Tensorized Bipartite Graph Learning for Multi-view Clustering

Wei Xia, Quanxue Gao, Qianqian Wang, Xinbo Gao, Chris Ding, and Dacheng Tao, *Fellow, IEEE*

Abstract—Despite the impressive clustering performance and efficiency in characterizing both the relationship between the data and cluster structure, most existing graph-based multi-view clustering methods still have the following drawbacks. They suffer from the expensive time burden due to both the construction of graphs and eigen-decomposition of Laplacian matrix. Moreover, none of them simultaneously considers the similarity of inter-view and similarity of intra-view. In this article, we propose a variance-based de-correlation anchor selection strategy for bipartite construction. The selected anchors not only cover the whole classes but also characterize the intrinsic structure of data. Following that, we present a *tensorized bipartite graph learning for multi-view clustering* (TBGL). Specifically, TBGL exploits the similarity of inter-view by minimizing the tensor Schatten *p*-norm, which well exploits both the spatial structure and complementary information embedded in the bipartite graphs of views. We exploit the similarity of intra-view by using the $\ell_{1,2}$-norm minimization regularization and connectivity constraint on each bipartite graph. So the learned graph not only well encodes discriminative information but also has the exact connected components which directly indicates the clusters of data. Moreover, we solve TBGL by an efficient algorithm which is time-economical and has good convergence. Extensive experimental results demonstrate that TBGL is superior to the state-of-the-art methods. Codes and datasets are available: https://github.com/xdweixia/TBGL-MVC.

Index Terms—Multi-view clustering, bipartite graph learning, tensor Schatten *p*-norm.

1 Introduction

In the real word applications, each object can be usually sensed and described from multiple views, and information embedded in different views are complementary and convey the common underlying clusters. Drawing the inspiration from this principle, multi-view clustering (MVC) has become an active topic in pattern analysis [1]–[7]. Spectral clustering (SC) is one of the most representative techniques for clustering due to its efficiency in characterizing both the complex structure and relationship among arbitrarily shaped data. It aims to divide graph into several disconnected sub-graphs such that the data in the same sub-graph have high similarity to each other, while data points in different sub-graphs have low similarity. Based on SC, many multi-view spectral clustering methods have been developed [8]–[10].

Although they have achieved impressive results for MVC task, all of them involve $n \times n$ graph construction and eigen-decomposition of Laplacian matrix whose computational complexity are $\mathcal{O}(Vn^2)$ and $\mathcal{O}(n^3)$, respectively, where V and n are the number of views and samples, respectively. Thus, they are inefficient or even fail in handling large-scale data which is ubiquitous in big data era.

To this end, bipartite graph based MVC methods have aroused widespread research interest. Bipartite graph can well present complex mechanisms of multi-view data by modeling the relationship between n data points and m ($m \ll n$) anchors and help to reduce both the computational complexity and storage complexity. Inspired by this, Li *et al*. [11] presented a bipartite graph-based fast method for the multi-view spectral clustering (MVSC). However, its performance heavily depends on the manually designed bipartite graphs of views. To cope with this problem, most existing methods [12], [13] get the consensus bipartite graph by linear combination of the bipartite graphs which are adaptively learned from the corresponding views. Despite the impressive performance of bipartite graph based clustering methods, **(1)** they cannot well encode the complementary information and spatial structure embedded in bipartite graphs of views, resulting in inferior results; **(2)** they only consider the similarity of intra-view while neglecting the similarity of inter-view; **(3)** they select anchors by random sampling technique or *K*-Means which are of randomness, resulting in an unstable and unsatisfactory performance.

To overcome the aforementioned problems, we present a novel anchor selection scheme, termed *variance-based de-correlation anchor selection* (VDA), and propose a *tensorized bipartite graph learning model for MVC* (TBGL) (See Fig. 1). Specifically, TBGL leverages the tensor Schatten *p*-norm constraint [14] on the third tensor, which consists of the bipartite graphs of views, to exploit the complementary information and spatial structure of views. Thus, the rank of the learned graph is very close to the target rank. By using the regularization of $\ell_{1,2}$-norm on bipartite graphs of views,

- *This work was supported in part by National Natural Science Foundation of China under Grant 62176203; in part by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) under Grant 202200035; in part by Natural Science Basic Research Plan in Shaanxi Province (Grant 2020JZ-19); in part by the Fundamental Research Funds for the Central Universities and the Innovation Fund of Xidian University. (Corresponding author: Q. Gao, e-mail: qxgao@xidian.edu.cn.)*
- *W. Xia, Q. Gao, and Q. Wang are with the State Key laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China.*
- *Chris Ding is with Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China.*
- *X. Gao is with the Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.*
- *D. Tao is with the UBTECH Sydney Artificial Intelligence Centre and the School of Information Technologies, Faculty of Engineering and Information Technologies, University of Sydney, Darlington, NSW 2008, Australia.*

Manuscript received XXXX; revised XXXX; accepted XXXX.

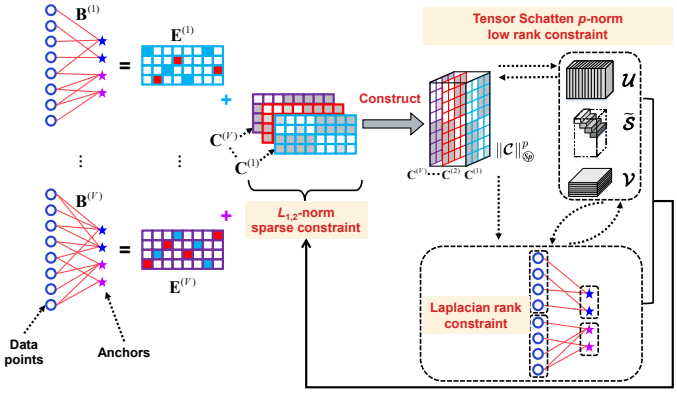


Fig. 1. The framework of TBGL, where $\mathbf{B}^{(v)}$ is the pre-defined graph of the v-th view; $\mathbf{C}^{(v)}$ is the learned graph of the v-th view; $\mathbf{E}^{(v)}$ is error.

the learned bipartite graphs well encode cluster structure and discriminative information. Moreover, we present an efficient optimization algorithm to solve TBGL with relatively low computational complexity. Comparing with existing well-studied graph based MVC methods, the contributions and novelties of TBGL could be summarized as follows:

- We learn tensorized bipartite graph by simultaneous considering inter-view and intra-view similarities with tensor Schatten p-norm and $\ell_{1,2}$-norm penalty, respectively. To our best knowledge, TBGL could be one of the first attempts to benefit multi-view bipartite graph learning with tensorized manner. Therefore, TBGL could provide some novel insights to the community of bipartite graph learning.
- To implement anchor selection stably and effectively, we propose a novel anchor selection scheme, which can ensure that the selected anchors not only cover the whole classes, but also characterize the intrinsic structure of data.
- We mathematically prove that the proposed algorithm converges to the KKT stationary point. Experimental results over the seven datasets indicate that TBGL outperforms the state-of-the-art methods.

Notations: In this article, we use bold calligraphy letters for 3rd-order tensors, *e.g.*, $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$; bold upper case letters for matrices, *e.g.*, $\mathbf{D}$; bold lower case letters for vectors, *e.g.*, $\mathbf{d}$; and lower case letters such as d_{ijk} for the entries of $\boldsymbol{\mathcal{D}}$. The i-th frontal slice of $\boldsymbol{\mathcal{D}}$ is $\boldsymbol{\mathcal{D}}^{(i)}$. $\overline{\boldsymbol{\mathcal{D}}}$ is the discrete Fast Fourier Transform (FFT) of $\boldsymbol{\mathcal{D}}$ along the third dimension, *i.e.*, $\overline{\boldsymbol{\mathcal{D}}} = fft(\boldsymbol{\mathcal{D}}, [\,], 3)$. Thus, $\boldsymbol{\mathcal{D}} = ifft(\overline{\boldsymbol{\mathcal{D}}}, [\,], 3)$. The trace of matrix $\mathbf{D} \in \mathbb{R}^{n \times m}$ is denoted by $\mathrm{tr}(\mathbf{D})$. The ℓ_1-norm of $\mathbf{D}$ is written as $\|\mathbf{D}\|_1$. The $\ell_{1,2}$-norm of matrix $\mathbf{D}$ is written as $\|\mathbf{D}\|_{1,2}^2 = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} |D_{i,j}|\right)^2$ [15]–[17]. $\mathbf{I}$ is an identity matrix.

2 Related Works and Background

In this section, we firstly revisit the single-view and multi-view bipartite graph clustering frameworks, respectively, then, we review the previous graph based multi-view clustering methods which are related to TBGL.

2.1 Single-view Oriented Bipartite Graph Clustering

We start with a brief review of the classical bipartite graph clustering. Given data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$ with K clusters, where d and n are the feature dimension and the number of samples, respectively. The bipartite graph can be defined as $\boldsymbol{G} = (\mathbf{X}, \mathbf{A}, \mathbf{B})$ [18], where $\mathbf{A} \in \mathbb{R}^{d \times m}$ represents the feature matrix of m ($m \ll n$) anchors; $\mathbf{B} \in \mathbb{R}^{n \times m}$ is the adjacency matrix of $\boldsymbol{G}$. Accordingly, the full adjacency matrix of the bipartite graph is

$$\mathbf{Z} = \begin{bmatrix} 0 & \mathbf{B} \\ \mathbf{B}^{\mathrm{T}} & 0 \end{bmatrix}. \tag{1}$$

Then we can calculate the normalized Laplacian matrix $\widetilde{\mathbf{L}}$ of $\mathbf{Z}$ by $\widetilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{Z}\mathbf{D}^{-\frac{1}{2}}$, where the degree matrix $\mathbf{D} \in \mathbb{R}^{(n+m)\times(n+m)}$ is a diagonal matrix, and $D_{ii} = \sum_j Z_{ij}$. Let $\mathbf{P} \in \mathbb{R}^{(n+m)\times K}$ be the indicator matrix, the objective function of the classic bipartite spectral graph partitioning [19] is

$$\min_{\mathbf{P}^{\mathrm{T}}\mathbf{P}=\mathbf{I}} \mathrm{tr}(\mathbf{P}^{\mathrm{T}}\widetilde{\mathbf{L}}\mathbf{P}). \tag{2}$$

where the clustering labels can be obtained by applying K-Means on the indicator matrix $\mathbf{P}$.

2.2 Multi-view Oriented Bipartite Graph Clustering

Since numerous real-world data is collected from different sources or represented by different types of features, several forms of bipartite graph based multi-view clustering methods are presented [11], [12], [20]. Let $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$ and $\{\mathbf{B}^{(v)}\}_{v=1}^{V}$ denote the data matrix and adjacency matrix of the v-th view, respectively, where $\mathbf{X}^{(v)} \in \mathbb{R}^{n \times d_v}$; $\mathbf{B}^{(v)} \in \mathbb{R}^{n \times m}$; d_v and m denote the number of feature dimensions and anchors in the v-th view, respectively; V is the number of views. One of the most representative multi-view clustering methods via bipartite graph can be concisely represented as

$$\begin{aligned} &\min_{\mathbf{P}^{\mathrm{T}}\mathbf{P}=\mathbf{I},\, \xi^{(v)}} \sum_{v=1}^{V} \xi^{(v)} \mathrm{tr}(\mathbf{P}^{\mathrm{T}}\widetilde{\mathbf{L}}^{(v)}\mathbf{P}) + \gamma\mathcal{R}(\boldsymbol{\xi}), \\ &s.t. \sum_{v=1}^{V} \xi^{(v)}=1,\ \xi^{(v)} \geq 0 \end{aligned} \tag{3}$$

where $\boldsymbol{\xi} = [\xi^1, \xi^2, \cdots, \xi^{(V)}]$ is the weighted vector; ξ^v is the weight of the v-th view, it reflects the importance of the v-th view for clustering; $\mathcal{R}(\cdot)$ is a regularizer that is used to keep the smooth of weights distribution; γ is a trade-off parameter. $\widetilde{\mathbf{L}}^{(v)}$=$\mathbf{I}$-$(\mathbf{D}^{(v)})^{-\frac{1}{2}}\mathbf{Z}^{(v)}(\mathbf{D}^{(v)})^{-\frac{1}{2}}$ is the normalized Laplacian matrix of v-th view, where $\mathbf{Z}^{(v)}$=$\begin{bmatrix} 0 & \mathbf{B}^{(v)} \\ (\mathbf{B}^{(v)})^{\mathrm{T}} & 0 \end{bmatrix}$ is full adjacency matrix of the bipartite graph; the diagonal matrix $\mathbf{D}^{(v)} \in \mathbb{R}^{(n+m)\times(n+m)}$ is the degree matrix of $\mathbf{Z}^{(v)}$. The optimal solution of $\mathbf{P}$ can be obtained by calculating the eigenvectors corresponding to the K smallest eigenvalues of $\mathbf{L} = \sum_{v=1}^{V} \xi^{(v)}\widetilde{\mathbf{L}}^{(v)}$.

The aforementioned method treats each row of $\mathbf{P}$ as a new representation of each data point and compute the clustering labels by employing the K-Means algorithm.

2.3 Related Works

One of the most representative methods is Co-regularized multi-view spectral clustering (Co-reg) [8]. Co-reg executes traditional SC on each view to obtain the view-specific indicator matrix, and then learn the consensus indicator matrix by minimizing the mismatch between indicator matrices.

To well exploit the spatial structure and complementary information embedded in multiple views, Xu *et al*. [9] proposed low-rank tensor constrained Co-reg (LTCSPC) by using the minimization of tensor nuclear norm based on tensor singular value decomposition (t-SVD) [21]. Although they have good performance, they treat all views equally, which is not reasonable in real-word applications. To improve the robustness of the algorithm, Nie *et al*. [10] adaptively assigned the weighted values for different views and developed auto-weighted graph learning method (AMGL).

However, the performance of the aforementioned methods heavily depends on the predefined graphs of different views. In real applications, it is very challenging to artificially design a suitable graph for each view. This reduces the flexibility of algorithms. For this consideration, Gao *et al*. [22] adaptively learned the view-consensus graph in the low-dimensional space by joint feature selection and adaptive neighbours. It explicitly assumes that each data in different views has the same neighbors. This assumption is very strict, leading to suboptimal performance. To relax this assumption, Zhan *et al*. [23] proposed multi-view graph learning for clustering (MVGL), which adaptively learns graph for each view, and then obtains a common graph by linear combination of the local graphs. However, these two steps are independent, which limits its performance. To tackle this problem, Nie *et al*. [24] presented self-weighted multi-view graph learning for clustering (SwMC), which integrates view-consensus graph learning and weighs learning for different views into a unified framework. To well exploit the complementary information of graphs, Wu *et al*. [25] proposed essential tensor learning for multi-view spectral clustering (ETLMSC), which learned the view-consensus graph by t-SVD based tensor nuclear norm minimization. Similarly, Xie *et al*. [26], [27] made the multi-view features form a 3rd-order tensor. And it utilized the self-representation strategy as a constraint and enforced the tensor multi-rank minimization for clustering.

Due to the computational complexity of the aforementioned methods is squared or cubic with the data size, thus, they are inefficient in handling big data. To improve the efficiency of graph construction and the Laplacian matrix eigen-decomposition, the existing works can be roughly divided into three categories: 1) matrix compression based methods [28], [29]; 2) kernel approximation based methods [30], [31]; 3) bipartite graph based methods [18], [19], [32]–[34], where the bipartite graph based methods have aroused widespread research interest. For example, Cai *et al*. [35] used a small-scale bipartite graph with the size of $n \times m$ to improve the efficiency of spectral clustering in handling large-scale data, where m ($m \ll n$) is the number of anchors. Nie *et al*. [36] proposed K-Multiple-Means (KMM) by extending K-Means with connectivity constraint. Nevertheless, all these methods are single-view oriented.

Recently, bipartite graphs based MVC methods have been emerging [13], [20], [37]. For example, Wang *et al*. proposed a fast multi-view subspace clustering method with consensus anchor guidance (FPMVS-CAG) [38], which simultaneously carries out anchor optimization and subspace bipartite graph construction. However, the clustering and graph learning are two independent processes, which limits the performance of FPMVS-CAG. To solve this problem, Li *et al*. [13] proposed a scalable and parameter-free multi-view clustering (SFMC) via the self-weighted graph fusion framework. SFMC integrates the Laplacian rank constraint and multi-view bipartite graph learning into a unified framework such that the learned graph has K-connected components. Thus, clustering results can be directly obtained by the connectivity of the learned graph. Despite the good performance of the aforementioned bipartite based MVC, they fail to simultaneously consider the inter-view and intra-view similarities. To cope with this issue, the proposed TBGL minimizes tensor Schatten p-norm and $\ell_{1,2}$-norm constraints on graphs which not only helps to explore the exploits the complementary information embedded in graphs of views, but also helps to encode discriminative information.

3 The Proposed TBGL

3.1 Problem Formulation and Objective

As the analyzed above, the existing bipartite graph based multi-view clustering methods have suffered from three main issues: (1) They cannot well explore both the complementary information and spatial structure embedded in bipartite graphs of different views. (2) They aim to learn the graph embedding of data from the multiple pre-defined bipartite graphs. However, the pre-defined bipartite graphs were built on random sampling or K-Means, which fails to capture the non-convex data distribution. (3) None of them simultaneously considers the similarity of the inter-view, which well exploits the complementary information embedded in graphs of views, and the similarity of intra-view which exploits the cluster structure of data.

To this end, we target at adaptively learning a new bipartite graph $\mathbf{C}^{(v)} \in \mathbb{R}^{n\times m}$ such that it well characterizes both the cluster structure and the relationship between m ($m \ll n$) anchors and n data points in the v-th view. Meanwhile, we aim at getting an implicitly view-consensus graph. It well explores the complementary information embedded in different views and has exactly K connected components, where K denotes the number of clusters. Thus, we can directly get the cluster labels.

To better formulate the objective function of TBGL, we first introduce Lemma 1 [39] and Definition 1.

Lemma 1. [39] The multiplicity K of the eigenvalue zeros of $\widetilde{\mathbf{L}}^{(v)}$ is equals to the number of connected components in the nonnegative graph associated with $\mathbf{F}^{(v)}$.

Remark 1. The graph $\mathbf{F}^{(v)}$ is a block anti-diagonal matrix, which is composed of the matrix $\mathbf{C}^{(v)}$ and its transposed matrix $(\mathbf{C}^{(v)})^{\mathrm{T}}$. Thus, $\mathbf{F}^{(v)}$ and $\mathbf{C}^{(v)}$ has the same number of connected components.

Definition 1. [14] Given $\mathcal{C} \in \mathbb{R}^{n_1\times n_2\times n_3}$, $h = \min(n_1, n_2)$, tensor Schatten p-norm of tensor $\mathcal{C}$ is defined as

$$\|\mathcal{C}\|_{\circledS_p} = \Big(\sum_{v=1}^{n_3} \|\overline{\mathcal{C}}^{(v)}\|_{\circledS_p}^{p}\Big)^{\frac{1}{p}} = \Big(\sum_{v=1}^{n_3}\sum_{j=1}^{h} \sigma_j(\overline{\mathcal{C}}^{(v)})^{p}\Big)^{\frac{1}{p}} \quad (4)$$

where $0 < p \leq 1$, $\sigma_j(\overline{\mathcal{C}}^{(v)})$ is the j-th singular value of $\overline{\mathcal{C}}^{(v)}$.

According to Lemma 1, if $\mathrm{rank}(\widetilde{\mathbf{L}}^{(v)}) = n + m - K$, then the corresponding graph $\mathbf{C}^{(v)}$ has K connected components. Moreover, considering different views have different contribution for clustering, we adaptively assign weight $\frac{1}{\xi^{(v)}}$ for

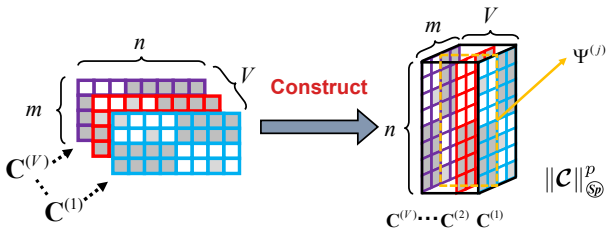


Fig. 2. Construction of tensor $\mathcal{C} \in \mathbb{R}^{n\times V\times m}$. $\mathbf{\Psi}^{(j)}$ denotes the j-th frontal slice of $\mathcal{C}$ ($j \in \{1, 2, \cdots, m\}$).

the Laplacian matrix of the v-th view. Then the weighted Laplacian matrix satisfies $\text{rank}(\widetilde{\mathbf{L}}_{\mathbf{C}}) = n + m - K$. In this case, the implicitly view-consensus graph $\mathbf{C} = \sum_{v=1}^{V} \frac{\mathbf{C}^{(v)}}{\xi^{(v)}} \Big/ \sum_{v=1}^{V} \frac{1}{\xi^{(v)}}$ has K-connected components. Then we can directly obtain the final clustering labels based on the connectivity of $\mathbf{C}$ without extra post-processing. For these purposes, we propose the following multi-view bipartite graph learning model:

$$\begin{aligned} \min_{\mathbf{C}^{(v)},\, \mathbf{E}^{(v)},\, \xi^{(v)}} & \|\mathcal{C}\|_{\circledS p}^{p} + \alpha \sum_{v=1}^{V} \|\mathbf{E}^{(v)}\|_1 + \gamma \sum_{v=1}^{V} \|\mathbf{C}^{(v)}\|_{1,2}^{2} \\ s.t. \quad & \mathbf{B}^{(v)} = \mathbf{C}^{(v)} + \mathbf{E}^{(v)},\, \mathbf{C}^{(v)} \geq 0,\, \mathbf{C}^{(v)}\mathbf{1} = \mathbf{1} \\ & \text{rank}(\widetilde{\mathbf{L}}^{(v)}) = n + m - K,\, \sum_{v=1}^{V} \xi^{(v)} = 1,\, \xi^{(v)} \geq 0 \end{aligned} \tag{5}$$

where $\mathcal{C} \in \mathbb{R}^{n\times V\times m}$, *i.e.*, $\mathcal{C}(:, v, :) = \mathbf{C}^{(v)}$; $\mathbf{E}^{(v)}$ is the error matrix of v-th view. $\widetilde{\mathbf{L}}^{(v)} = \mathbf{I} - \mathbf{D}_{\mathbf{F}^{(v)}}^{-\frac{1}{2}} \mathbf{F}^{(v)} \mathbf{D}_{\mathbf{F}^{(v)}}^{-\frac{1}{2}}$ is the normalized Laplacian matrix of $\mathbf{F}^{(v)} \in \mathbb{R}^{(n+m)\times(n+m)}$, which is defined as $\mathbf{F}^{(v)} = \begin{bmatrix} \mathbf{0} & \mathbf{C}^{(v)} \\ (\mathbf{C}^{(v)})^{\mathrm{T}} & \mathbf{0} \end{bmatrix}$. $\mathbf{D}_{\mathbf{F}^{(v)}}$ is a diagonal matrix whose diagonal elements are $\mathbf{D}_{\mathbf{F}^{(v)}}(i, i) = \sum_{j=1}^{n+m} \mathbf{F}^{(v)}(i, j)$; α and γ are two trade-off parameters.

Remark 2. **[The intuition and benefits of tensor Schatten *p*-norm]** Take the matrices for example, suppose $\sigma_1, \ldots, \sigma_h$ represents the singular values of matrix $\mathbf{C} \in \mathbb{R}^{n_1\times n_2}$ in the descending order, thus, when $p > 0$, the Schatten p-norm of $\mathbf{C}$ is $\|\mathbf{C}\|_{\circledS p}^{p} = \sigma_1^p + \cdots + \sigma_h^p$. When $p \to 0$, we can see $\lim_{p\to 0} \|\mathbf{C}\|_{\circledS p}^{p} = \#\{i\text{: } \sigma_i \neq 0\} = \text{rank}(\mathbf{C})$. Thus, comparing with existing nuclear norm, *i.e.*, $p = 1$, **Schatten *p*-norm minimization (which is a quasi-norm) can ensure the rank of the C more easily approximates the target rank**. As shown in Fig. 2, for tensor $\mathcal{C}$, the j-th frontal slice $\mathbf{\Psi}^{(j)}$ characterize the relationship between n samples and the j-th anchor in different views. The goal of multi-view learning is that $\mathbf{C}_{:,j}^{(1)}, \cdots, \mathbf{C}_{:,j}^{(V)}$ are as similar as possible, ideally. Moreover, there has a large difference between cluster structures of different views in practice. Thus, **tensor Schatten *p*-norm constraint on $\mathcal{C}$ can make sure that $\mathbf{\Psi}^{(j)}$ has spatial low-rank structure, which helps exploit the complementary information embedded in inter-views and get the view-consensus graph**.

Remark 3. **[The benefit of $\ell_{1,2}$-norm]** In the model (5), by imposing the $\ell_{1,2}$-norm penalty on $\mathbf{C}^{(v)}$, as γ increases, in each $C_{i,1}^{(v)}, \cdots, C_{i,m}^{(v)}$ of the i-th row $\mathbf{C}_i^{(v)}$, at least one component remains non-zero. By doing so, some discriminative components remain non-zero to provide certain flexibility in the learned graph $\mathbf{C}^{(v)}$, *i.e.*, making $\mathbf{C}^{(v)}$ well encode the discriminative information and cluster structure.

3.2 Optimization

To solve the model (5), it is difficult to directly cope with the Laplacian rank constraint, *i.e.*, $\text{rank}(\widetilde{\mathbf{L}}^{(v)}) = n + m - K$, which is a non-convex optimization problem. We can relax the rank constraint in the following way:

$$\|\widetilde{\mathbf{L}}^{(v)}\|_{\text{rank}=n+m-K} = \min \sum_{j\,=\,n+m-K+1}^{n+m} \sigma_j(\widetilde{\mathbf{L}}^{(v)}) \tag{6}$$

where $\sigma_j(\widetilde{\mathbf{L}}^{(v)})$ denotes the j-th singular value of $\widetilde{\mathbf{L}}^{(v)}$, and all singular values of $\widetilde{\mathbf{L}}^{(v)}$ are sorted in descending order, *i.e.*, $\sigma_1(\widetilde{\mathbf{L}}^{(v)}) \geq \sigma_2(\widetilde{\mathbf{L}}^{(v)}) \geq \cdots \geq \sigma_{n+m}(\widetilde{\mathbf{L}}^{(v)})$. If the right side of Eq. (6) is zero, then $\text{rank}(\widetilde{\mathbf{L}}^{(v)}) = n + m - K$. To tackle Eq. (6), we first introduce the following theorem.

Theorem 1. [40] If $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$ is a real symmetric matrix, then

$$\begin{aligned} \min_{\mathbf{p}_i^{\mathrm{T}}\mathbf{p}_j = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases}} \sum_{\ell=1}^{r} \mathbf{p}_\ell^{\mathrm{T}} \mathbf{\Pi} \mathbf{p}_\ell &= \min_{\mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I}} \text{tr}(\mathbf{P}^{\mathrm{T}} \mathbf{\Pi} \mathbf{P}) \\ &= \lambda_{n-r+1} + \cdots + \lambda_n \\ &= \sum_{\ell=n-r+1}^{n} \boldsymbol{\varphi}_\ell^{\mathrm{T}} \mathbf{\Pi} \boldsymbol{\varphi}_\ell = \text{tr}(\mathbf{\Omega}^{\mathrm{T}} \mathbf{\Pi} \mathbf{\Omega}) \end{aligned} \tag{7}$$

where $\mathbf{\Omega} = [\boldsymbol{\varphi}_{n-r+1}, \cdots, \boldsymbol{\varphi}_n]$, and $\lambda_1 \geq \cdots \geq \lambda_n$ are n eigenvalues of $\mathbf{\Pi}$. $\boldsymbol{\varphi}_1, \cdots, \boldsymbol{\varphi}_n$ are the orthonormal eigenvectors corresponding to $\lambda_1, \lambda_2, \cdots, \lambda_n$, respectively.

Since the Laplacian matrix $\widetilde{\mathbf{L}}^{(v)}$ is a positive semi-definite and real symmetric matrix, its eigenvalues and singular values are identical. According to Theorem 1, we have

$$\sum_{j\,=\,n+m-K+1}^{n+m} \sigma_j(\widetilde{\mathbf{L}}^{(v)}) = \min_{\mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I}} \text{tr}(\mathbf{P}^{\mathrm{T}} \widetilde{\mathbf{L}}^{(v)} \mathbf{P}) \tag{8}$$

where $\mathbf{P} = [\mathbf{p}_1; \cdots; \mathbf{p}_{n+m}] \in \mathbb{R}^{(n+m)\times K}$ is the indicator matrix of the v-th view. Then, the model (5) is equivalent to the following problem:

$$\begin{aligned} \min_{\substack{\mathbf{C}^{(v)},\, \mathbf{E}^{(v)} \\ \mathbf{P},\, \xi^{(v)}}} & \|\mathcal{C}\|_{\circledS p}^{p} + \alpha \sum_{v=1}^{V} \|\mathbf{E}^{(v)}\|_1 + \gamma \sum_{v=1}^{V} \|\mathbf{C}^{(v)}\|_{1,2}^{2} + \beta \text{tr}(\mathbf{P}^{\mathrm{T}} \widetilde{\mathbf{L}}_{\mathbf{C}} \mathbf{P}) \\ s.t. \quad & \mathbf{B}^{(v)} = \mathbf{C}^{(v)} + \mathbf{E}^{(v)},\, \sum_{v=1}^{V} \xi^{(v)} = 1,\, \xi^{(v)} \geq 0 \\ & \mathbf{C}^{(v)}\mathbf{1} = \mathbf{1},\, \mathbf{C}^{(v)} \geq 0,\, \mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I} \end{aligned} \tag{9}$$

where $\widetilde{\mathbf{L}}_{\mathbf{C}} = \sum_{v=1}^{V} \frac{1}{\xi^{(v)}} \widetilde{\mathbf{L}}^{(v)}$; β is a hidden parameter, which is adaptively updated as follows. We first initialize β with a small value, and update it according to the number of eigenvalue zero of $\widetilde{\mathbf{L}}_{\mathbf{C}}$ after each iteration. If this number is smaller than K, β is multiplied by 2; if it is greater than K+1, β is divided by 2; otherwise we terminate the iterations.

Inspired by the augmented Lagrange multiplier method, we introduce the auxiliary variables $\mathcal{J}$ and $\mathbf{O}^{(v)}$, and rewrite the model (9) as the following unconstrained problem:

$$\begin{aligned}\mathcal{L}&(\mathbf{C}^{(1)}, \cdots, \mathbf{C}^{(V)}, \mathcal{J}, \mathbf{O}^{(1)}, \cdots, \mathbf{O}^{(V)}, \mathbf{E}^{(1)}, \cdots, \mathbf{E}^{(V)}, \mathbf{P})\\ &= \|\mathcal{J}\|_{\circledS p}^{p} + \alpha\sum_{v=1}^{V}\|\mathbf{E}^{(v)}\|_1 + \gamma\sum_{v=1}^{V}\|\mathbf{O}^{(v)}\|_{1,2}^2 + \beta tr(\mathbf{P}^{\mathrm{T}}\widetilde{\mathbf{L}}_{\mathbf{C}}\mathbf{P})\\ &\quad+\sum_{v=1}^{V}(\langle\mathbf{Y}_3^{(v)}, \mathbf{C}^{(v)}-\mathbf{O}^{(v)}\rangle + \frac{\tau}{2}\|\mathbf{C}^{(v)}-\mathbf{O}^{(v)}\|_F^2)\\ &\quad+\sum_{v=1}^{V}(\langle\mathbf{Y}_1^{(v)}, \mathbf{B}^{(v)}-\mathbf{E}^{(v)}-\mathbf{C}^{(v)}\rangle + \frac{\mu}{2}\|\mathbf{B}^{(v)}-\mathbf{E}^{(v)}-\mathbf{C}^{(v)}\|_F^2)\\ &\quad+\langle\mathcal{Y}_2, \mathcal{C}-\mathcal{J}\rangle + \frac{\rho}{2}\|\mathcal{C}-\mathcal{J}\|_F^2\end{aligned} \tag{10}$$

where $\mathbf{Y}_1{}^{(v)}$, $\mathcal{Y}_2$ and $\mathbf{Y}_3{}^{(v)}$ represent Lagrange multipliers; μ, ρ and τ are the penalty parameters. Consequently, the optimization process could be separated into five steps:

• **Solving P with fixed $\mathbf{C}^{(v)}$, $\mathbf{E}^{(v)}$, $\mathbf{O}^{(v)}$, $\xi^{(v)}$ and $\mathcal{J}$.** In this case, the optimization w.r.t. $\mathbf{P}$ in Eq. (10) becomes

$$\mathbf{P}^* = \arg\min_{\mathbf{P}^{\mathrm{T}}\mathbf{P}=\mathbf{I}} \mathrm{tr}(\mathbf{P}^{\mathrm{T}}\widetilde{\mathbf{L}}_{\mathbf{C}}\mathbf{P}) \tag{11}$$

where $\widetilde{\mathbf{L}}_{\mathbf{C}} = \sum_{v=1}^{V}\frac{1}{\xi^{(v)}}\left(\mathbf{I}-\mathbf{D}_{\mathbf{F}^{(v)}}^{-\frac{1}{2}}\mathbf{F}^{(v)}\mathbf{D}_{\mathbf{F}^{(v)}}^{-\frac{1}{2}}\right)$.

To directly optimize Eq. (11), the computational complexity is $\mathcal{O}((n+m)^2K)$, resulting in failing to deal with large scale multi-view clustering. Instead of doing this, we herein provide an efficient algorithm. Substituting $\widetilde{\mathbf{L}}_{\mathbf{C}}$ into Eq. (11), and by using some simple matrix algebra, it is ready to see

$$tr(\mathbf{P}^{\mathrm{T}}\widetilde{\mathbf{L}}_{\mathbf{C}}\mathbf{P}) = \sum_{v=1}^{V}\frac{1}{\xi^{(v)}}[tr(\mathbf{P}^{\mathrm{T}}\mathbf{P}) - tr(\mathbf{P}^{\mathrm{T}}\mathbf{D}_{\mathbf{F}^{(v)}}^{-\frac{1}{2}}\mathbf{F}^{(v)}\mathbf{D}_{\mathbf{F}^{(v)}}^{-\frac{1}{2}}\mathbf{P})] \tag{12}$$

Since $\mathbf{P}^{\mathrm{T}}\mathbf{P}=\mathbf{I}$, then, Eq. (11) can be rewritten as

$$\max_{\mathbf{P}^{\mathrm{T}}\mathbf{P}=\mathbf{I}}\sum_{v=1}^{V}\frac{1}{\xi^{(v)}}tr(\mathbf{P}^{\mathrm{T}}\mathbf{D}_{\mathbf{F}^{(v)}}^{-\frac{1}{2}}\mathbf{F}^{(v)}\mathbf{D}_{\mathbf{F}^{(v)}}^{-\frac{1}{2}}\mathbf{P}). \tag{13}$$

Let $\mathbf{P} = [\mathbf{P}_U, \mathbf{P}_M]^{\mathrm{T}}$ and $\mathbf{D}_{\mathbf{F}}^{(v)} = diag(\mathbf{D}_U^{(v)}, \mathbf{D}_M^{(v)})$, where $\mathbf{P}_U \in \mathbb{R}^{n\times K}$ is the first n rows of $\mathbf{P}$ and $\mathbf{P}_M \in \mathbb{R}^{m\times K}$ is the remaining m rows of $\mathbf{P}$, $\mathbf{D}_U^{(v)} \in \mathbb{R}^{n\times\mathbf{P}_U}$ and $\mathbf{D}_M^{(v)} \in \mathbb{R}^{m\times m}$ are diagonal matrices whose diagonal elements are $\mathbf{D}_U^{(v)}(i,i)=\sum_{j=1}^{m}\mathbf{C}^{(v)}(i,j)$ and $\mathbf{D}_M^{(v)}(j,j)=\sum_{i=1}^{n}\mathbf{C}^{(v)}(i,j)$. Substituting them into Eq. (13), and from some simple matrix algebra, Eq. (13) becomes

$$\mathbf{P}^* = \arg\max_{\mathbf{P}_U^{\mathrm{T}}\mathbf{P}_U+\mathbf{P}_M^{\mathrm{T}}\mathbf{P}_M=\mathbf{I}} 2tr(\mathbf{P}_U^{\mathrm{T}}\mathbf{W}\mathbf{P}_M). \tag{14}$$

where $\mathbf{W} = \sum_{v=1}^{V}\frac{\mathbf{C}^{(v)}\mathbf{D}_M^{(v)\,-\frac{1}{2}}}{\xi^{(v)}}$. The optimal solution $\mathbf{P}^*$ in Eq. (14) can be efficiently obtained by Theorem 2.

Theorem 2. Suppose $\mathbf{W} \in \mathbb{R}^{n\times m}$, $\mathbf{P}_U \in \mathbb{R}^{n\times K}$, $\mathbf{P}_M \in \mathbb{R}^{m\times K}$. The optimal solutions to the following model:

$$\max_{\mathbf{P}_U^{\mathrm{T}}\mathbf{P}_U+\mathbf{P}_M^{\mathrm{T}}\mathbf{P}_M=\mathbf{I}} tr(\mathbf{P}_U^{\mathrm{T}}\mathbf{W}\mathbf{P}_M) \tag{15}$$

are $\mathbf{P}_U = \frac{\sqrt{2}}{2}\mathbf{U}_1$, $\mathbf{P}_M = \frac{\sqrt{2}}{2}\mathbf{V}_1$, where $\mathbf{U}_1$ and $\mathbf{V}_1$ are the leading K left and right singular vectors of $\mathbf{W}$, respectively.

According to Theorem 2, we have that the computational complexity is $\mathcal{O}(Vnm + m^2n)$ for solving the optimal solution $\mathbf{P}^*$. Compared with directly solving Eq. (11) whose computational complexity is $\mathcal{O}((n+m)^2K)$, TBGL is very efficient to handle large-scale multi-view data due to $m \ll n$. **The proof of Theorem 2 is in Appendix A.**

• **Solving $\mathbf{C}^{(v)}$ with fixed $\mathbf{E}^{(v)}$, $\mathcal{J}$, $\mathbf{O}^{(v)}$, $\xi^{(v)}$ and P.** Now, the optimization w.r.t. $\mathbf{C}^{(v)}$ in Eq. (10) is equivalent to

$$\begin{aligned}\min_{\mathbf{C}^{(v)}}&\langle\mathbf{Y}_2^{(v)}, \mathbf{C}^{(v)}-\mathbf{J}^{(v)}\rangle + \frac{\rho}{2}\|\mathbf{C}^{(v)}-\mathbf{J}^{(v)}\|_F^2 + \beta\mathrm{tr}(\mathbf{P}^{\mathrm{T}}\widetilde{\mathbf{L}}_{\mathbf{C}}\mathbf{P})\\ &+\sum_{v=1}^{V}(\langle\mathbf{Y}_3^{(v)}, \mathbf{C}^{(v)}-\mathbf{O}^{(v)}\rangle + \frac{\tau}{2}\|\mathbf{C}^{(v)}-\mathbf{O}^{(v)}\|_F^2)\\ &+\langle\mathbf{Y}_1^{(v)}, \mathbf{B}^{(v)}-\mathbf{E}^{(v)}-\mathbf{C}^{(v)}\rangle + \frac{\mu}{2}\|\mathbf{B}^{(v)}-\mathbf{E}^{(v)}-\mathbf{C}^{(v)}\|_F^2\\ &s.t. \quad \mathbf{C}^{(v)}\mathbf{1}=\mathbf{1}, \mathbf{C}^{(v)}\geq 0\end{aligned} \tag{16}$$

According to Eq. (14), the last term in Eq. (16) can be rewritten as

$$tr(\mathbf{P}^{\mathrm{T}}\widetilde{\mathbf{L}}\mathbf{P}) = \text{Constant} - 2\sum_{v=1}^{V}tr(\mathbf{C}^{(v)^{\mathrm{T}}}\mathbf{H}^{(v)^{\mathrm{T}}}) \tag{17}$$

where $\mathbf{H}^{(v)^{\mathrm{T}}} = \frac{\mathbf{D}_M^{(v)\,-\frac{1}{2}}}{\xi^{(v)}}\mathbf{P}_M\mathbf{P}_U^{\mathrm{T}}$. Substituting Eq. (17) into Eq. (16), and by some simple matrix algebra, Eq. (16) becomes

$$\min_{\mathbf{C}^{(v)}}\frac{\rho+\mu+\tau}{2}\|\mathbf{C}^{(v)} - \frac{\boldsymbol{\Lambda}}{\rho+\mu+\tau}\|_F^2, \ s.t.\ \mathbf{C}^{(v)}\mathbf{1}=\mathbf{1}, \mathbf{C}^{(v)}\geq 0 \tag{18}$$

where $\boldsymbol{\Lambda} = \rho\mathbf{G}^{(v)}+\mu\mathbf{Q}^{(v)}+\tau\Upsilon^{(v)}+2\beta\mathbf{H}^{(v)^{\mathrm{T}}}$; $\mathbf{G}^{(v)}=\mathbf{J}^{(v)}-\frac{1}{\rho}\mathbf{Y}_2^{(v)}$; $\Upsilon^{(v)}=\mathbf{O}^{(v)}-\frac{1}{\tau}\mathbf{Y}_3^{(v)}$; $\mathbf{Q}^{(v)} = \mathbf{B}^{(v)} - \mathbf{E}^{(v)} + \frac{1}{\mu}\mathbf{Y}_1^{(v)}$. To this end, the closed-form solution $\mathbf{C}^{(v)^*}$ is $\mathbf{c}_i^{(v)^*} = (\frac{\boldsymbol{\Lambda}_i}{\rho+\mu+\tau} + \gamma\mathbf{1})_+$ [41], where γ is the Lagrangian multiplier.

• **Solving $\mathbf{E}^{(v)}$ with fixed $\mathbf{C}^{(v)}$, $\mathcal{J}$, P and $\xi^{(v)}$.** In this case, the optimization w.r.t. $\mathbf{E}^{(v)}$ in (10) becomes

$$\mathbf{E}^{(v)^*} = \arg\min_{\mathbf{E}^{(v)}}\frac{\alpha}{\mu}\|\mathbf{E}^{(v)}\|_1 + \frac{1}{2}\|\mathbf{E}^{(v)} - \Gamma^{(v)}\|_F^2 \tag{19}$$

where $\Gamma^{(v)} = \mathbf{B}^{(v)} - \mathbf{C}^{(v)} + \frac{1}{\mu}\mathbf{Y}_1^{(v)}$. The optimal solution of Eq. (19) is $\mathbb{S}_{\frac{\alpha}{\mu}}[\Gamma^{(v)}]$, where $\mathbb{S}_{\frac{\alpha}{\mu}}[x] = \mathrm{sign}(x)\max(|x|-\frac{\alpha}{\mu}, 0)$ is the soft-thresholding operator [42].

• **Solving $\mathbf{O}^{(v)}$ with fixed $\mathbf{E}^{(v)}$, $\mathcal{J}$, $\mathbf{C}^{(v)}$, $\xi^{(v)}$ and P.** Now, the optimization w.r.t. $\mathbf{O}^{(v)}$ in Eq. (10) is equivalent to

$$\mathbf{O}^{(v)^*} = \arg\min_{\mathbf{O}^{(v)}}\frac{2\gamma}{\tau}\|\mathbf{O}^{(v)}\|_{1,2}^2 + \frac{1}{2}\|\mathbf{O}^{(v)} - (\mathbf{C}^{(v)} + \frac{\mathbf{Y}_3^{(v)}}{\tau})\|_F^2 \tag{20}$$

The optimal solution of Eq. (20) can be computed by [15].

• **Solving $\mathcal{J}$ with fixed $\mathbf{C}^{(v)}$, $\mathbf{E}^{(v)}$, P and $\xi^{(v)}$.** In this case, $\mathcal{J}$ can be solved by

$$\begin{aligned}\mathcal{J}^* &= \arg\min_{\mathcal{J}}\|\mathcal{J}\|_{\circledS p}^{p} + \langle\mathcal{Y}_1, \mathcal{C}-\mathcal{J}\rangle + \frac{\rho}{2}\|\mathcal{C}-\mathcal{J}\|_F^2\\ &= \arg\min_{\mathcal{J}}\frac{1}{\rho}\|\mathcal{J}\|_{\circledS p}^{p} + \frac{1}{2}\|\mathcal{C} + \frac{\mathcal{Y}_2}{\rho} - \mathcal{J}\|_F^2\end{aligned} \tag{21}$$

To solve Eq. (21), we first introduce the Theorem 3 [14].

Theorem 3. [14] Suppose $\mathcal{Z} \in \mathbb{R}^{n_1\times n_2\times n_3}$, $h = \min(n_1, n_2)$, let $\mathcal{Z} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$. For the following model:

$$\underset{\mathcal{X}}{\mathrm{argmin}}\frac{1}{2}\|\mathcal{X}-\mathcal{Z}\|_F^2 + \tau\|\mathcal{X}\|_{\circledS p}^{p} \tag{22}$$

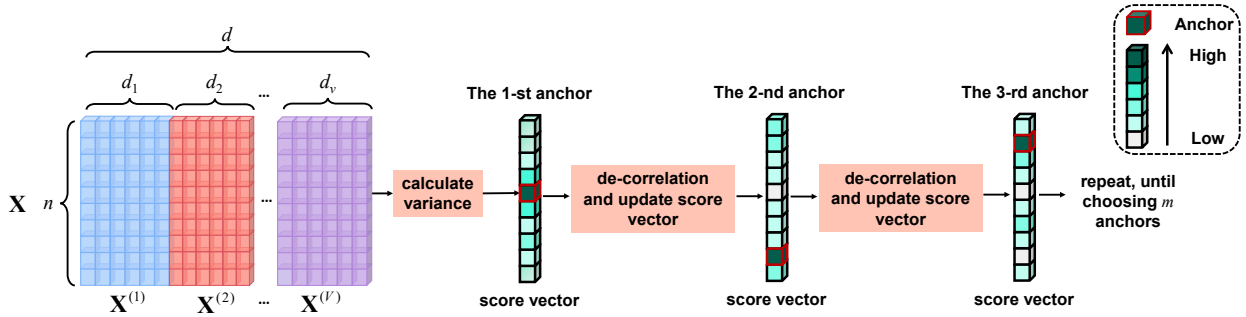


Fig. 3. Illustration of Variance-based De-correlation Anchor selection (VDA).

Algorithm 1: Procedure for solving TBGL

Input: Data matrices: $\{\mathbf{X}^{(v)}\}_{v=1}^{V} \in \mathbb{R}^{n\times d_v}$, anchors number m, and cluster number K, α.
Output: Graph $\mathbf{C}$ with K-connected components.

Select anchors $\{\mathbf{A}^{(v)}\}_{v=1}^{V} \in \mathbb{R}^{m\times d_v}$ by Algorithm 2;
Construct bipartite graphs $\{\mathbf{B}^{(v)}\}_{v=1}^{V} \in \mathbb{R}^{n\times m}$;
Initialize $\mathbf{C}^{(v)}$=$\mathbf{B}^{(v)}$, $\mathbf{E}^{(v)}=0$, $\mathbf{Y}_1^{(v)}$=0, $\mathcal{Y}_2$=0, $\mathcal{J}$=0, ρ=10^{-5}, μ=10^{-5}, η=1.1, $\xi^{(v)}=1/V$;
while *not converge* **do**
  Update $\mathbf{P}$ by solving Eq. (14);
  Update $\{\mathbf{C}^{(v)}\}_{v=1}^{V}$ by solving Eq. (18);
  Update $\{\mathbf{E}^{(v)}\}_{v=1}^{V}$ by solving Eq. (19);
  Update $\{\mathbf{O}^{(v)}\}_{v=1}^{V}$ by solving Eq. (20);
  Update $\mathcal{J}$ by using Eq. (24);
  Update $\xi^{(v)}$ by using Eq. (28);
  Update $\mathbf{Y}_1^{(v)}$, $\mathcal{Y}_2$, $\mathbf{Y}_3^{(v)}$, μ, ρ and τ:
  $\mathbf{Y}_1^{(v)} := \mathbf{Y}_1^{(v)} + \mu(\mathbf{B}^{(v)} - \mathbf{C}^{(v)} - \mathbf{E}^{(v)})$, μ=$\eta\mu$,
  $\mathcal{Y}_2 := \mathcal{Y}_2 + \rho(\mathcal{C} - \mathcal{J})$, $\rho := \eta\rho$,
  $\mathbf{Y}_3^{(v)}$=$\mathbf{Y}_3^{(v)} + \tau(\mathbf{C}^{(v)} - \mathbf{O}^{(v)})$, τ=$\eta\tau$;
end
Directly achieve the K clusters based on the connectivity of $\mathbf{C} = \sum_{v=1}^{V} \frac{\mathbf{C}^{(v)}}{\xi^{(v)}} / \sum_{v=1}^{V} \frac{1}{\xi^{(v)}}$;
return: Clustering results.

the optimal solution $\boldsymbol{\mathcal{X}}^*$ is

$$\boldsymbol{\mathcal{X}}^* = \Gamma_{\tau\cdot n_3}(\boldsymbol{\mathcal{Z}}) = \boldsymbol{\mathcal{U}} * ifft(\mathbf{P}_{\tau\cdot n_3}(\overline{\boldsymbol{\mathcal{Z}}})) * \boldsymbol{\mathcal{V}}^{\mathbf{T}} \tag{23}$$

where $\mathbf{P}_{\tau\cdot n_3}(\overline{\boldsymbol{\mathcal{Z}}}) \in \mathbb{R}^{n_1\times n_2\times n_3}$ is a tensor with the i-th frontal slice $\mathbf{P}_{\tau\cdot n_3}(\overline{\boldsymbol{\mathcal{Z}}}^{(i)})$ whose elements can be obtained by using the GST algorithm in Lemma 1 of [14].

According to Theorem 3, the solution of Eq. (21) is

$$\mathcal{J}^* = \Gamma_{\frac{1}{\rho}}(\mathcal{C} + \frac{1}{\rho}\mathcal{Y}_2). \tag{24}$$

• **Solving $\xi^{(v)}$ with fixed other variables.** In this case, the optimization w.r.t. $\xi^{(v)}$ in Eq. (10) is equivalent to

$$\min_{\xi^{(v)}} \sum_{v=1}^{V} \frac{tr(\mathbf{P}^{\mathbf{T}}\widetilde{\mathbf{L}}_{\mathbf{F}^{(v)}}\mathbf{P})}{\xi^{(v)}}, \quad \text{s.t.} \quad \sum_{v=1}^{V} \xi^{(v)} = 1, \xi^{(v)} \geq 0 \tag{25}$$

Let $\triangle^{(v)} = \sqrt{tr(\mathbf{P}^{\mathbf{T}}\widetilde{\mathbf{L}}_{\mathbf{F}^{(v)}}\mathbf{P})}$, (25) becomes

$$\min_{\xi^{(v)}} \sum_{v=1}^{V} \frac{\triangle^{(v)2}}{\xi^{(v)}}, \quad \text{s.t.} \quad \sum_{v=1}^{V} \xi^{(v)} = 1, \xi^{(v)} \geq 0 \tag{26}$$

Due to $\sum_{v=1}^{V} \xi^{(v)} = 1$, according to Cauchy-Schwartz's inequality, we have

$$\sum_{v=1}^{V} \frac{\triangle^{(v)2}}{\xi^{(v)}} = \left(\sum_{v=1}^{V} \frac{\triangle^{(v)2}}{\xi^{(v)}}\right)\left(\sum_{v=1}^{V} \xi^{(v)}\right) \geqslant \left(\sum_{v=1}^{V} \triangle^{(v)}\right)^2 \tag{27}$$

where equality holds iff $\sqrt{\xi^{(v)}} \propto \frac{\triangle^{(v)}}{\sqrt{\xi^{(v)}}}$. Moreover, the right-hand side of Eq. (27) is a constant, therefore $\forall v = 1, 2, \cdots, V$, the optimal $\xi^{(v)}$ is

$$\xi^{(v)} = \triangle^{(v)} / \sum_{v=1}^{V} \triangle^{(v)} \tag{28}$$

Finally, the optimization procedure for solving Eq. (5) is outlined in Algorithm 1. **The proof of Convergence Analysis is in Appendix B.**

4 Anchor Selection and Discussion

4.1 Anchor Selection

Given multi-view data matrices $\{\mathbf{X}^{(v)}\}_{v=1}^{V} \in \mathbb{R}^{n\times d_v}$, the way to construct adjacency matrices $\{\mathbf{B}^{(v)}\}_{v=1}^{V} \in \mathbb{R}^{n\times m}$ for multiple views is crucial. The core of the bipartite graphs construction is sampling m ($m \ll n$) representative data points, *i.e.*, anchors, from all sample points. The random sampling technique and the K-Means algorithm are two of the most representative strategies to select anchors [43], [44]. The random sampling technique randomly selects m data points from all samples as anchors. Despite being simple and efficient, the results of selected anchor points are of being occasional which leads to unstable and unsatisfactory clustering results. By contrast, the K-Means based anchor selection strategy directly leverages m centroids as anchors. As we all know, K-Means is sensitive to the initial centroids, and needs to be run many times independently to eliminate the randomness of results like random sampling. Generally speaking, the ideal anchors should have the following characteristics: **1)** they should well cover the whole classes of data and characterize the intrinsic structure of data; **2)** they should cover the entire data point clouds evenly.

Principal component analysis (PCA) [45] is one of the most representative data representation techniques. It can extract the $m \ll d$ most expressive features, where d is the dimension of the raw sample. This well characterizes both the whole original dimensional space and all classes, by selecting projection directions corresponding to the first m largest variances. Inspired by this, we leverage PCA to extract m most representative samples by viewing the sample

Algorithm 2: Procedure for VDA

Input: Data matrices: $\{\mathbf{X}^{(v)}\}_{v=1}^{V} \in \mathbb{R}^{n\times d_v}$, anchors number m.
Output: The anchor matrices: $\{\mathbf{A}^{(v)}\}_{v=1}^{V} \in \mathbb{R}^{m\times d_v}$.

Obtain $\mathbf{X}$ by concatenating $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$;
Calculate the variance $\Theta = [\theta_1, \theta_2, \cdots, \theta_n]$ of d-dimensional features of each sample;
Normalize Θ, $\mathrm{Ind} = \arg\max_i \theta_i$, $\mathbf{a}_1^{(v)} = \mathbf{x}_{\mathrm{Ind}}^{(v)}$;
for $j = 2$: m **do**
  Calculate δ_i between θ_i and θ_{Ind} by Eq. (31);
  Update θ_i by Eq. (32);
  Normalize Θ, $\mathrm{Ind} = \arg\max_i \theta_i$, $\mathbf{a}_j^{(v)} = \mathbf{x}_{\mathrm{Ind}}^{(v)}$;
end
return: Selected m anchors $\mathbf{A}^{(v)}$ of v-th view.

space and dimensional space as dimensional space and sample space, respectively. But it fails for large-scale data due to the high computational complexity of finding the projection directions. The features of each original sample reflect the intrinsic representation of objects to some extent. Based on this, we propose a novel anchors selection scheme (namely Variance-based De-correlation Anchor selection, VDA), which is simple and efficient. VDA views the features of the i-th sample as the embedding of the i-th projection direction, and selects the m representative data points according to their variance in the dimension space. It avoids finding the projection direction.

A naive scheme is to select m representative points corresponding to the first m largest variances. However, it cannot ensure the selected anchors well denote all points and cover all entire point cloud of data. The reason is that, in real applications, there are high correlation among samples. Thus the samples close to each other in the feature space provide little additional information. The anchor points should well cover the whole classes of data and characterize the intrinsic structure of data. To this end, our strategy follows the above principle to alternately select according to the correlation among samples, as illustrated in Fig. 3. Specifically, given data matrices $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$, we first concatenate the data matrix of each view along the feature dimension. Thus, the connected feature matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ can be represented as $\mathbf{X} = [\mathbf{X}^{(1)}; \mathbf{X}^{(2)}; \cdots; \mathbf{X}^{(v)}]$, where $d = \sum_{v=1}^{V} d_v$ is the dimension of connected feature matrix. Then the variance θ_i of the d-dimensional features of i-th sample can be calculated by using

$$\theta_i = \mathrm{Var}(\mathbf{x}_i), \tag{29}$$

where $\mathrm{Var}(\cdot)$ represents the calculating the variance; $\Theta = [\theta_1, \theta_2, \cdots, \theta_n] \in \mathbb{R}^n$ is the variance vector. After that, we normalize the score vector by dividing the largest score. Since the data point with large variance contains more information, we choose the data point with the largest score as the initial anchor point by solving

$$\mathrm{Ind} = \arg\max_i \theta_i. \tag{30}$$

Thus, the 1-st anchor in the v-th view is $\mathbf{a}_1^{(v)} = \mathbf{x}_{\mathrm{Ind}}^{(v)}$.

Noting that there are high correlation among the samples. And the samples close to each other in the feature space provide little additional information. To maximize the quantity of information provided by the selected anchors, we consider the correlation among samples. Let θ_{Index} denote the variance of the anchors selected from the previous round, the correlation δ_i between θ_{Index} and θ_i can be calculated as

$$\delta_i = \frac{1}{1 + \|\theta_{\mathrm{Index}} - \theta_i\|^2}, \tag{31}$$

where $\delta = [\delta_1, \delta_2, \cdots, \delta_n] \in \mathbb{R}^n$ is the correlation vector. Then we update the variance θ_i of the i-th sample by using

$$\theta_i = \theta_i \times (1 - \delta_i) \tag{32}$$

By leveraging this strategy, the data points with extremely high or low variance values would diminish, whereas the data points with medial scores would be exaggerated fairly in the next round. So the next selected anchor is of little chance being in the same clusters with the current selected anchor. Moreover, the variance of the selected data point is updated to 0, since we always select the sample with the highest variance θ_i. Therefore none of anchors can be chosen repeatedly.

After that, we normalize the updated score vector by dividing the largest score, the index of j-th anchor can be obtained by Eq. (30). Thus, the j-th anchor in the v-th view is $\mathbf{a}_j^{(v)} = \mathbf{x}_{\mathrm{Ind}}^{(v)}$. We alternately execute Eqs. (31 - 32) until m anchors are selected. We denote the selected m anchors of v-th view as $\mathbf{A}^{(v)} \in \mathbb{R}^{m\times d_v}$. The procedure for the proposed VDA is outlined in Algorithm 2.

Compared with the K-Means and random sampling technique based anchor selection strategies, VDA can provide more excellent performance for the large-scale data in the following aspects:

1) VDA does not require random initialization. So VDA can achieve stable anchor selection results.
2) Taking the correlation of the samples into consideration, VDA can ensure that the selected anchors not only cover the whole classes, but also characterize the intrinsic structure of data.

After obtaining m anchors, *i.e.*, $\mathbf{A}^{(v)}$ for each view, we can construct the adjacency matrix $\mathbf{B}^{(v)}$ by using many methods [18], [36], [46]. We herein leverage KMM [36] to construct $\mathbf{B}^{(v)}$ for each view.

TABLE 1
Storage complexity of TBGL, where *V, n, m, K* are the number of views, samples, anchors, and clusters, respectively

Variable	$\{\mathbf{C}^{(v)}, \mathbf{E}^{(v)}, \mathbf{O}^{(v)}, \mathbf{Y}_1^{(v)}, \mathbf{Y}_3^{(v)}\}_{v=1}^{V}$	$\mathbf{P}$	$\mathcal{J}$	$\mathcal{Y}_2$
Complexity	$\mathcal{O}((n+m)K)$	$\mathcal{O}(5Vnm)$	$\mathcal{O}(Vnm)$	$\mathcal{O}(Vnm)$
Total	$\mathcal{O}\left[n(7Vm+K)+mK\right]$			

4.2 Computational and Storage Complexities Analysis

Computational complexity: TBGL consists of two stages: 1) construction of graphs $\{\mathbf{B}^{(v)}\}_{v=1}^{V}$, 2) optimization by iterative solving Eq. (10). The first stage takes $\mathcal{O}(Vnmd+Vnm\log(m))$ time, where V, m and n are the number of views, anchors and samples, respectively. The second stage mainly focuses on five variables ($\mathbf{C}^{(v)}$, $\mathcal{J}$, $\mathbf{O}^{(v)}$, $\mathbf{E}^{(v)}$, $\mathbf{P}$). The complexity in updating these variables iteratively are $\mathcal{O}(VnmK+Vnm\log(m))$, $\mathcal{O}(Vnm\log(Vn)+V^2mn)$, $\mathcal{O}(Vnm)$,

TABLE 2
The clustering performances on MSRC-v5 and Handwritten4 datasets

Dataset	**MSRC**						
Metric	ACC	NMI	Purity	PER	REC	F-score	ARI
s-CLR (best) [41]	0.681±0.000	0.608±0.000	0.710±0.000	0.478±0.000	0.707±0.000	0.570±0.000	0.471±0.000
s-CLR-C [41]	0.590±0.000	0.509±0.000	0.614±0.000	0.451±0.000	0.482±0.000	0.466±0.000	0.377±0.000
Co-reg [8]	0.635±0.007	0.578±0.006	0.659±0.006	0.511±0.008	0.535±0.007	0.522±0.007	0.425±0.030
SwMC [24]	0.776±0.000	0.774±0.000	0.805±0.000	0.687±0.000	0.831±0.000	0.752±0.000	0.708±0.000
MVGL [23]	0.690±0.000	0.663±0.000	0.733±0.000	0.466±0.000	0.715±0.000	0.564±0.000	0.476±0.000
MVSC [11]	0.794±0.075	0.672±0.058	0.756±0.071	0.585±0.091	0.779±0.035	0.664±0.062	0.600±0.079
AMGL [10]	0.751±0.078	0.704±0.044	0.789±0.056	0.621±0.090	0.744±0.026	0.674±0.063	0.615±0.079
RMSC [47]	0.762±0.040	0.663±0.026	0.769±0.030	0.640±0.030	0.660 ±0.034	0.650±0.031	0.592±0.036
CSMSC [48]	0.758±0.007	0.735±0.010	0.793±0.008	0.736 ±0.014	0.673 ±0.008	0.703±0.010	0.653±0.012
SFMC [13]	0.810±0.000	0.721±0.000	0.810±0.000	0.657±0.000	0.782±0.000	0.714±0.000	0.663±0.000
t-SVD-MSC [26]	0.967±0.000	0.936±0.000	0.967±0.000	0.932±0.000	0.938±0.000	0.935±0.000	0.924±0.000
ETLMSC [25]	0.962±0.000	0.937±0.000	0.962±0.000	0.926±0.000	0.931±0.000	0.928±0.000	0.917±0.000
LTCSPC [9]	0.981±0.000	0.957±0.000	0.981±0.000	0.962±0.000	0.962±0.000	0.962±0.000	0.956±0.000
TBGL-C	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
TBGL	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
Dataset	**HW**						
Metric	ACC	NMI	Purity	PER	REC	F-score	ARI
s-CLR (best) [41]	0.698±0.000	0.731±0.000	0.731±0.000	0.592±0.000	0.803±0.000	0.681±0.000	0.640±0.000
s-CLR-C [41]	0.759±0.000	0.751±0.000	0.760±0.000	0.610±0.000	0.865±0.000	0.716±0.000	0.678±0.000
Co-reg [8]	0.784±0.010	0.758±0.004	0.795±0.008	0.698±0.010	0.724±0.005	0.710±0.007	0.667±0.037
SwMC [24]	0.758±0.000	0.833±0.000	0.792±0.000	0.686±0.000	0.867±0.000	0.766±0.000	0.737±0.000
MVGL [23]	0.811±0.000	0.809±0.000	0.831±0.000	0.721±0.000	0.826±0.000	0.770±0.000	0.743±0.000
MVSC [11]	0.796±0.059	0.820±0.030	0.808±0.044	0.715±0.082	0.838±0.035	0.769±0.046	0.741±0.053
AMGL [10]	0.704±0.045	0.762±0.040	0.732±0.042	0.591±0.081	0.781±0.022	0.670±0.060	0.628±0.070
RMSC [47]	0.681±0.043	0.661±0.022	0.713±0.037	0.582±0.035	0.617±0.026	0.599±0.030	0.533±0.034
CSMSC [48]	0.806±0.001	0.793±0.001	0.867±0.001	0.778±0.001	0.743±0.001	0.760±0.001	0.733±0.001
SFMC [13]	0.853±0.000	0.871±0.000	0.873±0.000	0.775±0.000	0.910±0.000	0.837±0.000	0.817±0.000
t-SVD-MSC [26]	0.988±0.000	0.972±0.000	0.988±0.000	0.976±0.000	0.976±0.000	0.976±0.000	0.974±0.000
ETLMSC [25]	0.938±0.001	0.893±0.001	0.938±0.001	0.886±0.001	0.890±0.001	0.888±0.001	0.876±0.001
LTCSPC [9]	0.988±0.000	0.973±0.000	0.988±0.000	0.975±0.000	0.975±0.000	0.975±0.000	0.972±0.000
TBGL-C	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
TBGL	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**

$\mathcal{O}(Vnm)$ and $\mathcal{O}(Vnm+m^2\mathrm{n})$, where K and t are the number of clusters and iteration, respectively. For $m \ll n$, the main complexity in this stage is $\mathcal{O}(m^2nt+Vnmt\log(Vn))$. Therefore, the main computational complexity of TBGL is $\mathcal{O}(m^2nt+Vnmd)$, which is linear to n.

Storage complexity: During the optimization procedure, TBGL needs to store $\{\mathbf{C}^{(v)}, \mathbf{E}^{(v)}, \mathbf{O}^{(v)}, \mathbf{Y}_1^{(v)}, \mathbf{Y}_3^{(v)}\}_{v=1}^{V}$, $\mathbf{P}$, $\mathcal{J}$ and $\mathcal{Y}_2$, their corresponding storage complexities are shown in Table 1. Thus, the storage complexity of TBGL is $\mathcal{O}\left[n(7Vm + K) + mK\right]$, which is much less than that of t-SVD-MSC and ETLMSC with $\mathcal{O}\left[n(3Vn + 2Vd)\right]$ and $\mathcal{O}(3Vn^2)$, respectively, since $m \ll n$ in practice.

5 Experiments

In this section, we conduct extensive experiments to verify the effectiveness of TBGL and other multi-view clustering methods. We run all experiments on a standard Windows 10 Server with two Intel (R) Xeon (R) Gold 6230 CPUs 2.1 GHz and 128 GB RAM, MATLAB R2020a.

5.1 Experimental Setup

Datasets: We use the following three synthetic toy datasets and seven real-world datasets to make experiments:

1) ***Two-moon*** dataset [36] has 2 clusters. For the balanced two-moon dataset, each cluster has 200 data points. The unbalanced two-moon dataset has a total of 300 data points. One class has 100 data points and the other class has 200 data points.
2) ***Three-ring*** [13] dataset has 3 clusters. For the balanced three-ring dataset, each cluster has 200 data points. For the unbalanced three-ring dataset, there are 650 data points, where three clusters have 50, 200 and 400 data points, respectively.
3) ***Synthetic*** dataset [13] has 3 views, each view is a synthetic block diagonal graph. The 1st and 2nd views have two diagonal blocks with different cluster structure, respectively; the 3rd view only includes Gaussian noise. The graph of each view adds the uniform random noise.
4) ***MSRC-v5*** (MSRC) [49] includes 7 kinds of objects with 210 images. We choose 24-dimension (D) CM feature, 576-D HOG feature, 512-D GIST feature, 256-D LBP feature, 254-D CENT feature as 5 views.
5) ***Handwritten4*** (HW) [50] has 10 digits with 2,000 images from UCI machine learning repository. 76-D FOU feature, 216-D FAC feature, 47-D ZER feature and 6-D MOR feature are employed as 4 views.
6) ***Mnist4*** [51] includes 4 categories handwritten digits, *i.e.*, from digit 0 to digit 3, with 4, 000 images. We utilize 30-D ISO feature, 9-D LDA feature and 30-D NPE feature as 3 views.
7) ***Caltech101-20*** (Cal101) [52] has 20 categories with 2,386 images. It is a subset of Caltech101 datasets. We employ 48-D GABOR feature, 40-D WM feature, 254-D CENT feature, 1, 984-D HOG feature, 512-D GIST feature and 928-D LBP feature as 6 views.
8) ***NUS-WIDE*** (NUS) [53] has 31 categories with 30,000 object images. 64-D CH feature, 225-D CM feature,

TABLE 3
The clustering performances on Mnist4 and Caltech101-20 datasets

Dataset	**Mnist4**						
Metric	ACC	NMI	Purity	PER	REC	F-score	ARI
s-CLR (best) [41]	0.843±0.000	0.762±0.000	0.744±0.000	0.640±0.000	0.824±0.000	0.721±0.000	0.655±0.000
s-CLR-C [41]	0.897±0.000	0.747±0.000	0.897±0.000	0.813±0.000	0.822±0.000	0.817±0.000	0.756±0.000
Co-reg [8]	0.785±0.003	0.602±0.001	0.786±0.002	0.670±0.002	0.696±0.002	0.682±0.001	0.575±0.002
SwMC [24]	0.914±0.000	0.799±0.000	0.912±0.000	0.844±0.000	0.852±0.000	0.848±0.000	0.799±0.000
MVGL [23]	0.912±0.000	0.785±0.000	0.910±0.000	0.795±0.000	0.804±0.000	0.800±0.000	0.733±0.000
MVSC [11]	0.733±0.115	0.651±0.069	0.780±0.070	0.650±0.092	0.773±0.041	0.704±0.066	0.592±0.096
AMGL [10]	0.910±0.000	0.785±0.000	0.910±0.000	0.836±0.000	0.843±0.000	0.840±0.000	0.786±0.000
RMSC [47]	0.705±0.000	0.486±0.000	0.705±0.000	0.590±0.000	0.606±0.000	0.598±0.000	0.462±0.000
CSMSC [48]	0.643±0.000	0.645±0.010	0.832±0.008	0.776±0.014	0.612±0.008	0.684±0.010	0.562±0.012
SFMC [13]	0.917±0.000	0.801±0.000	0.917±0.000	0.846±0.000	0.855±0.000	0.852±0.000	0.802±0.000
t-SVD-MSC [26]	0.653±0.000	0.657±0.000	0.743±0.000	0.625±0.000	0.802±0.000	0.703±0.000	0.587±0.000
ETLMSC [25]	0.934±0.000	0.847±0.000	0.934±0.000	0.878±0.000	0.885±0.000	0.881±0.000	0.842±0.000
LTCSPC [9]	0.929±0.000	0.813±0.000	0.929±0.000	0.863±0.000	0.869±0.000	0.866±0.000	0.821±0.000
TBGL-C	0.930±0.000	0.841±0.000	0.930±0.000	0.869±0.000	0.878±0.000	0.874±0.000	0.831±0.000
TBGL	**0.938±0.000**	**0.859±0.000**	**0.938±0.000**	**0.884±0.000**	**0.890±0.000**	**0.887±0.000**	**0.849±0.000**
Dataset	**Cal101**						
Metric	ACC	NMI	Purity	PER	REC	F-score	ARI
s-CLR (best) [41]	0.442±0.000	0.269±0.000	0.492±0.000	0.198±0.000	0.745±0.000	0.313±0.000	0.076±0.000
s-CLR-C [41]	0.596±0.000	0.429±0.000	0.653±0.000	0.313±0.000	**0.817±0.000**	0.453±0.000	0.285±0.000
Co-reg [8]	0.412±0.006	0.587±0.003	0.754±0.004	0.712±0.008	0.243±0.004	0.363±0.006	0.295±0.025
SwMC [24]	0.599±0.000	0.493±0.000	0.700±0.000	0.509±0.000	0.625±0.000	0.431±0.000	0.265±0.000
MVGL [23]	0.600±0.000	0.474±0.000	0.696±0.000	0.325±0.000	0.653±0.000	0.440±0.000	0.282±0.000
MVSC [11]	0.595±0.000	0.613±0.000	0.717±0.000	0.542±0.000	0.546±0.000	0.541±0.000	0.451±0.000
AMGL [10]	0.557±0.047	0.552±0.061	0.677±0.058	0.480±0.093	0.539±0.015	0.503±0.054	0.397±0.080
RMSC [47]	0.385±0.024	0.512±0.012	0.742±0.013	0.692±0.038	0.213±0.019	0.346±0.260	0.288±0.027
CSMSC [48]	0.474±0.037	0.648±0.011	0.563±0.031	0.290±0.034	0.730±0.037	0.415±0.039	0.356±0.040
SFMC [13]	0.642±0.000	0.595±0.000	0.748±0.000	0.586±0.000	0.677±0.000	0.628±0.000	0.461±0.000
t-SVD-MSC [26]	0.526±0.024	0.705±0.012	0.862±0.008	**0.871±0.022**	0.348±0.024	0.497±0.028	0.446±0.029
ETLMSC [25]	0.483±0.017	0.681±0.007	0.845±0.013	0.832±0.017	0.275±0.007	0.413±0.010	0.362±0.010
LTCSPC [9]	0.639±0.000	0.768±0.000	0.807±0.000	0.737±0.000	0.532±0.000	0.615±0.000	0.555±0.000
TBGL-C	0.757±0.000	0.735±0.000	0.848±0.000	0.574±0.000	0.672±0.000	0.619±0.000	0.538±0.000
TBGL	**0.789±0.000**	**0.806±0.000**	**0.877±0.000**	0.695±0.000	0.696±0.000	**0.686±0.000**	**0.627±0.000**

144-D CORR feature, 73-D EDH feature and 128-D WT feature are adopted as 5 views.

9) ***Reuters*** [54] has 18,758 documents of 6 categories. We adopt 21, 513-D English, 24, 892-D France, 34, 251-D German, 15, 506-D Italian and 11, 547-D Spanish as 5 views.
10) ***Noisy MNIST*** [55] has 70, 000 samples with 2 views, where the first view is original data, and the second view is constructed by random choosing within-class images with white Gaussian noise. Due to some baselines cannot deal with such a large-scale dataset, we randomly select 50,000 samples in experiments.

Baselines: We compare TBGL with 15 competitors, *i.e.*, single-view constrained Laplacian rank (s-CLR) [41], Co-reg [8], SwMC [24], MVGL [23], MVSC [11], AMGL [10], RMSC [47], CSMSC [48], LMVSC [12], ETLMSC [25], t-SVD-MSC [26], LTCSPC [9] and SFMC [13]. For an unbiased comparison, for all baselines, we adjusted the hyper-parameters according to the experimental settings reported in their paper to obtain the hyper-parameters corresponding to the best results on each dataset. Then, we independently repeat the involved methods 20 times and show the averages with the corresponding standard deviations.

Evaluation Metrics: The widely used 7 metrics are applied to evaluate the clustering performance, *i.e.*, 1) Accuracy (ACC); 2) Normalized Mutual Information (NMI); 3) Purity; 4) Precision (PRE); 5) Recall (REC); 6) F-score; and 7) Adjusted Rand Index (ARI). For all metrics, the higher value indicates the better clustering performance. For more detailed definitions about the metrics, please refer to [27].

5.2 Comparisons with State-of-the-art Methods

Tables 2, 3, 4 present the comparison in metrics of the above methods on 7 datasets, where the best and second best results in all methods are represented by **bold** value and underline value, respectively. For CLR with single-view setting, s-CLR (best) denotes the best results of CLR by employing features in different views, and s-CLR-C denotes the results of s-CLR on the concatenated view-features. TBGL-C is TBGL with the using the entire graph. From Tables 2, 3, we discover that:

- The single-view clustering method s-CLR (best) is inferior to multi-view clustering methods. This is because the information embedded in different views are complementary. And the multi-view methods well use this important formation for boosting clustering performance, while s-CLR does not.
- The multi-view clustering method Co-reg is overall inferior to the other multi-view methods, since it neglects the significant difference between different views for clustering. This is also because its performance heavily depend on the graphs, which are artificially defined. However, in real-world applications, it is difficult to artificially select a suitable graph for some complex data.
- The proposed TBGL and other tensorized methods, *i.e.*, ETLMSC, t-SVD-MSC, and LTCSPC are superior to the other methods. Tensorized methods well exploit the complementary information and the spatial structure information embedded in the graphs of different views, while other methods do not.

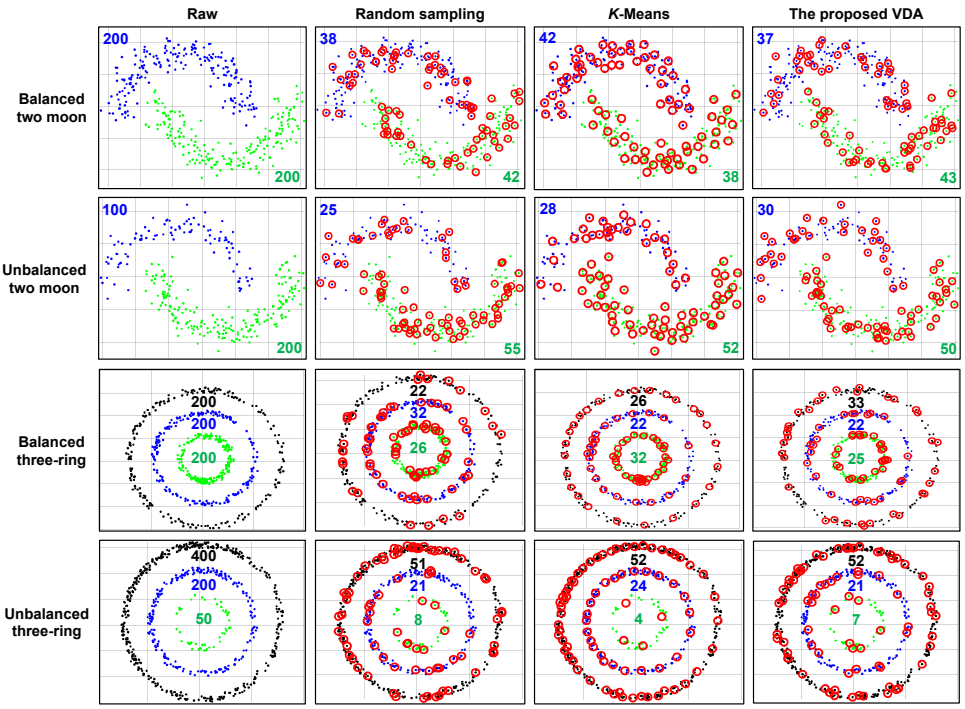


Fig. 4. The visualizations of anchor selection results of different methods on four toy datasets, where the points with same color represent a cluster; the corresponding colored numbers in the first column represent the number of data points in the cluster; the red circles represent the selected anchors; the corresponding colored numbers from the 2-nd to 4-th columns represent the number of selected anchors in the cluster.

TABLE 4
The clustering results and CPU time (sec.) on three large-scale datasets

Dataset	NUS							
Metric	ACC	NMI	Purity	PER	REC	F-score	ARI	Time
Co-reg	0.1194	0.1143	0.2144	0.1014	0.0623	0.0771	0.0360	3846.12
MVSC	0.1496	0.0752	0.1839	0.0865	0.1236	0.1017	0.0382	659.32
LMVSC	0.1140	0.0768	0.1708	0.1158	0.0680	0.0857	0.0161	**25.15**
t-SVD-MSC	OM [a]	OM	OM	OM	OM	OM	OM	OM
ETLMSC	OM	OM	OM	OM	OM	OM	OM	OM
SFMC	0.1689	0.0601	0.1904	0.0627	0.3595	0.1068	0.0124	181.72
TBGL	**0.2741**	**0.2023**	**0.2807**	**0.1308**	**0.4935**	**0.1566**	**0.0699**	552.35
Dataset	**Reuters**							
Metric	ACC	NMI	Purity	PER	REC	F-score	ARI	Time
Co-reg	0.5627	0.3261	0.5523	0.3751	0.4113	0.3954	0.2136	1219.59
MVSC	0.5958	0.3472	0.5741	0.4199	0.4832	0.4493	0.2858	581.32
LMVSC	0.5890	0.3346	0.6145	0.5364	0.3290	0.4151	0.2043	**150.51**
t-SVD-MSC	OM	OM	OM	OM	OM	OM	OM	OM
ETLMSC	OM	OM	OM	OM	OM	OM	OM	OM
SFMC	0.6023	0.3541	0.6042	0.4288	0.4917	0.4562	0.2967	494.68
TBGL	**0.7954**	**0.6595**	**0.7954**	**0.6160**	**0.9138**	**0.7359**	**0.6452**	697.43
Dataset	**Noisy MNIST**							
Metric	ACC	NMI	Purity	PER	REC	F-score	ARI	Time
Co-reg	OM	OM	OM	OM	OM	OM	OM	OM
MVSC	0.6795	0.7088	0.6795	0.5665	0.8006	0.4848	0.4010	1428.90
LMVSC	0.3885	0.3440	0.4344	0.3112	0.2998	0.3054	0.2263	**151.14**
t-SVD-MSC	OM	OM	OM	OM	OM	OM	OM	OM
ETLMSC	OM	OM	OM	OM	OM	OM	OM	OM
SFMC	0.6999	0.6811	0.7271	0.5035	0.7951	0.6166	0.5628	495.90
TBGL	**0.7638**	**0.7477**	**0.7812**	**0.6186**	**0.8447**	**0.6925**	**0.6386**	802.30

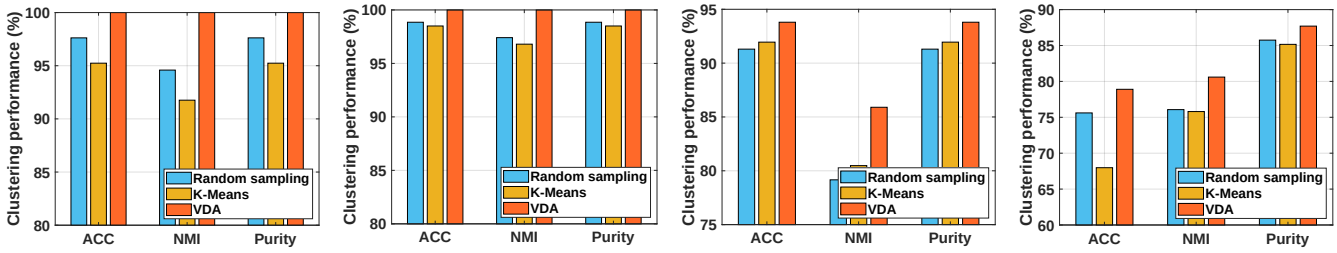
[a] *"OM" means "out-of-memory error"*

- The proposed TBGL is remarkably superior to SFMC and other non-tensorized methods. For example, on MSRC-v5 dataset, compared with SFMC, TBGL gains significant improvements around 19.0%, 27.9%, 19.0%, 34.3%, 21.8%, 28.6%, and 33.7% in terms of seven metrics, respectively. For MVC, an ideal view-consensus graph should have both the low-rank structure and K-connected components. To get the best clustering performance, the view-similar graphs among different views have not only high similarity but also high-similar spatial geometric structure. Our method explicitly takes into account these important information by minimizing the tensor Schatten p-norm, while SFMC does not. Moreover, in TBGL, the rank of the learned view-consensus graph approximates the target rank better than SFMC.
- Although TBGL is an anchor-based method, its performances also superior to ETLMSC, t-SVD-MSC and LTCSPC. This is mainly because TBGL simultaneously takes both the intra-view and inter-view similarity structures of the learned bipartite graphs into consideration by minimization tensor Schatten p-norm and $\ell_{1,2}$-norm penalty, thus the learned graph well characterizes the cluster structure and discriminative information. Moreover, when we use complete similarity graphs, *i.e.*, TBGL-C, it still gets relatively good results. Thus, the proposed techniques still work for the original full graph. In contrast, the clustering results of TBGL with using bipartite graph is better. These results indicate that bipartite graph helps to characterize the complex mechanisms of multi-view data and afford efficient clustering.

5.3 Experiments on Three Large-scale Datasets

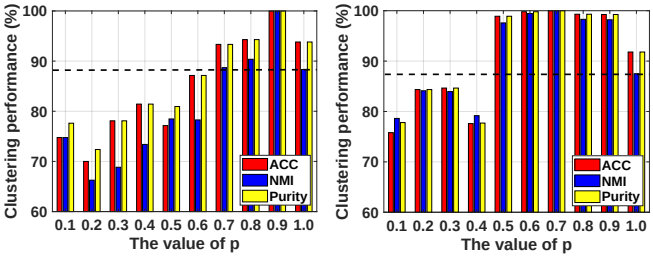
For the large scale datasets, due to CPU limitations, some methods, *e.g.*, SwMC, MVGL, AMGL, ETLMSC, t-SVD-MSC and LTCSPC, suffer from the out-of-memory issue. So we herein compare the clustering performances of our method with the partial competitors in Table 4. The number of anchors is set to 37, 400, and 1,000 on Reuters, NUS-WIDE, and Noisy MNIST dataset, respectively. As depicted in 4, we can draw conclusion that

- Comparing with non-tensorized method, *e.g.*, LMVSC and SFMC, it is true that the proposed TBGL takes



(a) MSRC-v5 (b) Handwritten4 (c) Mnist4 (d) Caltech101-20

Fig. 5. The clustering performances of the proposed method with different anchor selection methods on four real-world datasets.



(a) MSRC-v5 (b) Handwritten4

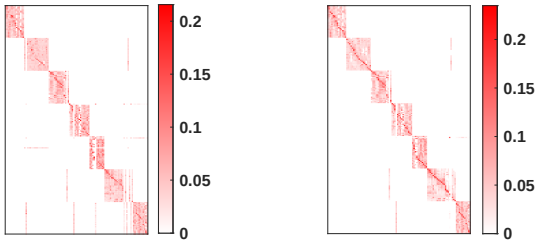
Fig. 6. The clustering performances of our method with the varying value of p on MSRC-v5 and Handwritten4 datasets.

TABLE 5
The clustering performances w./w.o. the $\ell_{1,2}$-norm constraint on six datasets, where the best results are represented by **bold** value

Dataset	Case	ACC	NMI	Purity	Dataset	Case	ACC	NMI	Purity
MSRC	w.o.	0.995	0.989	0.995	HW	w.o.	0.998	0.995	0.998
	w.	**1.000**	**1.000**	**1.000**		w.	**1.000**	**1.000**	**1.000**
Mnist4	w.o.	0.929	0.839	0.929	Cal101	w.o.	0.785	0.776	0.873
	w.	**0.938**	**0.859**	**0.938**		w.	**0.789**	**0.806**	**0.877**
NUS	w.o.	0.257	0.186	0.259	Reuters	w.o.	0.762	0.589	0.762
	w.	**0.274**	**0.202**	**0.281**		w.	**0.795**	**0.660**	**0.795**

a little more time. However, at the same time, the proposed TBGL achieves superior clustering results comparing with the non-tensorized methods on three challenging datasets;

- Though a tensorized method, the proposed method can still tackle large-scale datasets within an acceptable time frame. In contrast, t-SVD-MSC and ETLMSC suffer from out-of-memory issue. This is because we reduced the computational complexity required by the tensorized methods from $\mathcal{O}(n^3 + Vn^2 + Vn^2 \log(n))$ to $\mathcal{O}(m^2nt + Vnmd)$, where V, n, and m are the number of views, samples, anchors, respectively. Due to $m \ll n$ in practice, the proposed TBGL is more effective and more efficient.



(a) w.o. $\ell_{1,2}$-norm (b) w. $\ell_{1,2}$-norm

Fig. 7. The visualizations of the learned consensus graph w./w.o. $\ell_{1,2}$-norm penalty on MSRC dataset.

5.4 Ablation Studies

To evaluate the influence of different components in the proposed method, we conduct the following ablation studies:

Effectiveness of VDA: To this end, as reported in Fig. 4, we first compare the visualization results of different anchor selection methods on four toy datasets, where each cluster is represented by some points (see **blue**, **green**, and **black** points); each method selects 80 anchor points (see **red** circles). Compared with the anchor selection results of the random sampling and K-Means, we can observe that the proposed VDA always chooses proper anchors in terms of balanced and unbalanced toy datasets, which clearly demonstrates that the anchors selected via VDA can evenly cover the entire point clouds. Though the K-means method also gets good performances in some scenarios, it needs to calculate Euclidean distances among the features of all sample pints and anchors. When the dimension of the input data is high, such anchor selection scheme will be very time-consuming.

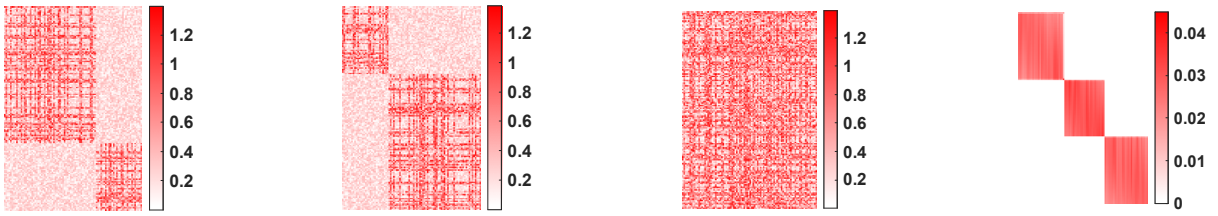
Moreover, we compare the clustering performance of the proposed method with different anchor selection scenarios on MSRC-v5, Handwritten4, Mnist4, and Caltech101-20 datasets. The clustering results are given in Fig. 5. One can observe that the clustering results under VDA scheme consistently outperform the results under two other anchor selection strategies. These results clearly verify that the anchor selection results offered by VDA to the multi-view clustering are more favorable and effective. This is because VDA takes the correlation of samples into account when choosing anchors, thus the selected anchors can fully cover the data points against all clusters.

Influence of Tensor Schatten p-norm: Taking MSRC-v5 and Handwritten4 datasets as examples, we analyze the impact of tensor Schatten p-norm for clustering. Specifically, we change p from 0.1 to 1.0 with the interval of 0.1, then we report the ACC, NMI and Purity. Note that, when p = 1.0, tensor Schatten p-norm degenerates into tensor nuclear norm. As shown in Fig. 6, we can observe that the results under different p are distinguishing mostly, and when p = 0.9 and p = 0.7, we obtain the best clustering results on MSRC-v5 and Handwritten4 dataset, respectively. This demonstrates that p has a significant influence on the clustering results. This is because that p exploits the significant difference among the singular values. Another reason is that the tensor Schatten p-norm approximate the target rank well in learning the view-consensus graph.

Influence of $\ell_{1,2}$-norm Regularization: In this section, we investigate the effectiveness of the introduced $\ell_{1,2}$-norm penalty. Table 5 lists the clustering results of the proposed method w./w.o. the $\ell_{1,2}$ regularization. From Table 5, it can

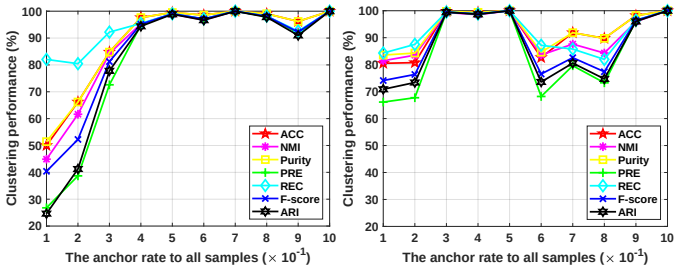
(a) ACC (b) NMI (c) Purity

Fig. 8. The clustering performances of TBGL w./w.o. self-weighted scheme on four real-world datasets.



(a) The 1-st input view (b) The 2-nd input view (c) The 3-rd input view (d) The learned graph

Fig. 9. The visualization of the learned bipartite graph via proposed method on synthetic dataset.



(a) MSRC-v5 (b) Handwritten4

Fig. 10. The performances of TBGL with varying the number of anchor points on MSRC-v5 and Handwritten4 datasets.

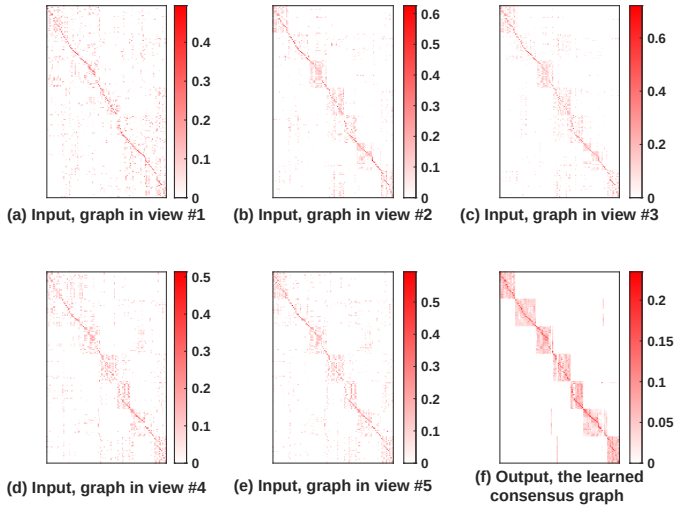


Fig. 11. The graphs visualizations on MSRC-v5 dataset.

be seen that the clustering performance is boosted via the $\ell_{1,2}$ regularization, especially on large datasets. Moreover, Fig. 7 shows the learned view consensus graph w./w.o. the $\ell_{1,2}$ regularization. From Fig. 7, it can be seen that the learned view consensus graph is sparser with the help of the $\ell_{1,2}$-norm constraint. These results indicate that the $\ell_{1,2}$ regularization is helpful to better characterize the similarity structure of intra-view, *i.e.*, well encoding the cluster structure and discriminative information.

5.5 Model Analysis

Effect of The Self-weighted Scheme: For the multi-view scenario, each view usually has significant different contribution for clustering. In this section, we first conduct controlled experiment to evaluate the influence of the proposed self-weighted scheme. As reported in Fig. 8 w.r.t. w./w.o. the self-weighted scheme, the proposed method equipped with self-weighted scheme always obtain the best clustering results on four datasets in terms of all three evaluation metrics. This is because the proposed method equipped with self-weighted scheme can well take into consideration the differences of the multiple views. This benefits from the adaptive weight assignment. The proposed method in absence of the self-weighted scheme generally fails, which results in the quick drop of the performance of the view-consensus graph learning.

Effect of The Number of Anchors: We empirically analyze the effect of the number of anchors for clustering on MSRC-v5 and Handwritten4 datasets. To this end, we turn the proportion that anchors take in the entire data points from 0.1 to 1.0 with the interval of 0.1, then we show seven metrics (ACC, NMI, Purity, PRE, REC, F-score and ARI) in Fig. 10. It is obviously observed that TBGL has a large fluctuation when varying the number of anchors. When the proportion is set to 0.5, TBGL obtains the best performance on MSRC-v5 dataset and Handwritten4 dataset. Moreover, we find that the metrics curves w.r.t. anchors proportion are not monotonously increasing. This indicates that it is not necessary to use numerous anchors for clustering.

Graph Visualization: We present the input graphs and the learned view-consensus graph of our method on MSRC-v5 dataset in Fig. 11, where (a) - (e) are input graphs corresponding to five views, (f) is the view-consensus graph. It can be seen that the connected components in the input graphs of all five views are not clear. By employing our proposed method, we can observe that the learned view-consensus graph has exact 7 connected components. It indicates that our method well characterizes the cluster structure. The above experimental results once again demonstrate that our proposed tensor Schatten p-norm regularization helps to

ensure the rank of the learned view-consensus graph more closer to the target rank.

Moreover, taking the synthetic data as an example, we visualize the view-consensus graph learned by the proposed method. Fig. 9 (a-c) present the input synthetic data of three views, respectively, where each input view adds uniform random noise. As reported in Fig. 9 (d), by leveraging the proposed tensorized bipartite graph learning method, the learned view-consensus graph has exactly 3 connected components. This result indicates that TBGL can well explore the cluster structure embedded in the noisy multi-view data via the self-weighted scheme.

6 Conclusion

In this article, we propose a *tensorized bipartite graph learning method for MVC* (TBGL). TBGL exploits similarity of the inter-view via tensor Schatten *p*-norm minimization, which well explores the complementary information embedded in the different views. Meanwhile, we combine the $\ell_{1,2}$-norm minimization regularization and connectivity constraint to explore the similarity of inter-view, thus, the learned graph not only well encodes discriminative information but also has the exact connected components. We solve our objective by an efficient algorithm. Extensive experiments on real-world datasets indicate the effectiveness of TBGL.

Appendix A
Proof of Theorem 2

Proof: According to Eq. (14), we have

$$\begin{aligned} tr(\mathbf{P}_N^{\mathrm{T}}\mathbf{W}\mathbf{P}_M) &= \frac{1}{2}(tr(\mathbf{P}_N^{\mathrm{T}}\mathbf{W}\mathbf{P}_M + tr(\mathbf{P}_M^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{P}_N)) \\ &= \frac{1}{2}tr\left(\begin{bmatrix}\mathbf{P}_N\\ \mathbf{P}_M\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{0} & \mathbf{W}\\ \mathbf{W}^{\mathrm{T}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{P}_N\\ \mathbf{P}_M\end{bmatrix}\right) \end{aligned} \tag{33}$$

Then Eq. (14) is equivalent to

$$\begin{aligned} &\arg\max_{\mathbf{P}_N,\mathbf{P}_M} \frac{1}{2}tr\left\{\begin{bmatrix}\mathbf{P}_N\\ \mathbf{P}_M\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{0} & \mathbf{W}\\ \mathbf{W}^{\mathrm{T}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{P}_N\\ \mathbf{P}_M\end{bmatrix}\right\} \\ &s.t.\ [\mathbf{P}_N\ \mathbf{P}_M]^{\mathrm{T}}[\mathbf{P}_N\ \mathbf{P}_M] = \mathbf{I} \end{aligned} \tag{34}$$

The optimal solution of Eq. (34) is

$$\frac{1}{2}\begin{bmatrix}\mathbf{0} & \mathbf{W}\\ \mathbf{W}^{\mathrm{T}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{P}_N\\ \mathbf{P}_M\end{bmatrix} = \begin{bmatrix}\mathbf{P}_N\\ \mathbf{P}_M\end{bmatrix}\mathbf{\Lambda} \tag{35}$$

where $\mathbf{\Lambda}$ is a diagonal matrix whose elements are composed of the eigenvalues of $\frac{1}{2}\begin{bmatrix}\mathbf{0} & \mathbf{W}\\ \mathbf{W}^{\mathrm{T}} & \mathbf{0}\end{bmatrix}$.

By simple matrix algebra, we have

$$\begin{cases} \frac{1}{2}\mathbf{W}\mathbf{P}_M = \mathbf{P}_N\mathbf{\Lambda} \\ \frac{1}{2}\mathbf{W}^{\mathrm{T}}\mathbf{P}_N = \mathbf{P}_M\mathbf{\Lambda} \end{cases} \tag{36}$$

Then it is ready to see

$$\begin{cases} (\sqrt{2}\Big/2\mathbf{W})^{\mathrm{T}}(\sqrt{2}\Big/2\mathbf{W})\mathbf{P}_M = \mathbf{P}_M(\sqrt{2}\mathbf{\Lambda})^2 \\ (\sqrt{2}\Big/2\mathbf{W})(\sqrt{2}\Big/2\mathbf{W})^{\mathrm{T}}\mathbf{P}_N = \mathbf{P}_N(\sqrt{2}\mathbf{\Lambda})^2 \end{cases} \tag{37}$$

According to Eq. (37), $\mathbf{P}_N$ and $\mathbf{P}_M$ are composed of the leading *K* left and right singular vectors of $\sqrt{2}\Big/2\mathbf{W}$. Denote by $\mathbf{U}_1$ and $\mathbf{V}_1$ are the leading *K* left and right singular vectors of $\mathbf{W}$, respectively, We have $\mathbf{P}_M = \frac{\sqrt{2}}{2}\mathbf{V}_1$, $\mathbf{P}_N = \frac{\sqrt{2}}{2}\mathbf{U}_1$. □

Appendix B
Proof of Convergence Analysis

Lemma 2 (Proposition 6.2 of [56]). Suppose $F : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}$ is represented as $F(X) = f \circ \sigma(X)$, where $X \in \mathbb{R}^{n_1\times n_2}$ with SVD $X = U\mathrm{diag}(\sigma_1, \ldots, \sigma_n)V^{\mathrm{T}}$, $n = \min(n_1, n_2)$, and f is differentiable. The gradient of $F(X)$ at X is

$$\frac{\partial F(X)}{\partial X} = U\mathrm{diag}(\theta)V^{\mathrm{T}}, \tag{38}$$

where $\theta = \frac{\partial f(y)}{\partial y}|_{y=\sigma(X)}$.

Theorem 4. [Convergence Analysis of **Algorithm 1**] Let $P_k = \{\mathbf{E}_k^{(v)}, \mathbf{O}_k^{(v)}, \mathbf{B}_k^{(v)}, \mathbf{C}_k^{(v)}, \mathbf{Y}_{1,k}^{(v)}, \mathbf{Y}_{3,k}^{(v)}, \mathcal{J}_k, \mathcal{Y}_{2,k}\},\ 1 \le k < \infty$ be a sequence generated by **Algorithm 1**, then

1) P_k is bounded;
2) Any accumulation point of P_k is a stationary KKT point.

B.1 Proof of the 1st part

Proof: To minimize $\mathbf{E}^{(v)}$ at step $k+1$, the optimal $\mathbf{E}_{k+1}^{(v)}$ should satisfy the first-order optimal condition of (19):

$$\frac{\alpha}{\mu_k}\partial\|\mathbf{E}_{k+1}^{(v)}\|_1 + \mathbf{E}_{k+1}^{(v)} - (\mathbf{B}_k^{(v)} - \mathbf{C}_k^{(v)} + \frac{1}{\mu_k}\mathbf{Y}_{1,k}^{(v)}) = 0.$$

By using the sub-gradient

$$\partial\|\mathbf{E}_{:,i}^{(v)}\|_1 = \begin{cases} \frac{\mathbf{E}_{:,i}^{(v)}}{\|\mathbf{E}_{:,i}^{(v)}\|_1}, & \mathbf{E}_{:,i}^{(v)} \neq 0 \\ \{\mathbf{E}_{:,i}^{(v)} \mid\mid \|\mathbf{E}_{:,i}^{(v)}\|_1 \le 1\}, & \mathbf{E}_{:,i}^{(v)} = 0 \end{cases},$$

and from the update rule

$$\begin{aligned} &\mathbf{Y}_{1,k+1}^{(v)} := \mathbf{Y}_{1,k}^{(v)} + \mu_k(\mathbf{B}_k^{(v)} - \mathbf{C}_k^{(v)} - \mathbf{E}_k^{(v)}) \\ &\Longrightarrow \alpha\partial\|\mathbf{E}_{k+1}^{(v)}\|_1 - \mathbf{Y}_{1,k+1}^{(v)} = 0, \end{aligned}$$

So, $\|\mathbf{Y}_{1,k+1}^{(v)}\|_1 \le N$ and $\mathbf{Y}_{1,k+1}^{(v)}$ is bounded.

To minimize $\mathbf{O}^{(v)}$ at step $k+1$, the optimal $\mathbf{O}_{k+1}^{(v)}$ should satisfy the first-order optimal condition of (20):

$$\frac{\gamma}{\tau_k}\partial\|\mathbf{O}_{k+1}^{(v)}\|_{1,2}^2 + (\mathbf{O}_{k+1}^{(v)} - \mathbf{C}_k^{(v)} - \frac{1}{\tau_k}\mathbf{Y}_{3,k}^{(v)}) = 0.$$

By using the sub-gradient

$$\begin{aligned} \partial\left\|\mathbf{O}_{k+1}^{(v)}(i,:)\right\|_1^2 = \{2\left\|\mathbf{O}_{k+1}^{(v)}(i,:)\right\|_1 h \mid \\ \|h\|_\infty \le 1, h\mathbf{O}_{i,:}^{(v)^T} = \left\|\mathbf{O}_{i,:}^{(v)}\right\|_1\}. \end{aligned} \tag{39}$$

and from the update rule

$$\begin{aligned} &\mathbf{Y}_{3,k+1}^{(v)} := \mathbf{Y}_{3,k}^{(v)} + \tau_k(\mathbf{C}_k^{(v)} - \mathbf{E}_k^{(v)}) \\ &\Longrightarrow \gamma\partial\|\mathbf{O}_{k+1}^{(v)}\|_{2,1} - \mathbf{Y}_{3,k+1}^{(v)} = 0, \end{aligned}$$

So, $\|\mathbf{Y}_{3,k+1}^{(v)}\|_\infty \le N$ and $\mathbf{Y}_{3,k+1}^{(v)}$ is bounded.

To minimize $\mathcal{J}$ at step $k+1$ in (21), the optimal $\mathcal{J}_{k+1}$ needs to satisfy

$$\nabla_{\mathcal{J}}\|\mathcal{J}_{k+1}\|_{\circledS p}^p + \rho_k(\mathcal{J}_{k+1} - \mathcal{C}_{k+1} - \frac{1}{\rho_k}\mathcal{Y}_{2,k}) = 0.$$

Recall that when $0 < p < 1$, in order to overcome the singularity of $(|\eta|^p)' = p\eta/|\eta|^{2-p}$ near $\eta = 0$, we consider for $0 < \epsilon \ll 1$ the approximation

$$\partial|\eta|^p \approx \frac{p\eta}{\max\{\epsilon^{2-p}, |\eta|^{2-p}\}}.$$

Letting $\overline{\mathcal{T}}^{(i)} = \overline{\mathcal{U}}^{(i)}\mathrm{diag}\left(\sigma_j(\overline{\mathcal{T}}^{(i)})\right)\overline{\mathcal{V}}^{(i)\mathrm{H}}$, then it follows from Lemma 2 that

$$\frac{\partial\|\overline{\mathcal{T}}^{(i)}\|_{\circledS p}^p}{\partial\overline{\mathcal{T}}^{(i)}} = \overline{\mathcal{U}}^{(i)}\mathrm{diag}\left(\frac{p\sigma_j(\overline{\mathcal{T}}^{(i)})}{\max\{\epsilon^{2-p}, |\sigma_j(\overline{\mathcal{T}}^{(i)})|^{2-p}\}}\right)\overline{\mathcal{V}}^{(i)\mathrm{H}}.$$

And then one can obtain

$$\frac{p\sigma_j(\overline{\mathcal{T}}^{(i)})}{\max\{\epsilon^{2-p}, |\sigma_j(\overline{\mathcal{T}}^{(i)})|^{2-p}\}} \le \frac{p}{\epsilon^{1-p}}$$
$$\Longrightarrow \left\|\frac{\partial\|\overline{\mathcal{T}}^{(i)}\|_{\circledS p}^p}{\partial\overline{\mathcal{T}}^{(i)}}\right\|_F^2 \le \sum_{i=1}^{N}\frac{p^2}{\epsilon^{2(1-p)}}.$$

So $\frac{\partial\|\overline{\mathcal{T}}\|_{\circledS p}^p}{\partial\overline{\mathcal{T}}}$ is bounded.

Let us denote $\widetilde{\mathbf{F}}_V = \frac{1}{\sqrt{V}}\mathbf{F}_V$, $\mathbf{F}_V$ is the discrete Fourier transform matrix of size $V\times V$, $\mathbf{F}_V^{\mathrm{H}}$ denotes its conjugate transpose. For $\mathcal{T} = \overline{\mathcal{T}}\times_3\widetilde{\mathbf{F}}_V$ and using the chain rule in matrix calculus, one can obtain that

$$\nabla_{\mathcal{T}}\|\mathcal{T}\|_{\circledS p}^p = \frac{\partial\|\mathcal{T}\|_{\circledS p}^p}{\partial\overline{\mathcal{T}}}\times_3\widetilde{\mathbf{F}}_V^{\mathrm{H}}$$

is bounded.

And it follows that

$$\mathcal{Y}_{2,k+1} = \mathcal{Y}_{2,k} + \rho_k(\mathcal{C}_{k+1} - \mathcal{T}_{k+1})$$
$$\Longrightarrow \nabla_{\mathcal{T}}\|\mathcal{T}_{k+1}\|_{\circledS p}^p = \mathcal{Y}_{2,k+1},$$

$\mathcal{Y}_{2,k+1}$ appears to be bounded.

Moreover, by using the updating rule

$$\mathbf{Y}_{1,k+1}^{(v)} := \mathbf{Y}_{1,k}^{(v)} + \mu_k(\mathbf{B}_k^{(v)} - \mathbf{C}_k^{(v)} - \mathbf{E}_k^{(v)}),$$
$$\mathbf{Y}_{3,k+1}^{(v)} := \mathbf{Y}_{3,k}^{(v)} + \tau_k(\mathbf{C}_k^{(v)} - \mathbf{O}_k^{(v)}),$$
$$\mathcal{Y}_{2,k} = \mathcal{Y}_{2,k-1} + \rho_{k-1}(\mathcal{C}_k - \mathcal{T}_k),$$

we can deduce

$$\begin{aligned}
&\mathcal{L}(\mathbf{E}_k^{(v)},\mathbf{O}_k^{(v)},\mathbf{B}_k^{(v)},\mathbf{C}_k^{(v)},\mathbf{Y}_{1,k}^{(v)},\mathbf{Y}_{3,k}^{(v)},\mathcal{T}_k,\mathcal{Y}_{2,k};\rho_k;\mu_k;\tau_k) \qquad (40)\\
&\le \mathcal{L}(\mathbf{E}_k^{(v)},\mathbf{O}_k^{(v)},\mathbf{B}_k^{(v)},\mathbf{C}_k^{(v)},\mathbf{Y}_{1,k-1}^{(v)},\mathbf{Y}_{3,k-1}^{(v)},\mathcal{T}_k,\mathcal{Y}_{2,k-1};\rho_{k-1};\mu_{k-1};\tau_{k-1})\\
&+\frac{\mu_k+\mu_{k-1}}{2\mu_{k-1}^2}\|\mathcal{Y}_{1,k}-\mathcal{Y}_{1,k-1}\|_F^2 + \frac{\|\mathcal{Y}_{1,k}\|_F^2}{2\mu_k} - \frac{\|\mathcal{Y}_{1,k-1}\|_F^2}{2\mu_{k-1}}\\
&+\frac{\rho_k+\rho_{k-1}}{2\rho_{k-1}^2}\|\mathcal{Y}_{2,k}-\mathcal{Y}_{2,k-1}\|_F^2 + \frac{\|\mathcal{Y}_{2,k}\|_F^2}{2\rho_k} - \frac{\|\mathcal{Y}_{2,k-1}\|_F^2}{2\rho_{k-1}}.\\
&+\frac{\tau_k+\tau_{k-1}}{2\tau_{k-1}^2}\|\mathcal{Y}_{3,k}-\mathcal{Y}_{3,k-1}\|_F^2 + \frac{\|\mathcal{Y}_{3,k}\|_F^2}{2\tau_k} - \frac{\|\mathcal{Y}_{3,k-1}\|_F^2}{2\tau_{k-1}}.
\end{aligned}$$

Thus, summing two sides of (40) from $k=1$ to n, we have

$$\begin{aligned}
&\mathcal{L}(\mathbf{E}_n^{(v)},\mathbf{O}_n^{(v)},\mathbf{B}_n^{(v)},\mathbf{C}_n^{(v)},\mathbf{Y}_{1,n}^{(v)},\mathbf{Y}_{3,n}^{(v)},\mathcal{T}_n,\mathcal{Y}_{2,n};\rho_n;\mu_n;\tau_n)\\
\le&\mathcal{L}(\mathbf{E}_0^{(v)},\mathbf{O}_0^{(v)},\mathbf{B}_0^{(v)},\mathbf{C}_0^{(v)},\mathbf{Y}_{1,0}^{(v)},\mathbf{Y}_{3,0}^{(v)},\mathcal{T}_0,\mathcal{Y}_{2,0};\rho_0;\mu_0;\tau_0)\\
&+\frac{\|\mathcal{Y}_{1,n}\|_F^2}{2\mu_n} - \frac{\|\mathcal{Y}_{1,0}\|_F^2}{2\mu_0} + \sum_{k=1}^{n}(\frac{\mu_k+\mu_{k-1}}{2\mu_{k-1}^2}\|\mathcal{Y}_{1,k}-\mathcal{Y}_{1,k-1}\|_F^2)\\
&+\frac{\|\mathcal{Y}_{2,n}\|_F^2}{2\rho_n} - \frac{\|\mathcal{Y}_{2,0}\|_F^2}{2\rho_0} + \sum_{k=1}^{n}(\frac{\rho_k+\rho_{k-1}}{2\rho_{k-1}^2}\|\mathcal{Y}_{2,k}-\mathcal{Y}_{2,k-1}\|_F^2)\\
&+\frac{\|\mathcal{Y}_{3,n}\|_F^2}{2\tau_n} - \frac{\|\mathcal{Y}_{3,0}\|_F^2}{2\tau_0} + \sum_{k=1}^{n}(\frac{\tau_k+\tau_{k-1}}{2\tau_{k-1}^2}\|\mathcal{Y}_{3,k}-\mathcal{Y}_{3,k-1}\|_F^2). \qquad (41)
\end{aligned}$$

Observe that

$$\sum_{k=1}^{\infty}\frac{\mu_k+\mu_{k-1}}{2\mu_{k-1}^2} < \infty,\ \sum_{k=1}^{\infty}\frac{\rho_k+\rho_{k-1}}{2\rho_{k-1}^2} < \infty,\ \sum_{k=1}^{\infty}\frac{\tau_k+\tau_{k-1}}{2\tau_{k-1}^2} < \infty,$$

we have the right-hand side of (41) is finite and thus $\mathcal{L}\left(\mathbf{E}_n^{(v)},\mathbf{O}_n^{(v)},\mathbf{B}_n^{(v)},\mathbf{C}_n^{(v)},\mathbf{Y}_{1,n}^{(v)},\mathbf{Y}_{3,n}^{(v)},\mathcal{T}_n,\mathcal{Y}_{2,n};\rho_n;\mu_n;\tau_n\right)$ is bounded. Notice

$$\begin{aligned}
&\mathcal{L}\left(\mathbf{E}_n^{(v)},\mathbf{O}_n^{(v)},\mathbf{B}_n^{(v)},\mathbf{C}_n^{(v)},\mathbf{Y}_{1,n}^{(v)},\mathbf{Y}_{3,n}^{(v)},\mathcal{T}_n,\mathcal{Y}_{2,n};\rho_n;\mu_n;\tau_n\right)\\
&=\sum_{v=1}^{V}\beta\mathrm{tr}(\mathbf{P}^{\mathrm{T}}\mathbf{L}_{\mathbf{C}_{n+1}}\mathbf{P}) + \sum_{v=1}^{V}\frac{\mu_n}{2}\|\mathbf{B}_n^{(v)} - \mathbf{C}_n^{(v)} - \mathbf{E}_n^{(v)} + \frac{\mathbf{Y}_{1,n}^{(v)}}{\mu_n}\|_F^2\\
&+\sum_{v=1}^{V}\frac{\tau_n}{2}\|\mathbf{C}_n^{(v)} - \mathbf{O}_n^{(v)} + \frac{\mathbf{Y}_{3,n}^{(v)}}{\tau_n}\|_F^2\\
&+\|\mathcal{T}_{n+1}\|_{\circledS p}^p + \frac{\rho_n}{2}\|\mathcal{C}_{n+1} - \mathcal{T}_{n+1} + \frac{\mathcal{Y}_n}{\rho_n}\|_F^2\\
&+\sum_{v=1}^{V}\alpha\|\mathbf{E}_n^{(v)}\|_1 + \sum_{v=1}^{V}\gamma\|\mathbf{O}_n^{(v)}\|_{1,2}^2, \qquad (42)
\end{aligned}$$

and each term of (42) is nonnegative. Following from the boundedness of $\mathcal{L}(\mathbf{E}_n^{(v)},\mathbf{O}_n^{(v)},\mathbf{B}_n^{(v)},\mathbf{C}_n^{(v)},\mathbf{Y}_{1,n}^{(v)},\mathbf{Y}_{3,n}^{(v)},\mathcal{T}_n,\mathcal{Y}_{2,n};\rho_n;\tau_n;\mu_n)$ we can deduce each term of (42) is bounded. And $\|\mathcal{T}_{n+1}\|_{\circledS p}^p$ being bounded implies that all singular values of $\mathcal{T}_{n+1}$ are bounded and hence $\|\mathcal{T}_{n+1}\|_F^2$ (the sum of squares of singular values) is bounded. Similarly, the sequence $\{\mathbf{E}_n^{(v)},\mathbf{O}_n^{(v)}\}$ are also bounded since all norms of $\mathbb{R}^n$ are equivalent. Considering the updating rule $(\mathcal{Y}_{2,k}-\mathcal{Y}_{2,k-1})/\rho_{k-1} = \mathcal{C}_k-\mathcal{T}_k$, it is ready to see $\mathcal{C}_k$ is bounded. □

B.2 Proof of the 2nd part

Proof: From Weierstrass-Bolzano theorem, there exists at least one accumulation point of the sequence P_k. We denote one of the points $P^* = \{\mathbf{E}^{(v)*},\mathbf{O}^{(v)*},\mathbf{B}^{(v)*},\mathbf{C}^{(v)*},\mathbf{Y}_1^{(v)*},\mathbf{Y}_3^{(v)*},\mathcal{T}^*,\mathcal{Y}_2^*\}$. Without loss of generality, we assume $\{\mathcal{P}_k\}_{k=1}^{+\infty}$ converge to P^*.

Note that from the updating rule for $\mathcal{Y}_2$, we have

$$\mathcal{Y}_{2,k+1} = \mathcal{Y}_{2,k} + \mu_k(\mathcal{C}_k - \mathcal{T}_k) \Longrightarrow \mathcal{T}^* = \mathcal{C}^*.$$

Similarly, by the updating rule of $\mathbf{Y}_1^{(v)}$, we have

$$\mathbf{Y}_{1,k+1}^{(v)} = \mathbf{Y}_{1,k}^{(v)} + \mu_k(\mathbf{B}_k^{(v)} - \mathbf{C}_k^{(v)} - \mathbf{E}_k^{(v)}) \Longrightarrow \mathbf{B}^{(v)*} - \mathbf{C}^{(v)*} = \mathbf{E}^{(v)*};$$

by the updating rule of $\mathbf{Y}_3^{(v)}$, we have

$$\mathbf{Y}_{3,k+1}^{(v)} = \mathbf{Y}_{3,k}^{(v)} + \tau_k(\mathbf{C}_k^{(v)} - \mathbf{O}_k^{(v)}) \Longrightarrow \mathbf{O}^{(v)*} - \mathbf{C}^{(v)*} = 0$$

In the $\mathcal{T}$-subproblem, we have

$$\nabla_{\mathcal{T}}\|\mathcal{T}_{k+1}\|_{\circledS p}^p = \mathcal{Y}_{2,k+1} \Longrightarrow \mathcal{Y}_2^* = \nabla_{\mathcal{T}}\|\mathcal{T}^*\|_{\circledS p}^p.$$

Similarly, in the $\mathbf{E}^{(v)}$-subproblem, we have

$$\mathbf{Y}_1^{(v)*} \in \alpha\partial\|\mathbf{E}^{(v)*}\|_1;$$

in the $\mathbf{O}^{(v)}$-subproblem, we have

$$\mathbf{Y}_3^{(v)*} \in \gamma\partial\|\mathbf{O}^{(v)*}\|_{1,2}.$$

Therefore, one can see that the sequences $\mathbf{E}^{(v)*},\mathbf{O}^{(v)*}$, $\mathbf{B}^{(v)*},\mathbf{C}^{(v)*},\mathbf{Y}_1^{(v)*},\mathbf{Y}_3^{(v)*},\mathcal{T}^*,\mathcal{Y}_2^*$ satisfy the KKT conditions. □

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers and AE for their constructive comments and suggestions. Also deserving recognition Dr. Han Zhang and Prof. Feiping Nie for providing the codes of SFMC and relevant datasets.

REFERENCES

[1] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, 2019.

[2] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "COMIC: multi-view clustering without parameter selection," in *ICML*, vol. 97, 2019, pp. 5092–5101.

[3] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 487–501, 2019.

[4] Q. Gao, W. Xia, Z. Wan, D. Xie, and P. Zhang, "Tensor-svd based graph learning for multi-view subspace clustering," in *AAAI*, 2020, pp. 3930–3937.

[5] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, 2020.

[6] C. Zhang, H. Fu, J. Wang, W. Li, X. Cao, and Q. Hu, "Tensorized multi-view subspace representation learning," *Int. J. Comput. Vis.*, vol. 128, no. 8, pp. 2344–2361, 2020.

[7] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, and E. Zhu, "Efficient and effective regularized incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2634–2646, 2021.

[8] A. Kumar, P. Rai, and H. D. III, "Co-regularized multi-view spectral clustering," in *NeurIPS*, 2011, pp. 1413–1421.

[9] H. Xu, X. Zhang, W. Xia, Q. Gao, and X. Gao, "Low-rank tensor constrained co-regularized multi-view spectral clustering," *Neural Networks*, vol. 132, pp. 245–252, 2020.

[10] F. Nie, J. Li, and X. Li, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *IJCAI*, 2016, pp. 1881–1887.

[11] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *AAAI*, 2015, pp. 2750–2756.

[12] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *AAAI*, 2020, pp. 4412–4419.

[13] X. Li, H. Zhang, R. Wang, and F. Nie, "Multiview clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 330–344, 2022.

[14] Q. Gao, P. Zhang, W. Xia, D. Xie, X. Gao, and D. Tao, "Enhanced tensor RPCA and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2133–2140, 2021.

[15] D. Ming and C. Ding, "Robust flexible feature selection via exclusive L21 regularization," in *IJCAI*, S. Kraus, Ed., 2019, pp. 3158–3164.

[16] D. Ming, C. Ding, and F. Nie, "A probabilistic derivation of LASSO and l12-norm feature selections," in *AAAI*, 2019, pp. 4586–4593.

[17] S. Deng, W. Xia, Q. Gao, and X. Gao, "Cross-view classification by joint adversarial learning and class-specificity distribution," *Pattern Recognit.*, vol. 110, p. 107633, 2021.

[18] W. Liu, J. He, and S. Chang, "Large graph construction for scalable semi-supervised learning," in *ICML*, 2010, pp. 679–686.

[19] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *ACM SIGKDD*, 2001, pp. 269–274.

[20] S. Huang, Z. Xu, I. W. Tsang, and Z. Kang, "Auto-weighted multi-view co-clustering with bipartite graphs," *Inf. Sci.*, vol. 512, pp. 18–30, 2020.

[21] W. Xia, X. Zhang, Q. Gao, X. Shu, J. Han, and X. Gao, "Multi-view subspace clustering by an enhanced tensor nuclear norm," *IEEE Trans. Cybern.*, vol. doi: 10.1109/TCYB.2021.3052352, 2021.

[22] Q. Gao, Z. Wan, Y. Liang, Q. Wang, Y. Liu, and L. Shao, "Multi-view projected clustering with graph learning," *Neural Networks*, vol. 126, pp. 335–346, 2020.

[23] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2887–2895, 2018.

[24] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *IJCAI*, 2017, pp. 2564–2570.

[25] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5910–5922, 2019.

[26] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, and Y. Qu, "On unifying multi-view self-representations for clustering by tensor multi-rank minimization," *Int. J. Comput. Vis.*, vol. 126, no. 11, pp. 1157–1179, 2018.

[27] Y. Xie, W. Zhang, Y. Qu, L. Dai, and D. Tao, "Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 572–586, 2020.

[28] B. Chen, B. Gao, T. Liu, Y. Chen, and W. Ma, "Fast spectral clustering of data using sequential matrix compression," in *ECML*, vol. 4212, 2006, pp. 590–597.

[29] T. Liu, H. Yang, X. Zheng, T. Qin, and W. Ma, "Fast large-scale spectral clustering by sequential shrinkage optimization," in *ECIR*, vol. 4425, 2007, pp. 319–330.

[30] C. C. Fowlkes, S. J. Belongie, F. R. K. Chung, and J. Malik, "Spectral grouping using the nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, 2004.

[31] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *NeurIPS*, 2007, pp. 1177–1184.

[32] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning A structured optimal bipartite graph for co-clustering," in *NeurIPS*, 2017, pp. 4129–4138.

[33] L. Wu, P. Chen, I. E. Yen, F. Xu, Y. Xia, and C. C. Aggarwal, "Scalable spectral clustering using random binning features," in *ACM SIGKDD*, 2018, pp. 2506–2515.

[34] P. Zhou, L. Du, and X. Li, "Self-paced consensus clustering with bipartite graph," in *IJCAI*, 2020, pp. 2133–2139.

[35] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, 2015.

[36] F. Nie, C. Wang, and X. Li, "K-multiple-means: A multiple-means clustering method with specified K clusters," in *ACM SIGKDD*, 2019, pp. 959–967.

[37] M. Sun, P. Zhang, S. Wang, S. Zhou, W. Tu, X. Liu, E. Zhu, and C. Wang, "Scalable multi-view subspace clustering with unified anchors," in *ACM MM*, 2021, pp. 3528–3536.

[38] S. Wang, X. Liu, X. Zhu, P. Zhang, Y. Zhang, F. Gao, and E. Zhu, "Fast parameter-free multi-view subspace clustering with consensus anchor guidance," *IEEE Trans. Image Process.*, vol. 31, pp. 556–568, 2022.

[39] K. Fan, "On a theorem of weyl concerning eigenvalues of linear transformations i," *Proc. Natl. Acad. Sci. USA*, vol. 35, no. 11, pp. 652–655, 1949.

[40] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[41] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *AAAI*, 2016, pp. 1969–1976.

[42] E. T. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for l_1-minimization: Methodology and convergence," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1107–1130, 2008.

[43] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, 2015.

[44] M. Wang, W. Fu, S. Hao, D. Tao, and X. Wu, "Scalable semi-supervised learning by efficient anchor graph regularization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1864–1877, 2016.

[45] I. T. Jolliffe, *Principal Component Analysis*, ser. Springer Series in Statistics. Springer, 1986.

[46] K. Fukunaga and P. M. Narendra, "A branch and bound algorithms for computing k-nearest neighbors," *IEEE Trans. Computers*, vol. 24, no. 7, pp. 750–753, 1975.

[47] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *AAAI*, 2014, pp. 2149–2155.

[48] S. Luo, C. Zhang, W. Zhang, and X. Cao, "Consistent and specific multi-view subspace clustering," in *AAAI*, 2018, pp. 3730–3737.

[49] J. M. Winn and N. Jojic, "LOCUS: learning object classes with unsupervised segmentation," in *IEEE ICCV*, 2005, pp. 756–763.

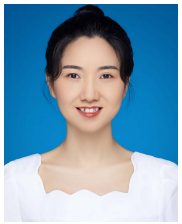
[50] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[51] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, 2012.

[52] F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Comput. Vis. Image Underst.*, vol. 106, no. 1, pp. 59–70, 2007.
[53] T. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: a real-world web image database from national university of singapore," in *ACM CIVR*, 2009.
[54] C. Apté, F. Damerau, and S. M. Weiss, "Automated learning of decision rules for text categorization," *ACM Trans. Inf. Syst.*, vol. 12, no. 3, pp. 233–251, 1994.
[55] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "On deep multi-view representation learning," in *ICML*, vol. 37, 2015, pp. 1083–1092.
[56] A. S. Lewis and H. S. Sendov, "Nonsmooth analysis of singular values. part i: Theory," *Set-Valued Analysis*, vol. 13, no. 3, pp. 213–241, 2005.

Wei Xia received the B.Eng. degree in Communication Engineering from Lanzhou University of Technology, Lanzhou, China, in 2018. He is currently pursuing the Ph.D. degree in communication and information system in Xidian University, Xi'an, China. His research interests include multimodal learning, representation learning, unsupervised and self-supervised learning.

Quanxue Gao received the B. Eng. degree from Xi'an Highway University, Xi'an, China, in 1998, the M.S. degree from the Gansu University of Technology, Lanzhou, China, in 2001, and the Ph.D. degree from Northwestern Polytechnical University, Xi'an China, in 2005. He was an associate research with the Biometrics Center, The Hong Kong Polytechnic University, Hong Kong from 2006 to 2007. From 2015 to 2016, he was a visiting scholar with the department of computer science, The University of Texas at Arlington, Arlington USA. He is currently a professor with the School of Telecommunications Engineering, Xidian University, and also a key member of State Key Laboratory of Integrated Services Networks. He has authored around 80 technical articles in refereed journals and proceedings, including IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cybernetics, CVPR, AAAI, and IJCAI. His current research interests include pattern recognition and machine learning.



Qianqian Wang received the B.Eng. degree in communication engineering from Lanzhou University of Technology, Lanzhou, China, in 2014, the Ph.D. degree from Xidian University, Xi'an, China, in 2019.

She was a Visiting Scholar with Northeastern University, Boston, MA, USA, from 2017 to 2018. She is currently a Lecturer with the School of Telecommunications Engineering, Xidian University. Her research interests include pattern recognition, principal component analysis, multi-view clustering, and partial multi-view clustering.

Xinbo Gao received the B.Eng., M.Sc. and Ph.D. degrees in electronic engineering, signal and information processing from Xidian University, Xi'an, China, in 1994, 1997, and 1999, respectively. From 1997 to 1998, he was a research fellow at the Department of Computer Science, Shizuoka University, Shizuoka, Japan. From 2000 to 2001, he was a post-doctoral research fellow at the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong. Since 2001, he has been at the School of Electronic Engineering, Xidian University. He is currently a Cheung Kong Professor of Ministry of Education of P. R. China, a Professor of Pattern Recognition and Intelligent System of Xidian University and a Professor of Computer Science and Technology of Chongqing University of Posts and Telecommunications. His current research interests include Image processing, computer vision, multimedia analysis, machine learning and pattern recognition. He has published six books and around 300 technical articles in refereed journals and proceedings. Prof. Gao is on the Editorial Boards of several journals, including Signal Processing (Elsevier) and Neurocomputing (Elsevier). He served as the General Chair/Co-Chair, Program Committee Chair/Co-Chair, or PC Member for around 30 major international conferences. He is a Fellow of the Institute of Engineering and Technology and a Fellow of the Chinese Institute of Electronics.

Chris Ding received his Ph.D. degree in Columbia University in 1987. He is currently a chair professor at Chinese University of Hong Kong, Shenzhen. Before that, he worked at California Institute of Technology, Jet Propulsion Lab, Lawrence Berkeley Lab, and University of Texas. His areas are machine learning, data mining, computational biology and high performance computing. He proposed L21 norm which is widely used in machine learning. Made significant contributions on principal component analysis, non-negative matrix factorization and feature selection. Designed the minimum redundancy maximum relevance feature selection algorithm that is widely adopted, e.g., by Uber; Two related papers were cited 11,700 times. Published over 200 research papers with over 50,000 citations.

Dacheng Tao is professor of computer science and ARC laureate fellow with the School of Computer Science and the Faculty of Engineering and Information Technologies, and the Inaugural director of the UBTECH Sydney Artificial Intelligence Centre, at The University of Sydney, Sydney, Australia. He mainly applies statistics and mathematics to artificial intelligence and data science. His research results have expounded in one monograph and more than 200 publications at prestigious journals and prominent conferences, such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, the IEEE Transactions on Neural Networks and Learning Systems, T-CYB, IJCV, JMLR, NIPS, ICML, CVPR, ICCV, ECCV, ICDM, and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDM'07, the best student paper award in IEEE ICDM'13, the 2014 ICDM 10-year highest-impact paper award, the 2017 IEEE Signal Processing Society Best Paper Award, and the distinguished paper award in the 2018 IJCAI. He received the 2015 Austrlian Scopus-Eureka Prize and the 2018 IEEE ICDM Research Contributions Award. He is a fellow of the Australian Academy of Science, AAAS, IEEE, IAPR, OSA and SPIE.