ± 0.0394 | 0.0169 |
| | $m = 20$ | 0.6400 ± 0.0795 | 0.7393 ± 0.0604 | 0.0148 |
| | $m = 10$ | 0.5721 ± 0.0996 | 0.6736 ± 0.0776 | 0.0124 |
| (RP) CLINK | $d = 90$ | 0.4003 ± 0.0000 | 0.5269 ± 0.0000 | 0.0023 |
| | $m = 50$ | 0.3985 ± 0.0734 | 0.5333 ± 0.0703 | 0.0014 |
| | $m = 20$ | 0.3575 ± 0.1124 | 0.4856 ± 0.1037 | 0.0014 |
| | $m = 10$ | 0.3468 ± 0.0802 | 0.4755 ± 0.0822 | 0.0011 |
| (RP) HDB | $d = 90$ | 0.4781 ± 0.0000 | 0.6425 ± 0.0000 | 0.0153 |
| | $m = 50$ | 0.4428 ± 0.0820 | 0.5984 ± 0.0604 | 0.0134 |
| | $m = 20$ | 0.4227 ± 0.1134 | 0.5867 ± 0.0790 | 0.0136 |
| | $m = 10$ | 0.3131 ± 0.1013 | 0.5030 ± 0.0743 | 0.0140 |
| (RP) MS | $d = 90$ | 0.6507 ± 0.0135 | 0.7484 ± 0.0221 | 0.0035 |
| | $m = 50$ | 0.5977 ± 0.0600 | 0.7014 ± 0.0514 | 0.0021 |
| | $m = 20$ | 0.5374 ± 0.0665 | 0.6444 ± 0.0683 | 0.0016 |
| | $m = 10$ | 0.4760 ± 0.0821 | 0.5717 ± 0.0919 | 0.0013 |

Table 18: Comparisons with baselines on data COIL-20. The results are averaged over 10 random projections. We use the format "mean ± standard deviation" to report the results of ARI and AMI.

| Model | | ARI | AMI | Time |
|---|---|---|---|---|
| (RP) CCM | $d = 1024$ | 0.8136 ± 0.0000 | 0.9165 ± 0.0000 | 36.1010 |
| | $m = 200$ | 0.8123 ± 0.0031 | 0.9164 ± 0.0008 | 7.1021 |
| | $m = 100$ | 0.8119 ± 0.0035 | 0.9152 ± 0.0031 | 3.5995 |
| | $m = 50$ | 0.8106 ± 0.0053 | 0.9144 ± 0.0042 | 2.2270 |
| | $m = 20$ | 0.8147 ± 0.0025 | 0.9166 ± 0.0002 | 1.3453 |
| | $m = 10$ | 0.8112 ± 0.0073 | 0.9133 ± 0.0058 | 0.8810 |
| (RP) KM++ | $d = 1024$ | 0.5937 ± 0.0250 | 0.7603 ± 0.0135 | 2.7407 |
| | $m = 200$ | 0.5730 ± 0.0333 | 0.7493 ± 0.0175 | 1.3477 |
| | $m = 100$ | 0.5622 ± 0.0323 | 0.7352 ± 0.0201 | 0.3493 |
| | $m = 50$ | 0.5484 ± 0.0265 | 0.7230 ± 0.0175 | 0.2461 |
| | $m = 20$ | 0.4908 ± 0.0230 | 0.6689 ± 0.0159 | 0.2047 |
| | $m = 10$ | 0.4443 ± 0.0227 | 0.6233 ± 0.0179 | 0.2081 |
| (RP) SC | $d = 1024$ | 0.8251 ± 0.0052 | 0.9234 ± 0.0054 | 0.4472 |
| | $m = 200$ | 0.8265 ± 0.0048 | 0.9258 ± 0.0030 | 0.2327 |
| | $m = 100$ | 0.8022 ± 0.0158 | 0.9109 ± 0.0087 | 0.0863 |
| | $m = 50$ | 0.7921 ± 0.0095 | 0.9032 ± 0.0055 | 0.0656 |
| | $m = 20$ | 0.7132 ± 0.0388 | 0.8505 ± 0.0198 | 0.0680 |
| | $m = 10$ | 0.5186 ± 0.0580 | 0.7156 ± 0.0343 | 0.0708 |
| (RP) CLINK | $d = 1024$ | 0.3538 ± 0.0000 | 0.5846 ± 0.0000 | 0.2086 |
| | $m = 200$ | 0.3483 ± 0.0294 | 0.5808 ± 0.0221 | 0.0520 |
| | $m = 100$ | 0.3814 ± 0.0342 | 0.6061 ± 0.0250 | 0.0234 |
| | $m = 50$ | 0.3601 ± 0.0365 | 0.5879 ± 0.0264 | 0.0155 |
| | $m = 20$ | 0.3708 ± 0.0399 | 0.5829 ± 0.0267 | 0.0110 |
| | $m = 10$ | 0.3376 ± 0.0319 | 0.5705 ± 0.0328 | 0.0161 |
| (RP) HDB | $d = 1024$ | 0.7703 ± 0.0000 | 0.8801 ± 0.0000 | 0.3163 |
| | $m = 200$ | 0.7303 ± 0.0297 | 0.8569 ± 0.0154 | 0.3337 |
| | $m = 100$ | 0.7259 ± 0.0304 | 0.8580 ± 0.0127 | 0.2909 |
| | $m = 50$ | 0.6806 ± 0.0266 | 0.8410 ± 0.0114 | 0.2893 |
| | $m = 20$ | 0.4924 ± 0.0757 | 0.7501 ± 0.0462 | 0.3133 |
| | $m = 10$ | 0.2132 ± 0.0386 | 0.6084 ± 0.0330 | 0.2991 |
| (RP) MS | $d = 1024$ | 0.5830 ± 0.0074 | 0.6597 ± 0.0100 | 2.5359 |
| | $m = 200$ | 0.5806 ± 0.0099 | 0.6651 ± 0.0093 | 1.2775 |
| | $m = 100$ | 0.5745 ± 0.0335 | 0.6627 ± 0.0395 | 0.2845 |
| | $m = 50$ | 0.5458 ± 0.0350 | 0.6392 ± 0.0406 | 0.0714 |
| | $m = 20$ | 0.4955 ± 0.0572 | 0.6040 ± 0.0353 | 0.0517 |
| | $m = 10$ | 0.3549 ± 0.0959 | 0.5738 ± 0.0693 | 0.0415 |

Table 19: Comparisons with baselines on data LUNG. The results are averaged over 10 random projections. We use the format "mean ± standard deviation" to report the results of ARI and AMI.

| **Model** | | **ARI** | **AMI** | **Time** |
|---|---|---|---|---|
| (RP) CCM | $d = 12625$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 3.9890 |
| | $m = 200$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.1521 |
| | $m = 100$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.1115 |
| | $m = 50$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.0863 |
| | $m = 20$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.0693 |
| | $m = 10$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.0588 |
| (RP) KM++ | $d = 12625$ | 0.9586 ± 0.0000 | 0.9426 ± 0.0000 | 0.2670 |
| | $m = 200$ | 0.9378 ± 0.0815 | 0.9312 ± 0.0592 | 0.0264 |
| | $m = 100$ | 0.9052 ± 0.1076 | 0.9013 ± 0.0937 | 0.0189 |
| | $m = 50$ | 0.8543 ± 0.1337 | 0.8694 ± 0.0991 | 0.0198 |
| | $m = 20$ | 0.8032 ± 0.1473 | 0.8060 ± 0.1204 | 0.0207 |
| | $m = 10$ | 0.7058 ± 0.1666 | 0.7258 ± 0.1298 | 0.0192 |
| (RP) SC | $d = 12625$ | 0.7998 ± 0.0000 | 0.7951 ± 0.0000 | 0.0221 |
| | $m = 200$ | 0.7979 ± 0.0739 | 0.8197 ± 0.0677 | 0.0138 |
| | $m = 100$ | 0.8587 ± 0.1161 | 0.8549 ± 0.1072 | 0.0127 |
| | $m = 50$ | 0.7832 ± 0.0572 | 0.8007 ± 0.0301 | 0.0112 |
| | $m = 20$ | 0.7202 ± 0.1051 | 0.7485 ± 0.0823 | 0.0116 |
| | $m = 10$ | 0.7229 ± 0.1262 | 0.7283 ± 0.1096 | 0.0120 |
| (RP) CLINK | $d = 12625$ | 1.0000 ± 0.0000 | 1.0000 ± 0.0000 | 0.0100 |
| | $m = 200$ | 0.8364 ± 0.1237 | 0.8114 ± 0.1222 | 0.0011 |
| | $m = 100$ | 0.8282 ± 0.1099 | 0.8117 ± 0.1084 | 0.0010 |
| | $m = 50$ | 0.7589 ± 0.1811 | 0.7751 ± 0.1292 | 0.0012 |
| | $m = 20$ | 0.7110 ± 0.1811 | 0.6997 ± 0.1292 | 0.0010 |
| | $m = 10$ | 0.5664 ± 0.2204 | 0.5647 ± 0.1977 | 0.0009 |
| (RP) HDB | $d = 12625$ | 0.4891 ± 0.0000 | 0.5490 ± 0.0000 | 0.0094 |
| | $m = 200$ | 0.4669 ± 0.0479 | 0.4325 ± 0.0669 | 0.0058 |
| | $m = 100$ | 0.4297 ± 0.0828 | 0.4254 ± 0.1043 | 0.0067 |
| | $m = 50$ | 0.4104 ± 0.1423 | 0.4212 ± 0.1303 | 0.0064 |
| | $m = 20$ | 0.3126 ± 0.1587 | 0.4024 ± 0.1387 | 0.0065 |
| | $m = 10$ | 0.3669 ± 0.1095 | 0.4282 ± 0.1130 | 0.0063 |
| (RP) MS | $d = 12625$ | 0.8999 ± 0.0654 | 0.8452 ± 0.0322 | 0.0419 |
| | $m = 200$ | 0.8431 ± 0.0748 | 0.7618 ± 0.0885 | 0.0012 |
| | $m = 100$ | 0.8481 ± 0.0637 | 0.7624 ± 0.1043 | 0.0007 |
| | $m = 50$ | 0.7923 ± 0.1105 | 0.7321 ± 0.1213 | 0.0006 |
| | $m = 20$ | 0.6979 ± 0.1649 | 0.6297 ± 0.1577 | 0.0006 |
| | $m = 10$ | 0.6542 ± 0.1466 | 0.5943 ± 0.1554 | 0.0009 |

Table 20: Comparisons with baselines on data MNIST. The results are averaged over 10 random projections. We use the format "mean ± standard deviation" to report the results of ARI and AMI.

| **Model** | | **ARI** | **AMI** | **Time** |
|---|---|---|---|---|
| (RP) CCM | $d = 784$ | 0.6468 ± 0.0000 | 0.7054 ± 0.0000 | 1240.4120 |
| | $m = 200$ | 0.6125 ± 0.0350 | 0.7134 ± 0.0249 | 180.3215 |
| | $m = 100$ | 0.6118 ± 0.0374 | 0.7265 ± 0.0315 | 133.4112 |
| | $m = 50$ | 0.6207 ± 0.0399 | 0.7102 ± 0.0232 | 59.2690 |
| | $m = 20$ | 0.6116 ± 0.0519 | 0.7152 ± 0.0416 | 39.5231 |
| | $m = 10$ | 0.6042 ± 0.0547 | 0.7305 ± 0.0242 | 22.2365 |
| (RP) KM++ | $d = 784$ | 0.3813 ± 0.0026 | 0.4991 ± 0.0015 | 60.3587 |
| | $m = 200$ | 0.3550 ± 0.0201 | 0.4738 ± 0.0130 | 18.3064 |
| | $m = 100$ | 0.3368 ± 0.0164 | 0.4509 ± 0.0124 | 8.0586 |
| | $m = 50$ | 0.2959 ± 0.0291 | 0.4041 ± 0.0229 | 4.6307 |
| | $m = 20$ | 0.2052 ± 0.0198 | 0.3046 ± 0.0179 | 2.9046 |
| | $m = 10$ | 0.1613 ± 0.0288 | 0.2439 ± 0.0365 | 2.1792 |
| (RP) SC | $d = 784$ | 0.6101 ± 0.0000 | 0.7196 ± 0.0000 | 15.2337 |
| | $m = 200$ | 0.5840 ± 0.0129 | 0.6898 ± 0.0092 | 4.5842 |
| | $m = 100$ | 0.5678 ± 0.0163 | 0.6745 ± 0.0105 | 2.8558 |
| | $m = 50$ | 0.4705 ± 0.0489 | 0.6061 ± 0.0310 | 2.0994 |
| | $m = 20$ | 0.2263 ± 0.0669 | 0.3953 ± 0.0424 | 2.2416 |
| | $m = 10$ | 0.1390 ± 0.0190 | 0.2548 ± 0.0230 | 2.3545 |
| (RP) CLINK | $d = 784$ | 0.1879 ± 0.0000 | 0.3227 ± 0.0000 | 10.7043 |
| | $m = 200$ | 0.1542 ± 0.0329 | 0.2891 ± 0.0236 | 2.9764 |
| | $m = 100$ | 0.1549 ± 0.0258 | 0.2914 ± 0.0303 | 1.9668 |
| | $m = 50$ | 0.1374 ± 0.0278 | 0.2573 ± 0.0323 | 1.5220 |
| | $m = 20$ | 0.1041 ± 0.0306 | 0.1963 ± 0.0325 | 1.3127 |
| | $m = 10$ | 0.0972 ± 0.0255 | 0.1572 ± 0.0363 | 1.2151 |
| (RP) MS | $d = 784$ | 0.5830 ± 0.0074 | 0.6597 ± 0.0100 | 2.5359 |
| | $m = 200$ | 0.3849 ± 0.0298 | 0.3861 ± 0.0413 | 67.6404 |
| | $m = 100$ | 0.3396 ± 0.0411 | 0.3513 ± 0.0421 | 38.5571 |
| | $m = 50$ | 0.2776 ± 0.0655 | 0.3258 ± 0.0815 | 30.9492 |
| | $m = 20$ | 0.1244 ± 0.0718 | 0.3102 ± 0.0833 | 17.3379 |
| | $m = 10$ | 0.0037 ± 0.0097 | 0.0552 ± 0.0571 | 2.9932 |

### A.10 Numerical Verification for the Model (RPCCM) with Uniform Weights for the MSG Problem Setting

In this section, we will verify the recovery guarantees of the model (RPCCM) with uniform weights for the MSG problem setting, as stated in Theorem 4 and Theorem 5. Since the robustness of random projections and the computational efficiency of the model (RPCCM) have been verified in Section 5, we will only test with a specific mixture of $K = 4$ Gaussians $\mathcal{N}(\mathbf{e}_\alpha, 0.01^2\mathbf{I}_{100})$ with $\mathbf{p}_\alpha = \frac{1}{4}$, for $\alpha = 1, \ldots, 4$. We will generate 10 datasets for verification, each containing 10000 points that are independently drawn from this distribution[9].

To simplify the analysis, we will fix the parameters $\theta = \sqrt{2d} = 10\sqrt{2}$ and $\eta = 0.1$. Based on these values, we can compute that $F(\theta, d) > 0.9999$, and $1 - \exp\left(-2\mathbf{p}_\alpha^2\eta^2 n\right) > 0.9999$. This implies that, with a probability higher than 0.9999, $|\tilde{I}_\alpha(d,\theta)| \geq 2250,\ \alpha = 1, \ldots, 4$. Using the above information, we can determine the values of $\gamma^G_{\min}$, $\gamma^G_{\max}$, and $r^G$ defined in (27), which are

$$\gamma^G_{\min} = 1.2570 \cdot 10^{-6},\ \gamma^G_{\max} = 7.0717 \cdot 10^{-5},\ r^G = 56.2585. \tag{59}$$

These values indicate that the model (CCM) could correctly recover the points indexed by $\tilde{I}_\alpha(d,\theta)$ for all $\alpha = 1, \ldots, 4$, provided that $\gamma$ is in the range $\left[\gamma^G_{\min}, \gamma^G_{\max}\right)$. Additionally, the large ratio $r^G$ indicates the feasibility of correct recovery for the model (RPCCM) using an appropriate embedding dimension. In this section, we will set $m = \lceil \epsilon^{-2}\log(n)\rceil$ (or $\tilde{m} = \lceil 2\epsilon^{-2}\log(K)\rceil$). The valid distortion range $(\epsilon^G_{\min}, \epsilon^G_{\sup})$ as defined in (35) (or $(\tilde{\epsilon}^G_{\min}, \tilde{\epsilon}^G_{\sup})$ in (37)), along with the tested values of distortions and embedding dimensions, are summarized in Table 21.

Table 21: Valid distortion range and tested values of distortions and embedding dimensions.

| **Embedding dimension** | **Valid distortion range** | **Tested values of $(\epsilon, m)$ (or $(\epsilon, \tilde{m})$)** |
|---|---|---|
| $m = \lceil \epsilon^{-2}\log(n)\rceil$ | [0.3035,1.7340) | $(0.5, 37), (1.0, 10), (1.5, 5)$ |
| $\tilde{m} = \lceil 2\epsilon^{-2}\log(K)\rceil$ | [0.1665, 0.8706) | $(0.4, 18), (0.6, 8), (0.8, 5)$ |

For each pair of $(\epsilon, m)$ (or $(\epsilon, \tilde{m})$), we will randomly sample 10 projection matrices for verification. We will compute $\hat{\gamma}^G_{\min}$ and $\hat{\gamma}^G_{\max}$ by (10) using each projection matrix for each dataset. Then, we will test the probability that (55) in Theorem 4 (or (56) in Theorem 5) is satisfied. The results of these tests are summarized in Table 22.

Table 22: Numerical verification for the correct recovery of the model (RPCCM) for 10 datasets generated from the same MSG setting. In the table, the value $p_{\tilde{I}_\alpha(d,\theta)}$ denotes the probability that (55) in Theorem 4 (or (56) in Theorem 5) is satisfied.

| **Case 1:** $m = \lceil \epsilon^{-2}\log(n)\rceil$ | | | | **Case 2:** $\tilde{m} = \lceil \epsilon^{-2}\log(K)\rceil$ | | | |
|---|---|---|---|---|---|---|---|
| $(\epsilon, m)$ | $(0.5, 37)$ | $(1.0, 10)$ | $(1.5, 5)$ | $(\epsilon, \tilde{m})$ | $(0.4, 18)$ | $(0.6, 8)$ | $(0.8, 5)$ |
| $p_{\tilde{I}_\alpha(d,\theta)}$ | 100/100 | 100/100 | 100/100 | $p_{\tilde{I}_\alpha(d,\theta)}$ | 100/100 | 100/100 | 100/100 |

[9] We want to remind that interested readers can refer to (Jiang et al., 2020) for detailed numerical results concerning the model (CCM) with uniform weights for the MSG problem setting.

From the results, we can observe that for all the tested distortions and embedding dimensions, the successful probability $p_{\tilde{I}_\alpha(d,\theta)}$ is 100/100, which implies that for all the 10 datasets, the model (RPCCM) could correctly recover the points indexed by $\tilde{I}_\alpha(d,\theta)$ for all $\alpha = 1, \ldots, 4$, provided $\gamma$ is in the range of (36) in Theorem 4 (or (38) in Theorem 5). Since the 10 datasets are generated from the same mixture of Gaussians, the above results verify the usefulness of our recovery guarantees for the MSG problem setting.

## References

Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

Dimitris Achlioptas and Frank McSherry. On spectral learning of mixtures of distributions. In *International Conference on Computational Learning Theory*, pages 458–469. Springer, 2005.

Nir Ailon and Bernard Chazelle. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009.

Nir Ailon and Edo Liberty. Fast dimension reduction using Rademacher series on dual BCH codes. *Discrete & Computational Geometry*, 42(4):615–630, 2009.

Nir Ailon and Edo Liberty. An almost optimal unrestricted fast Johnson-Lindenstrauss transform. *ACM Transactions on Algorithms*, 9(3):21, 2013.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 160–172. Springer, 2013.

Ricardo JGB Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data*, 10(1):1–51, 2015.

Eric C Chi and Kenneth Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013, 2015.

Eric C Chi and Stefan Steinerberger. Recovering trees with convex clustering. *SIAM Journal on Mathematics of Data Science*, 1(3):383–407, 2019.

Eric C. Chi, Brian R. Gaines, Will Wei Sun, Hua Zhou, and Jian Yang. Provable convex co-clustering of tensors. *Journal of Machine Learning Research*, 21(214):1–58, 2020.

Julien Chiquet, Pierre Gutierrez, and Guillem Rigaill. Fast tree inference with weighted fusion penalties. *Journal of Computational and Graphical Statistics*, 26(1):205–216, 2017.

Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, pages 163–172, 2015.

Michael B Cohen, TS Jayram, and Jelani Nelson. Simple analyses of the sparse Johnson-Lindenstrauss transform. In *1st Symposium on Simplicity in Algorithms (SOSA 2018)*, pages 15–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018.

Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

Ying Cui, Defeng Sun, and Kim-Chuan Toh. On the R-superlinear convergence of the KKT residuals generated by the augmented Lagrangian method for convex composite conic programming. *Mathematical Programming*, 178:381–415, 2019.

Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. A sparse Johnson-Lindenstrauss transform. In *Proceedings of the 42th ACM Symposium on Theory of Computing*, pages 341–350, 2010.

Sanjoy Dasgupta. Learning mixtures of gaussians. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 634–644. IEEE, 1999.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

Sanjoy Dasgupta and Leonard Schulman. A probabilistic analysis of em for mixtures of separated, spherical gaussians. *Journal of Machine Learning Research*, 8(2), 2007.

Kenneth R Davidson and Stanislaw J Szarek. Local operator theory, random matrices and banach spaces. *Handbook of the Geometry of Banach Spaces*, 1:317–366, 2001.

D. Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20: 364–366, 1977.

Daniel Dias, Sarajane Peres, and Helton Bscaro. Libras Movement. UCI Machine Learning Repository, 2009a.

Daniel B Dias, Renata CB Madeo, Thiago Rocha, Helton H Biscaro, and Sarajane M Peres. Hand movement recognition for brazilian sign language: a study using distance-based neural networks. In *2009 International Joint Conference on Neural Networks*, pages 697–704. IEEE, 2009b.

Alexander Dunlap and Jean-Christophe Mourrat. Local versions of sum-of-norms clustering. *SIAM Journal on Mathematics of Data Science*, 4(4):1250–1271, 2022.

K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1): 32–40, 1975.

Osman Güler. On the convergence of the proximal point algorithm for convex minimization. *SIAM Journal on Control and Optimization*, 29(2):403–419, 1991.

Toby Dylan Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th International Conference on Machine Learning*, pages 745–752, 2011.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2: 193–218, 1985.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 604–613, 1998.

Tao Jiang and Stephen Vavasis. Certifying clusters from sum-of-norms clustering. *arXiv preprint arXiv:2006.11355*, 2021.

Tao Jiang, Stephen Vavasis, and Chen Wen Zhai. Recovery of a mixture of Gaussians by sum-of-norms clustering. *Journal of Machine Learning Research*, 21(225):1–16, 2020.

Tao Jiang, Samuel Tan, and Stephen Vavasis. Re-embedding data to strengthen recovery guarantees of clustering. *arXiv preprint arXiv:2301.10901*, 2023.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Contemporary Mathematics*, volume 26, pages 189–206. American Mathematical Society, 1984.

Daniel M Kane and Jelani Nelson. A derandomized sparse Johnson-Lindenstrauss transform. *arXiv preprint arXiv:1006.3585*, 2010.

Daniel M Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1):1–23, 2014.

Kasper Green Larsen and Jelani Nelson. The Johnson-Lindenstrauss lemma is optimal for linear dimensionality reduction. In *43rd International Colloquium on Automata, Languages, and Programming*, pages 82:1–82:11, 2016.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Mihee Lee, Haipeng Shen, Jianhua Z Huang, and James S Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095, 2010.

Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. Clustering using sum-of-norms regularization: With application to particle filter output computation. In *2011 IEEE Statistical Signal Processing Workshop*, pages 201–204, 2011.

Konstantin Makarychev, Yury Makarychev, and Ilya Razenshteyn. Performance of Johnson–Lindenstrauss transform for k-means and k-medians clustering. *SIAM Journal on Computing*, 52(2):STOC19–269–STOC19–297, 2023.

Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing.* MIT press, 1999.

Jiří Matoušek. On variants of the Johnson-Lindenstrauss lemma. *Random Structures & Algorithms*, 33(2):142–156, 2008.

Sameer A Nene, Shree K Nayar, and Hiroshi Murase. Columbia object image library (coil-20). *Technical Report CUCS-005-96*, 1996.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14, 2001.

Ashkan Panahi, Devdatt Dubhashi, Fredrik D Johansson, and Chiranjib Bhattacharyya. Clustering by sum of norms: Stochastic incremental algorithm, convergence and cluster recovery. In *34th International Conference on Machine Learning*, pages 2769–2777, 2017.

Kristiaan Pelckmans, Joseph De Brabanter, Johan AK Suykens, and B De Moor. Convex clustering shrinkage. In *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*, 2005.

Peter Radchenko and Gourab Mukherjee. Convex clustering via $l_1$ fusion penalization. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 79(5):1527–1546, 2017.

R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976a.

R Tyrrell Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1:97–116, 1976b.

Mark Rudelson and Roman Vershynin. Non-asymptotic theory of random matrices: Extreme singular values. In *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*, pages 1576–1602, 2010.

Defeng Sun, Kim-Chuan Toh, and Yancheng Yuan. Convex clustering: Model, theoretical guarantee and efficient algorithm. *Journal of Machine Learning Research*, 22(9):1–32, 2021.

Kean Ming Tan and Daniela Witten. Statistical properties of convex clustering. *Electronic Journal of Statistics*, 9(2):2324–2347, 2015.

Sergei Vassilvitskii and David Arthur. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2006.

Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.

Suresh Venkatasubramanian and Qiushi Wang. The Johnson–Lindenstrauss transform: an empirical study. In *2011 Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 164–173. SIAM, 2011.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1073–1080, 2009.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854, 2010.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.

Yangyang Xu. Iteration complexity of inexact augmented lagrangian methods for constrained convex programming. *Mathematical Programming*, 185:199–244, 2021.

Yancheng Yuan, Defeng Sun, and Kim-Chuan Toh. An efficient semismooth Newton based algorithm for convex clustering. In *35th International Conference on Machine Learning*, pages 5718–5726, 2018.

Yancheng Yuan, Tsung-Hui Chang, Defeng Sun, and Kim-Chuan Toh. A dimension reduction technique for large-scale structured sparse optimization problems with application to convex clustering. *SIAM Journal on Optimization*, 32(3):2294–2318, 2022.

Changbo Zhu, Huan Xu, Chenlei Leng, and Shuicheng Yan. Convex optimization procedure for clustering: Theoretical revisit. *Advances in Neural Information Processing Systems*, 27:1619–1627, 2014.