Bidirectional Probabilistic Multi-Graph Learning and Decomposition for Multi-View Clustering

Xinxin Wang, Yongshan Zhang, *Senior Member, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*—Graph-based multi-view clustering has attracted remarkable attention due to its impressive performance. However, the typical framework consisting of graph learning and indicator generation may fail to align learned graphs with the underlying data structure due to the unidirectional pipeline from refined graphs to indicator generation. Another common problem is the inadequate prior information in graph learning methods. This paper proposes a Bidirectional Probabilistic Multi-graph Learning and Decomposition (BPMLD) method by establishing an explicit bidirectional pipeline between graph learning and indicator generation for multi-view clustering. Specifically, we design a confidence term based on clustering probability indicators and fuse it with graph learning to form clustering confidence driven graph learning. Meanwhile, graph tensor learning is introduced to recover the high-order correlations among the refined graphs. We further propose a multi-graph probability decomposition module to adaptively produce cluster indicators with probability representation from the refined graphs. The seamless integration between graph learning and indicator generation enables them to interact directly and enhance each other. To solve the proposed model, we design an effective optimization algorithm. Extensive experiments demonstrate the effectiveness of our method compared to state-of-the-art methods. The code is available at: https://github.com/W-Xinxin/BPMLD.**

***Index Terms*—Multi-view clustering, clustering confidence, tensor learning, graph decomposition.**

I. Introduction

With advancements in manufacturing processes, numerous data acquisition devices have been developed to capture information from diverse data sources, facilitating in-depth analysis and understanding of objects. Multi-view data are collected from different perspectives and provide various insights of the same object or scene. This has become increasingly vital in the field of image and video processing due to its richness of information. Significant advancements in various multi-view learning methods have facilitated applications, such as object recognition [1], image segmentation [2], and image processing [3], [4]. Typically, the core principle of multi-view learning is to devise appropriate strategies that explore consistent and complementary information among views to achieve optimal consensus [5], [6].

Multi-view clustering aims to group similar samples by exploring the relations among samples across multiple views in an unsupervised manner. For example, multi-view clustering algorithms can effectively categorize patients using medical images captured by different techniques such as MRI, CT scans, and X-rays. Despite extensive studies on this task over the past decades, two long-standing challenges remain: 1) how to effectively integrate the partial information from each view into a consensus that comprehensively and accurately represents the relations among data, and 2) how to identify and utilize the discriminative group information in the clustering process at a deeper level.

A great variety of multi-view clustering methods have been proposed to further investigate these two challenges. These methods can be roughly summarized in the following three categories. Multi-view K-means learning methods extend single-view K-means methods, and partition multi-view data based on distances to cluster centroids [7]. K-means learning is well-regarded for its low computational cost and intuitive theory. Numerous variants have emerged, *e.g.,* multi-view kernel K-means [8] and embedded K-means [9]. However, these methods may experience performance degradation because they rely solely on distance measures between samples and cluster centroids. This degradation becomes particularly pronounced when data are distributed in a non-Gaussian manifold, rendering commonly used distance-based priors largely ineffective. Multi-view subspace clustering methods focus on identifying multiple low-dimensional subspaces and utilize the resulting subspace representation matrices for grouping samples [10]. By summarizing existing methods [11], [12], [13], we observe that a common approach is to impose designed regular constraints on subspace representation matrices to recover the underlying subspace structures across multiple views. However, subspace learning methods often suffer from high computational costs due to the iterative computation of these matrices from high-dimensional data.

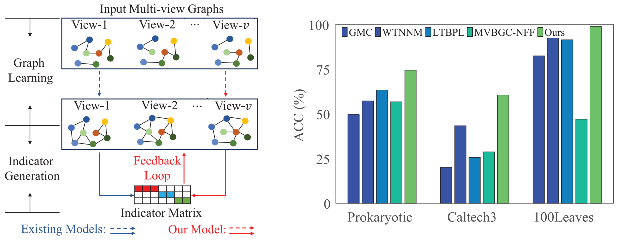
Unlike K-means clustering and subspace clustering methods, graph-based multi-view clustering mitigates the dependency on computational expense associated with data dimensionality. These methods typically start by constructing initial graphs for each view, followed by collaborative optimization strategies to achieve an optimal consensus among them. For instance, Nie et al. [14] proposed an element-wise distance measure for graph learning and introduced a

Received 17 May 2024; revised 4 November 2024 and 22 February 2025; accepted 13 May 2025. Date of publication 6 June 2025; date of current version 16 June 2025. This work was supported in part by the Science and Technology Development Fund, Macau, SAR, under Grant 0049/2022/A1 and Grant 0050/2024/AGJ; and in part by the University of Macau under Grant MYRG-GRG2024-00181-FST. The associate editor coordinating the review of this article and approving it for publication was Dr. Shiqi Wang. *(Corresponding author: Yicong Zhou.)*

Xinxin Wang and Yicong Zhou are with the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: xinxinwang1024@gmail.com; yicongzhou@um.edu.mo).

Yongshan Zhang is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China (e-mail: yszhang.cug@gmail.com).

Digital Object Identifier 10.1109/TIP.2025.3574924



(a) (b)

Fig. 1. Comparison between existing methods and our method. (a): Existing models present a unidirectional pipeline, while our model establishes a bidirectional one. (b): Recent graph-based methods, such as GMC, WTNNM, LTBPL, MVBGC-NFF, exhibit performance bottlenecks due to an implicit learning objective and inadequate prior issues.

connectivity constraint on the Laplacian matrix to achieve a global consensus affinity matrix. Zhan et al. [15] introduced a new disagreement measure to regularize initial graphs toward an optimal consensus graph. Gao et al. [16] employed a weighted tensor low-rank constraint to recover the low-rank spatial structure among multiple views. To explicitly account for both consistent and inconsistent information across views, the works in [17], [18] decomposed the initial graphs into diverse view-specific components and consistent components, then fused the consistent components to obtain the final consensus graph. Following this line of research, [19] further refined the inconsistent components into complementary components and noise components. For most graph-based multi-view clustering methods, spectral analysis and K-means algorithm are required to generate the final clustering indicators in additional steps after obtaining the final consensus graph.

Although existing methods have achieved extensive successes, these frameworks of graph-based multi-view clustering usually face two critical issues: inadequate prior and implicit learning objective, as shown in Fig. 1. To deal with inadequate prior issue, some existing works use pseudo labels to guide learning [20], [21]. These methods implicitly assume that the data distribution conforms to certain prior knowledge. However, data distributions in real-world applications are often quite complicated. Other methods fuse multiple priors into learning models to enhance model performance [22], [23]. For instance, [16] incorporates both column sparsity and global low-rank priors, while [24] combines global low-rank and singular value priors, and [11] considers dual structure information in graph learning. However, these methods still have some limitations. For example, linearly combining multiple priors introduces hyperparameter tuning challenge and manually defined priors may be biased. The second implicit learning objective issue arises because most existing methods employ a unidirectional pipeline [13], [16], [25]. Specifically, the refined affinity graphs dominate the quality of the cluster indicators. However, the resulting cluster indicators do not serve for the learning of the refined affinity graphs. Hence these two processes lack the necessary interactive feedback. As a result, the refined affinity graphs may not be suitable for the subsequent clustering tasks.

To address these issues simultaneously, we propose a unified framework called BPMLD. BPMLD achieves a bidirectional interaction between graph learning and clustering indicator generation to enhance clustering performance. We summarize our contributions as follows:

- We propose a clustering confidence driven graph learning module, which incorporates the highly confident information based on clustering probability indicators to remedy the inadequate prior issue.
- We propose a multi-graph decomposition module to decompose the refined affinity graphs into the consensus clustering probability indicators.
- Combining these two modules together, we propose a bidirectional probabilistic multi-graph learning and decomposition model named BPMLD, which establishes the explicit bidirectional refining pipeline to improve clustering performance.
- We devise an optimization algorithm to solve the quartic polynomial problem with constraints. Extensive experiments verify the effectiveness of our method.

II. Preliminaries

A. Notations

Throughout the paper, bold lower-case letters (e.g. $\mathbf{a}$), bold upper-case letters (e.g. $\mathbf{A}$) and calligraphy letters (e.g. $\mathcal{A}$) denote vectors, matrices, and tensors, respectively. For a matrix $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$, we use $\mathbf{A}_{i,:}$ to represent the i-th row vector of $\mathbf{A}$. The corresponding Frobenius norm, nuclear norm and ℓ_1-norm are defined as $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} \mathbf{A}_{ij}^2}$, $\|\mathbf{A}\|_* = \sum_i \sigma_i(\mathbf{A})$, and $\|\mathbf{A}\|_1 = \sum_{ij} \left|\mathbf{A}_{ij}\right|$, respectively, where $\mathbf{A}_{ij}$ is the element of $\mathbf{A}$, and $\sigma_i(\mathbf{A})$ is the i-th singular value of $\mathbf{A}$. For a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{A}^{(i,:,:)}, \mathcal{A}^{(:,j,:)}, \mathcal{A}^{(:,:,k)}$ represent the i-th horizontal, j-th lateral, and k-th frontal slice, respectively. For convenience, we use $\mathcal{A}^{(k)}$ to stand for the k-th frontal slice $\mathcal{A}^{(:,:,k)}$. The Frobenius norm and infinity norm of a 3-order tensor are defined as $\|\mathcal{A}\|_F = \sqrt{\sum_{ijk} \mathcal{A}_{ijk}^2}$ and $\|\mathcal{A}\|_\infty = \max_{ijk} \left|\mathcal{A}_{ijk}\right|$. The notation $\overline{\mathcal{A}}$ denotes the fast Fourier transformation (FFT) of $\mathcal{A}$ along the third dimension, *i.e.*, $\overline{\mathcal{A}} = fft(\mathcal{A}, [], 3)$. The tensor $\mathcal{A}$ can be recovered from $\overline{\mathcal{A}}$ by the inverse FFT operation, i.e., $\mathcal{A} = ifft(\overline{\mathcal{A}}, [], 3)$. Likewise, $\overline{\mathcal{A}}^{(i)}$ represents the i-th frontal slice of $\overline{\mathcal{A}}$. The block vectorization of $\mathcal{A}$ is given by $bvec(\mathcal{A}) = [\mathcal{A}^{(1)}; \dots; \mathcal{A}^{(n_3)}] \in \mathbb{R}^{n_1 n_3 \times n_2}$, and its inverse operation is defined as $ibvec(bvec(\mathcal{A})) = \mathcal{A}$. We then define the block circulant matrix $bcirc(\mathcal{A})$ of $\mathcal{A}$ as follows:

$$bcirc(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(n_3)} & \cdots & \mathcal{A}^{(2)} \\ \mathcal{A}^{(2)} & \mathcal{A}^{(1)} & \cdots & \mathcal{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}^{(n_3)} & \mathcal{A}^{(n_3-1)} & \cdots & \mathcal{A}^{(1)} \end{bmatrix} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}.$$

More details can be referred to [26]. Based on the above definitions, we further give the definitions of the tensor multi-rank and weighted tensor nuclear norm.

Definition 1 (Tensor Conjugate Transpose): For a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the conjugate transpose $\mathcal{A}^\top \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ can be obtained by conjugate-transposing each frontal slice of $\mathcal{A}$ and then reversing the order of the transposed frontal slices 2 through n_3.

Definition 2 (Identity Tensor): $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ is an identity tensor whose first frontal slice $\mathcal{I}^{(1)}$ is an $n \times n$ identity matrix, and the elements in the other frontal slices are all zeros.

Definition 3 (T-Product [27]): For two tensor $\mathcal{A}_1 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{A}_2 \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, we can compute the t-product as follows, $\mathcal{B} = \mathcal{A}_1 * \mathcal{A}_2 = ibvec(bcirc(\mathcal{A}_1).bvec(\mathcal{A}_2)) \in \mathbb{R}^{n_1 \times n_4 \times n_3}$.

In practice, the t-produce between $\mathcal{A}_1$ and $\mathcal{A}_2$ have an equivalence form in the Fourier domain, *i.e.*, $\mathcal{B} = \mathcal{A}_1 * \mathcal{A}_2$ and $\overline{\mathcal{B}} = \overline{\mathcal{A}}_1 \overline{\mathcal{A}}_2$ are equivalent.

Definition 4 (t-SVD [27]): For a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, it can be factorized by t-SVD as,

$$\mathcal{A} = \mathcal{U} * \Sigma * \mathcal{V}^{\top},$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors, (a tensor $\mathcal{P} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is orthogonal, if $\mathcal{P}^{\top} * \mathcal{P} = \mathcal{P} * \mathcal{P}^{\top} = \mathcal{I}$ holds), $\Sigma \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor, whose each frontal slice is diagonal.

The t-SVD of the tensor $\mathcal{A}$ can be equivalently computed by performing matrix SVD for each frontal slice $\overline{\mathcal{A}}^{(i)}$ in the Fourier domain, *i.e.*,

$$\overline{\mathcal{A}}^{(i)} = \overline{\mathcal{U}}^{(i)} \overline{\Sigma}^{(i)} \overline{\mathcal{V}}^{(i)^T}, \quad for \quad i = 1, 2, \ldots, n_3.$$

Definition 5 (Tensor Multi-Rank [28]): For a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its multi-rank can be defined as a vector $\mathbf{r} \in \mathbb{R}^{n_3 \times 1}$, whose the k-th element is the rank of the k-th frontal slices of $\overline{\mathcal{A}}^{(i)}$.

Definition 6 (Tensor Nuclear Norm [28]): For a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its tensor nuclear norm is defined as

$$||\mathcal{A}||_{\circledast} = \sum_{k=1}^{n_3} ||\overline{\mathcal{A}}^{(k)}||_* = \sum_{k=1}^{n_3} \sum_{i=1}^{\min(n_1,n_2)} \sigma_i(\overline{\mathcal{A}}^{(k)}).$$

It have been proven that the tensor nuclear norm is the tightest convex relaxation to l_1 norm of the tensor multi-rank [28].

Definition 7 (Weighted Tensor Nuclear Norm [16]): Considering the prior knowledge of singular values, the weighted tensor nuclear norm can be defined as

$$||\mathcal{A}||_{\omega,\circledast} = \sum_{k=1}^{n_3} \left\|\overline{\mathcal{A}}^{(k)}\right\|_{\omega,*} = \sum_{k=1}^{n3} \sum_{i=1}^{\min(n_1,n_2)} \omega_i \sigma_i \left(\overline{\mathcal{A}}^{(k)}\right),$$

where ω_i is the i-th element of the prior vector $\boldsymbol{\omega}$. In this paper, the dimension of the weighting factors $\boldsymbol{\omega}$ equals to the view number.

B. Naive Graph-Based Multi-View Clustering

Graph-based multi-view clustering **(GMVC)** methods jointly optimize multiple initial input similarity graphs toward a consensus that accurately reflects the underlying clustering structure. The typical graph learning paradigm can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{C}^v,\mathbf{E}^v} \ & \Omega(\mathbf{C}^v) + \lambda\Phi(\mathbf{E}^v) \\ s.t.\ & \mathbf{S}^v = \mathbf{C}^v + \mathbf{E}^v, \psi(\mathbf{C}^v), \end{aligned} \tag{1}$$

where $\{\mathbf{S}^v \in \mathbb{R}^{n \times n}\}_{v=1}^V$ are initial similarity graphs. However, their group structure is often obscured by the prevalent noise and data corruptions in real-world scenarios. Here, $\{\mathbf{C}^v \in \mathbb{R}^{n \times n}\}_{v=1}^V$ denote the refined graphs, and $\{\mathbf{E}^v \in \mathbb{R}^{n \times n}\}_{v=1}^V$ represent the corruptions removed from the initial graphs. $\Omega(\cdot)$ and $\Phi(\cdot)$ serve as certain regularization terms to refine the initial graphs and estimate corruptions, respectively. $\psi(\cdot)$ denotes specific numerical constraints, and λ is a trade-off factor balancing the two regularization terms. After optimizing Eq. (1) to obtain the refined graphs, the final affinity graph $\mathbf{Q} \in \mathbb{R}^{n \times n}$ can be computed as follows:

$$\mathbf{Q} = \frac{1}{V} \sum_{v=1}^{V} \frac{|\mathbf{C}^v| + |\mathbf{C}^v|^T}{2}. \tag{2}$$

After that, spectral clustering [29] or a connected component search method [30] can be applied to $\mathbf{Q}$ to obtain final clustering results. Despite achieving good performance, few methods address the limitations of the unidirectional pipeline and the issue of inadequate prior information.

III. Proposed Method

A. Overview

Fig. 2 illustrates the flowchart of the proposed BPMLD model. BPMLD includes two modules: (1) **Clustering Confidence Driven Graph Learning** (as shown in the left of Fig. 2): This module learns refined affinity graphs from the initial input graphs. It incorporates clustering confidence and tensor graph learning to facilitate the learning of refined graphs. (2) **Multi-graph Probability Decomposition** (as shown in the right of Fig. 2): This module adaptively computes consensus cluster probability indicators from the refined graphs in a single step and utilizes these indicators to calculate the clustering confidence of sample pairs. By integrating these two modules into a unified framework, the proposed BPMLD establishes a bidirectional refining pipeline between graph learning and clustering indicator generation. It effectively explores semantic information from graph structures as self-produced prior knowledge and uses this information to refine graph learning in each iteration. Finally, we take the optimized clustering probability indicators as clustering results.

B. Clustering Confidence Driven Graph Learning

Existing graph-based learning methods often employ a unidirectional learning pipeline. They utilize prior knowledge-driven learning procedures and result in implicit learning objectives. This limitation may lead to misalignments between affinity graphs and underlying data structure. Clustering indicators effectively reflect both data distribution and sample similarity. Introducing clustering feedback into the graph learning process enhances the reliability of the learning outcomes.

Given multiple initial input similarity graphs $\{S^v\}_{v=1}^V$, and assuming that noise or corruptions in the input graphs are sparse, the clustering confidence-driven graph learning with adaptive view weighting can be formulated as follows:

$$\min_{\mathbf{C}^v,\mathbf{E}^v,a_v} \sum_{v=1}^{V} a_v \sum_{i=1}^{n} \sum_{j=1}^{n} \left\|\mathbf{C}^v_{i,:} - \mathbf{C}^v_{j,:}\right\|_2^2 \mathbf{F}_{i,:}\mathbf{F}^T_{j,:} + \beta\, ||\mathcal{C}||_{\omega,\circledast} \\ + \lambda\, ||\mathbf{E}^v||_1$$

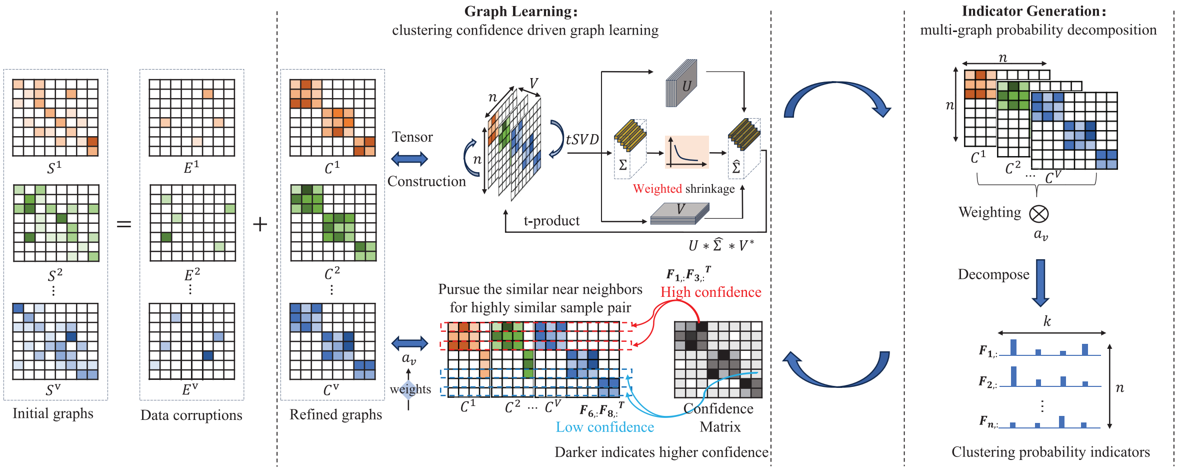


Fig. 2. The illustration of the proposed BPMLD model. The left side of the figure is: Clustering Confidence Driven Graph Learning module. It learns the refined affinity graphs $\mathbf{C}^1, \ldots, \mathbf{C}^V$ from input initial graphs $\mathbf{S}^1, \ldots, \mathbf{S}^V$. This module incorporates clustering confidence of samples and tensor graph learning to enhance the refinement process. The right side of the figure is: Multi-graph Probability Decomposition. This module adaptively balances the different refined affinity graphs to compute the consensus clustering probability indicators $\mathbf{F}$ that effectively facilitates the calculation of clustering confidence and forms a feedback loop through Eq. (5). Finally, the optimized $\mathbf{F}$ is taken as the final clustering result.

$$\begin{aligned} s.t.\ & \mathbf{S}^v = \mathbf{C}^v + \mathbf{E}^v, \mathbf{C}^v \geq 0, \mathbf{C}^v\mathbf{1} = \mathbf{1}, \\ & \mathcal{C} = \Psi(\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^V), \\ & \mathbf{F1} = \mathbf{1}, 0 \leq \mathbf{F}_i, :\leq 1, \mathbf{a1} = 1, a_v \geq 0, \end{aligned} \tag{3}$$

where $\{\mathbf{C}^v\}_{v=1}^V$ denote the refined affinity graphs, $\{\mathbf{E}^v\}_{v=1}^V$ present the data corruptions. The $\|\cdot\|_1$ norm characterizes the corruption term $\{\mathbf{E}^v\}_{v=1}^V$ stripped from the initial input graphs. Constraints $\{\mathbf{C}^v \geq 0, \mathbf{C}^v\mathbf{1} = \mathbf{1}\}$ denote the boundary constraint for the refined affinity graphs. Operator $\Psi(\cdot)$ stacks the refined affinity graphs $\{\mathbf{C}^v\}_{v=1}^V$ as frontal slices of a 3-order tensor $\mathcal{C}$ with dimensions $n \times n \times V$, and then rotates it to $n \times V \times n$. The term $\|\cdot\|_{\omega,\circledast}$ represents the emerging weighted tensor nuclear norm. It is imposed on $\mathcal{C}$ to capture the global low-rank properties across multiple views. The weight a_v balances the importance of each view. β and λ are non-negative trade-off parameters.

Matrix $\mathbf{F} \in \mathbb{R}^{n \times k}$ serves as the clustering probability indicators for n samples. It presents the probability of each sample belonging to one of the k clusters. The inner product of clustering indicators $\mathbf{F}_{i,:}\mathbf{F}_{j,:}^T$ effectively reflects the clustering confidence of the i-th sample and j-th samples belonging to the same category. $\mathbf{C}_{i,:}^v$ presents the nearest neighboring relationship of the i-th sample with other samples in the v-th graph. Minimizing the first term in Eq. (3) ensures that sample pairs with a high probability of belonging to the same category maintain the similar nearest neighboring relationships across each view. Regarding clustering confidence as prior knowledge, this module effectively addresses the inadequate prior issue.

C. Multi-Graph Probability Decomposition

To implement Eq. (3), we require the clustering probability indicator matrix $\mathbf{F}$. However, existing GMVC methods do not generate this matrix in a single step. To address this issue, we propose a graph decomposition approach to facilitate the generation of clustering probability indicators in a single step. Specifically, the clustering probability indicators can be obtained by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{F}} \ & \sum_{v=1}^{V} \left\| a_v\mathbf{C}^v - \mathbf{F}\mathbf{F}^T \right\|_F^2, \\ s.t.\ & \mathbf{F1} = \mathbf{1}, 0 \leq \mathbf{F}_{i,:} \leq 1, \end{aligned} \tag{4}$$

where $\mathbf{F}$ indicates the consensus clustering probability indicators. Constraints $\mathbf{F1} = \mathbf{1}, 0 \leq \mathbf{F}_{i,:} \leq 1$ ensure that the values are non-negative and sum to one. Additionally, view weights in Eq. (3) are utilized to balance the contribution of each view. Eq. (4) symmetrically decomposes multiple refined graphs into the consensus clustering probability indicators.

The resulting $\mathbf{F}$ effectively captures the semantic information of samples across multiple views. Its probability representation along with the inner product form ensures that the values of the clustering confidence term remain within the continuous interval $[0, 1]$. This is important for accurately reflecting semantic similarity in clustering scenarios. Therefore, $\mathbf{F}$ serves as a suitable intermediate variable for computing clustering confidence and guiding the learning of affinity graphs.

D. Overall Objective Function

Combining the two formulas discussed above, the final objective function of the proposed BPMLD can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{C}^v,\mathbf{E}^v,a_v,\mathbf{F}} \ & \sum_{v=1}^{V} \left\| a_v\mathbf{C}^v - \mathbf{F}\mathbf{F}^T \right\|_F^2 + \beta\|\mathcal{C}\|_{\omega,\circledast} + \lambda\|\mathbf{E}^v\|_1 \\ & + \alpha \sum_{v=1}^{V} a_v \sum_{i=1}^{n}\sum_{i=1}^{n} \left\| \mathbf{C}_i^v - \mathbf{C}_j^v \right\|_2^2 \mathbf{F}_{i,:}\mathbf{F}_{j,:}^T \\ s.t.\ & \mathbf{S}^v = \mathbf{C}^v + \mathbf{E}^v, \mathbf{C}^v \geq 0, \mathbf{C}^v\mathbf{1} = \mathbf{1}, \\ & \mathbf{F1} = \mathbf{1}, 0 \leq \mathbf{F}_{i,:} \leq 1, \mathbf{a1} = 1, a_v \geq 0 \\ & \mathcal{C} = \Psi(\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^V), \end{aligned} \tag{5}$$

where α are non-negative trade-off parameters.

Eq. (5) provides an explicit learning objective for both graph learning and clustering simultaneously. This facilitates the alignment between refined affinity graphs and underlying data distribution. This paradigm enables mutual refinement between both processes, and thus ultimately enhances the clustering performance.

IV. Optimization Algorithm

A. Optimization Strategy

It is intractable to optimize the formulation of Eq. (5) straightforwardly because of the existence of the quartic term and multiple constraint terms. We design an alternating optimization algorithm based on alternating direction method of multipliers (ADMM) [31] to solve each variable by fixing others. To make Eq. (5) separable, we introduce one auxiliary variable $\mathcal{J} = \mathcal{C}$. The corresponding augmented Lagrangian function is as follows:

$$\begin{aligned}\min_{\mathbf{C}^v,\mathbf{E}^v,a_v,\mathbf{F},\mathcal{J}} \quad & \sum_{v=1}^{V} \left\| a_v\mathbf{C}^v - \mathbf{F}\mathbf{F}^T \right\|_F^2 + \beta \|\mathcal{J}\|_{\omega,\circledast} + \lambda \|\mathbf{E}^v\|_1 \\ & + \alpha \sum_{v=1}^{V} a_v \sum_{i=1}^{n} \sum_{i=1}^{n} \left\| \mathbf{C}_i^v - \mathbf{C}_j^v \right\|_2^2 \mathbf{F}_{i,:}\mathbf{F}_{j,:}^T \\ & + \sum_{v=1}^{V} \frac{\rho}{2} \left\| \mathbf{S}^v - \mathbf{C}^v - \mathbf{E}^v + \frac{\mathbf{Y}_1^v}{\rho} \right\|_F^2 \\ & + \frac{\rho}{2} \left\| \mathcal{C} - \mathcal{J} + \frac{\mathcal{Y}_2}{\rho} \right\|_F^2 \\ s.t. \ & \mathbf{C}^v \geq 0, \mathbf{C}^v\mathbf{1} = \mathbf{1}, \mathcal{C} = \Psi(\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^V), \\ & \mathbf{F1} = \mathbf{1}, 0 \leq \mathbf{F}_{i,:} \leq 1, \mathbf{a1} = 1, a_v \geq 0, \end{aligned} \tag{6}$$

where $\mathbf{Y}_1^v \in \mathbb{R}^{n\times n}$ and $\mathcal{Y}_2 \in \mathbb{R}^{n\times V\times n}$ are Lagrange multipliers, and ρ is penalty factor.

Updating $\mathbf{E}^v$: Fixing other variables, we optimize $\mathbf{E}^v$ in each view as follow

$$\min_{\mathbf{E}^v} \frac{\lambda}{\rho} \|\mathbf{E}^v\|_1 + \frac{1}{2} \left\| \mathbf{E}^v - \left(\mathbf{S}^v - \mathbf{C}^v + \frac{\mathbf{Y}_1^v}{\rho} \right) \right\|_F^2. \tag{7}$$

It can be solved by the soft-threshold operator [32].

Updating $\mathbf{C}^v$: Fixing other variables, we update $\mathbf{C}^v$ in each view by solving the following problem,

$$\begin{aligned}\min_{\mathbf{C}^v} \quad & \left\| a_v\mathbf{C}^v - \mathbf{F}\mathbf{F}^T \right\|_F^2 + \alpha a_v \sum_{i=1}^{n} \sum_{i=1}^{n} \left\| \mathbf{C}_i^v - \mathbf{C}_j^v \right\|_2^2 \mathbf{F}_{i,:}\mathbf{F}_{j,:}^T \\ & + \frac{\rho}{2} \left\| \mathbf{C}^v - \left(\mathbf{S}^v - \mathbf{E}^v + \frac{\mathbf{Y}_1^v}{\rho} \right) \right\|_F^2 + \frac{\rho}{2} \left\| \mathbf{C}^v - \mathbf{J}^v + \frac{\mathbf{Y}_2^v}{\rho} \right\|_F^2 \\ s.t. \ & \mathbf{C}^v \geq 0, \mathbf{C}^v\mathbf{1} = \mathbf{1}. \end{aligned} \tag{8}$$

With the normalized condition $\mathbf{C}^v \geq 0, \mathbf{C}^v\mathbf{1} = \mathbf{1}$, it is difficult to find the optimal result directly. For efficiency, we introduce an auxiliary variable $\hat{\mathbf{C}}^v$. By removing the normalized condition, the problem can be reformulated as

$$\begin{aligned}\min_{\hat{\mathbf{C}}^v} \quad & \left\| a_v\hat{\mathbf{C}}^v - \mathbf{F}\mathbf{F}^T \right\|_F^2 + \alpha a_v \sum_{i=1}^{n} \sum_{i=1}^{n} \left\| \hat{\mathbf{C}}_i^v - \hat{\mathbf{C}}_j^v \right\|_2^2 \mathbf{F}_{i,:}\mathbf{F}_{j,:}^T \\ & + \frac{\rho}{2} \left\| \hat{\mathbf{C}}^v - \left(\mathbf{S}^v - \mathbf{E}^v + \frac{\mathbf{Y}_1^v}{\rho} \right) \right\|_F^2 + \frac{\rho}{2} \left\| \hat{\mathbf{C}}^v - \left(\mathbf{J}^v - \frac{\mathbf{Y}_2^v}{\rho} \right) \right\|_F^2. \end{aligned} \tag{9}$$

Taking the derivative with respect to $\hat{\mathbf{C}}^v$ and setting it to zero, the optimal solution of Eq. (9) is

$$\hat{\mathbf{C}}^v = (2a_v\mathbf{F}\mathbf{F}^T + \rho\mathbf{Q}^v + \rho\mathbf{P}^v)(2a_v^2\mathbf{I} + 4\alpha a_v\mathbf{L} + 2\rho\mathbf{I})^{-1}, \tag{10}$$

where $\mathbf{Q}^v = \mathbf{S}^v - \mathbf{E}^v + \frac{\mathbf{Y}_1^v}{\rho}$, and $\mathbf{P}^v = \mathbf{J}^v - \frac{\mathbf{Y}_2^v}{\rho}$.

With the solution of $\hat{\mathbf{C}}^v$, the optimal $\mathbf{C}^v$ can be obtained by solving the following problem

$$\min_{\mathbf{C}^v \geq 0, \mathbf{C}^v\mathbf{1}=\mathbf{1}} \left\| \mathbf{C}^v - \hat{\mathbf{C}}^v \right\|_F^2. \tag{11}$$

The problem of Eq. (11) is an Euclidean projection problem on the simplex space. It can be settled as in [33].

Updating $\mathcal{J}$: Fixing other variables, $\mathcal{J}$ can be updated by

$$\min_{\mathcal{J}} \beta \|\mathcal{J}\|_{\omega,\circledast} + \frac{\rho}{2} \|\mathcal{J} - \mathcal{B}\|_F^2, \tag{12}$$

where $\mathcal{B} = \mathcal{C} + \frac{\mathcal{Y}_2}{\rho}$. Based on Definition 6, we transform Eq. (12) into n_3 separated sub-problems in the frequency domain, and the k-th problem is

$$\min_{\overline{\mathcal{J}}^{(k)}} \sum_{j=1}^{V} \frac{\beta}{\rho} \omega_j \sigma_j(\overline{\mathcal{J}}^{(k)}) + \frac{1}{2} \left\| \overline{\mathcal{J}}^{(k)} - \overline{\mathcal{B}}^{(k)} \right\|_F^2. \tag{13}$$

The optimal solution of $\overline{\mathcal{J}}^{(k)}$ is

$$\overline{\mathcal{J}}^{(k)} = \mathbf{U}_{\overline{\mathcal{B}}^{(k)}} \Theta_{\frac{\beta}{\rho}\omega_j}(\Sigma_{\overline{\mathcal{B}}^{(k)}}) \mathbf{V}_{\overline{\mathcal{B}}^{(k)}}^T, \tag{14}$$

where $\mathbf{U}_{\overline{\mathcal{B}}^{(k)}} \Sigma_{\overline{\mathcal{B}}^{(k)}} \mathbf{V}_{\overline{\mathcal{B}}^{(k)}}^T$ is the SVD of $\overline{\mathcal{B}}^{(k)}$, and $\Theta(\cdot)$ is the common Singular Value Thresholding (SVT) operator. Once we obtain $\overline{\mathcal{J}}$, $\mathcal{J}$ can be recovered by inverse FFT operation.

Updating $\mathbf{a}$: Fixing other variables, $\mathbf{a}$ can be updated by solving the following problem,

$$\begin{aligned}\min_{\mathbf{a}} \sum_{v=1}^{V} \left\| a_v\mathbf{C}^v - \mathbf{F}\mathbf{F}^T \right\|_F^2 + \alpha \sum_{v=1}^{V} a_v b_v \\ s.t. \ \mathbf{a1} = 1, a_v \geq 0. \end{aligned} \tag{15}$$

where $b_v = \sum_{i=1}^{n} \sum_{i=1}^{n} \left\| \mathbf{C}_i^v - \mathbf{C}_j^v \right\|_2^2 \mathbf{F}_{i,:}\mathbf{F}_{j,:}^T$. Extending the Frobenius norm with trace and removing the irrelevant term, Eq. (15) can be rewritten as

$$\begin{aligned}\min_{\mathbf{a}} \sum_{v=1}^{V} a_v^2 \mathrm{Tr}\left(\mathbf{C}^{v^T}\mathbf{C}^v \right) - 2\ a_v \mathrm{Tr}\left(\mathbf{C}^{v^T}\mathbf{F}\mathbf{F}^T \right) + \alpha a_v b_v \\ s.t. \ \mathbf{a1} = 1, a_v \geq 0. \end{aligned} \tag{16}$$

Eq. (16) is a simple Quadratic Programming (QP) problem. We employ the off-the-shelf quadratic programming solver to address this problem.

Updating $\mathbf{F}$: Fixing other variables, we can update $\mathbf{F}$ by the following problem

$$\begin{aligned}\min_{\mathbf{F}} \sum_{v=1}^{V} \left\| a_v\mathbf{C}^v - \mathbf{F}\mathbf{F}^T \right\|_F^2 + \alpha a_v \mathrm{Tr}\left(\mathbf{D}_c^v\mathbf{F}\mathbf{F}^T \right) \\ s.t. \ \mathbf{F1} = \mathbf{1}, 0 \leq \mathbf{F}_{i,:} \leq 1, \end{aligned} \tag{17}$$

where $\mathbf{D}_c^v$ is the measure matrix obtained as $(\mathbf{D}_c^v)_{i,j} = \left\| \mathbf{C}_i^v - \mathbf{C}_j^v \right\|_2^2$. It is difficult to find the optimal $\mathbf{F}$ directly due to the existence of the quartic term. Under the consideration

Algorithm 1 BPMLD

```
Input: Initial graphs of the multi-view data with $V$ views
    $\{\mathbf{S}^1, \mathbf{S}^2, ..., \mathbf{S}^V\}$ and the number of classes $k$, parameters
    $\alpha, \beta, \lambda$, prior knowledge $\omega$.
Output: Clustering labels of data points.
Initialize $\mathbf{E}^v = \mathbf{C}^v = \mathbf{Y}_1^v = 0$, $\mathbf{J} = \mathbf{Y}_2 = 0$, $\mathbf{F} = 0$,
    $\rho = 0.0001, \gamma = 2, \rho_{max} = 10^8$
Initialize $a_v$ with $\frac{1}{V}$.
while not converge do
    Update $\mathbf{E}^v$ by solving Eq. (7);
    Update $\mathbf{C}^v$ by Eq. (11);
    Update $\mathbf{J}$ by solving Eq. (14);
    Update $\mathbf{a}$ by solving Eq. (16);
    Update $\mathbf{F}$ by Algorithm 2;
    Update $\mathbf{Y}_1^v, \mathbf{Y}_2$ and $\rho$ by Eq.(24) to Eq.(26);
    Check the converge conditions:
        $\|\mathbf{S}^v - \mathbf{C}^v - \mathbf{E}^v\|_\infty < 10^{-5}$, $\|\mathbf{C} - \mathbf{J}\|_\infty < 10^{-5}$
end while
Output the index of max elements of each row in $F$.
```

Algorithm 2 Optimizing **F**

```
Input: a, $\{\mathbf{C}^v\}_{v=1}^V$, $\{\mathbf{D}_c^v\}_{v=1}^V$.
Output: F
Initialize $\mathbf{F} = \mathbf{G} = \mathbf{Y}_3 = 0$, $\mu = 0.001$, $\rho = 1.3$, $\mu_{max} =$
    $10^5$.
while not converge do
    Update $\mathbf{G}$ by solving Eq. (19);
    Update $\mathbf{F}$ by Eq. (21);
    Update $\mathbf{Y}_3$ by Eq. (22);
    Update $\mu$ by Eq. (23);
    Check convergence conditions:
        $\|\mathbf{F} - \mathbf{G}\|_\infty < 10^{-5}$
end while
```

of efficiency, we introduce an auxiliary variable $\mathbf{G} = \mathbf{F}$, and use ADMM to solve this sub-problem. The corresponding augmented Lagrangian function is

$$\begin{aligned}\min_{\mathbf{F},\mathbf{G}} &\sum_{v=1}^{V} \left\|a_v\mathbf{C}^v - \mathbf{F}\mathbf{G}^T\right\|_F^2 + \alpha a_v \mathrm{Tr}\left(\mathbf{D}_c^v\mathbf{F}\mathbf{G}^T\right)\\ &+ \frac{\mu}{2}\left\|\mathbf{F} - \mathbf{G} + \frac{\mathbf{Y}_3}{\mu}\right\|_F^2\\ s.t.\ &\mathbf{F}\mathbf{1} = \mathbf{1}, 0 \le \mathbf{F}_{i,:} \le 1,\end{aligned} \tag{18}$$

where $\mathbf{Y}_3 \in \mathbb{R}^{n\times k}$ is Lagrange multiplier, and μ is penalty factor. When $\mathbf{F}$ is fixed, $\mathbf{G}$ can be solved as follows:

$$\mathbf{G} = \mathbf{R}(2V\mathbf{F}^T\mathbf{F} + \mu\mathbf{I})^{-1}, \tag{19}$$

where $\mathbf{R} = \mu\mathbf{F} + \mathbf{Y}_3 + \sum_{v=1}^{V} 2\ a_v\mathbf{C}_v^T\mathbf{F} - \alpha\sum_{v=1}^{V} a_v\mathbf{D}_c\mathbf{F}$. To solve $\mathbf{F}$ with normalization constraint by fixing $\mathbf{G}$, we reformulate the problem similarly to Eq. (8) by introducing auxiliary variable $\hat{\mathbf{F}}$. Thus, we solve $\hat{\mathbf{F}}$ without considering the normalization constraint as follows:

$$\hat{\mathbf{F}} = \mathbf{T}(2V\mathbf{G}^T\mathbf{G} + \mu\mathbf{I})^{-1}, \tag{20}$$

where $\mathbf{T} = \mu\mathbf{G} + \mathbf{Y}_3 + \sum_{v=1}^{V} 2\ a_v\mathbf{C}_v\mathbf{G} - \alpha\sum_{v=1}^{V} a_v\mathbf{D}_c^T\mathbf{G}$. With the solution of $\hat{\mathbf{F}}$, the optimal $\mathbf{F}$ can be obtained by solving the following problem,

$$\min_{\mathbf{F}^v\ge 0,\mathbf{F}^v\mathbf{1}=\mathbf{1}} \left\|\mathbf{F} - \hat{\mathbf{F}}\right\|_F^2. \tag{21}$$

It can be easily solved as Eq. (11). The Lagrange multiplier and penalty factor are updated as follows

$$\mathbf{Y}_3 = \mathbf{Y}_3 + \mu(\mathbf{F} - \mathbf{G}), \tag{22}$$

$$\mu = \min(\delta\mu, \mu_{max}), \tag{23}$$

where $\delta > 1$ is a learning rate, and μ_{max} is the upper bound of μ. We summarize the optimization of $\mathbf{F}$ in Algorithm 2.

Updating the rest variables: With the other variables fixed, the rest variables are alternately updated by

$$\mathbf{Y}_1^v = \mathbf{Y}_1^v + \rho(\mathbf{S}^v - \mathbf{C}^v - \mathbf{E}^v), \tag{24}$$

$$\mathcal{Y}_2 = \mathcal{Y}_2 + \rho(\mathcal{C} - \mathcal{J}), \tag{25}$$

$$\rho = \min(\gamma\rho, \rho_{max}), \tag{26}$$

where $\gamma > 1$ is a learning rate, and ρ_{max} is the upper bound of ρ.

Commonly, the penalty parameters μ and ρ should be initialized to a small positive number to achieve an optimal solution [31]. However, this may lead to a slow convergence rate. To accelerate the convergence rate, we employ the varying penalty strategy [34] by introducing extra parameter δ and γ in Eq. (23) and Eq. (26). Such that, the penalty parameters μ and ρ are enlarged with a positive learning rate until the upper bound is met. Through this strategy, the empirical convergence of the optimization algorithm is improved and the optimization performance will be less dependent on the initialization of μ and ρ.

Since Eq. (6) consists of two constraints, we define the residuals as follows:

$$\begin{aligned} r1 &= \max\{\|\mathbf{S}^v - \mathbf{C}^v - \mathbf{E}^v\|_\infty\},\\ r2 &= \|\mathbf{C} - \mathbf{J}\|_\infty, \end{aligned} \tag{27}$$

where $\|\cdot\|_\infty$ indicates the maximum elements in the matrix.

The stopping criterion is that the residuals are small enough along with the iterative optimization scheme. For clarity, the overall optimization procedure of Eq. (6) is summarized in Algorithm 1.

B. Algorithm Analysis

1) Convergence Analysis: In this paper, the proposed Algorithm 1 involves five blocks of variables, and the objective function of Eq. (6) is non-smooth with respect to these variables, making it difficult to prove theoretical convergence [35]. In practice, many methods employ ADMM-based algorithms to solve problems with more than six variable blocks [19], [36]. Furthermore, we empirically find that the proposed Algorithm 1 exhibits good convergence behaviors and generally converges within dozens of iterations using the varying penalty strategy when clustering real-world benchmark datasets. The residual curves of Algorithm 1 on two benchmark datasets are plotted in the Experiment section for detailed illustration.

2) Complexity Analysis: The proposed BPMLD model is optimized in an alternating manner. In each iteration, the optimization process is divided into five sub-problems, along with the multipliers and penalty update steps. For solving $\mathbf{E}^v$, it costs $\mathcal{O}(Vn^2)$ for element-wise thresholding. For updating $\mathbf{C}^v$, it mainly costs $\mathcal{O}(Vn^2)$ for Euclidean projection. For solving $\mathcal{J}$, it mainly costs $\mathcal{O}(Vn^2log(n))$ to perform FFT and inverse FFT on an $n \times V \times n$ tensor. When updating $\mathbf{F}$ in Algorithm 2, it mainly takes $\mathcal{O}(t_b n^2 k)$ on matrix multiplication, where t_b is the iteration number. For updating a_v, it needs $\mathcal{O}(V^3)$ to perform quadratic programming. Therefore, the overall computational complexity is approximately $\mathcal{O}(t_a(Vn^2 + Vn^2log(n) + t_b n^2 k + V^3))$, where t_a is the iteration number of the above five sub-problems.

C. Discussion

The proposed method evolves from graph learning but diverges from conventional graph-based methods. These conventional approaches rely on implicit learning objectives to learn affinity graphs, followed by spectral analysis of the Laplacian matrix and discretization to obtain final clustering results. Unlike previous methods, our method outputs soft clustering indicators in a single step by directly decomposing the refined affinity graphs.

Meanwhile, it uses these indicators to reflect clustering confidence among samples and applies them to effectively preserve clustering-oriented structure information within the affinity graph learning process. Both processes are unified in a single model. Moreover, our approach differs from symmetric nonnegative matrix factorization. This is because our method performs probability decomposition of the affinity graphs for the first time while simultaneously refining affinity graphs using clustering feedback. Compared with existing learning-based methods, our method achieves bidirectional interaction between graph learning and clustering indicator generation, and thus forms an explicit learning objective for both processes to enhance clustering performance. Compared with deep learning-based methods, our method is deterministic and easy to deploy.

V. Experiments

A. Experimental Settings

1) Datasets: We perform extensive experiments on the following ten datasets, including human face recognition databases (Yale, ORL), scene recognition database (MSRCv1), object recognition databases (Caltech3, Caltech7, and 100Leaves), image-text classification database (Wiki_fea), gene analysis databases (Prokaryotic, BDGP), and video analysis database (CCV). These benchmark datasets are popular and widely adopted in recent multi-view clustering studies. Among these, six datasets (Yale, Caltech7, 100Leaves, Wiki_fea, Caltech3, and BDGP) are also utilized in the original papers of competing methods for evaluation [16], [25], [33], [37], [38].

Specifically, **Yale**[1] [16] is a human face recognition database with 165 gray-scale facial images across 15 classes, described by three kinds of descriptors. **ORL**[2] [5] is also a human face recognition database containing 400 images from 40 distinct subjects. Each image is represented by three kinds of descriptors. **MSRCv1**[3] [39] is a scene recognition database with 210 images from seven scenes. Five kinds of features, i.e., LBP, HOG, GIST, CENTRIST, and SIFT are extracted from each image. **Caltech7**[4] [37] is a subset of the famous object recognition dataset Caltech101 [16]. It contains 1474 pictures of objects belonging to seven categories, and each picture is represented by six kinds of extracted features. **Caltech3** is also a subset of Caltech101, containing 512 images of objects from eleven classes, and three kinds of extracted features are used. **100Leaves**[5] [33] is an image identification database of plant leaves, including 1600 samples from 100 kinds of plants, represented by shape descriptor, texture histogram and fine-scale margin. **Wiki_fea**[6] [37] is an image-text classification database selected from Wikipedia. It consists of 2866 sections across ten categories, described by text and images, respectively. **Prokaryotic**[7] [40] is a bacterial classification database containing 551 prokaryotic species across four categories, presented in text, proteome, and genomic representation. **BDGP**[8] [38] is a biology gene database collected from 2500 instances of five varieties. It contains three kinds of Drosophila gut microbe genome sequences. **CCV**[9] [41] is a video analysis dataset containing 6773 YouTube videos across 20 semantic categories.

2) Competing Methods: We compare our method with one baseline method and eleven state-of-the-art methods across four types: (a) Matrix based full graph methods: SC (best) [29], GMC [33], CGD [25], EMGC^{2}F [39], and OMVCDR [38]; (b) Tensor based full graph methods: WTNNM [16], LTBPL [24]; (c) Anchor graph based methods: EMVGC-LG [11], MVBGC-NFF [13], and (d) Deep learning method: DealMVC [37], SCMVC [42], and DIVIDE [43]. We briefly introduce the methods that are not mentioned in previous sections. SC (best) is a baseline method that performs standard spectral clustering on each view, and report the best results. CGD recovers the unified graph in cross-view graph diffusion process and performs spectral clustering to obtain final results. EMGC^{2}F considers the credible consistent information among multiple views and directly learns a consistent clustering indicator matrix. DealMVC is a deep model that performs multi-view contrastive clustering considering local and global cross-view feature consistency.

Three common metrics, *i.e.,* clustering accuracy (ACC), normalized mutual information (NMI) and purity, are used to evaluate the clustering performance. For these evaluation metrics, higher values indicate better clustering performance.

3) Implementation Details: For fair comparison, we use the released source codes of competing methods in our

[1] https://cvc.yale.edu/projects/yalefaces/yalefaces.html

[2] https://cam-orl.co.uk/facedatabase.html

[3] https://www.microsoft.com/en-us/research/project/

[4] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[5] https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set

[6] http://www.svcl.ucsd.edu/projects/crossmodal/

[7] https://github.com/mbrbic/Multi-view-LRSSC/tree/master/datasets

[8] https://www.fruitfly.org/

[9] http://www.ee.columbia.edu/ln/dvmm/CCV/

TABLE I

CLUSTERING RESULTS OF DIFFERENT METHODS ON TEN DATASETS. THE 1ST/2ND BEST RESULTS ARE MARKED IN BOLD AND UNDERLINE, '-' MEANS EXECUTION FAILURE ON THAT DATASET

Methods	Yale	Caltech7	100Leaves	Wiki_fea	Prokaryotic	BDGP	CCV	Caltech3	ORL	MSRCv1
ACC										
SC(best)	$0.646_{\pm 0.003}$	$0.681_{\pm 0.000}$	$0.686_{\pm 0.008}$	$0.555_{\pm 0.000}$	$0.625_{\pm 0.037}$	$0.437_{\pm 0.000}$	$0.207_{\pm 0.003}$	$0.358_{\pm 0.000}$	$0.832_{\pm 0.008}$	$0.762_{\pm 0.000}$
GMC	$0.655_{\pm 0.000}$	$0.692_{\pm 0.000}$	$0.824_{\pm 0.000}$	$0.195_{\pm 0.000}$	$0.495_{\pm 0.000}$	-	$0.106_{\pm 0.000}$	$0.201_{\pm 0.000}$	$0.632_{\pm 0.000}$	$0.747_{\pm 0.000}$
CGD	$0.531_{\pm 0.003}$	$0.680_{\pm 0.000}$	$0.819_{\pm 0.011}$	$0.547_{\pm 0.000}$	$0.651_{\pm 0.000}$	$0.407_{\pm 0.000}$	$0.216_{\pm 0.000}$	$0.249_{\pm 0.002}$	$0.662_{\pm 0.022}$	$0.852_{\pm 0.000}$
EMGC^{2}F	$0.667_{\pm 0.000}$	$0.624_{\pm 0.000}$	$0.866_{\pm 0.000}$	$0.277_{\pm 0.000}$	$0.376_{\pm 0.000}$	$0.201_{\pm 0.000}$	$0.186_{\pm 0.000}$	$0.240_{\pm 0.000}$	$0.805_{\pm 0.000}$	$0.914_{\pm 0.000}$
WTNNM	$\underline{0.979_{\pm 0.000}}$	$0.519_{\pm 0.000}$	$\underline{0.924_{\pm 0.005}}$	$0.547_{\pm 0.002}$	$0.572_{\pm 0.000}$	$\underline{0.745_{\pm 0.000}}$	$\underline{0.379_{\pm 0.015}}$	$\underline{0.433_{\pm 0.015}}$	$0.945_{\pm 0.021}$	$0.871_{\pm 0.005}$
LTBPL	$0.939_{\pm 0.000}$	$\underline{0.870_{\pm 0.000}}$	$0.914_{\pm 0.000}$	$\underline{0.608_{\pm 0.000}}$	$0.633_{\pm 0.000}$	$0.202_{\pm 0.000}$	$0.122_{\pm 0.000}$	$0.256_{\pm 0.000}$	$\underline{0.957_{\pm 0.000}}$	$\underline{0.943_{\pm 0.000}}$
EMVGC-LG	$0.712_{\pm 0.018}$	$0.582_{\pm 0.000}$	$0.763_{\pm 0.001}$	$0.537_{\pm 0.000}$	$0.548_{\pm 0.034}$	$0.489_{\pm 0.007}$	$0.243_{\pm 0.000}$	$0.352_{\pm 0.003}$	$0.729_{\pm 0.021}$	$0.808_{\pm 0.022}$
MVBGC-NFF	$0.592_{\pm 0.048}$	$0.420_{\pm 0.031}$	$0.470_{\pm 0.014}$	$0.536_{\pm 0.031}$	$0.567_{\pm 0.002}$	$0.408_{\pm 0.039}$	$0.228_{\pm 0.006}$	$0.287_{\pm 0.015}$	$0.677_{\pm 0.039}$	$0.701_{\pm 0.034}$
DealMVC	$0.588_{\pm 0.000}$	$0.768_{\pm 0.000}$	$0.244_{\pm 0.000}$	$0.593_{\pm 0.000}$	$0.588_{\pm 0.000}$	$0.314_{\pm 0.000}$	$0.165_{\pm 0.000}$	$0.309_{\pm 0.000}$	$0.125_{\pm 0.000}$	$0.652_{\pm 0.000}$
SCMVC	$0.376_{\pm 0.015}$	$0.529_{\pm 0.017}$	$0.578_{\pm 0.035}$	$0.598_{\pm 0.017}$	$\underline{0.721_{\pm 0.061}}$	$0.484_{\pm 0.015}$	$0.257_{\pm 0.007}$	$0.232_{\pm 0.004}$	$0.412_{\pm 0.023}$	$0.523_{\pm 0.005}$
DIVIDE	$0.594_{\pm 0.025}$	$0.359_{\pm 0.023}$	$0.850_{\pm 0.016}$	$0.545_{\pm 0.027}$	$0.490_{\pm 0.032}$	$0.339_{\pm 0.011}$	$0.224_{\pm 0.005}$	$0.273_{\pm 0.010}$	$0.705_{\pm 0.028}$	$0.579_{\pm 0.039}$
OMVCDR	$0.648_{\pm 0.000}$	$0.553_{\pm 0.000}$	$0.683_{\pm 0.000}$	$0.534_{\pm 0.000}$	$0.546_{\pm 0.000}$	$0.569_{\pm 0.000}$	$0.151_{\pm 0.000}$	$0.293_{\pm 0.000}$	$0.598_{\pm 0.000}$	$0.624_{\pm 0.000}$
Ours	$\mathbf{0.982}_{\pm 0.000}$	$\mathbf{0.905}_{\pm 0.000}$	$\mathbf{0.989}_{\pm 0.000}$	$\mathbf{0.617}_{\pm 0.000}$	$\mathbf{0.744}_{\pm 0.000}$	$\mathbf{0.870}_{\pm 0.000}$	$\mathbf{0.415}_{\pm 0.000}$	$\mathbf{0.605}_{\pm 0.000}$	$\mathbf{0.977}_{\pm 0.000}$	$\mathbf{0.947}_{\pm 0.000}$
NMI										
SC(best)	$0.464_{\pm 0.005}$	$0.583_{\pm 0.000}$	$0.575_{\pm 0.006}$	$0.405_{\pm 0.000}$	$0.334_{\pm 0.020}$	$0.211_{\pm 0.000}$	$0.067_{\pm 0.002}$	$0.142_{\pm 0.000}$	$0.908_{\pm 0.006}$	$0.525_{\pm 0.000}$
GMC	$0.689_{\pm 0.000}$	$\underline{0.660_{\pm 0.000}}$	$0.929_{\pm 0.000}$	$0.078_{\pm 0.000}$	$0.193_{\pm 0.000}$	-	$0.004_{\pm 0.000}$	$0.162_{\pm 0.000}$	$0.857_{\pm 0.000}$	$0.771_{\pm 0.000}$
CGD	$0.608_{\pm 0.004}$	$0.569_{\pm 0.002}$	$0.929_{\pm 0.005}$	$\underline{0.547_{\pm 0.000}}$	$\underline{0.357_{\pm 0.000}}$	$0.207_{\pm 0.000}$	$0.193_{\pm 0.001}$	$0.170_{\pm 0.000}$	$0.798_{\pm 0.024}$	$0.759_{\pm 0.000}$
EMGC^{2}F	$0.697_{\pm 0.000}$	$0.582_{\pm 0.000}$	$0.939_{\pm 0.000}$	$0.175_{\pm 0.000}$	$0.123_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.123_{\pm 0.000}$	$0.134_{\pm 0.000}$	$0.910_{\pm 0.000}$	$0.833_{\pm 0.000}$
WTNNM	$\underline{0.975_{\pm 0.000}}$	$0.513_{\pm 0.000}$	$\underline{0.980_{\pm 0.003}}$	$0.506_{\pm 0.001}$	$0.245_{\pm 0.000}$	$0.582_{\pm 0.000}$	$\underline{0.331_{\pm 0.003}}$	$\underline{0.315_{\pm 0.012}}$	$0.983_{\pm 0.003}$	$0.827_{\pm 0.006}$
LTBPL	$0.967_{\pm 0.000}$	$0.623_{\pm 0.000}$	$0.950_{\pm 0.000}$	$\mathbf{0.549}_{\pm 0.000}$	$0.256_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.037_{\pm 0.000}$	$0.161_{\pm 0.000}$	$\mathbf{0.986}_{\pm 0.000}$	$\underline{0.896_{\pm 0.000}}$
EMVGC-LG	$0.750_{\pm 0.018}$	$0.603_{\pm 0.003}$	$0.905_{\pm 0.000}$	$0.470_{\pm 0.000}$	$0.256_{\pm 0.027}$	$0.315_{\pm 0.002}$	$0.187_{\pm 0.000}$	$0.271_{\pm 0.000}$	$0.875_{\pm 0.015}$	$0.717_{\pm 0.047}$
MVBGC-NFF	$0.646_{\pm 0.036}$	$0.427_{\pm 0.018}$	$0.717_{\pm 0.007}$	$0.512_{\pm 0.018}$	$0.313_{\pm 0.010}$	$0.190_{\pm 0.046}$	$0.184_{\pm 0.002}$	$0.198_{\pm 0.013}$	$0.851_{\pm 0.018}$	$0.623_{\pm 0.032}$
DealMVC	$0.515_{\pm 0.000}$	$0.472_{\pm 0.000}$	$0.644_{\pm 0.000}$	$0.545_{\pm 0.000}$	$0.230_{\pm 0.000}$	$\underline{0.644_{\pm 0.000}}$	$0.122_{\pm 0.000}$	$0.193_{\pm 0.000}$	$0.350_{\pm 0.000}$	$0.645_{\pm 0.000}$
SCMVC	$0.473_{\pm 0.053}$	$0.436_{\pm 0.004}$	$0.800_{\pm 0.019}$	$0.495_{\pm 0.044}$	$0.380_{\pm 0.078}$	$0.235_{\pm 0.011}$	$0.236_{\pm 0.004}$	$0.125_{\pm 0.001}$	$0.642_{\pm 0.028}$	$0.454_{\pm 0.025}$
DIVIDE	$0.621_{\pm 0.024}$	$0.409_{\pm 0.020}$	$0.932_{\pm 0.002}$	$0.507_{\pm 0.014}$	$0.252_{\pm 0.027}$	$0.130_{\pm 0.014}$	$0.195_{\pm 0.007}$	$0.213_{\pm 0.003}$	$0.846_{\pm 0.011}$	$0.484_{\pm 0.040}$
OMVCDR	$0.658_{\pm 0.000}$	$0.329_{\pm 0.000}$	$0.839_{\pm 0.000}$	$0.536_{\pm 0.000}$	$0305_{\pm 0.000}$	$0.356_{\pm 0.000}$	$0.116_{\pm 0.000}$	$0.186_{\pm 0.000}$	$0.773_{\pm 0.000}$	$0.475_{\pm 0.000}$
Ours	$\mathbf{0.980}_{\pm 0.000}$	$\mathbf{0.812}_{\pm 0.000}$	$\mathbf{0.994}_{\pm 0.000}$	$0.496_{\pm 0.000}$	$\mathbf{0.405}_{\pm 0.000}$	$\mathbf{0.768}_{\pm 0.000}$	$\mathbf{0.423}_{\pm 0.000}$	$\mathbf{0.603}_{\pm 0.000}$	$\underline{0.984_{\pm 0.000}}$	$\mathbf{0.913}_{\pm 0.000}$
Purity										
SC(best)	$0.671_{\pm 0.004}$	$0.596_{\pm 0.000}$	$0.836_{\pm 0.001}$	$0.546_{\pm 0.000}$	$0.367_{\pm 0.011}$	$0.338_{\pm 0.000}$	$0.183_{\pm 0.002}$	$0.289_{\pm 0.000}$	$0.851_{\pm 0.012}$	$0.612_{\pm 0.000}$
GMC	$0.661_{\pm 0.000}$	$0.885_{\pm 0.000}$	$0.851_{\pm 0.000}$	$0.202_{\pm 0.000}$	$0.584_{\pm 0.000}$	-	$0.106_{\pm 0.000}$	$0.230_{\pm 0.000}$	$0.715_{\pm 0.000}$	$0.790_{\pm 0.000}$
CGD	$0.549_{\pm 0.003}$	$0.868_{\pm 0.001}$	$0.849_{\pm 0.004}$	$0.622_{\pm 0.000}$	$0.746_{\pm 0.000}$	$0.407_{\pm 0.000}$	$0.247_{\pm 0.001}$	$0.274_{\pm 0.002}$	$0.707_{\pm 0.023}$	$0.852_{\pm 0.000}$
EMGC^{2}F	$0.673_{\pm 0.000}$	$\underline{0.920_{\pm 0.000}}$	$0.876_{\pm 0.000}$	$0.308_{\pm 0.000}$	$0.603_{\pm 0.000}$	$0.202_{\pm 0.000}$	$0.197_{\pm 0.000}$	$0.270_{\pm 0.000}$	$0.842_{\pm 0.000}$	$0.914_{\pm 0.000}$
WTNNM	$\underline{0.979_{\pm 0.000}}$	$0.868_{\pm 0.000}$	$\underline{0.944_{\pm 0.004}}$	$0.600_{\pm 0.000}$	$0.666_{\pm 0.000}$	$\underline{0.745_{\pm 0.000}}$	$\underline{0.399_{\pm 0.013}}$	$\underline{0.441_{\pm 0.014}}$	$0.958_{\pm 0.014}$	$0.871_{\pm 0.004}$
LTBPL	$0.939_{\pm 0.000}$	$0.872_{\pm 0.000}$	$0.924_{\pm 0.000}$	$0.621_{\pm 0.000}$	$0.646_{\pm 0.000}$	$0.202_{\pm 0.000}$	$0.124_{\pm 0.000}$	$0.268_{\pm 0.000}$	$\underline{0.970_{\pm 0.000}}$	$\underline{0.943_{\pm 0.000}}$
EMVGC-LG	$0.724_{\pm 0.018}$	$0.875_{\pm 0.000}$	$0.793_{\pm 0.001}$	$0.573_{\pm 0.000}$	$0.652_{\pm 0.001}$	$0.512_{\pm 0.007}$	$0.266_{\pm 0.000}$	$0.376_{\pm 0.004}$	$0.767_{\pm 0.019}$	$0.812_{\pm 0.024}$
MVBGC-NFF	$0.599_{\pm 0.045}$	$0.809_{\pm 0.011}$	$0.503_{\pm 0.014}$	$0.595_{\pm 0.030}$	$0.652_{\pm 0.012}$	$0.415_{\pm 0.039}$	$0.266_{\pm 0.006}$	$0.317_{\pm 0.011}$	$0.723_{\pm 0.032}$	$0.729_{\pm 0.028}$
DealMVC	$0.596_{\pm 0.000}$	$0.824_{\pm 0.000}$	$0.244_{\pm 0.000}$	$0.629_{\pm 0.000}$	$0.588_{\pm 0.000}$	$0.244_{\pm 0.000}$	$0.173_{\pm 0.000}$	$0.315_{\pm 0.000}$	$0.125_{\pm 0.000}$	$0.667_{\pm 0.000}$
SCMVC	$0.394_{\pm 0.017}$	$0.813_{\pm 0.001}$	$0.612_{\pm 0.037}$	$\underline{0.630_{\pm 0.026}}$	$\underline{0.771_{\pm 0.080}}$	$0.465_{\pm 0.013}$	$0.287_{\pm 0.006}$	$0.254_{\pm 0.000}$	$0.460_{\pm 0.028}$	$0.542_{\pm 0.012}$
DIVIDE	$0.600_{\pm 0.027}$	$0.773_{\pm 0.044}$	$0.866_{\pm 0.009}$	$0.599_{\pm 0.023}$	$0.630_{\pm 0.026}$	$0.364_{\pm 0.009}$	$0.256_{\pm 0.010}$	$0.313_{\pm 0.005}$	$0.746_{\pm 0.025}$	$0.594_{\pm 0.043}$
OMVCDR	$0.648_{\pm 0.000}$	$0.811_{\pm 0.000}$	$0.706_{\pm 0.000}$	$0.612_{\pm 0.000}$	$0.677_{\pm 0.000}$	$0.594_{\pm 0.000}$	$0.185_{\pm 0.000}$	$0.311_{\pm 0.000}$	$0.623_{\pm 0.000}$	$0.624_{\pm 0.000}$
Ours	$\mathbf{0.982}_{\pm 0.000}$	$\mathbf{0.976}_{\pm 0.000}$	$\mathbf{0.993}_{\pm 0.000}$	$\mathbf{0.727}_{\pm 0.000}$	$\mathbf{0.795}_{\pm 0.000}$	$\mathbf{0.870}_{\pm 0.000}$	$\mathbf{0.750}_{\pm 0.000}$	$\mathbf{0.828}_{\pm 0.000}$	$\mathbf{0.977}_{\pm 0.000}$	$\mathbf{0.947}_{\pm 0.000}$

experiments. We tune their parameters according to the suggestions in original papers, and repeat each algorithm ten times with the optimal parameters to report the average performance and standard deviations. For our method, the trade-off parameters α is tuned in the range of $[0.0001, 1]$, β is tuned in the range of $[0.1, 1000]$, λ is tuned in the range of $[0.1, 1000]$, and the weight ω is tuned in the range of $(0, 50]$. Without loss of generality, we use the same graph construction algorithm as GMC [33] to generate the initial view-specific graphs.

B. Performance Evaluation

1) Performance Comparison: Table I demonstrates the quantitative clustering results of thirteen methods across ten real benchmark datasets. Note that the proposed method and some competing methods (including GMC, EMGC^{2}F, LTBPL and OMVCDR) are deterministic and produce consistent outputs across multiple runs, leading to a standard deviation of zero. The same phenomenon can be observed in the original papers of competing methods [24], [33], [38]. Besides, the released source code of DealMVC has a fixed random seed, making the outputs reproducible and resulting in zero deviation. We can draw the following observations.

a) Full graph-based methods demonstrates superior performance compared to anchor graph-based methods. This advantage arises from the limitations inherent in simplified data representation of anchor graph. Such representations are constrained by the representation

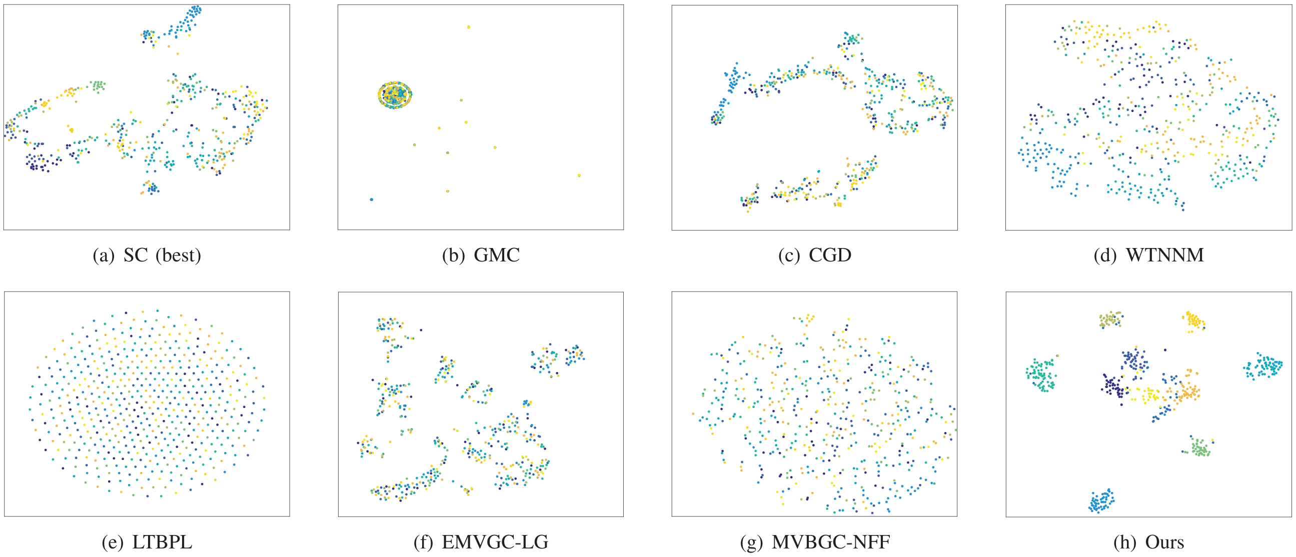

(a) SC (best) (b) GMC (c) CGD (d) WTNNM (e) LTBPL (f) EMVGC-LG (g) MVBGC-NFF (h) Ours

Fig. 3. T-SNE visualization of the consensus spectral representations from different methods on the Caltech3 dataset. The visualized representation of our method is the clustering probability indicator **F** in our learning model.

capacity of anchors, thereby fail to explore the comprehensive relationships among the data. For instance, on the BDGP and CCV datasets, our method outperforms EMVGC-LG and MVBGC-NFF by 38.1% and 46.2%, as well as 17.2% and 18.7% in terms of the ACC metric, respectively.

b) Tensor-based full graph methods generally outperform matrix-based full graph methods in most cases. This suggests that pursuing global low-rank structures among multiple affinity graphs is advantageous for enhancing clustering performance. For example, WTNNM, LTBPL, and our BPMLD all learn affinity full graphs in a tensor, achieving top-two performance in most cases. Although WTNNM and LTBPL integrate multiple priors into their learning processes, they still fail to achieve optimally refined affinity graphs for clustering. This may be because they rely on implicit learning objectives, which are insufficient for addressing issues related to inadequate prior information. For bacterial classification in the Prokaryotic dataset that has complicated underlying data distributions and significant differences among text, proteome, and genomic views, our method outperforms WTNNM and LTBPL by 17.2% and 11.1% in terms of the ACC metric, respectively.

c) Our BPMLD achieves the best performance in terms of three metrics against all baseline algorithms in most circumstances, which presents its superiority in multi-view datasets. Compared with deep learning-based methods, our method has deterministic optimization procedures and is easy to deploy. Moreover, our method fully leverages the spatial structural information among views. For example, our method achieves an ACC value of 61.7% on the Wiki_fea dataset, while DealMVC, SCMVC, and DIVIDE obtain 59.3%, 59.8%, and 54.5%, respectively.

d) Compared with other learning-based methods, our method establishes a bidirectional learning pipeline and introduces clustering confidence to refine graph learning, and thus effectively aligns affinity graphs with underlying data structure. For object recognition (*e.g.*, Caltech3 and 100Leaves datasets), our method outperforms GMC by 40.4% and 16.5% in terms of the ACC metric. For human face recognition (*e.g.*, ORL database) and scene recognition (*e.g.*, MSRCv1 database), our method surpasses CGD by 31.5% and 9.5%. By introducing the probability indicators, our method can well address sample uncertainty problem in various applications.

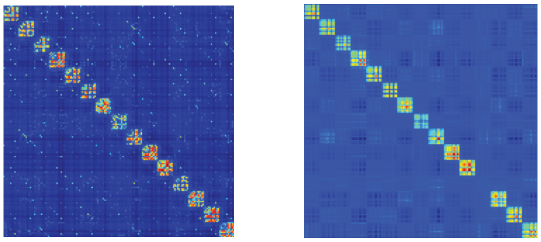
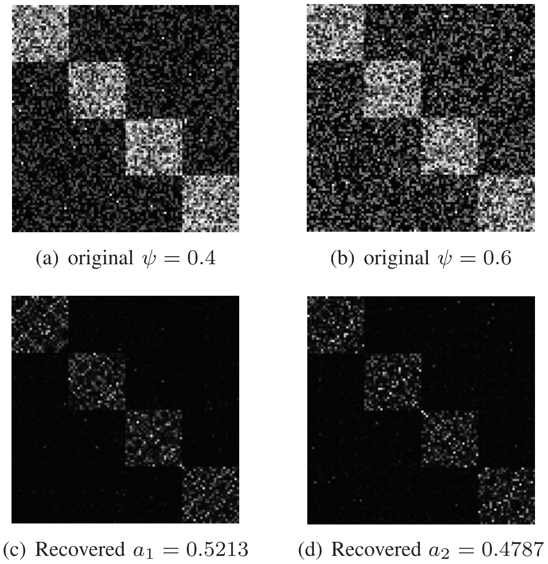

(a) Consensus graph (b) Confidence matrix

Fig. 4. Visualization of the consensus affinity graph and confidence matrix learned by our BPMLD on the Yale dataset.

To provide an intuitive comparison with the aforementioned competitors, we plot their learned consensus spectral embeddings in Fig. 3 using the t-SNE technique [44], where EMGC^{2}F and DealMVC are ignored since they directly produce discrete cluster indicators. We observe that our BPMLD learns more discriminative representations for the Caltech3 dataset.

Additionally, a block-diagonal structure in the refined affinity graph is desirable for graph-based clustering methods, ensuring that intra-cluster samples are densely connected while inter-cluster samples are disconnected. We plot the consensus affinity graph and confidence matrix learned from the Yale dataset in Fig. 4. The clear block-diagonal structures observed demonstrate the effectiveness of our proposed clustering confidence-driven strategy.



(a) original $\psi = 0.4$

(b) original $\psi = 0.6$

(c) Recovered $a_1 = 0.5213$

(d) Recovered $a_2 = 0.4787$

Fig. 5. Toy example with two views of varying noise levels. The upper row displays the similarity of samples on two views, each with a different level of noise. The lower row presents the recovered graphs with different weights. *i.e.*a_1 = 0.5213 for view 1 and a_2 = 0.4787 for view 2.

2) Graph Structure Refinement: To provide an intuitive visual illustration of the capability of our BPMLD, we create a two-view synthetic toy dataset, where each view is represented as a 100×100 matrix containing four 25×25 diagonal blocks [45]. The toy dataset is shown in Fig. 5(a) and Fig. 5(b). The values within each block represent the pairwise similarity of samples within a cluster, while points outside the blocks represent noise. Points in the blocks are randomly generated with the range of 0 and 1, and noise is generated within the range of 0 and ψ, with ψ set to 0.4 and 0.6, respectively. To increase the challenge, we introduce 25 randomly generated noisy points with an intensity of 1. Fig. 5(c) and Fig. 5(d) are the corresponding affinity graphs for these views after optimization. Our method effectively learns the clean block diagonal structure from the noisy input graphs. The highlights in different locations of the two recovered graphs indicate the model's ability to integrate complementary information from both views. Furthermore, the method assigns appropriate weights to each view based on their clustering capacity. For example, our BPMLD assigns a weight of 0.5213 to the view with a noise level of $\psi = 0.4$, while it assigns a weight of 0.4787 to the view with a noise level of $\psi = 0.6$. This further demonstrates the efficacy of our BPMLD model in integrating information.

C. Computational Efficiency Analysis

1) Time Complexity Comparison: To analyze the computational efficiency of the competing and proposed methods, we display their time complexities in Table II, excluding deep learning-based methods. Here, d, V, c, n, k, $|E|$ present the data dimension, number of views, number of neighbors, number of samples, number of clusters, and number of edges, respectively. As can be observed, EMVGC-LG, MVBGC-NFF, and OMVCDR show the linear time complexity for the number of samples, while SC, GMC and EMGC^{2}F represent the quadratic time complexity for the number of samples. Besides, CGD and WTNNM exhibit cubic time complexity for the number of samples. Our BPMLD and LTBPL show the efficiency between the quadratic and cubic time complexity for the number of samples.

TABLE II

Comparison of Time Complexity

Methods	Detailed Time Complexity
SC	$\mathcal{O}(Vn^2d + Vn^2k)$
GMC	$\mathcal{O}(t_a(Vcn + Vn^2 + kn + kn^2) + Vcdn)$
CGD	$\mathcal{O}(Vn^2d + t_an^3 + n^2k)$
EMGC^{2}F	$\mathcal{O}(\lvert E\rvert + t_aVn^2)$
WTNNM	$\mathcal{O}(t_a(Vn^3 + nd + Vn^2log(n)) + n^2k)$
LTBPL	$\mathcal{O}(t_a(Vn^2 + kn^2 + t_bV^2n^2 + Vn^2log(n) + t_cVn^2))$
EMVGC-LG	$\mathcal{O}(t_a(n(mdV + m^3) + Vm^3) + nmk)$
MVBGC-NFF	$\mathcal{O}(t_a(nm(m + Vd + Vd^2) + m^2Vd) + nmk)$
OMVCDR	$\mathcal{O}(t_aVd(nmd + md^2))$
Ours	$\mathcal{O}(t_a(Vn^2 + Vn^2log(n) + t_bn^2k + V^3))$

t_a presents the iteration number of outer loop, while t_b and t_c indicate the iteration number in sub-problems. d, V, c, n, k, $|E|$ denote the dimension of data, number of views, number of neighbors, number of samples, number of clusters, and number of edges respectively.

2) Running Time Comparison: The proposed BPMLD represents the geometric structure among samples using full graphs. This requires higher computational costs and longer optimization times. Table III presents the running times across ten datasets. Compared to some competing methods, the proposed BPMLD may take longer runtimes to achieve improved performance. Undoubtedly, larger datasets require longer running times. The EMGC^{2}F algorithm exhibits the shortest runtimes in most cases due to its relatively simple calculations. However, these heuristic algorithms may not provide competitive clustering performance. On the other hand, deep learning-based approaches often require high computational costs due to their complicated optimization processes. For instance, SCMVC method takes substantial time on most datasets. The proposed BPMLD, WTNNM and LTBPL utilize full tensor graphs to explore the spatial correlations among multiple views, making them highly time-consuming. Nonetheless, the proposed BPMLD separates graph construction from graph learning, resulting in lower computational costs compared to WTNNM and LTBPL on the Wiki_fea and BDGP datasets.

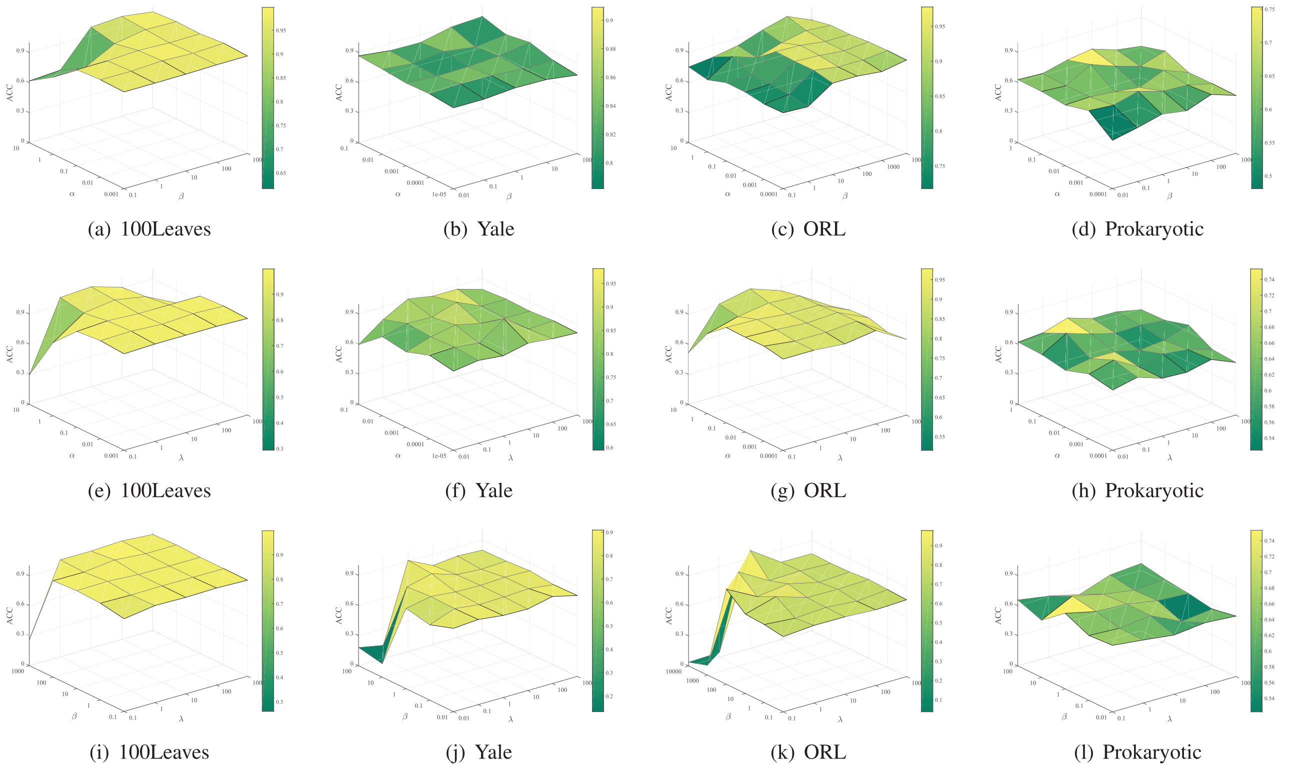
D. Model Analysis

1) Parameter Sensitivity Analysis: The proposed BPMLD has three trade-off parameters, *i.e.*, α, β, and λ. To thoroughly investigate their parameter sensitivity, we conduct sensitivity analyses on the 100Leaves, Yale, ORL, and Prokaryotic datasets. The clustering results of the ACC metric are illustrated in Fig. 6. As can be seen, BPMLD exhibits distinct parameter sensitivity across datasets due to different properties. Nevertheless, the broad optimal regions across these datasets indicate BPMLD's robustness to parameter selection in various clustering scenarios. For plant image identification in the 100Leaves dataset, BPMLD performs optimally when $\alpha \in [0.001, 1]$, and shows little sensitivity to β and λ.

TABLE III

RUNNING TIME COMPARISON OF DIFFERENT METHODS ON TEN MULTI-VIEW DATASETS. THE 1ST/2ND BEST RESULTS ARE MARKED IN BOLD AND UNDERLINE. (MEASURED BY SECOND)

Datasets	Yale	Caltech7	100Leaves	Wiki_fea	Prokaryotic	BDGP	CCV	Caltech3	ORL	MSRCv1
SC(best)	4.38	37.74	30.47	6.46	1.67	23.51	42.44	2.49	10.16	2.69
GMC	$\underline{0.25}$	6.04	$\underline{6.41}$	29.86	0.93	**2.14**	13.38	$\underline{0.73}$	$\underline{0.75}$	$\underline{0.45}$
CGD	0.86	39.58	86.51	136.02	2.83	89.73	1729.7	4.91	5.51	0.89
EMGC^{2}F	**0.05**	**0.88**	**0.24**	**0.68**	**0.08**	4.15	**4.44**	**0.06**	**0.22**	**0.26**
WTNNM	34.07	547.23	290.18	1002.98	116.42	885.37	5933.85	25.62	122.14	41.08
LTBPL	6.37	255.28	192.86	1054.50	41.47	540.89	10639.00	38.50	15.19	10.32
EMVGC-LG	4.27	7.99	48.77	17.58	2.15	16.12	33.24	7.57	8.32	5.87
MVBGC-NFF	6.60	$\underline{4.04}$	10.38	$\underline{1.79}$	$\underline{0.45}$	$\underline{2.69}$	$\underline{9.87}$	0.99	17.26	0.56
DealMVC	55.78	378.95	875.71	219.96	72.07	407.85	983.50	83.23	165.41	97.92
SCMVC	249.96	1297.88	5317.31	1196.42	340.96	235.24	7164.74	597.98	839.67	228.94
DIVIDE	216.03	230.19	216.88	218.69	193.52	228.44	402.57	173.06	175.39	190.73
OMVCDR	1.68	25.98	75.58	39.67	3.36	32.25	178.18	7.15	7.43	0.55
Ours	9.15	307.54	310.48	500.37	129.15	338.94	6218.5	226.88	23.30	7.47

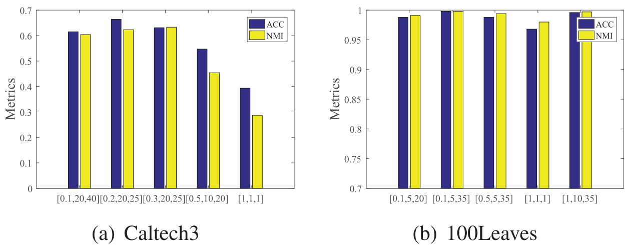


(a) 100Leaves (b) Yale (c) ORL (d) Prokaryotic

(e) 100Leaves (f) Yale (g) ORL (h) Prokaryotic

(i) 100Leaves (j) Yale (k) ORL (l) Prokaryotic

Fig. 6. Parameter sensitivity analyses: ACC scores obtained using different trade-off parameters α, β and λ across four datasets.



(a) Caltech3 (b) 100Leaves

Fig. 7. Parameter sensitivity analysis: Performance in terms of ACC and NMI with varying weighting factors ω on the three views of the Caltech3 and 100Leaves benchmark datasets.

For human face recognition in the ORL and Yale datasets, a higher value of β (like 10, and 100) generally yields a higher identification rate (as shown in Fig. 6(c)). However, when λ is set to a low value (like 0.01, and 0.1), a high value of β degrades the model performance (as shown in Fig. 6(j) and 6(k)). This is because human face recognition datasets are sensitive to noise [46]. A low value of λ fails to remove noise effectively, and thus results in the failure of tensor graph learning to explore the spatial structure among views. For bacterial classification in the Prokaryotic dataset, our method demonstrates stability when $\alpha \in [0.001, 1]$, with slight sensitivity to β and λ. Overall, these results highlight BPMLD's robustness across various real-world applications. Practically, tuning α within $[0.0001, 1]$, β within $[0.1, 1000]$, and λ within $[0.1, 1000]$ is usually feasible for achieving promising performance across different clustering scenarios. In addition, to study the weighting factor ω, we present the ω's varying choices on the Caltech3 and 100Leaves datasets in Fig. 7. We record the clustering performances in terms of ACC and NMI at different weight values. The results indicate that the weighting factor is crucial for improving clustering performance.

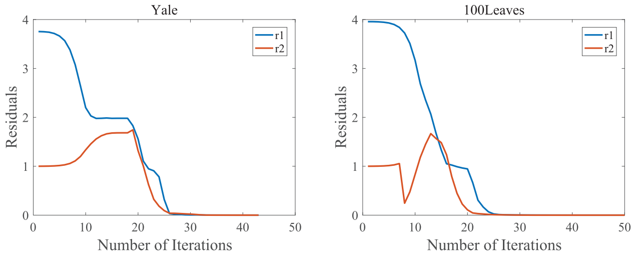


Fig. 8. Empirical convergence of our BPMLD model on the Yale and 100Leaves benchmark datasets.

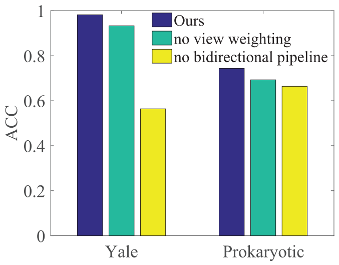


Fig. 9. Ablation study on two benchmark datasets. The clustering results in terms of the ACC metric for the proposed BPMLD and its two degraded models on the Yale and prokaryotic datasets.

2) Convergence Study: While providing a theoretical convergence proof for the proposed optimization algorithm under the ADMM framework is challenging, it is essential to analyze its empirical convergence on real-world benchmark datasets. According to Eq. (27), we plot the residual value curves across the Yale and 100Leaves datasets in Fig. 8. The results demonstrate that the proposed algorithm exhibits robust convergence behavior. Notably, the residuals become sufficiently small after approximately 25 iterations, indicating the high efficiency of the method. Overall, the empirical convergence analysis presented in Fig. 8offers valuable insights into the practical performance of the proposed algorithm, showcasing its reliability in converging within a reasonable number of iterations across diverse benchmark scenarios.

3) Ablation Study: The bidirectional refining pipeline is a primary contribution of this paper, and the view weighting scheme may also impact clustering performance. To further demonstrate the effectiveness of these two techniques *i.e.,* adaptive view weighting and the bidirectional refining pipeline, we present the results of the ablation study in Fig. 9. We conduct ablation experiments on the Yale and Prokaryotic benchmark datasets. We devise two degraded models: ‘no view weighting’ and ‘no bidirectional pipeline’. The ‘no view weighting’ model assigns equal weights to all views, while the ‘no bidirectional pipeline’ model is formulated as follows:

$$\begin{aligned}&\min_{\mathbf{C}^v,\mathbf{E}^v} \sum_{v=1}^{V} \beta \|\mathcal{C}\|_{\omega,\circledast} + \lambda \|\mathbf{E}^v\|_1\\&s.t.\ \mathbf{S}^v = \mathbf{C}^v + \mathbf{E}^v, \mathbf{C}^v \geq 0, \mathbf{C}^v\mathbf{1} = \mathbf{1},\\&\quad \mathcal{C} = \Psi(\mathbf{C}^1, \mathbf{C}^2, \dots, \mathbf{C}^V). \qquad (28)\end{aligned}$$

This model requires computing the affinity matrix using Eq. (2) and performing additional spectral clustering to obtain the final clustering results. From the experimental results in Fig. 9, we can conclude that the proposed method significantly outperforms both degraded models. This finding highlights the effectiveness of the proposed strategy and model.

VI. Conclusion

This paper proposed a bidirectional probabilistic multi-graph learning and decomposition (BPMLD) method for multi-view clustering. BPMLD establishes a bidirectional refining pipeline between graph learning and the clustering indicator generation. Specifically, we introduced a novel multi-graph probability decomposition module to generate clustering probability indicators in a single step. A clustering confidence matrix was constructed from these clustering probability indicators to reflect the clustering confidence among samples. Using this matrix, we proposed a clustering confidence driven graph learning module. These two modules are seamlessly integrated within a framework and alternately optimized to improve clustering performance. Extensive experiments on ten real-world benchmark datasets demonstrated the superiority of the proposed BPMLD model compared to state-of-the-art methods. Our future work will focus on employing compression techniques to address the scalability limitations.

References

[1] T. Yu, J. Meng, and J. Yuan, “Multi-view harmonized bilinear network for 3D object recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 186–194.

[2] A. Huang, L. Li, L. Zhang, Y. Niu, T. Zhao, and C.-W. Lin, “Multi-view graph embedding learning for image co-segmentation and co-localization,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 6, pp. 4942–4956, Jun. 2024.

[3] Y. Zhang, S. Yan, L. Zhang, and B. Du, “Fast projected fuzzy clustering with anchor guidance for multimodal remote sensing imagery,” *IEEE Trans. Image Process.*, vol. 33, pp. 4640–4653, 2024.

[4] Y. Zhang, S. Yan, X. Jiang, L. Zhang, Z. Cai, and J. Li, “Dual graph learning affinity propagation for multimodal remote sensing image clustering,” *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5521713.

[5] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, “Diversity-induced multi-view subspace clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, May 2015, pp. 586–594.

[6] Y. Zhang, J. Wu, C. Zhou, Z. Cai, J. Yang, and P. S. Yu, “Multi-view fusion with extreme learning machine for clustering,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 5, pp. 1–23, Sep. 2019.

[7] X. Cai, F. Nie, and H. Huang, “Multi-view K-means clustering on big data,” in *Proc. 23th Int. Joint Conf. Artif. Intell.*, May 2013, pp. 2598–2604.

[8] D. Guo, J. Zhang, X. Liu, Y. Cui, and C. Zhao, “Multiple kernel learning based multi-view spectral clustering,” in *Proc. 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 3774–3779.

[9] J. Xu, J. Han, and F. Nie, “Discriminatively embedded K-means for multi-view clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5356–5364.

[10] H. Gao, F. Nie, X. Li, and H. Huang, “Multi-view subspace clustering,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4238–4246.

[11] Y. Wen et al., “Efficient multi-view graph clustering with local and global structure preservation,” in *Proc. 31st ACM Int. Conf. Multimedia*. New York, NY, USA: ACM, Oct. 2023, pp. 3021–3030.

[12] Y. Chen, S. Wang, C. Peng, Z. Hua, and Y. Zhou, “Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering,” *IEEE Trans. Image Process.*, vol. 30, pp. 4022–4035, 2021.

[13] L. Li, J. Zhang, S. Wang, X. Liu, K. Li, and K. Li, “Multi-view bipartite graph clustering with coupled noisy feature filter,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12842–12854, Dec. 2023.

[14] F. Nie, J. Li, and X. Li, “Self-weighted multiview clustering with multiple graphs,” in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2564–2570.

[15] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1261–1270, Mar. 2019.

[16] Q. Gao, W. Xia, Z. Wan, D. Xie, and P. Zhang, "Tensor-SVD based graph learning for multi-view subspace clustering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 3930–3937.

[17] Y. Liang, D. Huang, and C. Wang, "Consistency meets inconsistency: A unified graph learning framework for multi-view clustering," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 1204–1209.

[18] S. Huang, I. W. Tsang, Z. Xu, and J. Lv, "Measuring diversity in graph learning: A unified framework for structured multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5869–5883, Dec. 2022.

[19] X. Xiao and Y.-J. Gong, "Accurate complementarity learning for graph-based multiview clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 16106–16118, Nov. 2024.

[20] D. Wang, S. Han, Q. Wang, L. He, Y. Tian, and X. Gao, "Pseudo-label guided collective matrix factorization for multiview clustering," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8681–8691, Sep. 2022.

[21] C. Liu, S. Wu, R. Li, D. Jiang, and H.-S. Wong, "Self-supervised graph completion for incomplete multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 9394–9406, Sep. 2023.

[22] H. Wang, J. Peng, W. Qin, J. Wang, and D. Meng, "Guaranteed tensor recovery fused low-rankness and smoothness," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10990–11007, Sep. 2023.

[23] S. Yu et al., "Sparse low-rank multi-view subspace clustering with consensus anchors and unified bipartite graph," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 1438–1452, Jan. 2025.

[24] M. Chen, C. Wang, and J. Lai, "Low-rank tensor based proximity learning for multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 5076–5090, May 2023.

[25] C. Tang et al., "CGD: Multi-view clustering via cross-view graph diffusion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 5924–5931.

[26] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 925–938, Apr. 2020.

[27] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, Aug. 2011.

[28] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 3842–3849.

[29] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, Jan. 2001, pp. 849–856.

[30] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM J. Comput.*, vol. 1, no. 2, pp. 146–160, 1972.

[31] S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[32] C. Zhang, H. Li, W. Lv, Z. Huang, Y. Gao, and C. Chen, "Enhanced tensor low-rank and sparse representation recovery for incomplete multi-view clustering," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 9, pp. 11174–11182.

[33] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, May 2019.

[34] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Adv. Neural Inf. Process. Syst.*, Sep. 2011, pp. 1–9.

[35] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Math. Program.*, vol. 155, no. 1, pp. 57–79, Jul. 2016.

[36] Y. Qin, G. Feng, Y. Ren, and X. Zhang, "Consistency-induced multiview subspace clustering," *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 832–844, Feb. 2023.

[37] X. Yang et al., "DealMVC: Dual contrastive calibration for multi-view clustering," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 337–346.

[38] X. Wan et al., "One-step multi-view clustering with diverse representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 3, pp. 5774–5786, Mar. 2025.

[39] D. Wu, J. Lu, F. Nie, R. Wang, and Y. Yuan, "EMGC^{2}F: Efficient multi-view graph clustering with comprehensive fusion," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 3566–3572.

[40] Y. Wen et al., "Unpaired multi-view graph clustering with cross-view structure matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 16049–16063, Nov. 2024.

[41] S. Wang et al., "Fast parameter-free multi-view subspace clustering with consensus anchor guidance," *IEEE Trans. Image Process.*, vol. 31, pp. 556–568, 2022.

[42] S. Wu et al., "Self-weighted contrastive fusion for deep multi-view clustering," *IEEE Trans. Multimedia*, vol. 26, pp. 9150–9162, 2024.

[43] Y. Lu, Y. Lin, M. Yang, D. Peng, P. Hu, and X. Peng, "Decoupled contrastive multi-view clustering with high-order random walks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, Sep. 2024, pp. 14193–14201.

[44] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[45] W. K. Wong, N. Han, X. Fang, S. Zhan, and J. Wen, "Clustering structure-induced robust multi-view graph recovery," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3584–3597, Oct. 2020.

[46] F. Wang et al., "The devil of face recognition is in the noise," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 765–780.

Xinxin Wang is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, University of Macau. His research interests include multi-view learning and hyperspectral image processing.

Yongshan Zhang (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in computer science from China University of Geosciences, Wuhan, China, in 2014 and 2019, respectively.

She has been a member at the BDSC Laboratory, University of Illinois at Chicago, and the VIP Laboratory, University of Macau. She is currently an Associate Professor with the School of Computer Science, China University of Geosciences. Her current research interests include machine learning and remote sensing image analysis.

Yicong Zhou (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Tufts University, Medford, MA, USA.

He is a Professor at the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, and artificial intelligence.

Dr. Zhou is a fellow of the Society of Photo-Optical Instrumentation Engineers (SPIE) and was recognized as one of "Highly Cited Researchers" in 2020, 2021, 2023, and 2024. He serves as a Senior Area Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.