Journal of Machine Learning Research 27 (2026) 1-44 Submitted 9/25; Revised 4/26; Published 5/26

Graph-based Clustering Revisited: A Relaxation of Kernel k-Means Perspective

Wenlong Lyu L_W_L@SEU.EDU.CN
School of Computer Science and Engineering
Southeast University
Nanjing, Jiangsu 211189, China

Yuheng Jia* YHJIA@SEU.EDU.CN
Division of Computer Science and Engineering
Southeast University
Nanjing, Jiangsu 211189, China

Hui Liu H2LIU@SFU.EDU.HK
Yam Pak Charitable Foundation School of Computing and Information Sciences
Saint Francis University
Hong Kong SAR

Junhui Hou JH.HOU@CITYU.EDU.HK
Department of Computer Science
City University of Hong Kong
Hong Kong SAR

Editor: Brian Kulis

Abstract

The well-known graph-based clustering methods, including spectral clustering, symmetric non-negative matrix factorization, and doubly stochastic normalization, can be viewed as relaxations of the kernel k-means approach. However, we posit that these methods excessively relax their inherent low-rank, nonnegative, doubly stochastic, and orthonormal constraints to ensure numerical feasibility, potentially limiting their clustering efficacy. In this paper, guided by our systematic theoretical analyses, we propose **Lo**w-**R**ank **D**oubly stochastic clustering (**LoRD**), a model that only relaxes the orthonormal constraint to derive a probabilistic clustering results. Furthermore, by theoretically establishing the equivalence between orthogonality and **B**lock diagonality under the doubly stochastic constraint, we propose **B-LoRD**. By integrating block diagonal regularization into LoRD, expressed as the maximization of the Frobenius norm, we enhance clustering performance. To ensure numerical solvability, we transform the non-convex doubly stochastic constraint into a linear convex constraint through the introduction of a class probability parameter. The theoretical demonstration of the gradient Lipschitz continuity of our LoRD and B-LoRD enables the proposal of a projected gradient algorithm whose exact iteration admits a sublinear convergence-rate bound and ensures first-order stationarity of every accumulation point for the exact projected gradient iteration. Extensive experiments underscore the effectiveness of our approaches. The code is publicly available at `https://github.com/lwl-learning/LoRD`.

*. Corresponding author.

Keywords: Graph-based clustering, low-rank, kerne k-means, block diagonal, doubly stochastic

1. Introduction

Clustering is a fundamental task in unsupervised learning, which aims to partition unlabeled samples into groups that reflect latent structure in the data, thereby supporting data summarization, pattern discovery, and downstream analysis. It has been widely used in applications such as image segmentation (Shi and Malik, 2000; Kim et al., 2024), document clustering (Cai et al., 2005), biological data analysis (Liu et al., 2024; Yang et al., 2025; Hu et al., 2021; Levine et al., 2015), and community discovery in networks (Schaeffer, 2007; Fortunato and Hric, 2016).

Among the many clustering paradigms, graph-based clustering (Schaeffer, 2007; Berahmand et al., 2025; Xue et al., 2024; Kang et al., 2021; Wu et al., 2022) partitions data according to pairwise affinities rather than Euclidean centroids. It is particularly useful when the cluster geometry is non-linear, when the input is more naturally represented by an affinity graph or similarity matrix, or when relations among samples are of primary interest (Von Luxburg, 2007; Dhillon et al., 2004; Shi and Malik, 2000).

A central challenge in graph-based clustering is to balance structural fidelity and numerical tractability. A desirable formulation should preserve the structural properties that make clusters identifiable, interpretable, and directly recoverable from the learned representation. Yet these same properties often appear as hard non-convex constraints that are difficult to optimize. This tension is especially visible in methods related to kernel k-means, where different relaxations lead to markedly different graph-based clustering models (Dhillon et al., 2004; Zass and Shashua, 2005, 2006; Kuang et al., 2012, 2015).

This perspective is useful because it reveals that many classical methods differ not only in their solvers, but also in the structural components they preserve. Spectral clustering (Von Luxburg, 2007) preserves orthogonality but relaxes nonnegativity and doubly stochasticity; SymNMF (Kuang et al., 2012, 2015) preserves a low-rank nonnegative factorization but relaxes doubly stochasticity; DSN (Zass and Shashua, 2005, 2006) preserves doubly stochasticity but relaxes the low-rank factorization. These choices directly affect whether the learned representation yields cluster assignments without post-processing, whether it admits a probabilistic interpretation, and how expensive the resulting optimization becomes.

In this paper, we revisit graph-based clustering from this structural viewpoint. This scope is practically relevant in settings where pairwise similarities or an affinity graph are the natural input, such as recent unsupervised image segmentation and spatial transcriptomics applications (Kim et al., 2024; Liu et al., 2024; Yang et al., 2025). Guided by this goal, we propose LoRD, which only relaxes the orthonormal constraint and thus yields probabilistic assignments, and B-LoRD, which further controls the learned block-diagonal structure. We introduce the kernel k-means view below as a unifying mathematical lens for these models. Fig. 1 summarizes this viewpoint and serves as a roadmap for the paper.

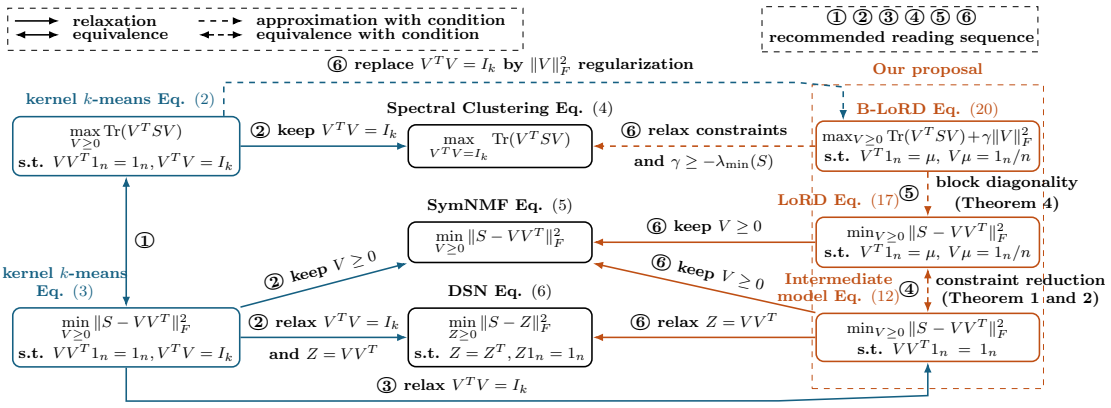


Figure 1: Schematic diagram of graph-based clustering methods. ①: The models in Eq. (2) and Eq. (3) are equivalent. ②: SC, SymNMF and DSN are overly relaxed kernel k-means. ③: The model in Eq. (12) only relaxes the least important orthonormal constraint $V^TV = I_k$. ④: The non-convex constraint $VV^T1_n = 1$ can be reduced to a convex constraint $V^T1_n = \mu, V\mu = 1_n/n$ by incorporating the class probability parameter μ (Theorems 1 and 2), making our LoRD numerically solvable and interpretable in terms of probability. ⑤: The k-block diagonality of VV^T can be adjusted by tuning $\gamma \in [-\lambda_{\max}(S), -\lambda_{\min}(S)]$ in B-LoRD. ⑥: kernel k-means, SC, SymNMF and DSC can be seen as a relaxation or approximation of our LoRD and B-LoRD.

1.1 A Unified View from Kernel k-means

We next make the above structural viewpoint precise through kernel k-means, which provides a convenient mathematical reference point for several representative graph-based clustering methods.

Let $\{x_1, \ldots, x_n\}$ be n data points to be grouped into k clusters $G_1, \ldots, G_k$, and $S_{ij} = \kappa(x_i, x_j)$ be a symmetric similarity matrix defined by a kernel function $\kappa(x_i, x_j)$, e.g., $\kappa(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\sigma^2)$ for the Gaussian kernel. Kernel k-means seeks to maximize intra-class similarity by partitioning data into clusters $G_1, \ldots, G_k$, as expressed by

$$\max_{G_1,\ldots,G_k} \sum_{r=1}^{k} \frac{1}{n_r} \sum_{x_i, x_j \in G_r} S_{ij}, \tag{1}$$

where $n_r = |G_r|$ represents the size of G_r. To transform Eq. (1) into matrix form, we introduce the class assignment matrix $V \in \mathbb{R}^{n\times k}$ with $V_{ij} = 1/\sqrt{n_j}$ if $x_i \in G_j$ and zero otherwise. Notably, the definition of V aligns with the constraints $V \geq 0, VV^T1_n = 1_n, V^TV = I_k$, where 1_n is an n-dimensional vector of ones, I_k is the identity matrix of size k, $V \geq 0$ means $\forall i, j, V_{ij} \geq 0$. Consequently, Eq. (1) can be equivalently expressed as:

$$\max_{V\geq 0} \mathrm{Tr}(V^TSV), \text{ s.t. } VV^T1_n = 1_n, V^TV = I_k. \tag{2}$$

Another equivalent form of Eq. (2) is written as (Ding et al., 2005):

$$\min_{V\geq 0} \|S - VV^T\|_F^2, \text{ s.t. } VV^T1_n = 1_n, V^TV = I_k, \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. These two formulations reveal four structural components that are central to graph-based clustering: the low-rank factorization $V \in \mathbb{R}^{n\times k}$, the nonnegative constraint $V \geq 0$, the doubly stochastic constraint $VV^T 1_n = 1_n$, and the orthonormal constraint $V^T V = I_k$. The low-rank structure ties the representation directly to k clusters, nonnegativity makes V cluster-indicative, doubly stochasticity supports a probabilistic interpretation, and orthogonality sharpens discriminability and is closely related to the block diagonality (Lu et al., 2018) of VV^T.

Kernel k-means is NP-hard (Aloise et al., 2009). The key question is therefore not whether one should relax the formulation, but which structural components should be relaxed and which should be preserved. Three representative answers are given by spectral clustering, SymNMF, and DSN.

1) Spectral clustering (SC) (Von Luxburg, 2007). SC retains only the orthonormal structure in Eq. (2), i.e.:

$$\max_{V^T V = I_k} \operatorname{Tr}(V^T S V). \tag{4}$$

The optimum of Eq. (4) is given by the eigenvectors of S corresponding to the largest k eigenvalues. However, SC relaxes both nonnegativity and doubly stochasticity, so the learned embedding does not directly provide cluster assignments and usually requires post-processing such as k-means.

2) Symmetric non-negative matrix factorization (SymNMF) (Kuang et al., 2012, 2015). SymNMF retains the low-rank nonnegative factorization in Eq. (3), i.e.:

$$\min_{V\geq 0} \|S - VV^T\|_F^2. \tag{5}$$

SymNMF can produce cluster-indicative factors without post-processing. However, because it relaxes doubly stochasticity and orthogonality, the probabilistic interpretation is lost and the original clustering structure is only partially preserved.

3) Doubly stochastic normalization (DSN) (Zass and Shashua, 2005, 2006). DSN retains the doubly stochastic structure at the level of Z, but relaxes both orthogonality and the low-rank factorization $Z = VV^T$ in Eq. (3) to solve the following convex problem:

$$\min_{Z\geq 0} \|S - Z\|_F^2, \text{ s.t. } Z = Z^T, Z1_n = 1_n. \tag{6}$$

The probabilistic interpretability of Z is analyzed in (Zass and Shashua, 2005). However, because the low-rank factorization is relaxed, DSN generally requires additional post-processing and incurs higher computational cost.

In summary, SC, SymNMF, and DSN can be viewed as three representative relaxation paths from kernel k-means, each preserving a different subset of the original structure. Our modeling choice is to move from hard partitions to probabilistic assignments by relaxing orthogonality while preserving the low-rank, nonnegative, and doubly stochastic structure. To make the resulting formulation tractable, we later show that the doubly stochastic constraint can be reduced to linear convex constraints through a class prior parameter μ. We further connect orthogonality with block diagonality, which motivates the B-LoRD extension. Here we focus on this high-level structural viewpoint; more detailed discussions on block-diagonal learning and related graph-based clustering methods are deferred to Sec. 2.

1.2 Contributions

The main contributions of this paper are summarized as follows.

1. We propose **LoRD**, a **Lo**w-**R**ank **D**oubly stochastic clustering model that preserves the low-rank, nonnegative, and doubly stochastic structure of kernel k-means while relaxing the orthonormal constraint. By introducing the class prior parameter μ, we reduce the original non-convex doubly stochastic constraint to linear convex constraints, which makes the model numerically tractable and yields a probabilistic interpretation of the learned assignments (Theorems 1 and 2).

2. We propose **B-LoRD**, which incorporates block diagonal regularization into LoRD. Under the doubly stochastic constraint, the block-diagonal structure of VV^T can be controlled via $\|V\|_F^2$ (Theorem 4), so that B-LoRD can strengthen or weaken the learned block diagonality through the hyper-parameter γ.

3. We develop an efficient projected gradient descent algorithm for LoRD and B-LoRD, with $\mathcal{O}(n^2k)$ complexity per iteration, which can be reduced to $\mathcal{O}(n\log(n)k)$ by exploiting the sparsity of S. We show that the objective functions are gradient Lipschitz continuous (Theorem 5), that the exact iteration admits a sublinear convergence-rate bound (Lemma 7), and that every accumulation point of the exact projected gradient iteration is a first-order stationary point (Lemma 8).

4. Extensive experiments on both synthetic and real-world datasets demonstrate that LoRD and B-LoRD are highly competitive with representative graph-based clustering baselines. The results further show that the objective values of LoRD and B-LoRD are better aligned with clustering accuracy than those of more heavily relaxed methods.

The remainder of this paper is organized as follows. In Sec. 2, we review block-diagonal learning and representative graph-based clustering methods related to our work. In Sec. 3, we propose **Lo**w-**R**ank **D**oubly stochastic clustering (LoRD) and **B**lock diagonality regularized LoRD (B-LoRD), which are then numerically solved by an efficient yet effective projected gradient descent algorithm in Sec. 4. In Sec. 5, we evaluate the performance of our LoRD and B-LoRD in both synthetic and real-world datasets. Finally, we conclude this paper in Sec. 6.

2. Related Work

2.1 Block Diagonal Structure

A similarity matrix $S \in \mathbb{R}^{n\times n}$ exhibits an ideal clustering structure when it has exactly k connected components, where k is the number of clusters and each connected component corresponds to one cluster. Such an S can be expressed as a k-block diagonal matrix (Feng et al., 2014; Lu et al., 2018) as follows:

$$S = \begin{bmatrix} S_1 & 0 & \cdots & 0 \\ 0 & S_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_k \end{bmatrix}, \text{ where } S_i \in \mathbb{R}^{n_i\times n_i}. \tag{7}$$

According to the spectral graph theorem (Von Luxburg, 2007), the number of connected components of S is equal to the multiplicity k of the eigenvalue 0 of the Laplacian matrix $L_S = \text{Diag}(S1_n) - S$, where $\text{Diag}(z)$ is a diagonal matrix with z as its diagonal elements. Building on this insight, the k-block diagonal structure of S can be achieved by constraining $\text{rank}(L_S) = n-k$ (Wang et al., 2016). However, the constraint is difficult to directly handle in optimization. The most common approach is to relax it into a regularization $\|S\|_{\boxed{k}}$ using Ky Fan's theorem (Wang et al., 2016; Xie et al., 2017; Nie et al., 2014), i.e.,

$$\|S\|_{\boxed{k}} := \sum_{i=n-k+1}^{n} \lambda_i(L_S) = \min_{\substack{0 \preceq W \preceq I_n \\ \text{Tr}(W)=k}} \langle L_S, W \rangle, \tag{8}$$

where $\lambda_i(L_S)$ is the i-th largest eigenvalue of L_S, $0 \preceq W \preceq I_n$ means that $0 \leq \lambda_{\min}(W) \leq \lambda_{\max}(W) \leq 1$. Despite this relaxation, $\|S\|_{\boxed{k}}$ requires an auxiliary variable W to be alternatively optimized. Therefore, learning a k-block diagonal structure remains an optimization challenge.

When combining the doubly stochastic constraint $Z = Z^T$, $Z1_n = 1_n$ in DSN, the block diagonality can also be boosted in different ways, possibly easier than $\|Z\|_{\boxed{k}}$. For example, motivated by the Davis-Kahan theorem, two constraints $\sigma_k(Z) \geq c_1$ and $\sigma_{k+1}(Z) \leq c_2$ are introduced in (Park and Kim, 2017), where $\sigma_k(Z)$ is the k-largest singular value of Z, $c_1, c_2 \in [0, 1]$ are hyper-parameters. When c_1 is close to one, $\lambda_{n-k+1}(L_Z) = 1-\lambda_k(Z) \leq 1-c_1^2$ is close to zero. Thus, $\sigma_k(Z) \geq c_1$ can be seen as a relaxation of the k-block diagonal constraint $\text{rank}(L_Z) \leq n-k$. More recently, (Julien, 2022) noticed that if a matrix Z is both doubly stochastic and idempotent (i.e., $Z^2 = Z$), then Z is block diagonal. Thus, the idempotent condition $Z^2 = Z$ is added as a constraint. However, two common issues exist in the above methods: 1) High computational complexity ($\mathcal{O}(n^3)$), as the low-rank structure of Z was relaxed; 2) They only focus on enhancing block diagonality, as the regularization coefficient is nonnegative.

In this paper, we demonstrate that when further combining the low-rank structure ($Z = VV^T$), the block diagonality of VV^T can be enhanced (resp. weakened) by maximizing (resp. minimizing) $\|V\|_F^2$.

2.2 Graph-based Clustering Methods

Complementary to the block-diagonal viewpoint above, representative graph-based clustering methods can also be compared by which structural components of kernel k-means they preserve or relax.

Semi-definite programming (SDP) (Peng and Wei, 2007; Kulis et al., 2007) provides a convex relaxation of kernel k-means, formulated as:

$$\max_{Z} \langle S, Z \rangle, \quad \text{s.t.} \quad Z \succeq 0, Z \geq 0, Z = Z^T, Z1_n = 1_n, \text{Tr}(Z) = k. \tag{9}$$

The gap between SDP and kernel k-means lies in the idempotency constraint $Z = Z^2$ (Kulis et al., 2007), which is relaxed in SDP. Owing to its convexity, SDP enjoys well-established statistical guarantees (Giraud and Verzelen, 2019; Chen and Yang, 2021). However, due to its high complexity (i.e., $\mathcal{O}(n^{3.5})$ per iteration (Sun et al., 2020)), SDP is impractical for real-world datasets.

To reduce the complexity of SDP, Zhuang et al. (2024) proposed NLR, which leverages a low-rank factorization $Z = UU^T$ and directly optimizes over $U \in \mathbb{R}^{n\times r}$ with $r \geq k$. The NLR formulation is given by:

$$\max_{U\geq 0} \mathrm{Tr}(U^T SU), \text{ s.t. } \|U\|_F^2 = k, UU^T 1_n = 1_n. \tag{10}$$

Comparing Eq.(10) with Eq.(2), NLR can be viewed as kernel k-means where the orthogonality constraint $V^T V = I_k$ with $V \in \mathbb{R}^{n\times k}$ is relaxed to $\|V\|_F^2 = k$ and extended to $V \in \mathbb{R}^{n\times r}$. Benefiting from the low-rank structure, the per-iteration complexity of NLR is reduced to $\mathcal{O}(n^2 rt)$, where t denotes the number of primal descent steps. The primary optimization challenge arises from the quadratic, non-convex constraint $UU^T 1_n = 1_n$. Consequently, the algorithm in Zhuang et al. (2024) requires careful tuning of both the step size and regularization coefficient, and the constraint $UU^T 1_n = 1_n$ is not strictly enforced.

The doubly stochastic constraint can also be handled effectively via the Majorization-Minimization (MM) framework. For instance, DCD (Yang and Oja, 2012; Yang et al., 2016) is a graph-based clustering method that imposes both low-rank and doubly stochastic structures, formulated as:

$$\min_{W\geq 0} D_{KL}(S||W\mathrm{Diag}^{-1}(W^T 1_n)W^T), \quad \text{s.t.} \quad W1_k = 1_n, \tag{11}$$

where $D_{KL}(\cdot|\cdot)$ denotes the KL divergence. The resulting similarity matrix is both low-rank and doubly stochastic. The per-iteration complexity of DCD is $\mathcal{O}(nkq)$, where q denotes the sparsity of the input similarity matrix S (typically $q = \mathcal{O}(\log n)$).

In contrast to DCD, we introduce a class prior probability vector $\mu \in \mathbb{R}^k$, which reduces the doubly stochastic constraint $VV^T 1_n = 1_n$ to linear and convex conditions $V^T 1_n = \mu, V\mu = 1_n/n$. By exploiting the Lipschitz continuity of the gradients in our formulation, we design a projected gradient descent algorithm that achieves the same $\mathcal{O}(nkq)$ complexity for sparse S.

3. Proposed Models

3.1 LoRD: Graph-Based Probabilistic Clustering

In contrast to the rigid partitions sought by kernel k-means, probabilistic clustering (Zass and Shashua, 2005) aims to determine the probability that x_i belongs to a typical cluster, i.e., $\mathrm{P}(y_i = j|x_i), j = 1, \ldots, k$. However, the orthonormal constraint $V^T V = I_k$ in kernel k-means forces V to be a hard clustering result, i.e., each row of V has only one non-zero element. To this end, we relax $V^T V = I_k$ in Eq. (3) to obtain a soft clustering result, i.e.,

$$\min_{V\in\mathbb{R}^{n\times k}} \|S - VV^T\|_F^2, \text{ s.t. } V \geq 0, VV^T 1_n = 1_n. \tag{12}$$

Unlike SC, SymNMF, and DSN, Eq. (12) solely relaxes the least crucial orthogonality constraint, which is necessary to obtain probabilistic clustering. However, Eq. (12) is difficult to optimize due to the non-convex quadratic constraint $VV^T 1_n = 1_n$. To address this, we first express the constraint space of Eq. (12) as Ω:

$$\Omega := \{V \in \mathbb{R}^{n\times k} \mid V \geq 0, VV^T 1_n = 1_n/n\}, \tag{13}$$

where we replace $VV^T1_n = 1_n$ with $VV^T1_n = 1_n/n$, which is equivalent to a scalar product of V. To reduce the quadratic constraint in Ω, we denote $\mu = V^T1_n$, so that $VV^T1_n = 1_n/n$ is equivalently written as $V\mu = 1_n/n$. In other words, we construct a subspace of Ω determined by μ:

$$\Omega(\mu) := \{V \in \mathbb{R}^{n\times k} | V \geq 0, V^T1_n = \mu, V\mu = 1_n/n\}. \tag{14}$$

To ensure $\Omega(\mu)$ is a subspace of Ω, we must have $\mu \geq 0$ and $\|\mu\|_2^2 = 1_n^T VV^T1_n = 1$, indicating that μ should lie on the space of $\mathbb{S}_+^k = \{\mu \in \mathbb{R}^k \mid \mu \geq 0, \|\mu\|_2^2 = 1\}$, i.e., the nonnegative unit sphere embedded in $\mathbb{R}^k$. The relationship between $\Omega(\mu)$ and Ω is formally stated in Theorem 1 below:

Theorem 1 *When μ is varied over $\mathbb{S}_+^k$, the family of space $\Omega(\mu)$ is a partition of Ω, i.e.:*

- *$\forall \mu \in \mathbb{S}_+^k$, $\Omega(\mu)$ is non-empty, i.e., $\Omega(\mu) \neq \emptyset$.*
- *When μ is varied over $\mathbb{S}_+^k$, Ω is the union of $\Omega(\mu)$, i.e., $\Omega = \bigcup_{\mu\in\mathbb{S}_+^k} \Omega(\mu)$.*
- *$\forall \mu, \nu \in \mathbb{S}_+^k$ where $\mu \neq \nu$, the intersection of $\Omega(\mu)$ and $\Omega(\nu)$ is empty, i.e., $\Omega(\mu) \cap \Omega(\nu) = \emptyset$.*

For a better understanding of Theorem 1, the relationship between Ω and $\Omega(\mu)$ is schematically illustrated in the inset figure: any $V \in \Omega$ (non-convex, blue face) lies on $\Omega(\mu)$ (convex, orange segment) that determined by a certain $\mu \in \mathbb{S}_+^k$. Therefore, it is natural to ask: *What is the physical meaning of μ, and which μ should we expect.* The answers to these questions are given in Theorem 2 below.

Theorem 2 *Let $\mathrm{P}(c_j)$ be the prior probability of the j-th class, $\mathrm{P}(x_i)$ be the prior probability of x_i, assumed to be uniform, i.e., $\mathrm{P}(x_i) = 1/n$. When $\mu = [\sqrt{\mathrm{P}(c_1)}, \ldots, \sqrt{\mathrm{P}(c_k)}]^T$, any $V \in \Omega(\mu)$ can be expressed as:*

$$V_{ij} = \frac{\mathrm{P}(y_i = j|x_i)\mathrm{P}(x_i)}{\sqrt{\mathrm{P}(c_j)}}. \tag{15}$$

Thus, V_{ij} is associated with the conditional probability of x_i belonging to the j-th class $\mathrm{P}(y_i = j|x_i)$, indicating that V corresponds to a probabilistic clustering result.

Furthermore, the pairwise probability matrix can be recovered by $Z = n^2V\mathrm{Diag}(\mu\odot\mu)V^T$, such that

$$Z_{ij} = \mathrm{P}(y_i = y_j|x_i, x_j). \tag{16}$$

In other words, Z_{ij} describes the conditional probability of x_i and x_j belonging to the same class.

In Theorem 2, the $1/n$ in the assumption $\mathrm{P}(x_i) = 1_n/n$ is drawn from $VV^T1_n = 1_n/n$, indicating that if $\mathrm{P}(x_1), \ldots, \mathrm{P}(x_n)$ are known, it may make sense to replace the doubly stochastic constraint with $VV^T1_n = [\mathrm{P}(x_1), \ldots, \mathrm{P}(x_n)]^T$. More importantly, Theorem 1 and Theorem 2 state that when $\mathrm{P}(c_1), \ldots, \mathrm{P}(c_k)$ are known, we expect the learned V lies on the $\Omega(\mu)$, where $\mu = [\sqrt{\mathrm{P}(c_1)}, \ldots, \sqrt{\mathrm{P}(c_k)}]^T$, and **the constraint $V \in \Omega$ is equivalently**

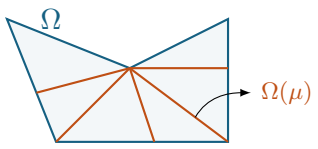
reduced to $V \in \Omega(\mu)$. Building on this insight, we propose a low-rank doubly stochastic clustering (LoRD) model, which is formulated as

$$\min_{V \in \Omega(\mu)} \|S - VV^T\|_F^2. \tag{17}$$

Compared to Eq. (12), Eq. (17) is numerically solvable because $\Omega(\mu)$ is linear and convex. In practice, as the class prior probability is generally unknown, we can simply set $\mu = [1/\sqrt{k}, \ldots, 1/\sqrt{k}]^T$. Our experiments in Sec. 5.6 show that the proposed model is robust to the value of μ.

Remark 3 *For LoRD in Eq. (17), the optimization space $\Omega(\mu)$ only relaxes the least important orthonormal constraint $V^TV = I_k$ of kernel k-means in Eq. (3). For numerical solvability, we further reduce Ω to $\Omega(\mu)$ (Theorem 1), where μ is a user-specified parameter associated with the prior probability of the class (Theorem 2). As a result, LoRD can learn a probabilistic clustering result (Theorem 2).*

3.2 B-LoRD: Adjusting k-Block Diagonality of VV^T



Although LoRD achieves probabilistic clustering, it remains unclear how to control the distribution of $\mathrm{P}(y_i = j|x_i)$ (sharp or uniform), which is closely related to the orthogonality of V: when V is fully orthogonal, we have the sharpest clustering probability $\mathrm{P}(y_i = j|x_i) = 1$ if $x_i \in G_j$ and zero otherwise; when V is least orthogonal, we have the uniform clustering probability $\mathrm{P}(y_i = j|x_i) = 1/k, j = 1, \ldots, k$. Interestingly, we find that the orthogonality of V is equivalent to the k-block diagonality of VV^T under the doubly stochastic constraint, as described by the following theorem:

Theorem 4 *For any $V \in \Omega$ (which naturally includes any $V \in \Omega(\mu)$), the following equality holds:*

$$\|VV^T\|_{\boxed{k}} = \frac{k}{n} - \|V\|_F^2. \tag{18}$$

Specifically, the least k-block diagonal case ($\|VV^T\|_{\boxed{k}}$ is maximized) occurs when $V \in \{1_n\mu^T/n \mid \mu \in \mathbb{S}_+^k\}$, and the fully k-block diagonal case ($\|VV^T\|_{\boxed{k}}$ is minimized to zero) occurs when V is orthogonal.

Besides, the objective function in Eq. (17) is equivalent to

$$\|S - VV^T\|_F^2 = \|S\|_F^2 - 2\mathrm{Tr}(V^TSV) + \|VV^T\|_F^2, \tag{19}$$

where $\|S\|_F^2$ can be treated as a constant, and the role of minimizing $\|VV^T\|_F^2$ under the constraint $V \in \Omega(\mu)$ is to weaken the block diagonality of VV^T. To see this, we have $\|VV^T\|_F^2 = \sum_{j=1}^k \sigma_j^4(V) \geq \frac{1}{n^2}$, where the lower bound is achieved when $V = 1_n\mu^T/n$ with $\sigma(V) = [\frac{1}{\sqrt{n}}, 0, \ldots, 0]^T$.

Motivated by the above analysis, we propose a low-rank block diagonal doubly stochastic clustering (B-LoRD) model (replace $\|VV^T\|_F^2$ in Eq. (17) with $-2\gamma\|V\|_F^2$):

$$\max_{V \in \Omega(\mu)} \mathrm{Tr}(V^TSV) + \gamma\|V\|_F^2, \tag{20}$$

where γ is a hyper-parameter that controls the block diagonality of VV^T. Specifically, the objective function of Eq. (20) can be written as $\mathrm{Tr}(V^T(S+\gamma I_n)V)$, indicating that the value of γ should lie in the range $[-\lambda_{\max}(S), -\lambda_{\min}(S)]$. When $\gamma \leq -\lambda_{\max}(S)$, Eq. (20) becomes a convex optimization problem, and its global optimum is trivial: $V = 1_n\mu^T/n$. When $\gamma = -\lambda_{\min}(S)$, we observe that the learned V is almost orthogonal (see Fig. 4 for details), indicating that γ is sufficiently large. Note that γ can be negative, which means the block diagonality of VV^T is weakened.

3.3 Relation to Other Graph-Based Clustering Methods

Kernel k-means. Our model in Eq. (12), which is equivalent to LoRD in Eq. (17), assuming the class prior probability μ, relaxes only the least important orthogonality constraint $V^TV = I_k$ in Eq. (3). Moreover, our B-LoRD in Eq. (20) replaces the orthonormal constraint $V^TV = I_k$ in Eq. (2) with a regularization term $\gamma\|V\|_F^2$. Therefore, when γ is sufficiently large (e.g., $\gamma \geq -\lambda_{\min}(S)$), our LoRD and B-LoRD can be regarded as a relaxation and an approximation of kernel k-means, respectively.

SC. SC can be interpreted as a relaxation of our B-LoRD in Eq. (20) in the case where γ is sufficiently large and the constraint $V \in \Omega(\mu)$ is relaxed. As a consequence, SC requires a post-processing step to obtain the final clustering result.

SymNMF. SymNMF can be thought of as a relaxation of Eq. (12) and Eq. (17), where the constraints $V \in \Omega$ and $V \in \Omega(\mu)$ are relaxed to $V \geq 0$. This relaxation leads to the loss of both probabilistic interpretability and discriminative capability. As analyzed in Sec. 5.2, minimizing the objective function value of SymNMF does not significantly improve clustering performance. In contrast, our LoRD and B-LoRD demonstrate meaningful performance gains.

DSN. Building upon our model in Eq. (12), DSN further parameterizes $Z = VV^T$ to obtain a convex optimization problem. However, this comes at the cost of requiring post-processing to extract the clustering results and incurring higher computational complexity, typically $\mathcal{O}(n^3)$, due to the relaxation of the low-rank structure. In comparison, our LoRD and B-LoRD achieve lower computational complexity at $\mathcal{O}(n^2k)$.

GWL. Additionally, our B-LoRD in Eq. (20) can be reformulated as a Gromov-Wasserstein learning problem (Chowdhury and Needham, 2021), which is an optimal transport (Montesuma et al., 2024) based approach to clustering. A detailed analysis of this connection is provided in the following subsection.

3.4 B-LoRD VS. Gromov-Wasserstein Learning in Optimal Transport

Optimal transport (OT) (Montesuma et al., 2024) has received a lot of attention in the machine learning community, as it learns a transport plan P on the joint probability space:

$$\Pi(\alpha, \beta) := \{P \in \mathbb{R}^{n\times k} \mid P \geq 0, P1_k = \alpha, P^T1_n = \beta\}, \tag{21}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are marginal probabilities satisfying $\alpha^T1_n = \beta^T1_k = 1$.

Gromov-Wasserstein learning (GWL) (Xu et al., 2019; Chowdhury and Needham, 2021; Van Assel et al., 2024) is an OT-based approach to graph partition, which solves

$$\min_{P\in\Pi(\alpha,\beta)} \sum_{i,j=1}^{n}\sum_{a,b=1}^{k} \ell(S_{ij},C_{ab})P_{ia}P_{jb}, \tag{22}$$

where $S\in\mathbb{R}^{n\times n}$ is the source graph that describes the similarities between samples, $C\in\mathbb{R}^{k\times k}$ is the target graph that describes the similarities between clusters, and ℓ is a loss function.

Let $D=\mathrm{Diag}(\mu)$, for any $V\in\Omega(\mu)$ such that $V\geq 0, V\mu=1_n/n, V^T1_n=\mu$, we have $P=VD\in\Pi(1_n/n,\mu\odot\mu)$, where $\odot$ is the Hadamard product. By parameterizing $V=PD^{-1}$, our B-LoRD becomes

$$\max_{P\in\Pi(1_n/n,\mu\odot\mu)} \mathrm{Tr}(D^{-1}P^TSPD^{-1})+\gamma\|PD^{-1}\|_F^2. \tag{23}$$

Interestingly, Eq. (23) is mathematically similar to the GWL. To demonstrate this, Eq. (23) can be transformed into

$$\min_{P\in\Pi(1_n/n,\mu\odot\mu)} -\sum_{i,j=1}^{n}\sum_{a=1}^{k}(S+\gamma I_n)_{ij}\mu_a^{-2}P_{ia}P_{ja}, \tag{24}$$

where D^{-1} is regarded as a target graph in which each cluster is only similar to itself, and the loss function is $\ell(S_{ij},D_{aa})=-(S+\gamma I_n)_{ij}\mu_a^{-2}$.

4. Numerical Optimization

4.1 Optimization Framework

We propose a projected gradient method to solve Eq. (17) and Eq. (20). Let $f_1(V)$ and $-f_2(V)$ be the objective functions of Eq. (17) and Eq. (20), respectively. Note that we transform Eq. (20) into the problem of minimizing $f_2(V)$ for a consistent description with Eq. (17). The gradients of $f_1(V)$ and $f_2(V)$ are

$$\nabla_1(V)=4(VV^T-S)V \quad\text{and}\quad \nabla_2(V)=-2(SV+\gamma V), \tag{25}$$

respectively, and they have the following property.

Theorem 5 *For $V\in\Omega$ (naturally includes $V\in\Omega(\mu)$), ∇_1 and ∇_2 are Lipschitz continuous, where the Lipschitz constant L_1 and L_2 are:*

$$L_1=4\left(3/n+\|S\|_{\mathrm{op}}\right),\quad L_2=2\left\|S+\gamma I_n\right\|_{\mathrm{op}}, \tag{26}$$

where $\|\cdot\|_{\mathrm{op}}$ is the operator norm, i.e., the largest singular value of a matrix.

In the remainder of this paper, we will omit the subscripts of f,∇, and L if they are clear in context. Since ∇ is Lipschitz continuous, the step size of the projected descent can

be automatically set to $1/L$, leading to the update formulas to solve Eq. (17) and Eq. (20) at the t-th iteration as

$$V^{t+1} = \mathcal{P}_{\Omega(\mu)}(V^t - \nabla(V^t)/L), \tag{27}$$

where $\mathcal{P}_{\Omega(\mu)}$ is the orthogonal projector onto $\Omega(\mu)$:

$$\mathcal{P}_{\Omega(\mu)}(U) = \underset{V \in \Omega(\mu)}{\arg\min} \|V - U\|_F^2, \tag{28}$$

which can be calculated by applying the Dykstra algorithm (Boyle and Dykstra, 1986) introduced in the next subsection. Note that $\mathcal{P}_{\Omega(\mu)}(U)$ is well defined, that is, the optimization problem in Eq. (28) has a unique optimum as $\Omega(\mu)$ is convex.

Moreover, to initialize $V^0 \in \Omega(\mu)$, the Sinkhorn-Knopp algorithm (Sinkhorn, 1964) is applied, such that for any $U \in \mathbb{R}^{n\times k}$, $\text{Sinkhorn}(U, \mu) \in \Omega(\mu)$. The introduction of the Sinkhorn-Knopp algorithm can be found in Sec. 4.3.

The overall projected gradient descent algorithm for solving LoRD in Eq. (17) and B-LoRD in Eq. (20) is summarized in Alg. 1, where we repeat Eq. (27) until reach the maximum iteration count or $\|V^{t+1} - V^t\|_F / \|V\|_F \leq \delta_v$. In Alg. 1, $\text{rand}(n, k)$ returns a random matrix of $n \times k$ in the range of $[0, 1]$. In B-LoRD, instead of tuning the hyper-parameter $\gamma \in [-\lambda_{\max}(S), -\lambda_{\min}(S)]$, we use $\tau \in [0, 1]$ to calculate $\gamma = -\lambda_{\max}(S) + \tau(\lambda_{\max}(S) - \lambda_{\min}(S))$ for convenience.

Algorithm 1: LoRD Eq. (17) and B-LoRD Eq. (20)

Input: $S \in \mathbb{R}^{n\times n}$, $\mu \in \mathbb{S}_+^k$, $\tau \in [0, 1]$, $t_{\max} = 4000$, $\delta_v = 10^{-4}$

Initialize $V^0 = \text{Sinkhorn}(\text{rand}(n, k), \mu)$ (Alg. 3), $t = 0$

$\gamma = -\lambda_{\max}(S) + \tau(\lambda_{\max}(S) - \lambda_{\min}(S))$

Calculate Lipschitz constant L according to Eq. (26)

repeat

  $V^{t+1} = \mathcal{P}_{\Omega(\mu)}(V^t - \nabla(V^t)/L)$ (Alg. 2)

  $t = t + 1$

until $t = t_{\max}$ ***or*** $\|V^{t+1} - V^t\|_F / \|V^t\|_F \leq \delta_v$;

4.2 Projection onto $\Omega(\mu)$

In Alg. 1, the projection onto $\Omega(\mu)$ is a crucial step, but the closed-form expression for $\mathcal{P}_{\Omega(\mu)}(U)$ is difficult to derive. To this end, we adopt the Dykstra algorithm (Boyle and Dykstra, 1986) to compute $\mathcal{P}_{\Omega(\mu)}(U)$, which is a powerful tool for solving projections onto the intersection of convex sets, provided that the projection onto each convex set can be easily computed. Indeed, $\Omega(\mu)$ can be seen as the intersection of two convex sets: $\mathbb{R}_+^{n\times k} := \{V \in \mathbb{R}^{n\times k} | V \geq 0\}$ and $\Omega_0(\mu) := \{V \in \mathbb{R}^{n\times k} | V^T 1_n = \mu, V\mu = 1_n/n\}$, each of which has a closed-form expression of projector as follows:

Lemma 6 *For any $U \in \mathbb{R}^{n\times k}$, we have:*

$$\mathcal{P}_{\mathbb{R}_+^{n\times k}}(U) = \max(U, 0) \quad \text{and} \quad \mathcal{P}_{\Omega_0(\mu)}(U) = U + \frac{1_n^T U\mu + 1}{n} 1_n \mu^T - \frac{1_n 1_n^T}{n} U - U\mu\mu^T. \tag{29}$$

Based on this, the modified Dykstra algorithm is proposed in Alg. 2, where $b_{\max}$ and δ_d are predefined maximum iteration count and convergence tolerance, respectively, and $\min(V)$ represents the minimal element of V. In Alg. 2, we use an adaptive step strategy (Combettes and Pesquet, 2009) to accelerate convergence: as the iteration count b grows from 0 to ∞, the step size β grows from 1 to 2.

Algorithm 2: Modified Dykstra algorithm for solving $\mathcal{P}_{\Omega(\mu)}(U)$

Input: $U \in \mathbb{R}^{n\times k}$, $\mu \in \mathbb{S}_+^k$, $b_{\max} = 1000$, $\delta_d = 10^{-5}$
Output: V

Initialize $V = U,\ Z = \text{zeros}(n,k),\ b = 0,\ \alpha_b = 1$
repeat
  $b = b + 1$
  $\alpha_{b+1} = \frac{1}{2}\left(1 + \sqrt{4\alpha_b^2 + 1}\right)$
  $\beta = 1 + \frac{1-\alpha_b}{\alpha_{b+1}}$
  $Y = (1-\beta)V + \beta \max(V - Z, 0)$
  $Z = Y - V + Z$
  $V = (1-\beta)Y + \beta\left[Y + \frac{1_n^T Y \mu}{n} 1_n \mu^T - \frac{1_n 1_n^T}{n} Y - Y\mu\mu^T\right]$
until $b = b_{\max}$ ***or*** $-\min(V) \le \delta_d \min(\max(\mu), 1/(n\min(\mu)))$;
$V = \max(V, 0)$

4.3 Initialization Method

Given a strictly positive matrix $U \in \mathbb{R}^{n\times k}$, the Sinkhorn-Knopp algorithm (Sinkhorn, 1964) seeks two diagonal matrices D_r, D_l such that $D_r U D_l \in \Pi(\alpha, \beta)$. Motivated by the relationship between $\Omega(\mu)$ and $\Pi(1_n/n, \mu \odot \mu)$, we first apply the Sinkhorn-Knopp algorithm to generate a random $P \in \Pi(1_n/n, \mu \odot \mu)$, and then normalize $V^0 = P\text{Diag}^{-1}(\mu)$ to obtain a random $V^0 \in \Omega(\mu)$. The pseudocode in the MATLAB syntax is provided in Alg. 3.

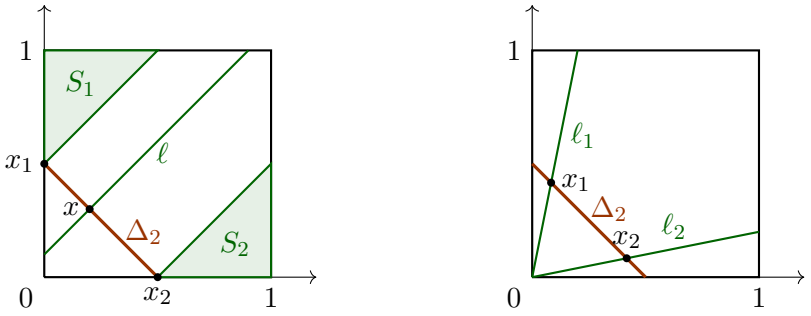
Algorithm 3: Sinkhorn-Knopp algorithm for normalization $U \in \mathbb{R}^{n\times k}$ into $\Omega(\mu)$

Input: $U \in \mathbb{R}^{n\times k}$, $\mu \in \mathbb{S}_+^k$, $s_{\max} = 1000$, $\delta_s = 10^{-16}$
Output: V

Initialize $\ell = 1_n,\ s = 0$
$P = \max(U \odot \mu^T, 10^{-20})$ `// Make sure` P `is strict positive`
repeat
  $s = s + 1$
  $r = (\mu \odot \mu) \oslash (P^T \ell)$
  $\ell = (1_n/n) \oslash (Pr)$
until $s = s_{\max}$ ***or*** $\max(\max(|(P^T\ell) \odot r - \mu \odot \mu|), \max(|(Pr) \odot \ell - 1_n/n|)) \le \delta_s$;
$P = \text{Diag}(l) \odot P\text{Diag}(r)$
$V = P\text{Diag}^{-1}(\mu)$

4.4 Why not the Dykstra Algorithm for Initialization?

One might wonder why the Dykstra algorithm is not used to initialize V^0, given that it is already a tool for projecting any $U \in \mathbb{R}^{n\times k}$ onto $\Omega(\mu)$. Since the objective functions of LoRD in Eq. (17) and B-LoRD in Eq. (20) are non-convex, we perform multiple initializations of $V^0 \in \Omega(\mu)$ to mitigate the risk of converging to poor stationary points. Ideally, these initializations would be drawn uniformly from $\Omega(\mu)$. However, to the best of our knowledge, uniformly sampling from the set of doubly stochastic matrices remains an open problem (Cappellini et al., 2009).



(a) Dykstra algorithm (b) Sinkhorn-Knopp algorithm

Figure 2: The schematic diagram of the initialization method.

In our experiment, we found that many initializations based on the Dykstra algorithm converge to poor stationary points, i.e., results corresponding to high objective function values and low clustering performance. In contrast, initializations based on the Sinkhorn-Knopp algorithm are more likely to avoid poor stationary points. The reasons might be as follows. We simplify $\Omega(\mu)$ by setting $n = 2$ and $k = 1$, so $\Omega(\mu)$ becomes the two-dimensional simplex $\Delta_2 = \{v \in \mathbb{R}^2 \mid v \geq 0, v(1) + v(2) = 0.5\}$. In Fig. 2a, suppose x is an interior point of Δ_2, i.e., $x > 0, x(1)+x(2) = 0.5$, and $x_1 = (0, 0.5)$ and $x_2 = (0.5, 0)$ are the boundaries of Δ_2. The probability density of sampling x is proportional to the length of the corresponding line segment ℓ (passes through x and has a slope of 1). However, the probability density of sampling x_1 and x_2 is proportional to the area of the corresponding regions S_1 and S_2, respectively. As a result, the Dykstra algorithm-based initialization has a high probability of obtaining a boundary point. In Fig. 2b, the probability density of sampling $x_1, x_2 \in \Delta_2$ is proportional to the length of the corresponding line segment ℓ_1, ℓ_2, respectively, (passes through x_1, x_2 and $(0, 0)$). Therefore, the Sinkhorn-Knopp algorithm-based initialization can generate more uniform samples.

4.5 Convergence Analysis

The convergence of Alg. 1 is described as follows.

Lemma 7 (Beck, 2017, Theorem 10.15). *Suppose $f(V)$ is gradient L-Lipschitz continuous, and $\Omega(\mu) \subseteq \mathbb{R}^{n\times k}$ is closed, convex and nonempty. Let V^* be a global optimum of Eq.* (17)

or Eq. (20)*. At the t-th iteration of Alg. 1, the following inequality holds:*

$$\min_{0\le i\le t}\|V^{i+1}-V^i\|_F \le \sqrt{\frac{2}{L}\frac{f(V^0)-f(V^*)}{t+1}}. \tag{30}$$

Lemma 7 states the convergence condition $\frac{\|V^{t+1}-V^t\|_F}{\|V^t\|_F} \le \delta_v$ is always satisfied when t is sufficiently large.

For the exact projected gradient iteration, we can further establish the following first-order stationarity result.

Lemma 8 (Li and Lin, 2015, Theorem 1). *Let $f(V)$ denote the objective function of Eq.* (17) *or Eq.* (20)*, and assume that ∇f is L-Lipschitz continuous on $\Omega(\mu)$. Recall that $\Omega(\mu)$ is nonempty by Theorem 1; moreover, it is closed, convex, and compact. Consider the exact projected gradient iteration*

$$V^{t+1} = \mathcal{P}_{\Omega(\mu)}\left(V^t - \nabla f(V^t)/L\right). \tag{31}$$

Then every accumulation point $\bar{V}$ of $\{V^t\}_{t\ge 0}$ is a first-order stationary point of

$$\min_{V\in\Omega(\mu)} f(V), \tag{32}$$

namely,

$$\langle \nabla f(\bar{V}), V-\bar{V}\rangle \ge 0, \quad \forall V\in\Omega(\mu), \tag{33}$$

or equivalently,

$$0 \in \nabla f(\bar{V}) + N_{\Omega(\mu)}(\bar{V}), \tag{34}$$

where $N_{\Omega(\mu)}(\bar{V})$ denotes the normal cone of $\Omega(\mu)$ at $\bar{V}$.

4.6 Complexity Analysis

Under the general setting of graph-based clustering, i.e., S is an $n\times n$ matrix without special structure, the calculation of $\nabla(V)$ requires $\mathcal{O}(n^2k)$ complexity. To solve $\mathcal{P}_{\Omega(\mu)}(U)$, the calculations of Y, Z, V in Alg. 2 have $\mathcal{O}(nk)$ complexity per iteration. Therefore, the complexity of Alg. 2 is $\mathcal{O}(nkb_{\text{avg}})$, where b_{avg} is the mean number of iterations of Alg. 2. The value of b_{avg} is approximately 50 in most cases of our experiments A.4.

In practice, to avoid $\mathcal{O}(n^2)$ complexity, S is typically constructed as a sparse q-nearest neghbor (q-NN) graph (Hou et al., 2022; Park and Kim, 2017; Wang et al., 2016), where q is set to $\lfloor \log_2(n)\rfloor + 1$ as sugessted by (Von Luxburg, 2007). Under this setting, the calculation of $\nabla(V)$ only requires $\mathcal{O}(n\log(n)k)$ complexity, and storing S and V only require $\mathcal{O}(n\log n)$ and $\mathcal{O}(nk)$ memory, respectively. Therefore, Alg. 1 requires $\mathcal{O}(nk(\log n + b_{\text{avg}}))$ time complexity per iteration and $\mathcal{O}(n(\log n + k))$ memory. Moreover, the Alg. 1 only involves matrix product operations, which enables well GPU compatibility and scalability for large-scale datasets.

Compared to the DSN-based block diagonal enhancement methods (Wang et al., 2016; Park and Kim, 2017; Julien, 2022) whose complexities are $\mathcal{O}(n^3)$, our Alg. 1 is more efficient benefiting from the low-rank structure of VV^T.

Table 1: Dominant optimization costs after graph construction.

Method	Type	Dense Graph	Sparse Graph	Notation
k-means		$O(nmk)$	$O(nmk)$	
KKM		$O(n^2)$	–	
GKKM	nested iter.	$O(n^3kt)$	–	$t \approx 15$: inner KKM iters
SC	iteration-free	$O(n^3)$	$O(\|\mathcal{E}\|k)$	
NCut	iteration-free	$O(n^3)$	$O(\|\mathcal{E}\|k)$	
SR		$O(nk^2)$	$O(nk^2)$	Iterate after SC
DBSC		$O(n^2k + nk^2 + k^3)$	$O(\|\mathcal{E}\|k + nk^2)$	
DirectSC		$O(n^2kt)$	$O(\|\mathcal{E}\|kt)$	$t \approx 5k$: Lanczos iterations
SymNMF		$O(n^2k)$	$O(\|\mathcal{E}\|k + nk^2)$	
PHALS		$O(n^2k + nk^2)$	$O(\|\mathcal{E}\|k + nk^2)$	
S^3NMF		$O((n^2k + nk^2)b)$	$O((\|\mathcal{E}\|k + nk^2)b)$	$b \approx 20$: initial factors
NLR	nested iter.	$O(n^2rt)$	$O(\|\mathcal{E}\|rt)$	$r \approx 20k$: factor rank $t \approx 100$: primal steps
DSN		$O(n^2)$	$O(n^2)$	
DvD		$O(n^3)$	$O(n^3)$	
DSNI		$O(n^3)$	$O(n^3)$	
DSDC	scaling	$O(n^2b + nm^2 + m^3)$	–	$b \approx 100$: scaling iters
SDP		$O(n^{3.5})$	–	
DCD		$O(n^2k)$	$O(\|\mathcal{E}\|k)$	
LoRD	nested iter.	$O(n^2k + nkb)$	$O(\|\mathcal{E}\|k + nkb)$	$b \approx 50$: Dykstra iters
B-LoRD	nested iter.	$O(n^2k + nkb)$	$O(\|\mathcal{E}\|k + nkb)$	$b \approx 150$: Dykstra iters

All costs exclude graph construction. For iterative methods we report per-iteration complexity; for iteration-free methods we report the dominant one-shot cost.
$|\mathcal{E}|$ denotes the number of nonzeros in the sample graph. For the q-NN graphs used in our paper, $|\mathcal{E}| \approx nq$ with $q = \lfloor \log_2(n) \rfloor + 1$, typically about 8–13 on our datasets.
Reported parameter values are taken from the original papers or public code when available; otherwise they are approximate.

To further position our method relative to representative baselines, Table 1 summarizes the dominant optimization costs after graph construction. The table makes clear that LoRD and B-LoRD are not intended to outperform vanilla k-means in raw speed, but they avoid the cubic cost of heavier DSN/SDP-style relaxations while preserving more of the original clustering structure.

5. Experiments

5.1 Experimental Settings

Datasets. We adopted 12 datasets as described in Table 2. As our methods require the input of a prior class probability μ, we divide the datasets into six class-balanced datasets and six class-imbalanced datasets with varying imbalance rates (IBR) defined in Eq. (36) for better analysis.

Compared methods. We compared the proposed methods with the following methods:
Kernel k-means-based methods:

- Kernel k-means (KKM) (Dhillon et al., 2004): Approximately solves the kernel k-means problem Eq. (2) with multiple random restarts.

Table 2: Descriptions of Datasets

Code	Dataset	Dimension	# Sample (n)	# Cluster (k)	IBR
D1	YaleB	32×32	165	15	0
D2	ORL	32×32	400	40	0
D3	CHART	60	600	6	0
D4	USPS-1000	16×16	1000	10	0
D5	Isolet	617	7797	26	0
D6	COIL100	32×32	7200	100	0
D7	Semeion	16×16	1593	10	3×10^{-5}
D8	MNIST	28×28	70000	10	0.0006
D9	MNIST-2000	28×28	2000	10	0.0014
D10	Wine	13	178	3	0.0114
D11	Yeast	8	1484	10	0.2503
D12	Ecoli	7	336	8	0.2704

- Global kernel k-means (GKKM) (Tzortzis and Likas, 2009): A deterministic algorithm for solving KKM that uses an incremental approach to obtain clustering results, making it more likely to avoid poor local minima and find a near-optimal solution.

Spectral clustering (SC)-based methods:

- Spectral clustering (SC) (Alg. 3 in (Von Luxburg, 2007)): A relaxation of kernel k-means that only keeps the orthogonality constraint.
- Normalize Cut (NCut) (Shi and Malik, 2000): A variant of SC that transforms graph partitioning into solving the eigenvectors of the normalized graph Laplacian matrix to achieve optimal segmentation by minimizing inter-class similarity and maximizing intra-class similarity.
- Spectral rotation (SR) (Huang et al., 2013): An improvement of SC. Instead of post-processing via k-means, SR imposes an additional orthonormal constraint to better approximate the optimal continuous solution.
- Discrete and balanced spectral clustering (DBSC) (Wang et al., 2023): An improvement of SC, which can jointly learn the spectral factor and clustering result, with an adjustable balance rate for clusters.
- Direct spectral clustering (DirectSC) (Nie et al., 2024): An improvement of SC, which can adaptively learn an improved similarity graph as well as the corresponding spectral factor from an initial low-quality similarity graph. Both the learned similarity graph and spectral factor can be used to directly obtain the final clustering result.

SymNMF-based methods:

- SymNMF (Kuang et al., 2012, 2015): A relaxation of kernel k-means that only keeps nonnegative constraint. The multiplicative update algorithm described in (Long et al., 2007) is applied to solve Eq. (5).
- PHALS (Hou et al., 2022): An efficient algorithm to solve SymNMF.

- Self-supervised SymNMF (S^3NMF) (Jia et al., 2021): Progressively boosts the quality from an initial low-quality similarity matrix by combining multiple class assignment matrices.
- NLR (Zhuang et al., 2024): A non-convex Burer-Monteiro factorization approach to solve the (kernel) k-means problem.

Doubly stochastic normalization (DSN)-based methods:

- Doubly stochastic normalization (DSN) (Zass and Shashua, 2006): A relaxation of kernel k-means that relaxes the orthogonality constraint and low-rank structure.
- Structured doubly stochastic clustering (SDS) (Wang et al., 2016): A DSN method with an enhanced block diagonal structure by incorporating the block diagonal regularization $\|Z\|_{\boxed{k}}$.
- DvD (Park and Kim, 2017): A DSN method with an enhanced block diagonal structure based on Davis-Kahan theorem.
- DSNI (Julien, 2022): A DSN method with an enhanced block diagonal structure by incorporating idempotent regularization of Z.
- Doubly stochastic distance clustering (DSDC) (He and Zhang, 2023): A scalable method that replaces doubly stochastic similarity matrix with a doubly stochastic Euclidean matrix.

Others graph-based clustering methods:

- SDP (Peng and Wei, 2007; Kulis et al., 2007): A convex relaxation of kernel k-means that relaxes the idempotency constraint and low-rank structure.
- DCD (Yang and Oja, 2012; Yang et al., 2016): Repleace the L_2-norm with KL divergence to measure the discrepancy between the input similarity matrix and the learned similarity matrix.

Construction of S. We construct the similarity matrix S for each dataset using the q-nearest neighbors (q-NN) graph weighted with the self-tuning method (Zelnik-Manor and Perona, 2004). Let $x_i \in N_q(x_j)$ represent the sample x_i that belongs to the q-NN of x_j, S is defined as:

$$S_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|_2^2}{\sigma_i \sigma_j}\right), & \text{if } x_i \in N_q(x_j) \text{ or } x_j \in N_q(x_i) \\ 0, & \text{otherwise} \end{cases}, \tag{35}$$

where σ_i is set to the Euclidean distance between x_i and its 7-th nearest neighbor, and q is set to be $\lfloor \log_2(n) \rfloor + 1$ sugessted by (Von Luxburg, 2007).

Additionally, in LoRD, we normalize $S \leftarrow S/1_n^T S 1_n$ because $\forall V \in \Omega, 1_n^T V V^T 1_n = 1$; in KKM and GKKM, we use a fully connected graph (i.e., set $q = n$) because they require S to be positive semidefinite. More precisely, the classical kernel k-means interpretation relies on a kernel similarity matrix and thus on the positive semidefiniteness of S. By contrast,

LoRD and B-LoRD are motivated by the relaxation perspective of kernel k-means but are ultimately used here as graph-based clustering models. Once we move to the relaxed formulations, the optimization problem and the projected-gradient solver only require a symmetric affinity matrix that encodes pairwise relations, rather than a PSD kernel matrix.

Initialization method. For GKKM, SDS, SC, DirectSC, DSN, SDS, DSNI, and DSDC methods, no random initialization is required, and they only need the constructed S (GKKM, SC, DirectSC, DSN, SDS, and DSNI) or X (DSDC) as input. For KKM, DCD, NLR and S^{3}NMF, we adopt initialization methods provided in the original papers. For the other methods, we first generate $V \in \mathbb{R}^{n\times k}$ with elements uniformly sampled in the range $[0, 1]$, and

- In LoRD and B-LoRD, we use $V^0 = \text{Sinkhorn}(V)$ (described in Alg. 3) to normalize V onto $\Omega(\mu)$.
- In SymNMF and PHALS, we normalize $V^0 = \frac{\sqrt{\langle S, VV^T\rangle}}{\|VV^T\|_F} V$.
- In SR and DBSC, we binarize $V^0_{ij} = 1$ if $j = \arg\max_{\hat{j}} V_{i\hat{j}}$, and 0 otherwise, for each i-th row of V^0.

Result selection. For each method that requires random initialization, we run 50 initializations and report the result corresponding to the minimal or maximal objective function value. The SDP, DSN, SDS, DvD, and DSNI methods require SC as post-processing to obtain the clustering result, so we run SC 50 times and report the average performance for these methods.

Hyper-paramaters tuning. For a fair comparison, the hyper-paramaters tuning methods for DCD, DBSC, DirectSC, S^{3}NMF, SDS, DvD, DSNI, DSDC are provided in the original papers. For KKM, GKKM, SC, SR, SymNMF, PHALS, DSN, and LoRD, there are no hyper-parameters to tune. For NLR, we carefully tune the hyper-parameter α and β to satisfy the constraint $VV^T 1_n = 1_n$ and guarantee convergence. For the proposed B-LoRD, we use τ tuned in $\{0.01, 0.02, \ldots, 1\}$ to calculate γ.

For simplicity, we present a subset of results here. We refer readers to the *Appendix* for detailed synthetic experiments in A.1, hyperparameter analysis in Appendix 5.4, and convergence analysis in Appendix 5.5.

5.2 Clustering Results

Table 3 shows the clustering performance in terms of ACC of all methods. We also refer readers to Appendix A.3 for the results in terms of other metrics, including NMI, PUR, and F1. From these results, we can observe that

- Our B-LoRD significantly outperforms the compared methods, achieving the highest ACC values in most cases (35/48), and the second highest in 5/48 cases.
- Our LoRD outperforms the hyper-parameter-free methods (KKM, SC, SR, SymNMF, PHALS, and DSC) and is competitive with block diagonality enhanced methods (NLR, DvD, and DSNI).

Table 3: Comparisons of clustering performance in terms of ACC. The last column refers to the average ACC of the nine datasets, excluding Isolet, COIL100, and MNIST. The best and second-best results are highlighted in **bold** and underlined, respectively.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	Avg.
KKM	0.485	0.588	0.835	0.494	0.547	0.473	0.572	0.657	0.606	0.944	0.297	0.569	0.599
GKKM	0.339	0.488	0.568	0.507	0.609	0.238	0.536	–	0.595	0.944	0.317	0.640	0.548
SDP	0.504	0.665	0.680	0.539	–	–	0.670	–	0.669	0.916	0.330	0.538	0.603
DCD	0.442	0.645	0.570	0.539	0.534	0.522	0.564	0.682	0.684	0.961	0.309	0.595	0.590
SC	0.466	0.640	0.568	0.542	0.542	0.589	0.665	0.682	0.682	0.949	0.333	0.533	0.598
SR	0.467	0.638	0.568	0.535	0.530	0.547	0.553	0.670	0.677	0.949	0.371	0.610	0.596
NCut	0.447	0.623	0.568	0.521	0.522	0.511	0.546	0.674	0.649	0.949	0.378	0.606	0.587
DBSC	0.473	0.658	0.842	0.552	0.542	0.545	0.618	–	0.625	0.944	0.364	0.539	0.624
DirectSC	0.429	0.598	0.645	0.464	0.432	–	0.458	–	0.451	0.899	0.359	0.631	0.548
SymNMF	0.473	0.645	0.842	0.549	0.537	0.493	0.615	0.525	0.641	0.916	0.334	0.524	0.615
PHALS	0.473	0.645	0.800	0.520	0.546	0.527	0.630	0.610	0.644	0.927	0.362	0.518	0.613
S^3NMF	0.475	0.630	0.810	0.623	–	–	0.704	–	0.664	0.935	0.348	0.578	0.641
NLR	0.491	0.649	0.645	0.418	0.352	–	0.497	–	0.457	0.938	0.364	0.649	0.568
DSN	0.469	0.640	0.568	0.547	0.540	0.588	0.664	–	0.681	0.949	0.328	0.534	0.598
SDS	0.503	0.658	0.847	0.574	–	–	0.693	–	0.666	0.961	0.387	0.711	0.667
DvD	0.442	0.601	0.608	0.510	–	–	0.517	–	0.451	**0.966**	0.367	0.616	0.564
DSNI	0.465	0.632	0.563	0.610	–	–	0.670	–	0.695	0.899	0.325	0.589	0.605
DSDC	0.405	0.549	0.602	0.478	0.561	0.518	0.619	0.557	0.579	0.888	0.324	0.550	0.555
LoRD (ours)	0.467	0.655	0.878	0.606	0.593	0.496	0.755	0.943	0.657	0.944	0.303	0.455	0.636
B-LoRD (ours)	**0.515**	**0.685**	**0.905**	**0.740**	**0.644**	**0.647**	**0.783**	**0.964**	**0.745**	0.955	**0.412**	**0.741**	**0.720**

- On the balanced datasets, our B-LoRD achieves the highest ACC values in most cases (20/24), and the second highest in the rest (4/24), due to its inherent advantage ($\mu_0 = \mu^*$) in this case.
- On the imbalanced datasets, our B-LoRD still outperforms the compared methods. For example, on the Yeast and Ecoli dataset with the highest IBR, B-LoRD achieves the highest ACC, PUR, and F1. These results show that B-LoRD is robust to imbalanced datasets. More results can be found in Appendix 5.4.
- The block diagonality enhanced methods outperform the others, especially our B-LoRD and SDS, as they both apply k-block diagonal regularization Eq. (7). Compared to SDS, our B-LoRD exploits the low-rank doubly stochastic matrix to simplify the computation, thereby improving computational efficiency.

The corresponding observed runtime statistics are reported in Appendix A.2, where the mean and standard deviation tables further clarify the intended computational positioning.

5.3 Correlation between Objective Function Value and ACC

Fig. 3 and Table 4 show the correlation between the objective function values of models and their ACCs (the full results are available in Appendix A.3). Since different methods optimize different relaxations and objective functions, their raw objective values are not directly comparable across methods. Accordingly, we use the objective values only to evaluate the within-method alignment between the attained objective and the resulting clustering quality, rather than as a cross-method performance metric.

Table 4: The R^2 between the objective function value and the ACC learned by 50 initializations. The values higher than 0.5 are bold.

R^2	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	Avg.
KKM	0.08	**0.53**	**0.75**	0.23	0.32	**0.62**	0.24	0.47	0.31	**0.96**	0.12	−0.02	0.38(3)
SR	0.00	0.33	0.35	**0.94**	0.02	**0.72**	−0.83	NaN	0.31	NaN	0.01	0.27	0.21
SymNMF	0.37	0.21	0.21	0.18	0.00	0.09	0.30	−0.02	0.27	**0.75**	0.00	0.06	0.20
PHALS	0.02	0.28	0.47	−0.01	0.07	0.09	0.07	−0.16	0.12	**0.75**	0.07	−0.13	0.14
LoRD (ours)	0.01	0.02	**0.71**	0.40	**0.53**	**0.62**	**0.75**	**0.91**	0.42	**1.00**	0.24	0.09	0.48(2)
B-LoRD (ours)	0.14	0.31	**0.52**	0.37	**0.64**	**0.60**	**0.80**	**0.91**	**0.51**	**1.00**	0.45	0.00	0.49(1)

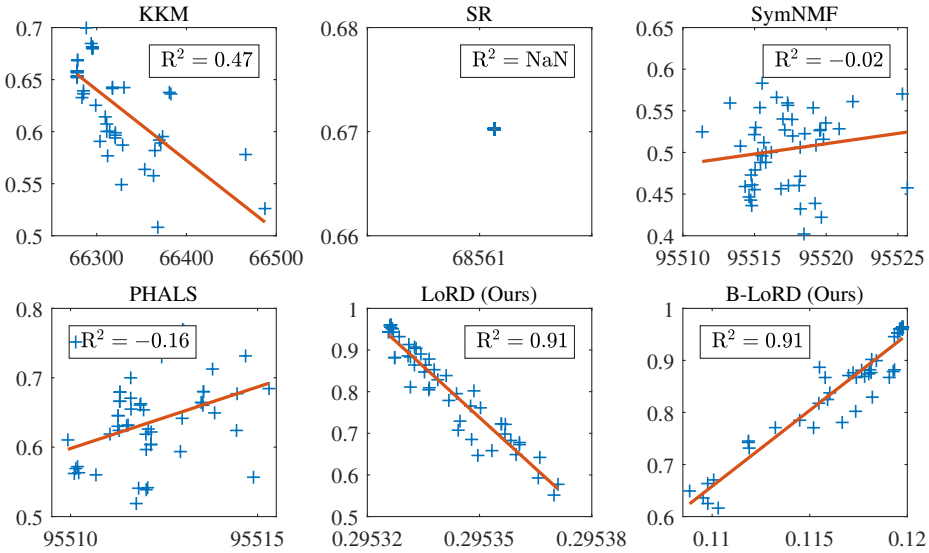


Figure 3: The relationship between the objective function value (x-axis) and the clustering ACC (y-axis) on the MNIST dataset, zoom in for details.

The strength of this correlation is quantitatively measured by the coefficient of determination (R^2). From these results, we observe that the objective function values of KKM, our LoRD, and our B-LoRD are highly correlated with the clustering performance, while SR, SymNMF, and PHALS are not. This may be because SR, SymNMF, and PHALS relax the double stochastic constraint, making V unable to represent clusters partition well. Compared to KKM, our LoRD and B-LoRD further reduce the optimization space by specifying class prior probability μ, which likely explains why the R^2s of LoRD and B-LoRD are higher than that of KKM.

5.4 Hyper-parameters Analysis

How does τ control the block diagonality of VV^T? To investigate how τ (used to calculate γ) controls the block diagonality of VV^T learned by B-LoRD, we visualize VV^T with different τ in Fig. 4, which can be found that:

- The block diagonality of VV^T increases as τ increases, with the rate of increase varying across datasets, which may be related to some properties of S.

Table 5: The gap of ACC on imbalanced datasets.

Datasets IBR	Semeion 3×10^{-5}	MNIST-2000 0.0014	Wine 0.0114	Yeast 0.2503	Ecoli 0.2704
LoRD-μ_0	0.755	0.657	0.944	0.303	0.455
LoRD-μ^*	0.755	0.669	0.916	0.385	0.717
LoRD-gap	0	0.008	−0.028	0.082	0.262
B-LoRD-μ_0	0.783	0.745	0.955	0.412	0.741
B-LoRD-μ^*	0.783	0.742	0.933	0.470	0.801
B-LoRD-gap	0	−0.003	−0.022	0.058	0.060

- When τ is sufficiently small, e.g., $\tau = 0.1$, the learned $V \approx 1_n \mu^T / n$ and $VV^T \approx 1_n 1_n^T / n^2$ with no block diagonality. In theory, this trivial solution always occurs when $\tau = 0$.
- When τ is large, the learned VV^T has high block diagonality, e.g., $\tau = 0.7$ on the ORL dataset. In particular, when $\tau = 1$, the learned VV^T is almost block diagonal, i.e., each row of the learned V has only one non-zero element.

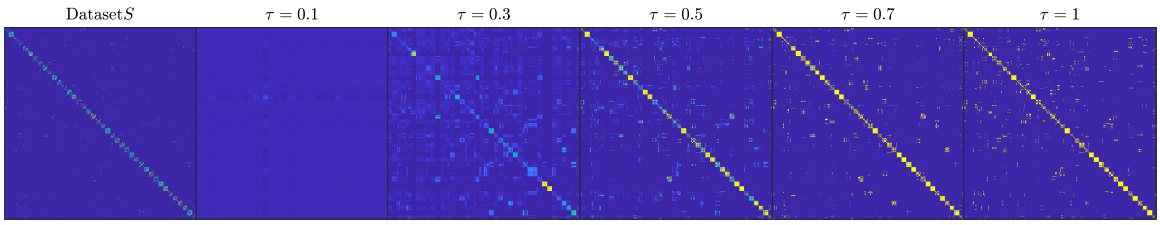


Figure 4: The visualization of learned VV^T (normalized to $[0, 1]$) with different τ of B-LoRD on the ORL datasets, zoom in for details.

The influence of τ on the ACC: The result is shown in Fig. 5, where $\tau_0 = \frac{\lambda_{\max}(S)}{\lambda_{\max}(S) - \lambda_{\min}(S)}$. When $\tau > \tau_0$, $\gamma > 0$ and the block diagonality is enhanced; when $\tau < \tau_0$, $\gamma < 0$ and the block diagonality is weakened. Moreover, we visualized the learned $Z = n^2 V \mathrm{Diag}(\mu \odot \mu) V^T$ of LoRD and B-LoRD on each dataset in Fig. 6, where the result of B-LoRD corresponds to the best τ achieving the highest ACC. From Fig. 5 and Fig. 6,

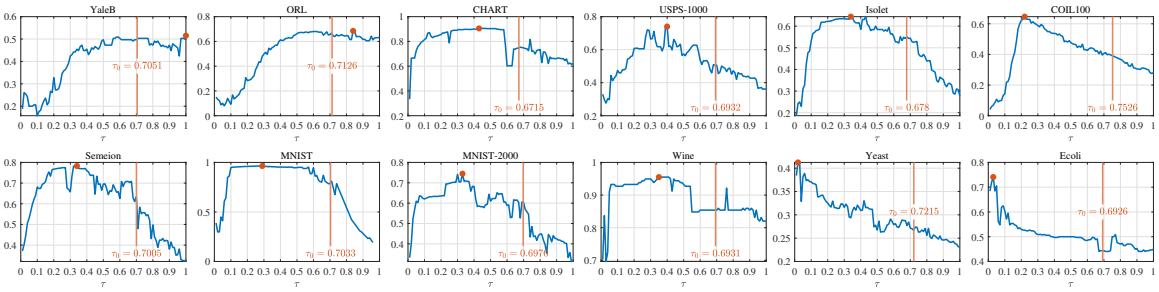


Figure 5: Values of ACC (y-axis) of B-LoRD with different values of τ (x-axis).

it can be seen that

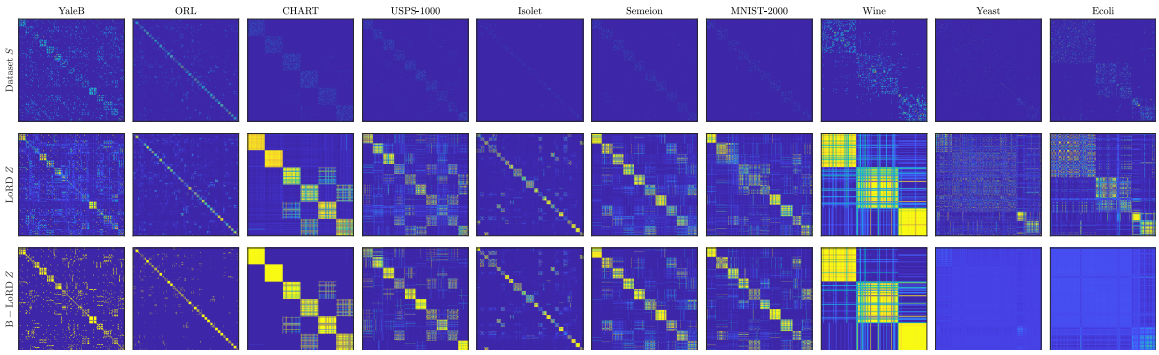


Figure 6: The visualization of learned Z (normalized to $[0, 1]$) of LoRD and B-LoRD on each datasets.

- When the dataset is balanced, the optimal τ (corresponding to the highest ACC) is generally high, and the learned Z exhibits high block-diagonality, especially on the YaleB, ORL, and CHART datasets.
- When the dataset is imbalanced, B-LoRD cannot find a suitable uniform partition, making small τ perform well, especially on the Yeast and Ecoli datasets.

Practical adaptive choosing strategy: As shown in Figure 5, ACC is very sensitive to τ, which requires an adaptive hyperparameter strategy. Here we provide two practical guideline or choosing τ as follows:

- **Using sample size n:** Empircal observations indicate that τ decreases as n increases, leading to the first approximation strategy $\hat{\tau}_1 = \min\{2n^{-0.24}, 1\}$.
- **Using n and block-diagonality of S:** Furthermore, τ decreases with lower block-diagonality of S. We quantify the block-diagonality using $b = \sum_{i=1}^{k} \lambda_i(L_S)/\mathrm{Tr}(L_S)$, where $L_S = \mathrm{Diag}(S\mathbf{1}_n) - S$ is the Laplacian of S. This metric can be efficiently computed when S is sparse. Combining b and n, we propose the second approximation $\hat{\tau}_2 = \min\{0.34 \exp(50b - 0.03 \log n), 1\}$.

The values of $\hat{\tau}_1$, $\hat{\tau}_2$ and τ^* on each dataset are shown in Table 6, where MAE means Mean Absolute Error.

Table 6: The values of $\hat{\tau}_1$, $\hat{\tau}_2$ and τ^*.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	MAE
$\hat{\tau}_1$	0.59	0.47	0.43	0.38	0.23	0.24	0.34	0.14	0.32	0.58	0.35	0.50	0.16
$\hat{\tau}_2$	0.83	0.95	0.28	0.29	0.26	0.26	0.28	0.24	0.27	0.30	0.28	0.30	0.13
τ^*	0.83	0.62	0.44	0.28	0.34	0.22	0.42	0.34	0.38	0.43	0.04	0.03	0

The clustering ACC of B-LoRD corresponding to $\hat{\tau}_1$, $\hat{\tau}_2$ and τ^* on each dataset are shown in Table 7.

From Table 6 and Table 7, it can be observed that when using $\hat{\tau}_1$ and $\hat{\tau}_2$, the MAE values are 0.16 and 0.13 respectively, while the average ACC decreases by only 0.05 and 0.045 accrodingly. These results suggest the effectiveness of the proposed adaptive strategies.

Table 7: The clustering ACC values of B-LoRD corresponding to $\hat{\tau}_1$, $\hat{\tau}_2$ and τ^*.

ACC	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	Avg.
$\hat{\tau}_1$	0.059	0.637	0.905	0.633	0.627	0.629	0.782	0.954	0.716	0.848	0.304	0.500	0.670
$\hat{\tau}_2$	0.485	0.628	0.890	0.719	0.633	0.608	0.693	0.962	0.702	0.949	0.321	0.509	0.675
τ^*	0.515	0.685	0.905	0.740	0.644	0.647	0.783	0.964	0.745	0.955	0.412	0.741	0.720

5.5 Convergence Analysis

The convergence behaviors of the proposed Alg. 1 are shown in Fig. 7 and Fig. 8, where Fig. 8 corresponds to the result with the optimal hyper-parameter. From these figures, we observe that the objective function value monotonically decreases in Fig. 7 and monotonically increases in Fig. 8, typically converging within a few hundred iterations.

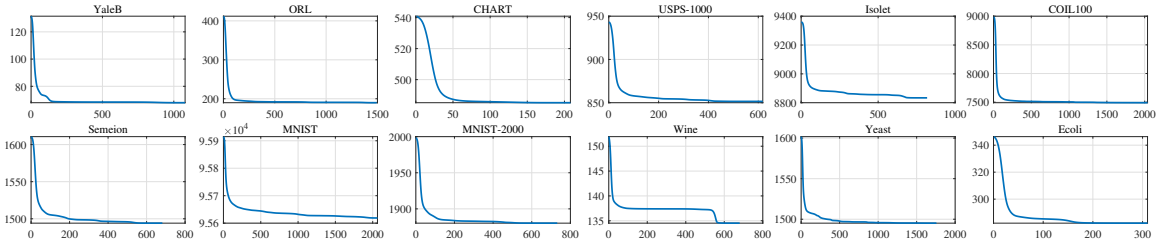


Figure 7: Convergence curves of LoRD on ten datasets. For each dataset, the x-axis represents the iteration count, and the y-axis represents the objective function of Eq. (17).

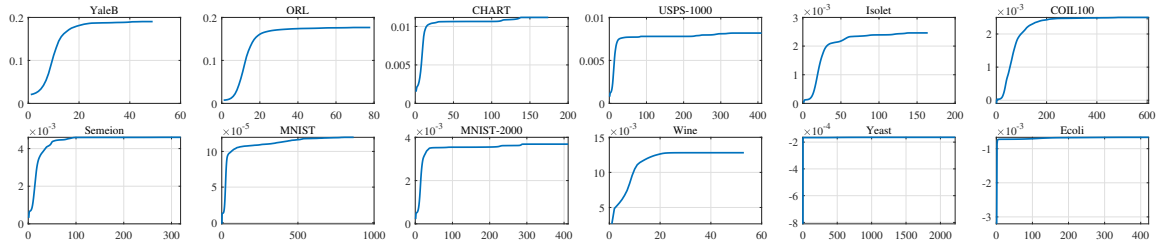


Figure 8: Convergence curves of B-LoRD on ten datasets. For each dataset, the x-axis represents the iteration count, and the y-axis represents the objective function of Eq. (20).

5.6 Robustness to Data Imbalance

In our experiments, we use $\mu_0 = [1/\sqrt{k}, \ldots, 1/\sqrt{k}]^T$ because $\mu^* = [\sqrt{\pi_1}, \ldots, \sqrt{\pi_k}]^T$ is unknown. This setting may lead to misleading clustering results when the dataset is significantly imbalanced. To analyze the performance gap of LoRD and B-LoRD between μ_0 and μ^* on imbalanced datasets, we define the imbalance rate (IBR) as:

$$\mathrm{IBR} = 1 - \mathcal{H}(\boldsymbol{\pi})/\log k, \tag{36}$$

where $\mathcal{H}(\boldsymbol{\pi}) = -\sum_{i=1}^{k} \boldsymbol{\pi}_i \log \boldsymbol{\pi}_i$ is the entropy of $\boldsymbol{\pi}$, the normalization factor $\log k$ ensures that $\mathrm{IBR} \in [0, 1]$.

As shown in Table 5, the performance gap generally increases as IBR increases. Meanwhile, B-LoRD is more robust to IBR than LoRD, because the block diagonality of VV^T can be adapted by tuning $\tau \in [0, 1]$ to alleviate this effect. Please see the detailed discussion in Appendix 5.4.

5.7 Synthetic Experiment

We generated 200 samples from four Gaussian distributions:

$$\mathcal{N}\left(\begin{bmatrix}-2\\2\end{bmatrix}, \begin{bmatrix}0.25 & 0\\0 & 0.25\end{bmatrix}\right), \quad \mathcal{N}\left(\begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right),$$
$$\mathcal{N}\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right), \quad \mathcal{N}\left(\begin{bmatrix}2\\-2\end{bmatrix}, \begin{bmatrix}2.25 & 0\\0 & 2.25\end{bmatrix}\right).$$

We set five different values of $\boldsymbol{\pi}$ to obtain different IBRs, as shown in Table 8.

Table 8: Clustering ACC of synthetic experiment.

$n\boldsymbol{\pi}$	IBR	GMM		LoRD		B-LoRD	
		μ^*	μ_0	μ^*	μ_0	μ^*	μ_0
$[50, 50, 50, 50]$	0	0.935	0.935	0.940	0.940	**0.960**	**0.960**
$[40, 50, 50, 60]$	0.0073	**0.960**	0.945	0.945	0.920	0.945	0.935
$[30, 45, 55, 70]$	0.0315	**0.925**	0.895	0.860	0.800	0.865	0.880
$[20, 40, 60, 80]$	0.0768	0.780	0.835	0.870	0.740	**0.890**	0.790
$[10, 30, 60, 100]$	0.1761	0.840	0.745	0.885	0.620	**0.900**	0.700

The clustering result for the case of $n\boldsymbol{\pi} = [20, 40, 60, 80]$ is plotted in Fig. 9, where the similarity matrix $S = X^T X$.

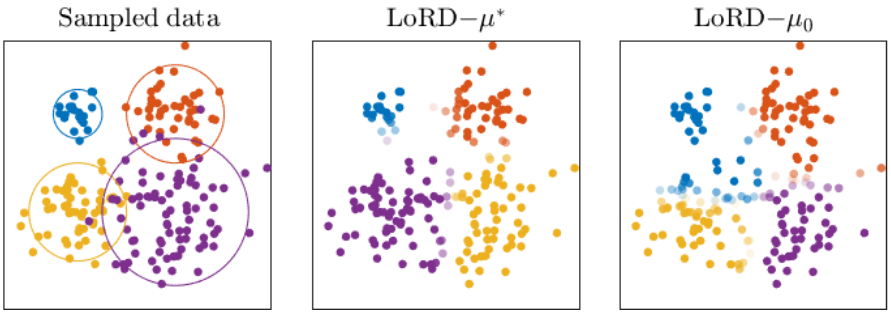


Figure 9: The sampled data and the clustering result of LoRD for the case of $n\boldsymbol{\pi} = [20, 40, 60, 80]$. The color and opacity of a data point represent its cluster and clustering probability, respectively.

From Table 8 and Fig. 9, we can observe that:

- Regardless of whether μ^* or μ_0 is given in LoRD, samples close to the cluster center have a high clustering probability, while samples at the intersection of multiple clusters have low clustering probability.
- When μ_0 deviates from μ^*, LoRD cannot find a suitable uniform partition, resulting in a low clustering probability for a large number of samples.

- As the IBR increases, the performance gap between LoRD and B-LoRD for given μ^* and μ_0 increases.

Please see the detailed results and analyses in Appendix A.1.

6. Conclusion and Discussion

In this paper, we have introduced LoRD, a novel graph-based clustering model, by relaxing the least crucial orthonormal constraint of kernel k-means, which is further enhanced by integrating adjustable block diagonality, leading to B-LoRD. To tackle numerical challenges, we theoretically elucidated how the non-convex doubly stochastic constraint can be reduced to a convex constraint by introducing the class probability parameter μ. Additionally, leveraging the gradient Lipschitz continuity property, we devised a projected gradient algorithm for the effective resolution of LoRD and B-LoRD, established a sublinear convergence-rate bound for its exact iteration, and ensured first-order stationarity of every accumulation point for the exact projected gradient iteration.

From a practical viewpoint, our goal is not to replace vanilla k-means, but to provide a more structure-preserving and still computationally manageable alternative within graph-based clustering. The added complexity and runtime analyses show that LoRD and B-LoRD are slower than vanilla k-means, as expected, yet substantially lighter than heavier DSN/SDP-style relaxations because they preserve the low-rank factorization throughout optimization.

Despite the effectiveness of LoRD and B-LoRD, a practical hurdle remains as μ^* is typically unknown in real-world applications. Moving forward, our research will delve into methods for estimating μ^* accurately to alleviate the impact of estimated biases on model performance. Another practical bottleneck is the projection onto $\Omega(\mu)$ via the modified Dykstra algorithm, which remains the main target for further acceleration.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants U24A20322, 62576094 and 62422118. This work is also supported by Hong Kong UGC under grants UGC/FDS11/E03/24, UGC/FDS11/E03/25, and Hong Kong Research Grants Council under Grant 11219324. This research work is also supported by the Big Data Computing Center of Southeast University.

Appendix A. Additional Experimental Results

A.1 Additional Synthetic Experiment

In Table 9, 10 and 11, we list the clustering performances of GMM, LoRD and B-LoRD in the synthetic experiment, respectively. Moreover, the clustering results of LoRD are shown in Fig. 10.

Table 9: Clustering performance of GMM in synthetic experiment.

s.p.c.	IBR	ACC			NMI			PUR			F1		
		μ^*	μ_0	gap	μ^*	μ_0	gap	μ^*	μ_0	gap	μ^*	μ_0	gap
$[50, 50, 50, 50]$	0	0.935	0.935	0	0.832	0.832	0	0.935	0.935	0	0.879	0.879	0
$[40, 50, 50, 60]$	0.0073	0.960	0.945	0.015	0.877	0.847	0.030	0.960	0.945	0.015	0.916	0.887	0.029
$[30, 45, 55, 70]$	0.0315	0.925	0.895	0.030	0.783	0.739	0.044	0.925	0.895	0.030	0.847	0.791	0.056
$[20, 40, 60, 80]$	0.0768	0.780	0.835	-0.055	0.672	0.708	-0.036	0.820	0.835	-0.015	0.747	0.739	0.078
$[10, 30, 60, 100]$	0.1761	0.840	0.745	0.095	0.657	0.506	0.151	0.840	0.745	0.095	0.713	0.623	0.090

Table 10: Clustering performance of LoRD in synthetic experiment.

s.p.c.	IBR	ACC			NMI			PUR			F1		
		μ^*	μ_0	gap	μ^*	μ_0	gap	μ^*	μ_0	gap	μ^*	μ_0	gap
$[50, 50, 50, 50]$	0	0.940	0.940	0	0.838	0.838	0	0.940	0.940	0	0.887	0.887	0
$[40, 50, 50, 60]$	0.0073	0.945	0.920	0.025	0.844	0.786	0.058	0.945	0.920	0.025	0.887	0.845	0.042
$[30, 45, 55, 70]$	0.0315	0.860	0.800	0.060	0.721	0.590	0.131	0.860	0.800	0.060	0.735	0.659	0.076
$[20, 40, 60, 80]$	0.0768	0.870	0.740	0.130	0.724	0.552	0.172	0.870	0.740	0.130	0.749	0.597	0.152
$[10, 30, 60, 100]$	0.1761	0.885	0.620	0.265	0.651	0.496	0.155	0.885	0.775	0.110	0.802	0.552	0.250

Table 11: Clustering performance of B-LoRD in synthetic experiment.

s.p.c.	IBR	ACC			NMI			PUR			F1		
		μ^*	μ_0	gap	μ^*	μ_0	gap	μ^*	μ_0	gap	μ^*	μ_0	gap
$[50, 50, 50, 50]$	0	0.960	0.960	0	0.884	0.884	0	0.960	0.960	0	0.923	0.923	0
$[40, 50, 50, 60]$	0.0073	0.945	0.935	0.010	0.861	0.840	0.021	0.945	0.935	0.010	0.888	0.871	0.017
$[30, 45, 55, 70]$	0.0315	0.865	0.880	-0.015	0.740	0.711	0.029	0.865	0.880	-0.015	0.743	0.770	-0.027
$[20, 40, 60, 80]$	0.0768	0.890	0.790	0.100	0.752	0.631	0.121	0.890	0.820	0.070	0.787	0.682	0.105
$[10, 30, 60, 100]$	0.1761	0.900	0.700	0.200	0.681	0.535	0.146	0.900	0.805	0.095	0.823	0.634	0.189

Additionally, in Table 9, 10 and 11, it can be seen that B-LoRD is more robust to the deviation between μ_0 and μ^*. To study its mechanism, we provide the hyper-parameter analysis of B-LoRD in the synthetic experiment, as shown in Fig. 11, and the clustering result under optimal hyper-parameter is shown in Fig. 12. From these results, we observe that:

- When μ^* is known, the B-LoRD achieves high ACC when τ is large, i.e., the learned VV^T exhibits high k-block diagonality.
- When μ_0 deviates from μ^*, B-LoRD cannot find a suitable uniform partition, resulting in better performance for a smaller τ. This is because, when the k-block diagonality of VV^T is weakened, the learned partition ratios do not strictly obey μ_0. For example, as shown in the third row of Fig. 12, the cases of $n\pi = [20, 40, 60, 80]$ and

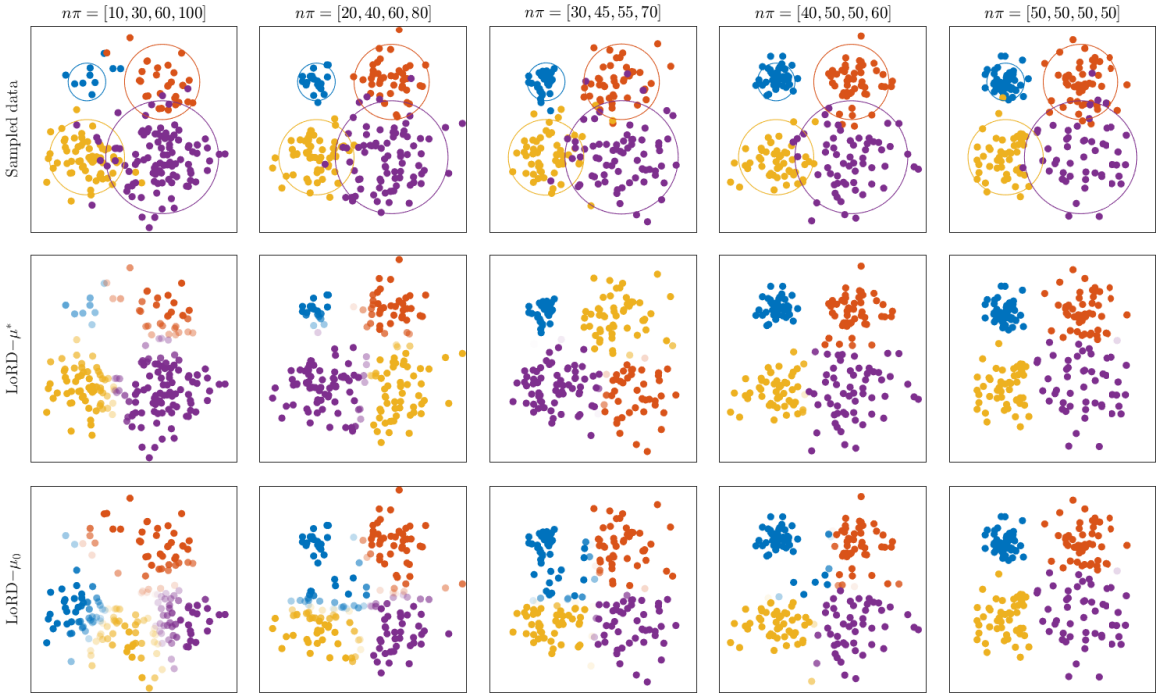


Figure 10: Clustering result of LoRD in synthetic experiment. The four clusters are marked with different colors, and the opacity is set to the cluster probability $\mathrm{P}(y_j|x_i)$.

$n\pi = [30, 45, 55, 70]$, the partition corresponding to the blue-colored cluster is almost correct. Moreover, in the case of $n\pi = [10, 30, 60, 100]$, the blue-colored cluster vanishes. Therefore, by tuning τ, B-LoRD can enhance or weaken the block diagonality, and reduce the impact of the deviation between μ_0 and μ^*.

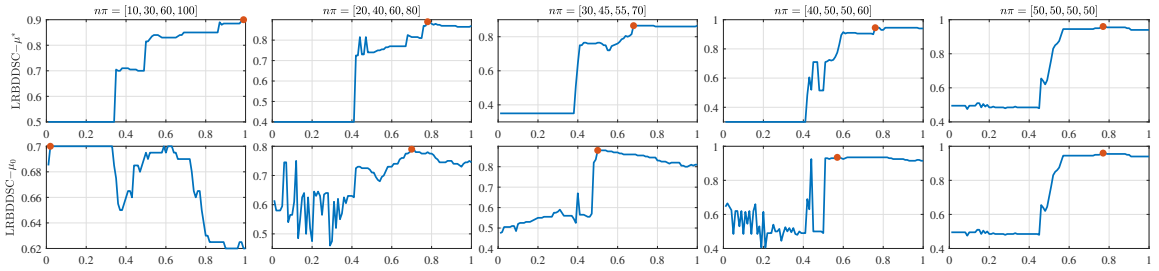


Figure 11: Values of ACC (y-axis) of B-LoRD with different values of τ (x-axis) in synthetic experiment. The optimal τ corresponding to highest ACC is marked with an orange point.

A.2 Runtime Analysis

To complement the asymptotic comparison in Table 1, we additionally report the observed wall-clock runtimes for the methods and datasets for which complete measurements are available, with the mean shown on the first line and the standard deviation shown below

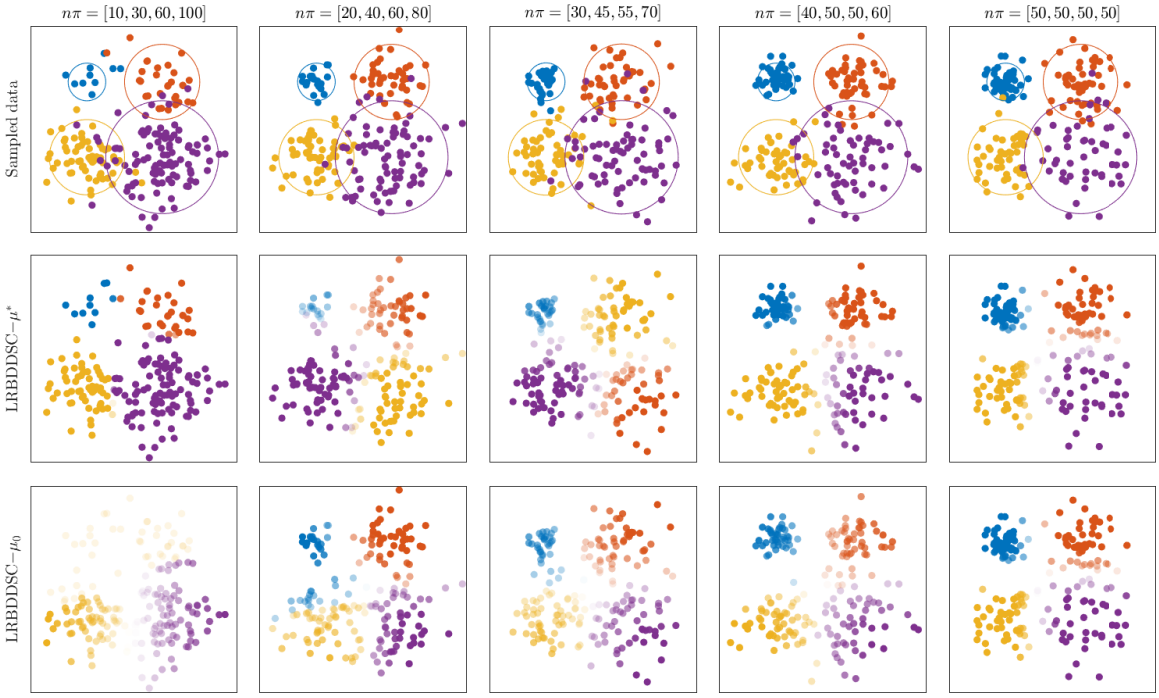


Figure 12: Clustering results of B-LoRD in synthetic datasets. The four clusters are marked with different colors, and the opacity is set to the cluster probability $\mathrm{P}(y_j|x_i)$.

as ± std. COIL100 and MNIST are omitted from this runtime table, and SDP/SDS are not included because those runs were not completed at the time of writing. All runtime measurements were obtained on Windows 10 with an Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz. For each method on each dataset, we ran 20 trials from different initializations and report the resulting mean and standard deviation. The initialization protocol follows the **Initialization method** paragraph in Sec. 5.1.

Taken together, Table 1 and Table 12 clarify the intended computational positioning. Vanilla k-means is consistently the fastest method, as expected. LoRD and B-LoRD are not meant to beat k-means in speed; rather, within the graph-based comparison class they trade additional runtime for a more structure-preserving formulation and stronger clustering performance. Their dominant extra practical cost remains the projection onto $\Omega(\mu)$ implemented by the modified Dykstra algorithm.

A.3 Clustering Results

The clustering NMI, PUR and F1 scores of all methods on each dataset are shown in Table 13, Table 14 and Table 15, respectively. The analyses of these results are consistent with the discussions in Sec. 5.2.

A.4 Complexity of Dykstra Algorithm 2

As analyzed in Sec. 4.6, the complexity of the Dykstra Algorithm 2 is $\mathcal{O}(nkb_{\text{avg}})$, where b_{avg} is the average iteration count. In this subsection, we summarize b_{avg} for LoRD and

Table 12: Observed wall-clock runtime (seconds): mean on the first line and ± standard deviation on the second line.

Method	YaleB	ORL	CHART	USPS-1000	Isolet	Semeion	MNIST-2000	Wine	Yeast	Ecoli
k-means	7.81 ms ±4.33 ms	11.1 ms ±3.31 ms	4.03 ms ±0.85 ms	9.57 ms ±1.95 ms	0.46 ±0.12	14.1 ms ±2.61 ms	76.8 ms ±20.5 ms	3.44 ms ±0.22 ms	5.34 ms ±1.79 ms	4.35 ms ±0.68 ms
KKM	1.67 ms ±0.51 ms	4.87 ms ±1.02 ms	19.4 ms ±8.37 ms	84.3 ms ±22.8 ms	8.7 ±3.05	0.38 ±0.3	0.59 ±0.22	1.47 ms ±0.75 ms	0.21 ±62.3 ms	4.34 ms ±1.66 ms
GKKM	23.6 ms ±2.02 ms	0.26 ±9.15 ms	0.12 ±13.8 ms	0.58 ±3.08 ms	153 ±4.15	2.59 ±2.02	5.33 ±0.36	7.61 ms ±0.13 ms	1.51 ±34 ms	46.6 ms ±0.45 ms
SDP	13.4 ±0.16	107 ±1.58	171 ±3.82	450 ±1.36	2.79 h ±15.9	577 ±1.51	928 ±27.0	6.62 ±15 ms	1090 ±2.95	22.9 ±0.23
SR	76 ms ±0.2	0.1 ±5.95 ms	26.1 ms ±1.38 ms	0.26 ±8.71 ms	16.7 ±0.1	0.52 ±11.2 ms	0.71 ±4.03 ms	11.1 ms ±1.06 ms	0.39 ±6.68 ms	24.2 ms ±2.47 ms
SymNMF	0.19 ±50 ms	0.54 ±0.15	0.1 ±21.5 ms	0.22 ±62.4 ms	2.75 ±0.5	0.22 ±44 ms	0.22 ±42.7 ms	69.6 ms ±26.8 ms	0.24 ±38.8 ms	0.12 ±27.6 ms
PHALS	53.2 ms ±18.3 ms	0.34 ±0.12	43 ms ±12 ms	92.8 ms ±28 ms	2.53 ±0.45	0.13 ±28.9 ms	0.15 ±34.4 ms	18.2 ms ±6.32 ms	0.14 ±32.8 ms	44.6 ms ±16.5 ms
S^3NMF	14.4 ±2.71	66.4 ±17.8	53.5 ±14.9	218 ±54.3	150 ±0.2	396 ±101	587 ±135	4.61 ±0.78	334 ±86.5	24.3 ±5
DSN	1.22 ±0.17	2.69 ±0.16	0.7 ±27.6 ms	1.84 ±54.6 ms	37.5 ±1.7	2.63 ±58.3 ms	3.45 ±48.7 ms	0.35 ±19.4 ms	3.61 ±83.7 ms	1.06 ±28.2 ms
SDS	5.06 ±0.12	34.5 ±0.98	65.4 ±0.89	133 ±1.95	7922 ±164	301 ±1.57	495 ±34.4	5.48 ±0.12	285 ±7.42	11.7 ±0.08
LoRD	0.66 ±0.29	5.44 ±2.17	1.21 ±0.9	4.56 ±2.62	122 ±41.9	6.65 ±2.48	9.54 ±3.34	0.12 ±0.05	7.42 ±3.01	0.7 ±0.36
B-LoRD	0.85 ±0.38	8.26 ±2.35	0.98 ±0.41	1.83 ±0.46	65.7 ±24.4	2.81 ±1.07	4.39 ±1.98	0.16 ±0.09	3.04 ±1.3	0.37 ±0.18

Table 13: NMI on each dataset.

NMI	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	Avg.
KKM	0.534	0.765	0.794	0.525	0.741	0.762	0.578	0.582	0.561	0.799	0.239	0.592	0.599
GKKM	0.415	0.692	0.788	0.543	0.763	0.622	0.534	–	0.565	0.822	0.239	0.618	0.580
SDP	0.555	0.807	0.819	0.576	–	–	0.590	–	0.612	0.763	0.260	0.550	–
DCD	0.513	0.797	0.806	0.592	0.758	0.807	0.620	0.746	0.680	0.863	0.251	0.619	0.638
SC	0.537	0.798	0.795	0.587	0.764	0.823	0.660	0.755	0.670	0.825	0.265	0.588	0.636
SR	0.534	0.794	0.795	0.579	0.765	0.814	0.607	0.743	0.665	0.825	0.278	0.632	0.634
NCut	0.520	0.796	0.800	0.590	0.761	0.793	0.614	0.750	0.647	0.825	**0.284**	0.619	0.634
DBSC	0.547	0.801	0.814	0.558	0.758	0.792	0.647	–	0.590	0.828	0.282	0.581	0.628
DirectSC	0.518	0.792	0.809	0.522	0.637	–	0.495	–	0.423	0.718	0.257	0.620	0.573
SymNMF	0.532	0.806	0.814	0.539	0.726	0.774	0.614	0.532	0.573	0.781	0.266	0.574	0.611
PHALS	0.534	0.803	0.767	0.528	0.755	0.798	0.647	0.689	0.609	0.798	0.279	0.575	0.616
S^3NMF	0.525	0.791	0.816	0.562	–	–	0.665	–	0.636	0.819	0.260	0.592	0.630
NLR	0.536	0.800	0.591	0.517	0.541	–	0.550	–	0.485	0.802	0.240	**0.800**	0.591
DSN	0.540	0.799	0.795	0.589	0.765	0.822	0.660	–	0.669	0.825	0.264	0.589	0.637
SDS	**0.580**	**0.835**	0.845	0.651	–	–	0.671	–	0.665	0.865	0.270	0.637	0.669
DvD	0.499	0.783	0.696	0.478	–	–	0.570	–	0.527	**0.877**	0.232	0.573	0.582
DSNI	0.540	0.806	0.805	0.647	–	–	0.655	–	**0.697**	0.696	0.263	0.610	0.635
DSDC	0.487	0.759	0.796	0.453	0.733	0.779	0.560	0.500	0.522	0.713	0.247	0.585	0.569
LoRD (ours)	0.518	0.798	0.827	0.579	0.758	0.774	0.681	0.883	0.614	0.807	0.256	0.530	0.623
B-LoRD (ours)	0.564	0.824	**0.850**	**0.678**	**0.794**	**0.831**	**0.696**	**0.910**	0.683	0.853	0.279	0.621	**0.672**

Table 14: PUR on each dataset.

PUR	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	Avg.
KKM	0.491	0.628	0.835	0.547	0.607	0.530	0.618	0.657	0.651	0.944	0.513	0.801	0.670
GKKM	0.364	0.568	0.667	0.560	0.643	0.339	0.553	–	0.646	0.944	0.497	0.813	0.624
SDP	0.516	0.687	0.762	0.604	–	–	0.646	–	0.677	0.916	0.526	0.797	–
DCD	0.449	0.675	0.667	0.608	0.604	0.602	0.647	0.738	0.733	0.961	0.503	0.830	0.675
SC	0.467	0.669	0.667	0.615	0.608	0.648	0.714	0.750	0.732	0.949	0.519	0.826	0.684
SR	0.467	0.670	0.667	0.609	0.601	0.636	0.635	0.737	0.727	0.949	0.514	0.839	0.675
NCut	0.453	0.667	0.667	0.596	0.593	0.596	0.621	0.737	0.699	0.949	0.516	0.834	0.675
DBSC	0.473	0.685	0.842	0.594	0.612	0.599	0.689	–	0.663	0.944	0.549	0.824	0.696
DirectSC	0.452	0.647	0.667	0.497	0.489	–	0.505	–	0.480	0.899	0.520	0.827	0.610
SymNMF	0.479	0.690	0.842	0.588	0.587	0.556	0.670	0.586	0.674	0.916	0.530	0.824	0.690
PHALS	0.479	0.683	0.800	0.567	0.616	0.588	0.690	0.676	0.690	0.927	0.553	0.819	0.690
S^3NMF	0.485	0.664	0.817	0.629	–	–	0.712	–	0.678	0.935	0.539	0.827	0.698
NLR	0.503	0.687	0.645	0.502	0.430	–	0.560	–	0.489	0.938	0.458	0.687	0.608
DSN	0.470	0.669	0.667	0.619	0.609	0.647	0.713	–	0.732	0.949	0.520	0.826	0.685
SDS	0.515	**0.710**	0.847	0.638	–	–	0.740	–	0.719	0.961	0.545	**0.843**	0.724
DvD	0.455	0.668	0.680	0.510	–	–	0.602	–	0.522	**0.966**	0.449	0.816	0.630
DSNI	0.472	0.669	0.667	0.649	–	–	0.708	–	**0.747**	0.899	0.528	0.817	0.684
DSDC	0.418	0.600	0.695	0.489	0.609	0.559	0.644	0.601	0.626	0.888	0.521	0.817	0.633
LoRD (ours)	0.479	0.683	0.878	0.613	0.610	0.538	0.755	0.943	0.657	0.944	0.522	0.783	0.702
B-LoRD (ours)	**0.521**	0.703	**0.905**	**0.748**	**0.658**	**0.659**	**0.783**	**0.964**	**0.747**	0.955	**0.561**	0.833	**0.751**

Table 15: F1-score on each dataset.

F1	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	Avg.
KKM	0.325	0.453	0.752	0.413	0.518	0.404	0.471	0.540	0.486	0.888	0.243	0.502	0.504
GKKM	0.165	0.216	0.688	0.409	0.566	0.109	0.436	–	0.478	0.889	0.249	0.557	0.454
SDP	0.339	0.538	0.718	0.450	–	–	0.478	–	0.513	0.838	0.267	0.445	–
DCD	0.298	0.522	0.695	0.482	0.524	0.455	0.498	0.668	0.603	0.920	0.243	0.528	0.532
SC	0.317	0.520	0.691	0.481	0.520	0.528	0.567	0.667	0.600	0.898	0.264	0.476	0.535
SR	0.317	0.512	0.691	0.470	0.500	0.446	0.482	0.653	0.592	0.898	0.296	0.541	0.533
NCut	0.299	0.485	0.693	0.463	0.480	0.313	0.475	0.646	0.547	0.898	0.301	0.536	0.533
DBSC	0.328	0.538	0.765	0.447	0.525	0.475	0.562	–	0.528	0.887	0.281	0.467	0.534
DirectSC	0.293	0.460	0.699	0.380	0.369	–	0.388	–	0.343	0.811	0.286	0.548	0.468
SymNMF	0.313	0.540	0.765	0.436	0.493	0.418	0.536	0.454	0.525	0.838	0.268	0.465	0.521
PHALS	0.313	0.534	0.708	0.411	0.523	0.461	0.570	0.605	0.549	0.857	0.281	0.461	0.520
S^3NMF	0.320	0.511	0.757	0.482	–	–	0.605	–	0.583	0.880	0.268	0.509	0.546
NLR	0.315	0.470	0.514	0.313	0.152	–	0.320	–	0.264	0.877	0.322	0.470	0.429
DSN	0.321	0.521	0.691	0.484	0.517	–	0.566	–	0.599	0.898	0.262	0.477	0.535
SDS	**0.368**	0.551	0.771	0.521	–	–	0.601	–	0.587	0.923	**0.398**	0.673	0.599
DvD	0.251	0.302	0.587	0.342	–	–	0.323	–	0.338	**0.931**	0.318	0.533	0.436
DSNI	0.321	0.514	0.694	0.523	–	–	0.567	–	0.622	0.809	0.255	0.541	0.538
DSDC	0.264	0.414	0.697	0.364	0.516	0.474	0.484	0.451	0.465	0.793	0.243	0.491	0.468
LoRD (ours)	0.308	0.535	0.802	0.491	0.560	0.436	0.638	0.893	0.548	0.888	0.243	0.430	0.543
B-LoRD (ours)	0.355	**0.593**	0.837	**0.633**	**0.608**	**0.576**	**0.656**	**0.930**	**0.640**	0.913	0.365	**0.744**	**0.637**

B-LoRD on each dataset, as shown in Fig. 13. From Fig. 13, we can be observe that:

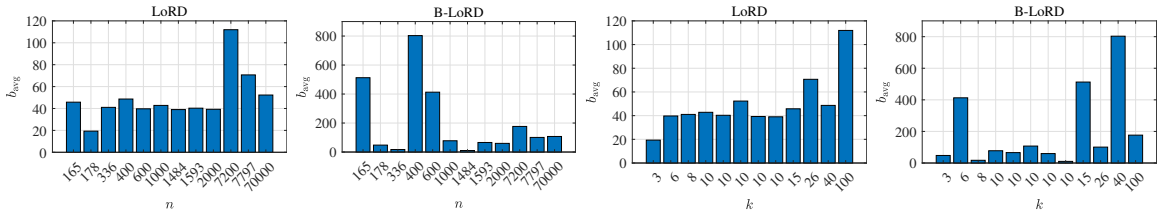


Figure 13: The correlation between b_{avg} (y-axis) and number of samples n and classes k (x-axis, resorted in ascent order).

The b_{avg} of LoRD is approximately 50, which is independent of n but proportional to k. The b_{avg} of B-LoRD varies more significantly, ranging from approximately 50 to 500, and appears to be independent of both n and k.

A.5 Correlation between Objective Function Value and ACC

The relationship between the objective function value and the clustering ACC of SR, SymNMF, PHALS, LoRD and B-LoRD are described in Fig. 14 to Fig. 19, respectively. In general, the correlation (measured by R^2) in KKM, LoRD and B-LoRD is stronger than SR, SymNMF and PHALS, because the doubly stochastic constraint is relaxed in SR, SymNMF and PHALS.

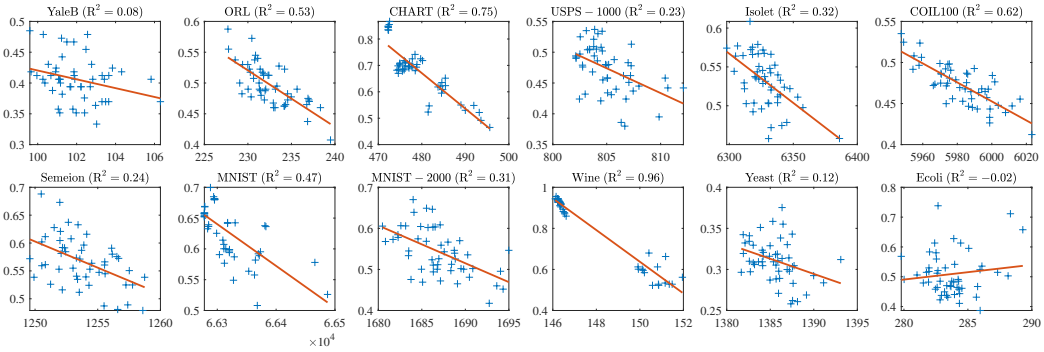


Figure 14: Correlation of objective function and clustering ACC in KKM.

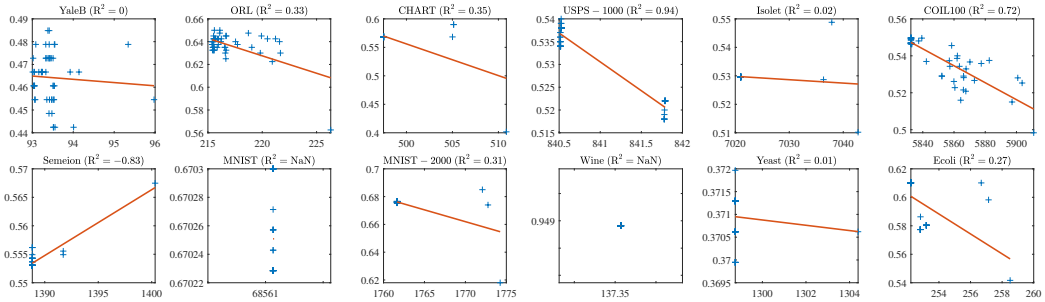


Figure 15: Correlation of objective function and clustering ACC in SR.

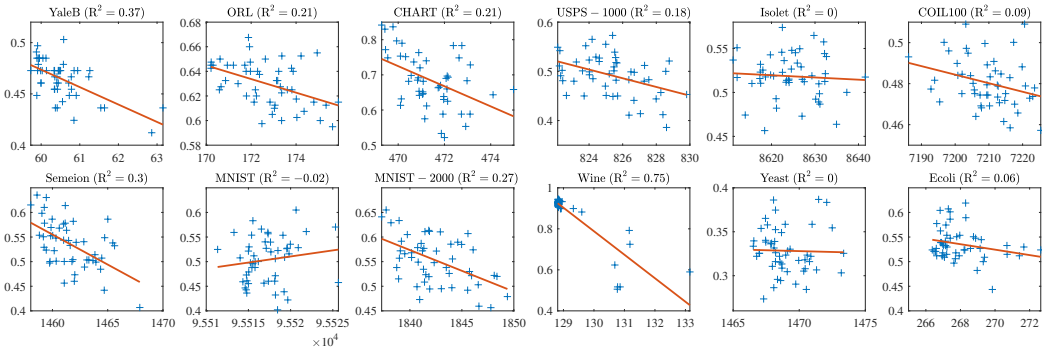


Figure 16: Correlation of objective function and clustering ACC in SymNMF.

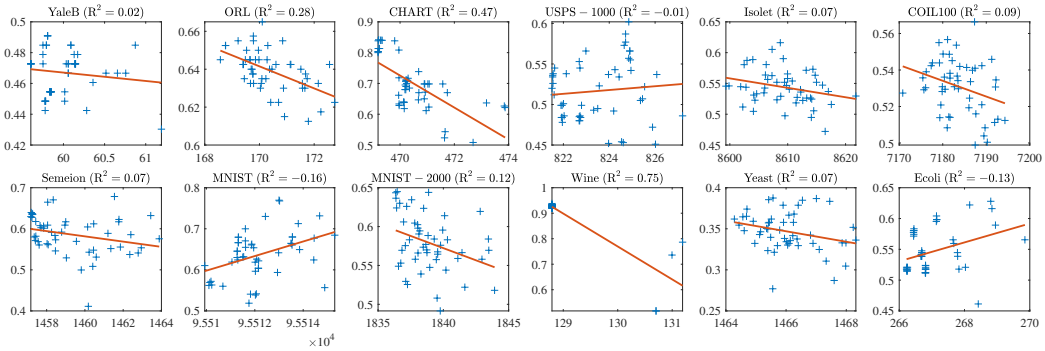


Figure 17: Correlation of objective function and clustering ACC in PHALS.

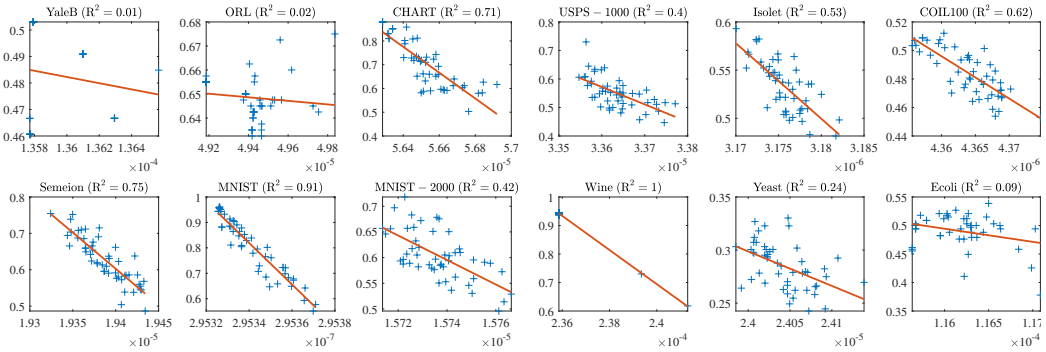


Figure 18: Correlation of objective function and clustering ACC in LoRD.

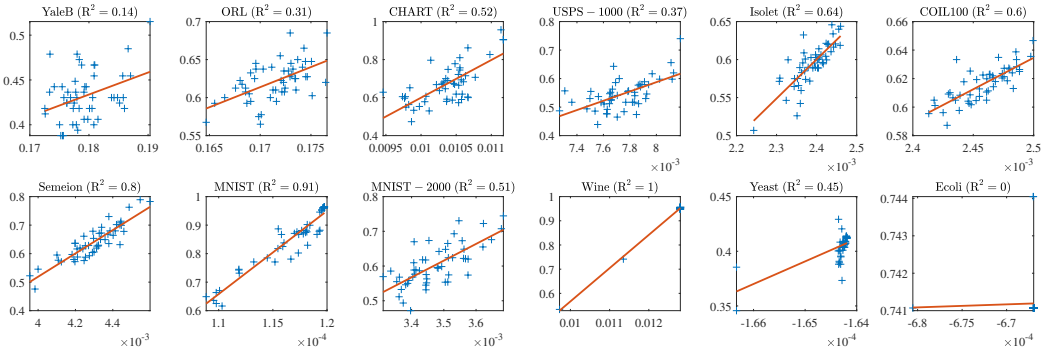


Figure 19: Correlation of objective function and clustering ACC in B-LoRD.

Appendix B. Proofs

B.1 Proof of Theorem 1

Proof The proof is straightforward, the three conditions in Theorem 1 are proven as follows:

- First, for all $\mu \in \mathbb{S}_+^k$, we can construct $1_n\mu^T/n \in \Omega(\mu)$, which shows that $\Omega(\mu) \neq \emptyset$.
- Second, $\Omega(\mu)$ is a subspace of Ω, which shows that $\bigcup_{\mu\in\mathbb{S}_+^k} \Omega(\mu) \subseteq \Omega$. Moreover, for all $V \in \Omega$, we have $1_n^T VV^T 1_n = 1 \Rightarrow V^T 1_n \in \mathbb{S}_+^k$, thus $V \in \Omega(V^T 1_n)$, which implies that $\Omega \subseteq \bigcup_{\mu\in\mathbb{S}_+^k} \Omega(\mu)$. Thus, $\Omega = \bigcup_{\mu\in\mathbb{S}_+^k} \Omega(\mu)$ holds.
- Third, suppose $V \in \Omega(\mu)$ and $V \in \Omega(\nu)$, we have $V^T 1_n = \mu = \nu$, which contradicts the condition $\mu \neq \nu$. Therefore, $\Omega(\mu) \cap \Omega(\nu) = \emptyset$.

■

B.2 Proof of Theorem 2

Proof Let $\mu = [\sqrt{\mathrm{P}(c_1)}, \ldots, \sqrt{\mathrm{P}(c_j)}]^T$, $\mathrm{P}(x_i) = 1_n/n$ and $V_{ij} = \frac{\mathrm{P}(y_i=j|x_i)\mathrm{P}(x_i)}{\sqrt{\mathrm{P}(c_j)}}$. According to $\mathrm{P}(y_i = j|x_i)\mathrm{P}(x_i) = \mathrm{P}(x_i|y_i = j)\mathrm{P}(c_j)$, we have:

$$\begin{aligned}(V\mu)_i &= \mathrm{P}(x_i)\sum_{j=1}^{k}\mathrm{P}(y_i = j|x_i) = \mathrm{P}(x_i) = \frac{1}{n}.\\ (V^T 1_n)_j &= \sum_{i=1}^{n}\frac{\mathrm{P}(y_i = j|x_i)\mathrm{P}(x_i)}{\sqrt{\mathrm{P}(c_j)}} = \sqrt{\mathrm{P}(c_j)}\sum_{i=1}^{n}\mathrm{P}(x_i|y_i = j) = \sqrt{\mathrm{P}(c_j)} = \mu_j,\end{aligned} \tag{37}$$

which indicate that $V \in \Omega(\mu)$.

Moreover, let $Z = n^2 V\mathrm{Diag}(\mu\odot\mu)V^T$. Under the conditional independence assumption, i.e., $\mathrm{P}(y_i|x_i) = \mathrm{P}(y_i|x_i, x_j)$ and $\mathrm{P}(y_i, y_j|x_i, x_j) = \mathrm{P}(y_i|x_i, x_j)\mathrm{P}(y_j|x_i, x_j)$, we have

$$Z_{ij} = \sum_{a=1}^{k}\mathrm{P}(y_i = a|x_i)\mathrm{P}(y_j = a|x_j) = \sum_{a=1}^{k}\mathrm{P}(y_i = a, y_j = a|x_i, x_j) = \mathrm{P}(y_i = y_j|x_i, x_j). \tag{38}$$

■

B.3 Proof of Theorem 4

Proof Given $V \in \Omega$, we have the Laplacian of VV^T is:

$$L_{VV^T} := \mathrm{Diag}(VV^T 1_n) - VV^T = 1_n/n - VV^T. \tag{39}$$

Therfore, the first $n-k$ largest eigenvalues of L_{VV^T} are all $1/n$, and the last k eiganvalues are:

$$\lambda_{n-i+1}(L_{VV^T}) = \frac{1}{n} - \lambda_i(VV^T) = \frac{1}{n} - \sigma_i^2(V), \quad i = 1, \ldots, k. \tag{40}$$

Accordingly, $\|VV^T\|_{\boxed{k}}$ can be simplified as:

$$\sum_{i=1}^{k} \lambda_{n-i+1}(L_{VV^T}) = \frac{k}{n} - \sum_{i=1}^{k} \sigma_i^2(V) = \frac{k}{n} - \|V\|_F^2. \tag{41}$$

Moreover, according to $\mathcal{P}_{\Omega_0(\mu)}(U)$ given in Lemma 6, the least k-block diagonal case is $1_n\mu^T/n = \mathcal{P}_{\Omega_0(\mu)}(0_{n\times k}) = \arg\min_{V\in\Omega(\mu)} \|V\|_F^2$, where $0_{n\times k}$ is an $n \times k$ matrix with all zeros. For all $\mu \in \mathbb{S}_+^k$, the k-block diagonality of $1_n\mu^T/n$ are equal, i.e., $\|1_n\mu^T/n\|_F^2 = 1/n$.

The fully k-block diagonal case occurs when V is orthogonal. For example, given a partition $G_1, \ldots, G_k$, let $V_{ij} = 1/\sqrt{n \times n_j}$ if $x_i \in G_j$ and zero otherwise. Then, $\|V\|_F^2 = \frac{k}{n}$, which implies that $\|VV^T\|_{\boxed{k}} = 0$. ■

B.4 Proof of Theorem 5

Proof To show that $\nabla_1 = 4(VV^T - S)V$ and $\nabla_2 = -2(SV + \gamma V)$ are L_1- and L_2-Lipschitz continuous on Ω, respectively, we need to prove:

$$\forall V, U \in \Omega(\mu), \quad \begin{cases} \|\nabla_1(V) - \nabla_1(U)\|_F \le 4\left(3/n + \|S\|_{\text{op}}\right)\|V - U\|_F \\ \|\nabla_2(V) - \nabla_2(U)\|_F \le 2\|S + \gamma I_n\|_{\text{op}}\|V - U\|_F \end{cases}, \tag{42}$$

where $\|\cdot\|_{\text{op}}$ denotes the operator norm, i.e., the largest singular value of matrix. For ∇_2, the proof is straightforward:

$$\|\nabla_2(V) - \nabla_2(U)\|_F = \|2(S + \gamma I_n)(V - U)\|_F \le 2\|S + \gamma I_n\|_{\text{op}}\|V - U\|_F. \tag{43}$$

For ∇_1, we have:

$$\begin{aligned} \|\nabla_1(V) - \nabla_1(U)\|_F &= \|4(VV^TV - UU^TU) - 4S(V - U)\|_F \\ &\le 4\|VV^TV - UU^TU\|_F + 4\|S\|_{\text{op}}\|V - U\|_F. \end{aligned} \tag{44}$$

The upper bound of $\|VV^TV - UU^TU\|_F$ can be derived as follows:

$$\begin{aligned} \|VV^TV - UU^TU\|_F &= \|VV^T(V - U) + V(V - U)^TU + (V - U)UU^T\|_F \\ &\le \|VV^T(V - U)\|_F + \|V(V - U)^TU\|_F + \|(V - U)UU^T\|_F \\ &\le \left(\|VV^T\|_{\text{op}} + \|V\|_{\text{op}}\|U\|_{\text{op}} + \|UU^T\|_{\text{op}}\right)\|V - U\|_F \\ &\le 3\|VV^T\|_{\text{op}}\|V - U\|_F \\ &= \frac{3}{n}\|V - U\|_F, \end{aligned} \tag{45}$$

where $\|VV^T\|_{\text{op}} = 1/n$ because $VV^T1_n = 1_n/n$. Substituting Eq. (45) into Eq. (44), we finally obtain:

$$\|\nabla_1(V) - \nabla_1(U)\|_F \le 4\left(3/n + \|S\|_{\text{op}}\right)\|V - U\|_F. \tag{46}$$

■

B.5 Proof of Lemma 6

Proof The projection problem of $U \in \mathbb{R}^{n\times k}$ onto $\Omega_0(\mu)$ is formulated as:

$$\mathcal{P}_{\Omega_0(\mu)}(U) = \underset{V^T 1_n=\mu, V\mu=1_n/n}{\arg\min} \frac{1}{2}\|V-U\|_F^2. \tag{47}$$

Let $\alpha \in \mathbb{R}^k$ and $\beta \in \mathbb{R}^n$ be the lagrange multiplier for constraint $V^T 1_n = \mu$ and $V\mu = 1_n/n$, respectively, the Lagrangian $\mathcal{L}(V,\alpha,\beta)$ is:

$$\mathcal{L}(V,\alpha,\beta) = \frac{1}{2}\|V-U\|_F^2 + \alpha^T(V^T 1_n - \mu) + \beta^T(V\mu - 1_n/n). \tag{48}$$

The partial derivative of $\mathcal{L}$ w.r.t. V satisfies:

$$\frac{\partial \mathcal{L}}{\partial V} = V - U + 1_n\alpha^T + \beta\mu^T = 0 \Longrightarrow V = U - 1_n\alpha^T - \beta\mu^T. \tag{49}$$

By applying the constraint conditions, we have:

$$\begin{cases} V^T 1_n = \mu \\ V\mu = 1_n/n \end{cases} \Longrightarrow \begin{cases} U^T 1_n - n\alpha - \mu 1_n^T\beta = \mu \\ U\mu - 1_n\mu^T\alpha - \beta = 1_n/n \end{cases} \Longrightarrow \begin{bmatrix} nI_k & \mu 1_n^T \\ 1_n\mu^T & I_n \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} U^T 1_n - \mu \\ U\mu - 1_n/n \end{bmatrix}. \tag{50}$$

Therefore, α and β can be obtained by solving the linear equation in Eq. (50). By applying the LDU decomposition of the block matrix, we have:

$$\begin{bmatrix} nI_k & \mu 1_n^T \\ 1_n\mu^T & I_n \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} I_k & \mu 1_n^T \\ 0 & I_n \end{bmatrix} \begin{bmatrix} nI_k - n\mu\mu^T & 0 \\ 0 & I_n \end{bmatrix} \begin{bmatrix} I_k & 0 \\ 1_n\mu^T & I_n \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} U^T 1_n - \mu \\ U\mu - 1_n/n \end{bmatrix}, \tag{51}$$

which can be further simplified as:

$$\begin{aligned} \begin{bmatrix} nI_k - n\mu\mu^T & 0 \\ 0 & I_n \end{bmatrix} \begin{bmatrix} \alpha \\ 1_n\mu^T\alpha + \beta \end{bmatrix} &= \begin{bmatrix} I_k & \mu 1_n^T \\ 0 & I_n \end{bmatrix}^{-1} \begin{bmatrix} U^T 1_n - \mu \\ U\mu - 1_n/n \end{bmatrix} \\ \Longrightarrow \quad \begin{bmatrix} n(I_k - \mu\mu^T)\alpha \\ 1_n\mu^T\alpha + \beta \end{bmatrix} &= \begin{bmatrix} (I_k - \mu\mu^T)U^T 1_n \\ U\mu - 1_n/n \end{bmatrix}. \end{aligned} \tag{52}$$

Note that $\mathrm{rank}(I_k - \mu\mu^T) = k-1$, and the null space of $I_k - \mu\mu^T$ is spaned by the vector μ. Therefore, the solution of the linear equation Eq. (50) is:

$$\begin{cases} \alpha = U^T 1_n/n + c\mu \\ \beta = (I_n - 1_n 1_n^T/n)U\mu - 1_n/n - c1_n \end{cases}, \tag{53}$$

where $c \in \mathbb{R}$ is arbitrary. Nevertheless, c does not appeared in the expression of V. To see this, by substituting Eq. (53) into Eq. (49), we finally get:

$$V = U + \frac{1_n^T U\mu + 1}{n} 1_n\mu^T - \frac{1_n 1_n^T}{n} U - U\mu\mu^T. \tag{54}$$

■

B.6 Proof of Lemma 7

Proof Suppose $\Omega(\mu) \subseteq \mathbb{R}^{n\times k}$ is closed, convex and nonempty. The projector $\mathcal{P}_{\Omega(\mu)}(U) = \arg\min_{V\in\Omega(\mu)} \|V-U\|_F^2$ satisfies the following important property:

$$\forall V \in \Omega(\mu), U \in \mathbb{R}^{n\times k}, \quad \left\langle \mathcal{P}_{\Omega(\mu)}(U) - U, V - \mathcal{P}_{\Omega(\mu)}(U) \right\rangle \geq 0. \tag{55}$$

Given $f(V)$ with a gradient $\nabla f(V)$ that is L-Lipschitz continuous, $f(V)$ has a quadratic upper bound:

$$\begin{aligned} f(V^{t+1}) &\leq f(V^t) + \langle \nabla f(V^t), V^{t+1} - V^t \rangle + \frac{L}{2}\|V^{t+1} - V^t\|_F^2 \\ &= f(V^t) - \frac{L}{2}\|V^{t+1} - V^t\|_F^2 + L\left\langle V^{t+1} - \left(V^t - \nabla f(V^t)/L\right), V^{t+1} - V^t \right\rangle. \end{aligned} \tag{56}$$

Recall that $V^{t+1} = \mathcal{P}_{\Omega(\mu)}(V^t - \nabla f(V^t)/L)$. Substituting $V^t - \nabla f(V^t)/L$ and V^t into U and V in Eq. (55), we have:

$$\left\langle V^{t+1} - \left(V^t - \nabla f(V^t)/L\right), V^{t+1} - V^t \right\rangle \leq 0. \tag{57}$$

Therefore, we get:

$$\begin{aligned} & f(V^t) - f(V^{t+1}) \geq \frac{L}{2}\|V^{t+1} - V^t\|_F^2 \\ \Longrightarrow \quad & f(V^0) - f(V^{t+1}) = \sum_{i=0}^{t} f(V^i) - f(V^{i+1}) \geq \frac{L}{2}\sum_{i=0}^{t}\|V^{i+1} - V^i\|_F^2. \end{aligned} \tag{58}$$

Additionally, by applying $f(V^0) - f(V^*) \geq f(V^0) - f(V^{t+1})$ and $\sum_{i=0}^{t}\|V^{i+1} - V^i\|_F^2 \geq (t+1)\min_{0\leq i\leq t}\|V^{i+1} - V^i\|_F^2$, we finally get:

$$\min_{0\leq i\leq t}\|V^{i+1} - V^i\|_F \leq \sqrt{\frac{2}{L}\frac{f(V^0) - f(V^*)}{t+1}}. \tag{59}$$

■

B.7 Proof of Lemma 8

Proof We first show that $\Omega(\mu)$ is compact. Since

$$\Omega(\mu) = \left\{ V \in \mathbb{R}^{n\times k} \mid V \geq 0,\ V^T 1_n = \mu,\ V\mu = 1_n/n \right\}, \tag{60}$$

it is an intersection of closed sets, hence closed. Moreover, for any $V \in \Omega(\mu)$ and any (i,j), the nonnegativity of V and the column-sum constraint imply

$$0 \leq V_{ij} \leq \sum_{r=1}^{n} V_{rj} = \mu_j. \tag{61}$$

Thus $\Omega(\mu)$ is bounded. Since $\Omega(\mu)$ is also nonempty by Theorem 1, it is compact in the finite-dimensional space $\mathbb{R}^{n\times k}$.

By the optimality condition of the Euclidean projector, for any $U \in \mathbb{R}^{n\times k}$ and any $V \in \Omega(\mu)$,

$$\left\langle \mathcal{P}_{\Omega(\mu)}(U) - U, V - \mathcal{P}_{\Omega(\mu)}(U) \right\rangle \geq 0. \tag{62}$$

Applying Eq. (62) with

$$U = V^t - \nabla f(V^t)/L, \quad \mathcal{P}_{\Omega(\mu)}(U) = V^{t+1}, \tag{63}$$

we obtain

$$\left\langle V^{t+1} - \left(V^t - \nabla f(V^t)/L\right), V - V^{t+1} \right\rangle \geq 0, \quad \forall V \in \Omega(\mu). \tag{64}$$

Since ∇f is L-Lipschitz continuous on $\Omega(\mu)$, the descent lemma gives

$$f(V^{t+1}) \leq f(V^t) + \langle \nabla f(V^t), V^{t+1} - V^t \rangle + \frac{L}{2}\|V^{t+1} - V^t\|_F^2. \tag{65}$$

Taking $V = V^t$ in Eq. (64) yields

$$\langle \nabla f(V^t), V^{t+1} - V^t \rangle \leq -L\|V^{t+1} - V^t\|_F^2. \tag{66}$$

Substituting Eq. (66) into Eq. (65), we get

$$f(V^{t+1}) \leq f(V^t) - \frac{L}{2}\|V^{t+1} - V^t\|_F^2, \quad \forall t \geq 0. \tag{67}$$

Since f is continuous on the compact set $\Omega(\mu)$, it is bounded below on $\Omega(\mu)$. Let $\underline{f} := \min_{V\in\Omega(\mu)} f(V)$. Summing Eq. (67) from $t = 0$ to T gives

$$\frac{L}{2}\sum_{t=0}^{T}\|V^{t+1} - V^t\|_F^2 \leq f(V^0) - f(V^{T+1}) \leq f(V^0) - \underline{f}. \tag{68}$$

Letting $T \to \infty$, we obtain

$$\sum_{t=0}^{\infty}\|V^{t+1} - V^t\|_F^2 < \infty. \tag{69}$$

In particular, $\|V^{t+1} - V^t\|_F \to 0$ as $t \to \infty$.

Since $\{V^t\}_{t\geq 0} \subseteq \Omega(\mu)$ and $\Omega(\mu)$ is compact, the sequence $\{V^t\}_{t\geq 0}$ admits at least one accumulation point. Let $\bar{V}$ be an accumulation point, and let $\{V^{t_j}\}_{j\geq 0}$ be a subsequence such that

$$V^{t_j} \to \bar{V} \quad \text{as } j \to \infty. \tag{70}$$

Hence

$$\begin{aligned}\|V^{t_j+1} - \bar{V}\|_F &\leq \|V^{t_j+1} - V^{t_j}\|_F + \|V^{t_j} - \bar{V}\|_F \\ &\to 0,\end{aligned} \tag{71}$$

that is, $V^{t_j+1} \to \bar{V}$.

Rewriting Eq. (64), we obtain

$$\left\langle \nabla f(V^t) + L(V^{t+1} - V^t), V - V^{t+1} \right\rangle \geq 0, \quad \forall V \in \Omega(\mu). \tag{72}$$

Fix any $V \in \Omega(\mu)$. Taking $t = t_j$ in Eq. (72) and letting $j \to \infty$, by the continuity of ∇f, together with $V^{t_j} \to \bar{V}$, $V^{t_j+1} \to \bar{V}$, and $\|V^{t_j+1} - V^{t_j}\|_F \to 0$, we obtain

$$\langle \nabla f(\bar{V}), V - \bar{V} \rangle \geq 0, \quad \forall V \in \Omega(\mu). \tag{73}$$

This is exactly the first-order optimality condition for the constrained problem

$$\min_{V \in \Omega(\mu)} f(V), \tag{74}$$

and is equivalent to

$$0 \in \nabla f(\bar{V}) + N_{\Omega(\mu)}(\bar{V}). \tag{75}$$

Therefore, every accumulation point of the exact projected gradient iteration is a first-order stationary point. ■

References

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75:245–248, 2009.

Amir Beck. *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization, 2017.

Kamal Berahmand, Farid Saberi-Movahed, Razieh Sheikhpour, Yuefeng Li, and Mahdi Jalili. A comprehensive survey on spectral clustering with graph structure learning. *arXiv preprint arXiv:2501.13597*, 2025.

James P Boyle and Richard L Dykstra. A method for finding projections onto the intersection of convex sets in hilbert spaces. In *Advances in Order Restricted Statistical Inference: Proceedings of the Symposium on Order Restricted Statistical Inference held in Iowa City, Iowa, September 11–13, 1985*, pages 28–47. Springer, 1986.

Deng Cai, Xiaofei He, and Jiawei Han. Document clustering using locality preserving indexing. *IEEE transactions on knowledge and data engineering*, 17(12):1624–1637, 2005.

Valerio Cappellini, Hans-Jürgen Sommers, Wojciech Bruzda, and Karol Życzkowski. Random bistochastic matrices. *Journal of Physics A: Mathematical and Theoretical*, 42(36): 365209, 2009.

Xiaohui Chen and Yun Yang. Cutoff for exact recovery of gaussian mixture models. *IEEE Transactions on Information Theory*, 67(6):4223–4238, 2021.

Samir Chowdhury and Tom Needham. Generalized spectral clustering via gromov-wasserstein learning. In *International Conference on Artificial Intelligence and Statistics*, pages 712–720. PMLR, 2021.

Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, 2009.

Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556, 2004.

Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 606–610. SIAM, 2005.

Jiashi Feng, Zhouchen Lin, Huan Xu, and Shuicheng Yan. Robust subspace segmentation with block-diagonal prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3818–3825, 2014.

Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics reports*, 659:1–44, 2016.

Christophe Giraud and Nicolas Verzelen. Partial recovery bounds for clustering with the relaxed k-means. *Mathematical Statistics and Learning*, 1(3):317–374, 2019.

Li He and Hong Zhang. Doubly stochastic distance clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(11):6721–6732, 2023.

Liangshao Hou, Delin Chu, and Li-Zhi Liao. A progressive hierarchical alternating least squares method for symmetric nonnegative matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5355–5369, 2022.

Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J Irwin, Edward B Lee, Russell T Shinohara, and Mingyao Li. Spagcn: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature methods*, 18(11):1342–1351, 2021.

Jin Huang, Feiping Nie, and Heng Huang. Spectral rotation versus k-means in spectral clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, pages 431–437, 2013.

Yuheng Jia, Hui Liu, Junhui Hou, Sam Kwong, and Qingfu Zhang. Self-supervised symmetric nonnegative matrix factorization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(7):4526–4537, 2021.

Ah-Pine Julien. Learning doubly stochastic and nearly idempotent affinity matrix for graph-based clustering. *European Journal of Operational Research*, 299(3):1069–1078, 2022. ISSN 0377-2217.

Zhao Kang, Chong Peng, Qiang Cheng, Xinwang Liu, Xi Peng, Zenglin Xu, and Ling Tian. Structured graph learning for clustering and semi-supervised classification. *Pattern Recognition*, 110:107627, 2021.

Chanyoung Kim, Woojung Han, Dayun Ju, and Seong Jae Hwang. Eagle: Eigen aggregation learning for object-centric unsupervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3523–3533, 2024.

Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 106–117. SIAM, 2012.

Da Kuang, Sangwoon Yun, and Haesun Park. Symnmf: nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization*, 62: 545–574, 2015.

Brian Kulis, Arun C Surendran, and John C Platt. Fast low-rank semidefinite programming for embedding and clustering. In *Artificial Intelligence and Statistics*, pages 235–242. PMLR, 2007.

Jacob H Levine, Erin F Simonds, Sean C Bendall, Kara L Davis, El-ad D Amir, Michelle D Tadmor, Oren Litvin, Harris G Fienberg, Astraea Jager, Eli R Zunder, et al. Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197, 2015.

Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. *Advances in neural information processing systems*, 28, 2015.

Wei Liu, Bo Wang, Yuting Bai, Xiao Liang, Li Xue, and Jiawei Luo. Spagic: graph-informed clustering in spatial transcriptomics via self-supervised contrastive learning. *Briefings in bioinformatics*, 25(6):bbae578, 2024.

Bo Long, Zhongfei Zhang, Xiaoyun Wu, and Philip S Yu. Relational clustering by symmetric convex coding. In *Proceedings of the 24th international conference on Machine learning*, pages 569–576, 2007.

Canyi Lu, Jiashi Feng, Zhouchen Lin, Tao Mei, and Shuicheng Yan. Subspace clustering by block diagonal representation. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):487–501, 2018.

Eduardo Fernandes Montesuma, Fred Maurice Ngole Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 977–986, 2014.

Feiping Nie, Chaodie Liu, Rong Wang, and Xuelong Li. A novel and effective method to directly solve spectral clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Jiwoong Park and Taejeong Kim. Learning doubly stochastic affinity matrix via davis-kahan theorem. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 377–384. IEEE, 2017.

Jiming Peng and Yu Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM journal on optimization*, 18(1):186–205, 2007.

Satu Elisa Schaeffer. Graph clustering. *Computer science review*, 1(1):27–64, 2007.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

Defeng Sun, Kim-Chuan Toh, Yancheng Yuan, and Xin-Yuan Zhao. Sdpnal+: A matlab software for semidefinite programming with bound constraints (version 1.0). *Optimization Methods and Software*, 35(1):87–115, 2020.

Grigorios F Tzortzis and Aristidis C Likas. The global kernel k-means algorithm for clustering in feature space. *IEEE transactions on neural networks*, 20(7):1181–1194, 2009.

Hugues Van Assel, Cédric Vincent-Cuaz, Nicolas Courty, Rémi Flamary, Pascal Frossard, and Titouan Vayer. Distributional reduction: Unifying dimensionality reduction and clustering with gromov-wasserstein projection. *arXiv preprint arXiv:2402.02239*, 2024.

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

Rong Wang, Huimin Chen, Yihang Lu, Qianrong Zhang, Feiping Nie, and Xuelong Li. Discrete and balanced spectral clustering with scalability. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Xiaoqian Wang, Feiping Nie, and Heng Huang. Structured doubly stochastic matrix for graph based clustering: Structured doubly stochastic matrix. In *Proceedings of the 22nd ACM SIGKDD International conference on Knowledge discovery and data mining*, pages 1245–1254, 2016.

Danyang Wu, Feiping Nie, Jitao Lu, Rong Wang, and Xuelong Li. Effective clustering via structured graph learning. *IEEE Transactions on Knowledge and Data Engineering*, 35 (8):7909–7920, 2022.

Xingyu Xie, Xianglin Guo, Guangcan Liu, and Jun Wang. Implicit block diagonal low-rank representation. *IEEE Transactions on Image Processing*, 27(1):477–489, 2017.

Hongteng Xu, Dixin Luo, and Lawrence Carin. Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems*, 32, 2019.

Jingjing Xue, Liyin Xing, Yuting Wang, Xinyi Fan, Lingyi Kong, Qi Zhang, Feiping Nie, and Xuelong Li. A comprehensive survey of fast graph clustering. *Vicinagearth*, 1(1):7, 2024.

Yitao Yang, Yang Cui, Xin Zeng, Yubo Zhang, Martin Loza, Sung-Joon Park, and Kenta Nakai. Staig: Spatial transcriptomics analysis via image-aided graph contrastive learning for domain exploration and alignment-free integration. *Nature Communications*, 16(1): 1067, 2025.

Zhirong Yang and Erkki Oja. Clustering by low-rank doubly stochastic matrix decomposition. *arXiv preprint arXiv:1206.4676*, 2012.

Zhirong Yang, Jukka Corander, and Erkki Oja. Low-rank doubly stochastic matrix decomposition for cluster analysis. *Journal of Machine Learning Research*, 17(187):1–25, 2016.

Ron Zass and Amnon Shashua. A unifying approach to hard and probabilistic clustering. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 294–301. IEEE, 2005.

Ron Zass and Amnon Shashua. Doubly stochastic normalization for spectral clustering. *Advances in neural information processing systems*, 19, 2006.

Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. *Advances in neural information processing systems*, 17, 2004.

Yubo Zhuang, Xiaohui Chen, Yun Yang, and Richard Y. Zhang. Statistically optimal k-means clustering via nonnegative low-rank semidefinite programming. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=v7ZPwoHU1j`.