Fine-Grained Graph Learning for Multi-View Subspace Clustering

Yidi Wang, Xiaobing Pei, and Haoxi Zhan

Abstract—Multi-view subspace clustering (MSC) is a popular unsupervised method by integrating heterogeneous information to reveal the intrinsic clustering structure hidden across views. Usually, MSC methods use graphs (or affinity matrices) fusion to learn a common structure, and further apply graph-based approaches to clustering. Despite progress, most of the methods do not establish the connection between graph learning and clustering. Meanwhile, conventional graph fusion strategies assign coarse-grained weights to combine multi-graph, ignoring the importance of local structure. In this article, we propose a fine-grained graph learning framework for multi-view subspace clustering (FGL-MSC) to address these issues. To utilize the multi-view information sufficiently, we design a specific graph learning method by introducing graph regularization and a local structure fusion pattern. The main challenge is how to optimize the fine-grained fusion weights while generating the learned graph that fits the clustering task, thus making the clustering representation meaningful and competitive. Accordingly, an iterative algorithm is proposed to solve the above joint optimization problem, which obtains the learned graph, the clustering representation, and the fusion weights simultaneously. Extensive experiments on eight real-world datasets show that the proposed framework has comparable performance to the state-of-the-art methods.

Index Terms—Multi-view learning, subspace clustering, graph learning, joint optimization.

I. INTRODUCTION

A

effective unsupervised learning method, multi-view clustering has attracted wide attention in recent years [1], [2]. Among this literature, the graph-based clustering methods are brought forward and have achieved remarkable improvements in a series of applications. Graph-based approaches have been widely adopted in the early period of single-view analysis. The process can be interpreted as a two-stage algorithm: constructing graphs from features and using graph Laplacian to solve a quadratic optimization problem [3]. These methods can generally converge to a global optimal solution and identify arbitrary shapes of clusters. Despite progress, graph-based clustering methods are often limited by the quality of the generated graphs. In this regard, many graph construction methods are proposed. The classic ones are the fully-connected graph constructed by the Gaussian kernel function and the k-nearest neighbor graph which only retains part of edges [4]. In addition, there are local discriminant graph [5], pairwise similarity graph [6], adaptive neighbors graph [7], and so on.

A different graph construction approach based on subspace learning is proposed by [8], and it further developed a new subspace clustering pattern e.g. [9], [10], [11], [12], [13]. Subspace clustering methods can capture more global manifold information by using a self-expression matrix, which can deal with noise and missing points [14]. With the widespread use of neural networks, more works have extended subspace clustering to deep learning and achieved better results than traditional models e.g. [15], [16], [17], [18], [19].

Through the in-depth study of clustering, more and more researchers find that single-view data cannot completely describe the internal structure of data. Therefore, multi-view clustering methods are gradually developed. [20] and [21] suggest that the multi-view clustering models are generally based on consistency across views. [22] further concludes that the consensus principle and the complementary principle play important roles in the success of multi-view clustering. Specifically, the consensus principle means that each view admits a common clustering structure and the complementary principle means that each view contains different degrees of variation in the distribution of samples. Hence the approaches of learning one potential common subspace or graph structure and further deriving the cluster indicator are feasible by integrating multi-view information.

Practically, there are three prevailing schemes to cope with the integration of multi-view information. The first kind tries to strike a balance between consistency and inconsistency, by assigning weights for views to fuse rich information, such as the multi-view subspace clustering model [23], auto-weighted multiple graph learning model [24], multi-view graph learning model [25], multi-view non-negative embedding and spectral embedding model [26], and its constraint form [27]. The second kind is to decompose the original graph structure captured on a single view into two parts, that is, a common consensus graph and an independent graph, and optimize to obtain the graph structures and fusion weights of both parts, such as consistent and specific multi-view subspace clustering model [21], a consistency-induced multi-view subspace clustering model [28], and a unified graph alignment framework [29]. In addition, many tensor-based analytical methods have emerged in recent years.
Learning and clustering representation into account together.

2) We regard the unified graph learning process as an aggregation of local structures for each sample, which considers fine-grained weights to all samples for different views, alleviating the cross-talk of local structures. For the unified graph, a Frobenius norm is also introduced to control its sparsity.

3) We design an effective algorithm to solve the optimization of FGL-MSC, and theoretically show its convergence and computational complexity. The experiments on eight real-world datasets show that our method has comparable performance with the SOTA methods.

The rest of this article is structured as follows. Section II provides the notations and some related works. Section III presents the model formulation and its optimization algorithm. The experiments on benchmark datasets are shown in Section IV. And Section V concludes the article.

II. NOTATIONS AND RELATED WORKS

In this section, we first define the notations used throughout the article and briefly review three representative multi-view clustering models. The functional form of spectral clustering is also provided.

Notation: Given a multi-view data \( \mathcal{X} = \{X^1, \ldots, X^t\} \subseteq \mathbb{R}^{d \times n} \), \( t, n, d_v \) denote the number of views, the number of samples, and the feature dimension of \( X^v \), respectively. \( W = \{W^1, \ldots, W^t\} \) is the self-expression matrices generated from different subspaces. The clustering representation (or the input of clustering tasks) is denoted by \( F \).

For an arbitrary matrix \( A, A_{ij} \) denotes the \( ij \)-th element and \( A_i \) is the \( i \)-th column of \( A \). \( \lambda_1 \) and \( \lambda_2 \) denote the \( l_1 \)-norm and \( l_2 \)-norm of vector \( A \), respectively. \( \|A\|_1 \) denotes the transpose of \( A \), and \( \|A\|_F = \sqrt{\sum_{i,j} |A_{ij}|^2} \). The Frobenius norm of \( A \) is \( \|A\|_F = \sqrt{\sum_{i,j} |A_{ij}|^2} \). The Frobenius norm of vector \( A \) is \( \|A\|_2 \). \( \lambda \geq 0 \) means all elements in \( A \) are non-negative.

In addition, \( I \) denotes a column vector whose elements are all 1. \( I \) denotes an identity matrix with the proper size. All matrices are represented in capital letters.

A. Multi-View Subspace Clustering

Subspace clustering assumes that samples can be linearly represented by other samples lying in the same subspace, so as to mine the correlation between different data. Mathematically, given \( X \in \mathbb{R}^{d \times n} \) with \( n \) samples and \( d \) is the feature dimension. The subspace clustering can be formulated as [38]:

\[
\min_{W} \|X - XW\|_F^2 + \lambda \Omega(W),
\]

where \( W \in \mathbb{R}^{n \times n} \) is the self-expression matrix, \( \Omega(\cdot) \) is a regularized function, and \( \lambda > 0 \) is a balance parameter. The MVSC model [23] extended (1) to the multi-view problems. The specific form is:

\[
\min_{W^v, F} \|X^v - X^vW^v\|_F^2 + \lambda \sum_v Tr(F^T L^v F), \quad \text{s.t.} \quad W^v \geq 0, W^v 1 = 1, F^T F = I,
\]
where $L^v = D^v - W^v$ is the graph Laplacian matrix, and $D^v$ is the degree matrix with $D^v_{ii} = \sum_{j=1}^{c} W^v_{ij}$. (2), learn the self-expression matrix for each view, and then obtains the common clustering representation directly by the rank constraint term. The information integration of this method is reflected in the optimization of clustering representation $F$.

**B. Multi-Graph Fusion for Multi-View Spectral Clustering**

GFSC [39] proposes a new graph fusion mechanism to integrate different subspaces into one graph, and optimize clustering representation together:

\[
\begin{aligned}
\min_{W^v, S, F} & \sum_{v=1}^{n} \left( \|X^v - X^v W^v\|_F^2 + \alpha \|W^v\|_F^2 \right) \\
& + \beta \lambda_v \|W^v - S\|_F^2) + \gamma \text{Tr} \left( F^T L_S F \right) \\
\text{s.t.} & \quad W^v \geq 0, F^T F = I,
\end{aligned}
\]

(3)

where $S$ is a consensus graph, which approximates the self-expression matrix of each individual view and maintains an explicit clustering structure via the spectral clustering term. $L_S$ is the Laplacian matrix of $S$, with the same definition as (2). The graph fusion weight $\lambda_v$ is determined by the inverse distance between each individual graph $W^v$ and the consensus graph $S$. When the $\lambda$ is set to $\frac{1}{2}$, the GFSC model degrades to the MVSC model.

**C. Graph Learning for Multi-View Clustering**

Different from the above two integrating methods, the MVGL model [25] proposes a node-level graph learning method. The key in MVGL is a two-step graph learning process. Firstly, it calculates a set of single view graphs by adaptive neighbors graph [7] as follows:

\[
\begin{aligned}
\min_{W^v} & \sum_{i,j=1}^{n} W^v_{ij} \|X^v_i - X^v_j\|_2^2 + \lambda \|W^v\|_F^2 \\
\text{s.t.} & \quad W^v \geq 0, W^v_i 1 = 1.
\end{aligned}
\]

(4)

Then assuming that $G$ is the unified graph fused by all the $W^v$, it proposes a global graph learning function:

\[
\begin{aligned}
\min_{G, \alpha} & \sum_{i=1}^{n} \left( \|G_i - \sum_{v=1}^{t} \alpha^v_i W^v_i\|_2^2 \right) + \gamma \text{Tr} (F^T L G F) \\
\text{s.t.} & \quad \forall i, \sum_{v=1}^{t} \alpha^v_i = 1, \alpha^v_i \geq 0,
\end{aligned}
\]

where $\alpha$ is the fine-grained fusion weights, and $L_G$ denotes the Laplacian matrix of $G$. Based on (5), MVGL turns the local structure fusion into a least square approximation problem, and provides an efficient optimization algorithm.

**D. Reformulation of Spectral Clustering**

The aforementioned methods use the reformulation of spectral clustering to learn the graph and clustering representation simultaneously. Here we make a brief description of this popular technology.

Let $c$ denote the cluster number and $F \in \mathbb{R}^{n \times c}$ be the clustering representation vector. Given a graph Laplacian matrix $L = D - W$, where $W$ is the generated graph from data and $D$ is a degree matrix of $W$, [24] provides the functional form of spectral clustering to obtain $F$:

\[
\min \text{Tr}(F^T L F) \quad \text{s.t.} \quad F^T F = I.
\]

(6)

The optimal solution $F$ to the (6) is formed by the $c$ eigenvectors of $L$ corresponding to the $c$ smallest eigenvalues. Furthermore, given $F$, the construction of an optimal graph with $c$ connected components is also equivalent to (6) according to following equation [7]:

$$
\sum_{i=1}^{c} \sigma_i(L) = \min_{F^TF=I} \text{Tr}(F^T L F),
$$

(7)
where $\sigma_i(L)$ is the $i$-th smallest eigenvalue of $L$. Therefore, (6) is introduced into many multi-view clustering frameworks to participate in joint optimization as a rank constraint of graphs.

III. METHODOLOGY

In this section, we first derive the proposed model formulation, then present its optimization algorithm, and finally provide the overall analysis of the model.

A. Model Formulation

As shown in Fig. 2, FGL-MSC can be divided into three stages.

1) Refined Graph Learning: As aforementioned, the self-expression matrix $W^v$ obtained by (1) can capture global information but ignores the local structure. In recent works, the distance metric graphs are frequently used to complement the self-expression matrix in the description of the local structure [40], [41], [42]. The processed (or refined) graph shows stronger robustness than the original self-expression matrix. Inspired by these investigations, we design a two-part learner to extract useful structure from $\mathcal{V}$. The learner’s form can be described as:

$$\begin{align*}
\min_{\Omega_1(W^v, Z^v) + \lambda \Omega_2(G, Z^v, \alpha_v)} \\
\text{s.t. } W^v \geq 0, Z^v \geq 0, G \geq 0,
\end{align*}$$

where $G$ is the unified graph, $\alpha_v$ is fusion weights, and $Z^v$ denotes the refined graph. $\lambda > 0$ is a hyper-parameter.

In the first part of (8), we design a graph regularization term to ensure the retention of global information. Meanwhile, $Z^v$ is initialized by the adaptive neighbors graph [7], which extracts the local structure efficiently. In the same way as (4), The initialization of $Z^v$ can be described as follows:

$$\begin{align*}
\min_{Z^v} \sum_{i,j=1}^{t} Z^v_{ij} |X^v_i - X^v_j|^2_F + \gamma ||Z^v||_F^2 \\
\text{s.t. } Z^v \geq 0, \forall i, Z^v_i \mathbf{1} = 1,
\end{align*}$$

2) Graph Fusion: The second part of (8) is a graph fusion step. Generally, $\alpha$ to each view is often optimized independently, which depends a lot on the overall quality of its view. To address the rough weights assignment caused by this dependence, we adopt the fine-grained fusion weights proposed by [25].

To overcome the inconvenience of vector representation, We transform the tensor $Z$ at the data level:

$$\bar{Z} = \text{trans}(Z),$$

where the transformation function trans($Z$) can be described as $Z^T = [(Z^1)^T, \ldots, (Z^n)^T]^T$. The obtained $\bar{Z} = \{Z^1, \ldots, Z^n\} \in \mathbb{R}^{t \times n}$ denotes multi-view topology for all samples, which can be considered as a cross-view structural representation. Therefore, the specific formulation of (8) is as follows:

$$\begin{align*}
\min_{G, \mathcal{A}} \sum_{v=1}^{t} ||W^v - Z^v||_F^2 + \lambda \sum_{i=1}^{n} ||G^T_i - A^T_i \bar{Z}_i||_2^2 \\
\text{s.t. } \forall i, A^T_i \mathbf{1} = 1, A_i \geq 0, G \geq 0, G\mathbf{1} = \mathbf{1},
\end{align*}$$

where $A_{iv}$ is the weight for the $i$-th sample in the $v$-th view. To ensure that the weights are meaningful, we add a constraint that the sum of each column of $A$ is one. $A_i$ is optimized independently. $\lambda > 0$ is a trade-off hyper-parameter. This problem is a classical convex quadratic programming (QP), which can be solved by some QP optimizers. Our specific optimization algorithm will be described in detail later.

3) The Unified Model: Furthermore, taking advantage of the functional spectral clustering, we establish the connection between graph learning and clustering. The unified objective function combining two tasks can optimize them simultaneously. Considering (2) and (11), the joint optimization problem is as follows:

$$\begin{align*}
\min_{G, \mathcal{A}} \sum_{v=1}^{t} \left\{ ||X^v - X^v W^v||_F^2 + \alpha ||W^v - Z^v||_F^2 \right\} + \lambda \sum_{i=1}^{n} ||G^T_i - A^T_i \bar{Z}_i||_2^2 + \eta \text{Tr}(F^T L F) \\
\text{s.t. } \forall i, A^T_i \mathbf{1} = 1, A_i \geq 0, G \geq 0, G\mathbf{1} = \mathbf{1}, F^T F = I,
\end{align*}$$

where $\alpha > 0, \lambda > 0, \eta > 0$ are three trade-off hyper-parameters, and $L$ is the Laplacian matrix of $G$. Note that (12) is prone to yielding the trivial solution with respect to $G$, i.e., all non-zero elements are located on the diagonal of $G$, while $L = 0$. So we use the Frobenius norm of $G$ to smooth its elements.

Then our Fine-grained graph Learning for Multi-view Subspace Clustering (FGL-MSC) can be formulated as:

$$\begin{align*}
\min_{G, \mathcal{A}} \sum_{v=1}^{t} \left\{ ||X^v - X^v W^v||_F^2 + \alpha ||W^v - Z^v||_F^2 \right\} + ||W^v||_1 + \lambda \sum_{i=1}^{n} ||G^T_i - A^T_i \bar{Z}_i||_2^2 + \gamma ||G||_F^2 + \eta \text{Tr}(F^T L F) \\
\text{s.t. } \forall i, A^T_i \mathbf{1} = 1, A_i \geq 0, G \geq 0, G\mathbf{1} = \mathbf{1}, F^T F = I,
\end{align*}$$

where $\gamma > 0$ is also a hyper-parameter for the regularization of $G$. Note that $\gamma$ is difficult to be tuned, so we propose an effective algorithm to optimize $G$, which is shown in the next subsection. This model enjoys the following properties:

1) According to (10), The graph transformation operation transforms generalized sparse graphs into representations of nodes that contain high-level structural information. The point-level graph learning term reconstructs them as one sparse graph. Essentially, it’s a dimensional reduction from $n \times n \times t$ to $n \times n$, which is a different opinion from existing methods.

2) The fine-grained weights, i.e., $A$ in (11) can be seen as the reconstruction coefficient during the dimensional reduction. The constraint on this coefficient can ensure the structural feature selection and extraction capability so that the learned structural representation not only retains the latent consistency but also reduces the noise.
The joint framework (13) fulfills the tasks of subspace capturing, graph learning, clustering, and point-level weights learning in a single objective function, which enhances the connection between different tasks.

B. Model Optimization

We introduce an alternating iteration algorithm to optimize the problem (13). We decompose it as five subproblems and update them alternatively.

1) \( W^v \)-Subproblem: When \( Z^v, G, A, F \) are fixed, we can obtain the subproblem of (13) with respect to \( W^v \):

\[
\begin{align*}
\min_{W^v} & \quad \|X^v - X^vW^v\|_F^2 + \alpha \|W^v - Z^v\|_F^2 + \|W^v\|_1 \\
\text{s.t.} & \quad W^v \geq 0.
\end{align*}
\]

(14)

It is obvious that (14) can not be derived directly due to the non-differentiable property of the Lasso regularization. But there are still many ways to solve it, such as the Proximal Gradient Algorithm (PGA). [40] investigates a lot in solving the same kind of problems and proposes an efficient algorithm under non-negative data, which has achieved better results. Similarly, we can solve the (14) as follows:

\[
(W^v_{ij})^{k+1} = (W^v_{ij})^k \frac{2((X^v)^T X^v + \alpha Z^v)_{ij}}{(\beta^v_{ij})^k + 1},
\]

(15)

where \((\beta^v_{ij})^k = (X^v)^T X^v + (X^v)^T X^v + (X^v)^T (X^v)^T + 2\alpha (W^v)^k\). The optimal \( W^v \) can be easily obtained through several iterations.

2) \( Z^v \)-Subproblem: When \( W^v, G, A, F \) are fixed, we can obtain the subproblem of (13) with respect to \( Z^v \):

\[
\begin{align*}
\min_{Z^v} & \quad t \sum_{v=1}^t \alpha \|W^v - Z^v\|_F^2 + \lambda \sum_{i=1}^n \|G^T_i - A^T_i \tilde{Z}^i\|_2^2 \\
\text{s.t.} & \quad Z^v \geq 0.
\end{align*}
\]

(16)

For the sake of optimization, (16) can be rewritten as:

\[
\begin{align*}
\min_{Z^v} & \quad \alpha \|\tilde{W}^v - \tilde{Z}^v\|_F^2 + \lambda \|G^T - A^T \tilde{Z}^v\|_2^2 \\
\text{s.t.} & \quad \tilde{Z}^v \geq 0,
\end{align*}
\]

(17)

where \( \tilde{W}^v \) and \( \tilde{Z}^v \) are obtained in the same form as in (10). Setting the first-order derivative of (17) with respect to \( \tilde{Z}^v \) to zero, it yields:

\[
\tilde{Z}^v = (\alpha I + \lambda A_i A_i^T)^{-1}(\alpha \tilde{W}^v + \lambda A_i G_i^T).
\]

(18)

The sparse graphs \( Z^v \) can be obtained by \( \text{trans}^{-1}(\tilde{Z}) \). In order for the new \( Z^v \) to behave as graphs, normalization and symmetrization are used.

3) \( G \)-Subproblem: When \( W^v, Z^v, A, F \) are fixed, we can obtain the subproblem of (13) with respect to \( G \):

\[
\begin{align*}
\min_{G} & \quad \lambda \sum_{i=1}^n \|G^T_i - A^T_i \tilde{Z}^i\|_F^2 + \gamma \|G\|_F^2 + \eta \text{Tr}(F^T LF) \\
\text{s.t.} & \quad G \geq 0, G1 = 1.
\end{align*}
\]

(19)

To solve (19), we use an equality:

\[
\text{Tr}(F^T LF) = \sum_{i,j=1}^n \frac{1}{2}||F_i - F_j||_2^2 g_{ij}.
\]

(20)

We denote \( H \) as the distance matrix of \( F \), that is, \( h_{ij} = ||F_i - F_j||_2^2 \). Eq. (19) can be reformulated as:

\[
\begin{align*}
\min_{G_i} & \quad \left( -4\lambda A^T_i \tilde{Z}^i + \eta H_i^T \right) \|G_i\|_2^2 \\
\text{s.t.} & \quad G_i \geq 0, G_i1 = 1.
\end{align*}
\]

(21)

The problem (21) can be solved as the Euclidean projection problem:

\[
\begin{align*}
\min_{G_i} & \quad \left( -4\lambda A^T_i \tilde{Z}^i + \eta H_i^T \right) \|G_i\|_2^2 \\
\text{s.t.} & \quad G_i \geq 0, G_i1 = 1.
\end{align*}
\]

(22)

The augmented Lagrangian function of (22) is:

\[
L(G_i, \sigma_1, \sigma_2) = \left( -4\lambda A^T_i \tilde{Z}^i + \eta H_i^T \right) \|G_i\|_2^2 + \sigma_1 (G_i1 - 1) + \sigma_2 G_i,
\]

(23)

where \( \sigma_1, \sigma_2 > 0 \) are the Lagrangian multipliers.

Generally, assume that the optimal \( G_i \) has only \( m \) non-negative elements. Let \( Q_i^T = (\eta H_i^T - 4\lambda A^T_i \tilde{Z}^i) \), and be ordered from small to large, we can approximately set the \( G \) as follows:

\[
G_{ij} = \frac{Q_{i,m+1} - Q_{i,j}}{m Q_{i,m+1} - \sum_{j=1}^m Q_{i,j}}.
\]

(24)

Using (24), we replace the harder-to-tune parameter \( \gamma \) in the original problem with an integer \( m \). Meanwhile, the average degree of a learned graph \( G \) is controllable, i.e., only \( m \) neighbors for each node are reported in \( G \).

4) \( A \)-Subproblem: When \( W^v, Z^v, G, F \) are fixed, we can obtain the subproblem of (13) with respect to \( A \):

\[
\begin{align*}
\min_{A} & \quad \|G_i^T - A^T_i \tilde{Z}^i\|_2^2 \\
\text{s.t.} & \quad A_i \geq 0, A_i^T 1 = 1.
\end{align*}
\]

(25)

After reorganization, (25) can be written as:

\[
\begin{align*}
\min_{A} & \quad ||A_i^T T^i||_2^2 \\
\text{s.t.} & \quad A_i \geq 0, A_i^T 1 = 1,
\end{align*}
\]

(26)

where \( T^i = (1G^T - \tilde{Z}^i) \in \mathbb{R}^{l \times n} \). Setting the first-order derivative of (26), it yields:

\[
A_i = \frac{(T^i (T^i)^T)^{-1} 1}{1^T (T^i (T^i)^T)^{-1} 1}.
\]

(27)

5) \( F \)-Subproblem: Given \( W^v, Z^v, G, A \), the subproblem of \( F \) becomes problem (6). The optimal \( F \) is formed by the \( c \) eigenvectors of \( L \) corresponding to the \( c \) smallest eigenvalues.
C. Discussions

Here we can give a clear interpretation of the mechanism of the proposed graph learning method. The initialization of graph $G$ in (13) is defined as a mean of all the adaptive neighbors graphs generated from each view. During the optimization, the refined graph $\tilde{Z}$ is learned to strike a balance between the informative and redundant structures will be gradually digested and diluted. Finally, the unified graph $Gij$ is given in the form of probabilities by (24), the value of which depends on two aspects, the weighted sum of the edges between $i$ and $j$ in all refined graphs, and the Euclidean distance of their clustering representation through spectral decomposition. In addition, the optimization algorithm we use can also control the sparsity of the unified graph by adjusting the parameter $m$. Here we theoretically analyze the convergence and computational complexity of the optimization algorithm, i.e., Algorithm 1.

1) Convergence Analysis: The convergence of Algorithm 1 is given by Theorem 1.

Theorem 1: The alternate updating strategy in Algorithm 1 monotonically decrease the value of (13) in each iteration until convergence.

Proof: Firstly, the convergence of (15) is provided in [40]. Let $l$ denotes the iterating times, the overall objective function $\mathcal{L}(W, Z, G, A, F)$ monotonically decreases:

$$\mathcal{L}((W)^{l+1}, Z, G, A, F) \leq \mathcal{L}((W)^{l}, Z, G, A, F).$$

(28)

Next, The Hessian matrix of (17) is:

$$\frac{\partial^2 \mathcal{L}(\tilde{Z})}{\partial (\tilde{Z})^2} = \alpha I + \lambda A_{i1} A_{1i}^T.$$

(29)

It is clear that $\alpha ||\tilde{W}^i - \tilde{Z}^i||_F^2 + \lambda ||G_{i1}^T - A_{1i}^T \tilde{Z}^i||_2^2 \geq 0$ and (29) is positive semi-definite. Therefore (17) is a convex function. The overall objective function monotonically decreases:

$$\mathcal{L}(W, (Z)^{l+1}, G, A, F) \leq \mathcal{L}(W, (Z)^{l}, G, A, F).$$

(30)

When others are fixed and updating $G$ or $A$, (22) and (25) are both convex functions because the second-order derivatives of them with respect to each variant are greater than 0. So the overall objective function monotonically decreases:

$$\mathcal{L}(W, Z, (G)^{l+1}, A, F) \leq \mathcal{L}(W, Z, G^{l}, A, F)$$

$$\mathcal{L}(W, Z, G, (A)^{l+1}, F) \leq \mathcal{L}(W, Z, G, A^{l}, F).$$

(31)

Finally, according to [43], the Hessian matrix of the Lagrangian function of (6) is also positive semi-definite. So (6) is a convex function, and the overall objective function monotonically decreases:

$$\mathcal{L}(W, Z, G, A, F^{l+1}) \leq \mathcal{L}(W, Z, G, A, F^{l}).$$

(32)

This completes the proof.

2) Computational Complexity Analysis: The first step of FGL-MSC is to solve (9), whose cost is $O(n^2)$. The second step is to solve (15), which costs $O(n^3)$ based on [40]. In the third and the fourth step of FGL-MSC, we need $O((t_2 + t_3 + t_4^2)n)$ and $O(n^2 + t_2^2)$ to calculate $G$ and $Z$, respectively. The fifth step is to solve (27), we need $O(tn)$ to calculate $T^i$, and the cost of (27) is $O(t^2n + t^3)$. At last, the eigendecomposition problem cost $O(cn^2)$. Therefore, the total computational complexity of FGL-MSC is as follows:

$$O(n^2 + l(n^3 + (1 + 3l + 2l^2 + c)n^2 + 2l^2n)),$$

(33)

where $l$ is the iteration of the above updating steps. Because of $n \gg t$, $n \gg c$, and $n \gg l$, the main computational complexity of FGL-MSC is at the same level as most multi-view clustering methods. The computational complexity mainly locates in computing (15). Due to the existence of Lasso regularization, more efficient optimization strategies should be investigated.

IV. Experiment

In this section, we carry out clustering experiments on 8 benchmark datasets, comparing FGL-MSC with 10 related multi-view clustering methods in terms of three performance evaluation metrics. All the experiments are performed using MATLAB 2021b on a Linux Server with an Intel Xeon 2.10 GHz CPU and 128 GB RAM.

A. Datasets and Experimental Setup

To demonstrate the efficacy of the proposed framework FGL-MSC totally, we conduct experiments on five small-scale (the number of nodes is less than 2000) benchmark datasets and three large-scale datasets. We normalized the data in the range $[0, 1]$, taking into account the metric differences between different forms of features [44]. The detailed information of small-scale datasets is as follows:

Algorithm 1: The General Algorithm of FGL-MSC.

Input: $t$ views data $\{X^1, \ldots, X^t\}$ with $X^v \in \mathbb{R}^{d_v \times n}$, cluster number $c$.

Parameter: $\alpha, \lambda, \eta > 0$, and the number of neighbors in $G$ is set to 10.

Output: the unified graph $G$, the fine-grained fusion weight matrix $A$, the clustering representation $F$.

1: Initialize the weight $a_{ij} = \frac{1}{t}$.
2: Compute $Z^v$ by solving (9), and $G$ is the average of all $Z^v$.
3: Compute $F$ by solving (6).
4: while convergence condition does not meet do
5: Update $W^v$ by (15).
6: Update $Z^v$ by (18).
7: Update $G$ by (24).
8: Update $A$ by (27).
9: Update $F$ by solving (6).
10: end while
11: Apply $K$-means to $F$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
**MSRC-v1** [45] dataset contains 210 nodes with 7 classes: tree, airplane, face, car, building, bicycle, and cow. Each class has 30 nodes. The following 4 views are available: CM, GISTM, LBP, and CENT.

**BBCSport** dataset is composed of news articles in 5 topical areas from the BBC website, which is associated with 2 views.

**WebKB** dataset is a web page classification data set. It contains 203 nodes with binary classes. The following 3 views are available: The Web page text content called PAGE, the anchor text on links called LINK, and the ‘Page-Link’ content.

**100leaves** dataset contains 1600 samples with 100 plant species and each class has 16 samples. There are 3 published features can be used for clustering: shape descriptor, texture histogram, and fine-scale margin.

**ORL** dataset contains 400 face images with 40 classes. Three views can be used: 6750 dimension Gabor, 4096 dimension intensity, and 3304 dimensions LBP.

Likewise, Three large-scale datasets are as follows: **Caltech101-20** (Caltech20) dataset contains 2386 nodes with 20 classes, 6 views can be used: Gabor, Wavelet moments, Cenhst, HOG, GIST, and LBP.

**Scene-15** (scene15) [46] dataset consists of 4485 nodes that cover 15 categories: suburb, bedroom, kitchen, industrial, living, room, forest, inside city, coast, office, highway, tall building, mountain, street, open country, and store. 3 views are available.

**Hdigit** dataset contains 10000 nodes. This handwritten digits (0-9) data set is from two sources, i.e., MNIST Handwritten Digits and USPS Handwritten Digits.

The proposed FGL-MSC is evaluated in three wide evaluation metrics, including accuracy (ACC), normalized mutual information (NMI), and adjusted Rand index (ARI). For all measures, the higher the scores, the better the model performs.

We compare FGL-MSC with 10 state-of-the-art multi-view clustering methods. **Spectral Clustering (SC)** [3] performs the standard spectral clustering in each view. The best results of them are reported. **AMGL** [24] proposes a parameter-free framework to implement the multi-view SC model. **MLAN** [47] extends the adaptive neighbors graph proposed by [7] to multi-view graph learning. **MVGL** [25] proposes a novel fine-grained graph fusion pattern for a set of adaptive neighbors graphs generated from each view. **GMC** [48] conducts a general framework that ensembles graph learning, graph fusion, and clustering. **LMS** [12] is a representative multi-view subspace clustering method that seeks a common latent representation from the original features and employs subspace clustering. **GFSC** [39] combines subspace clustering and graph fusion to create a jointly multi-view clustering framework. **CGL** [49] proposes a graph learning method to conduct clustering by simultaneously learning spectral embedding matrices and low-rank tensor representation. **CNESE** [27] is a constrained multi-view spectral clustering model via integrating non-negative embedding and spectral embedding. **CimSC** [28] is a consistency-induced multi-view subspace clustering framework. All source codes are downloaded from their author’s pages, and their detailed configurations are referenced from the experimental setups provided in their articles. For a fair comparison, we run each experiment 10 times to report the mean and standard deviation.

**B. Clustering Results**

The experimental results (ACC, NMI, and ARI) of different methods on small-scale datasets are reported in Table I, respectively. The results on large-scale datasets are presented in Tables II, III, and IV, respectively. The best results are marked in boldface. According to these results, we have the following observations:

1. Comparing the basic SC with others, the best result of SC tends to lag behind multi-view methods. It illustrates the importance of investigating multi-view models, which exploit information that cannot be detected in a single view.

2. On the one hand, FGL-MSC shows competitive performance on the above datasets. In terms of ACC and ARI, our model outperforms all baseline models except CimSC on MSRC-v1, BBCSport, ORL, 100leaves, and Hdigit. For WebKB, our FGL-MSC exceeds AMGL, MLAN, MVGL, LMS, and CGL concerning all metrics. For Caltech20, our method performs better than MLAN, GMC, and CNESE concerning all metrics. For scene15, FGL-MSC outperforms AMGL, MLAN, MVGL, GMC, and GFSC concerning all metrics. On the other hand, fine-grained weights can cause the model to rely more on the quality of the multi-view data. But on large-scale datasets, our model still gets the best results in terms of ACC on Hdigit. It also exceeds the performances of AMGL, MLAN, GMC, and GFSC on most metrics. Note that despite better results of LMS, CGL, and CimSC, their time and space consumption is more substantial than our model. These results can prove that our strategy is successful.

3. Our model performs better than MVGL for almost all measures on every dataset. The difference between FGL-MSC and MVGL is that our model builds a joint learning framework for the single graph on each view, the unified graph, the fusion weights, and the clustering representation, which enhance each other. The results show the benefits of our model.

4. Our model is ahead of GFSC for most measures on six datasets. Though GFSC exploits a novel graph-level fusion method, the learned graph is probably not as accurate as the fine-grained one in most cases. This fully demonstrates the efficacy of our model.

**C. Parameter Analysis**

Our proposed model has four parameters $\alpha, \lambda, \eta,$ and $m$ that need to be tuned. Among them, $m$ determines the degree of nodes in the learned graph $G$. The setting of $m$ usually depends on the size of the dataset. In our experiments, $m$ is set to 1000
for Hdigit and 10 for the other seven datasets. We also found that for many datasets, the performance tends to be the best when $\alpha = 10^{-2}$. Taking 100leaves, MSRC-v1, and ORL for examples, we show the results under different parameters in Fig. 3.

In summary, the optimal parameters ($\alpha$, $\lambda$, $\eta$) are set to $(10^{-2}, 1, 10)$, $(10^{-2}, 10, 10^{-1})$, $(10^{-2}, 1, 10)$, $(10^{-2}, 10^{-1}, 10)$, $(10^{-2}, 1, 10^{-2})$, $(10^{-2}, 10, 10^{-1})$, $(10^{-2}, 10^{-1}, 10)$, $(10^{-2}, 1, 10^{-2})$, $(10^{-2}, 10, 10^{-1})$, $(10^{-2}, 10^{-1}, 10)$. The bold entities indicate the best results from the comparison experiment.
Fig. 3. Parameters analysis for FGL-MSC on 100leaves, MSRC-v1, and ORL.

TABLE IV
CLUSTERING RESULTS (ARI) ON LARGE-SCALE DATASETS (MEAN ± STANDARD DEVIATION)

| Methods    | Caltech20 | scene15 | Hdigit |
|------------|-----------|---------|--------|
| SC(best)   | 0.1670±0.02 | 0.1217±0.01 | 0.5614±0.02 |
| AMGL       | 0.3707±0.06 | 0.1358±0.01 | 0.9457±0.06 |
| MLAN       | 0.1925±0.00 | 0.0054±0.00 | 0.8430±0.00 |
| MVGL       | 0.4596±0.00 | 0.1463±0.00 | 0.9420±0.00 |
| GMC        | 0.1284±0.00 | 0.0042±0.00 | 0.9458±0.00 |
| LMSC       | 0.3397±0.03 | 0.2138±0.01 | 0.9493±0.00 |
| GFSC       | 0.4003±0.07 | 0.1549±0.01 | 0.9271±0.00 |
| CGL        | 0.4408±0.03 | 0.2391±0.01 | 0.9507±0.02 |
| CNESE      | 0.2564±0.01 | 0.2047±0.00 | 0.8800±0.04 |
| CMSC       | 0.4461±0.05 | 0.2165±0.01 | 0.9242±0.02 |
| FGL-MSC    | 0.2641±0.01 | 0.1762±0.01 | 0.9521±0.00 |

The bold entities indicate the best results from the comparison experiment.

10^{-2}, (10^{-1}, 10^{-1}, 10), (10^{-2}, 1, 10^{-2}), (10^{-2}, 10^{-1}, 10^{-2}) for MSRC-v1, ORL, 100leaves, BBCSport, WebKB, Caltech20, scene15, Hdigit, respectively. It is obvious that our method performs well in a wide range of parameter values.

D. Ablation Study

To understand the role of the individual modules in FGL-MSC, we propose two variants and conduct experiments on three datasets. The two variants are as follows:

**FGL-Z**: Graph Learning with a two-step updating for $Z^v$ in (13). Firstly, the self-expression matrix $W^v$ is optimized independently by (14). Then the objective function (13) becomes:

$$
\begin{align*}
\min_{G,F,A} & \sum_{v=1}^{l} \|W^v - Z^v\|_F^2 + \lambda \sum_{i=1}^{n} \|G_i^T - A_i^T \tilde{Z}^i\|_F^2 \\
& + \gamma \|G\|_F^2 + \eta Tr(F^T L F) \\
\text{s.t.} & \ G \geq 0, \ A \geq 0, \ A_i^T 1 = 1, \ G 1 = 1, \ F^T F = I. \tag{34}
\end{align*}
$$

In (34), the initialization for $Z^v$ is to set $Z^v = W^v$. From a holistic view, the variant FGL-Z is similar to FGL-MSC, except for whether $W^v$ is involved in the joint optimization of the models. The learned graph $W^v$ (or $Z^v$) is fused directly in the
second step, while the downstream task can no longer have an impact on the update of $W^v$.

FGL-F: Graph learning and clustering separately. Specifically, the rank constraint is not considered for the fine-grained graph fusion scheme. The parameter $\eta$ is set to 0, and we implement the spectral clustering method after obtaining $G$ in FGL-F, the objective function of FGL-F can be expressed as follows:

$$
\min_{G,A} \sum_{v=1}^t (||X^v - X^vW^v||_F^2 + \alpha||W^v - Z^v||_F^2 + ||W^v||_1) + \lambda \sum_{i=1}^n ||G^T_i - A^T_i Z^v||_2^2 + \gamma ||G||_F^2
$$

s.t. $G \geq 0, A \geq 0, A^T_{1} = 1, G^1 = 1, (35)$

where $G$ is the input of a classic spectral clustering.

FGL-ZF: Based on the variant FGL-Z, the rank constraint term is also removed. We get the optimal $W^v$ by solving (14), and calculate the unified graph $G$ by (36). The clustering result can be obtained by spectral clustering on $G$.

$$
\min_{G,A} \sum_{v=1}^t ||W^v - Z^v||_F^2 + \lambda \sum_{i=1}^n ||G^T_i - A^T_i Z^v||_2^2
$$

s.t. $G \geq 0, A \geq 0, A^T_{1} = 1, G^1 = 1, (36)$

Table V demonstrates that our unified framework containing the graph refinement and rank constraint modules works best under the same dataset and parameters setting. According to the results of FGL-ZF and FGL-F, the joint optimization for $W^v$ and $Z^v$ outperforms the two-step scheme, but the graph learned by variant FGL-F does not obtain the appropriate cluster structure, with a loss of performance. Comparing the results of FGL-ZF and FGL-Z, the rank constraint term improves the accuracy of clustering. But the variant FGL-Z can not work well because the graphs generated by the self-expression term do not take full advantage of the multi-view information, thus fitting the clustering task not well enough.

E. Effectiveness of Fine-Grained Graph Learning

Here we design experiments on synthetic data and three real-world datasets to verify the effectiveness of the fine-grained graph fusion scheme. We introduce a variant of the graph fusion stage, i.e., graph-level fusion instead of fine-grained graph fusion, and the new model is shown in (37).

$$
\min_{G,F,a} \sum_{v=1}^t (||X^v - X^vW^v||_F^2 + \alpha||W^v - Z^v||_F^2 + ||W^v||_1) + \lambda \sum_{i=1}^n ||G^T_i - A^T_i Z^v||_2^2 + \gamma ||G||_F^2 + \eta Tr(F^T L F)
$$

s.t. $G \geq 0, a \geq 0, a^T_{1} = 1, G^1 = 1, F^T F = I, (37)$

the optimization of (37) differs from that of FGL-MSC in the update of the fusion weight $a$.

$$
\min_{a} ||G - \sum_{v=1}^t a_v Z^v||_F^2
$$

s.t. $a \geq 0, a^T_{1} = 1, (38)$

[50] provides an efficient algorithm for solving (38), and we substitute the weight $a$ derived from this algorithm into the update process for the remaining variables to obtain the model results for the graph-level fusion.

First, We make a toy dataset containing 7 nodes with 4 sets of their feature views. To visualize clearly, we demonstrate each original graph $W^v$ generated by (9) and (15), as shown in Fig. 4.

![Fig. 4. Visualization of the original graphs $W^v$. In the clustering task, nodes 1–4 are in one class and 5–7 are in another.](image)

![Fig. 5. Unified graphs generated by the graph-level fusion method and FGL-MSC.](image)
Then we conduct the graph-level fusion method and FGL-MSC to obtain their unified graphs separately. The unified graphs are shown in Fig. 5. Obviously, there are still redundant edges between two clusters in the unified graph obtained by the graph-level fusion method, which will have an impact on the clustering performance. And the fine-grained graph fusion scheme splits two clusters successfully. Therefore, our proposed model has the ability to filter local structures and learn potential clustering representations. Then we visualize the learned unified graphs by FGL-MSC on MSRC-v1, Caltech20, and Hdigit dataset in Fig. 6. It is observed that the number of connected subgraphs on each unified graph is approximated by the number of clusters. Also, the unified graphs are sparse with few redundant edges between clusters. These experimental results can demonstrate the effectiveness of the fine-grained graph fusion scheme and explain the good performance achieved by FGL-MSC.

F. Convergence Analysis

We investigate the convergence analysis of our model by representing the loss value calculated by (13) with the increasing iteration. The results on ORL and Caltech20 are reported in Fig. 7. It is clear that the loss value is decreasing stably during the optimization. When the iteration runs to 20 rounds, the rates of change are significantly lower. When iterating to 50 rounds, the loss values plateau. The iteration set in our experiments is 10 for computational efficiency considerations. But the convergence results tell us that FGL-MSC could achieve a better performance than the above experiments. Generally speaking, our model has good convergence and achieves cheerful performance after a few iterations.

V. CONCLUSION

In this article, we propose a new fine-grained graph learning framework for multi-view subspace clustering. To avoid the deterioration of the learning process from incorrect neighbors and redundant structures and to elevate the clustering performance, we employ a fine-grained graph fusion method, which is incorporated into the joint framework. This framework generates the fine-grained weights, the unified graph, and the clustering representation simultaneously. An effective optimization strategy with guaranteed convergence was provided to solve the joint objective function. Then, Extensive experiments on some benchmark datasets demonstrated the effectiveness of our method for multi-view subspace clustering.

REFERENCES

[1] Y. Yang and H. Wang, “Multi-view clustering: A survey,” Big Data Mining Anal., vol. 1, no. 2, pp. 83–107, 2018.
[2] G. Chao, S. Sun, and J. Bi, “A survey on multiview clustering,” IEEE Trans. Artif. Intell., vol. 2, no. 2, pp. 146–168, Apr. 2021.
[3] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2001, pp. 849–856.
[4] X. He and P. Niyogi, “Locality preserving projections,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2003, pp. 153–160.
[5] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, “Image clustering using local discriminant models and global integration,” IEEE Trans. Image Process., vol. 19, no. 10, pp. 2761–2773, Oct. 2010.
[6] Z. Zhao, L. Wang, H. Liu, and J. Ye, “On similarity preserving feature selection,” IEEE Trans. Knowl. Data Eng., vol. 25, no. 3, pp. 619–632, Mar. 2013.
[7] F. Nie, X. Wang, and H. Huang, “Clustering and projected clustering with adaptive neighbors,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2014, pp. 977–986.
[8] E. Elhamifar and R. Vidal, “Sparse subspace clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2009, pp. 2790–2797.
[9] H. Hu, Z. Lin, J. Feng, and J. Zhou, “Smooth representation clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 3834–3841.
[10] J. Wang, X. Wang, F. Tian, C. H. Liu, and H. Yu, “Constrained low-rank representation for robust subspace clustering.” IEEE Trans. Cybern., vol. 47, no. 12, pp. 4534–4546, Dec. 2017.
[11] S. Liu et al., “Efficient one-pass multi-view subspace clustering with consensus anchors,” AAAI, pp. 7576–7584, 2022.
[12] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, “Latent multi-view subspace clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 4333–4341.
[13] C. Zhang et al., “Generalized latent multi-view subspace clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 1, pp. 86–99, Jan. 2020.
[14] Z. Li, J. Liu, J. Tang, and H. Lu, “Robust structured subspace learning for data representation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 10, pp. 2085–2098, Oct. 2015.
[15] P. Zhu, B. Hui, C. Zhang, D. Du, L. Wen, and Q. Hu, “Multi-view deep subspace clustering networks,” 2019, arXiv:1908.01978.
[16] X. Peng, J. Feng, S. Xiao, J. Lu, Z. Yi, and S. Yan, “Deep sparse subspace clustering,” 2017, arXiv:1709.08374.
[17] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. D. Reid, “Deep subspace clustering networks,” in Proc. Annu. Conf. Neural Inf. Process. Syst., 2017, pp. 24–33.
[18] Q. Wang, J. Cheng, Q. Gao, G. Zhao, and L. Jiao, “Deep multi-view subspace clustering with unified and discriminative learning,” IEEE Trans. Multimedia, vol. 23, pp. 3483–3493, 2021.
[19] Q. Zheng, J. Zhu, Y. Ma, Z. Li, and Z. Tian, “Multi-view subspace clustering networks with local and global graph information,” Neurocomputing, pp. 23–33.
[20] 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, 2019, pp. 1204–1209.
[21] S. Luo, C. Zhang, W. Zhang, and X. Cao, “Consistent and specific multi-view subspace clustering,” in Proc. AAAI Conf. Artif. Intell., 2018, pp. 3730–3737.
[22] M. Chen et al., “Representation learning in multi-view clustering: A literature review,” Data Sci. Eng., vol. 7, no. 3, pp. 225–241, 2022.
[23] H. Gao, F. Nie, X. Li, and H. Huang, “Multi-view subspace clustering,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 4238–4246.
[24] F. Nie, J. Li, and X. Li, “Parameter-free auto-weighted multiple graph learning: A framework for multi-view clustering and semi-supervised classification,” in Proc. Int. Joint Conf. Artif. Intell., 2016, pp. 1881–1887.
[25] K. Zhan, C. Zhang, J. Guan, and J. Wang, “Graph learning for multi-view clustering,” IEEE Trans. Cybern., vol. 48, no. 10, pp. 2887–2895, Oct. 2018.
[26] Z. Hu, F. Nie, R. Wang, and X. Li, “Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding,” Inf. Fusion, vol. 55, pp. 251–259, 2020.
[27] S. E. Hajjar, F. Dornaika, and F. Abdallah, “Multi-view spectral clustering via constrained nonnegative embedding,” Inf. Fusion, vol. 78, pp. 209–217, 2022.
[28] Y. Qin, G. Feng, Y. Ren, and X. Zhang, “Consistency-induced multi-view subspace clustering,” IEEE Trans. Cybern., vol. 53, no. 2, pp. 832–844, Feb. 2023.
[29] Z. Gu and S. Feng, “Individuiality meets commonality: A unified graph learning framework for multi-view clustering,” ACM Trans. Knowl. Discov. Data, vol. 17, no. 1, pp. 1:1–1:21, 2023.
[30] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, “Low-rank tensor constrained multi-view subspace clustering,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1582–1590.
[31] 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, Dec. 2019.
[32] Q. Gao, W. Xia, Z. Wan, D. Xie, and P. Zhang, “Tensor-SVD based graph learning for multi-view subspace clustering,” in Proc. 34th AAAI Conf. Artif. Intell., pp. 15–23, 2021.
[33] M. S. Chen, C. D. Wang, and J. H. 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.
[34] L. Fu, J. Yang, C. Chen, and C. Zhang, “Low-rank tensor approximation with local structure for multi-view intrinsic subspace clustering,” Inf. Sci., vol. 606, pp. 877–891, 2022.
[35] Y. Chen, S. Wang, J. Su, and J. Chen, “Corentropy-induced tensor learning for multi-view subspace clustering,” in Proc. IEEE Int. Conf. Data Mining, 2022, pp. 897–902.
[36] G. A. Khan, J. Hu, T. Li, B. Diallo, and H. Wang, “Multi-view data clustering via non-negative matrix factorization with manifold regularization,” Int. J. Mach. Learn. Cybern., vol. 13, no. 3, pp. 677–689, 2022.
[37] G. Zhong, T. Shu, G. Huang, and X. Yan, “Multi-view spectral clustering by simultaneous consensus graph learning and discretization,” Knowl. Based Syst., vol. 235, 2022, Art. no. 107632.
[38] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 171–184, Jan. 2013.
[39] Z. Kang et al., “Multi-graph fusion for multi-view spectral clustering,” Knowl. Based Syst., vol. 189, 2020, Art. no. 105102.
[40] S. B. Chen, C. H. Q. Ding, and B. Luo, “Similarity learning of manifold data,” IEEE Trans. Cybern., vol. 45, no. 9, pp. 1744–1756, Sep. 2015.
[41] R. Wang, S. Chen, B. Luo, and J. Zhang, “Multi-view similarity learning of manifold data,” in Proc. 10th Int. Conf. Image Graph., 2019, pp. 631–643.

Z. Kang et al., “Structured graph learning for clustering and semi-supervised classification,” Pattern Recognit., vol. 110, 2021, Art. no. 107627.

M. W. Newman, N. Libraty, O. On, K. A. On, and K. A. On, “The laplacian spectrum of graphs,” Graph Theory Combinations Appl., vol. 18, no. 7, pp. 871–898, 1991.

X. Cai, F. Nie, and H. Huang, “Multi-view k-means clustering on Big Data,” in Proc. Int. Joint Conf. Artif. Intell., 2013, pp. 2598–2604.

Y. J. Lee and K. Grauman, “Foreground focus: Unsupervised learning from partially matching images,” Int. J. Comput. Vis., vol. 85, no. 2, pp. 143–166, 2009.

S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2006, pp. 2169–2178.

F. Nie, G. Cai, and X. Li, “Multi-view clustering and semi-supervised classification with adaptive neighbours,” in Proc. AAAI Conf. Artif. Intell., 2017, pp. 2408–2414.

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, Jun. 2020.

W. Zhou, H. Wang, and Y. Yang, “Consensus graph learning for incomplete multi-view clustering,” in Proc. Adv. Knowl. Discov. Data Mining, 23rd Pacific-Asia Conf., 2019, pp. 529–540.

P. Nie, L. Tian, R. Wang, and X. Li, “Multiview semi-supervised learning model for image classification,” IEEE Trans. Knowl. Data Eng., vol. 32, no. 12, pp. 2389–2400, Dec. 2020.

Yaozong Wang received the B.E. degree in software engineering from the Jiangxi University of Science and Technology, Wuhan, China, in 2006. From 2000 to 2003, he was a Research Engineer and a Software Developer with the Research and Development Department, Huawei Technologies Company Ltd., Shenzhen, China. He is currently a Professor with the School of Software, Huazhong University of Science and Technology. His research interests include machine learning, data mining, and software engineering.

Xiaobao Pei received the Ph.D. degree in computer software and theory from the Huazhong University of Science and Technology, Wuhan, China, in 2006. From 2000 to 2003, he was a Research Engineer and a Software Developer with the Research and Development Department, Huawei Technologies Company Ltd., Shenzhen, China. He is currently a Professor with the School of Software, Huazhong University of Science and Technology. His research interests include machine learning, data mining, and software engineering.