Learning Orthogonal Latent Representations for Multi-View Clustering

Xiaolin Xiao, *Member, IEEE*, Yue-Jiao Gong, *Senior Member, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*—In the field of multi-view clustering, latent representations are often employed to address the challenge posed by low-quality data. Traditional approaches typically assume that multiple views are fully dependent, directly learning a common latent representation from the observed data. However, this assumption is overly restrictive in real-world scenarios and may overlook valuable information, as the independence of different views can reveal critical view-specific characteristics. To overcome this limitation, we propose learning Orthogonal Latent Representations for Multi-View Clustering (OLR-MVC), which jointly captures both cross-view dependence and independence. Specifically, our model maps multi-view data into shared and private latent spaces using distinct projection bases. To accurately capture both dependence and independence, we enforce orthogonality between the shared and private latent representations while also encouraging pairwise orthogonality among private representations. Furthermore, we leverage the self-expressive property of these latent representations to capture global data structures. Extensive experimental evaluations demonstrate that OLR-MVC outperforms state-of-the-art multi-view clustering methods.**

***Index Terms*—Dependence, independence, orthogonal latent representation, multi-view clustering, self-expressiveness.**

I. Introduction

FOR data clustering, relying on a single feature often introduces bias, as feature extraction is typically an unsupervised process that may not align with the underlying cluster memberships. To address this issue, an emerging research field focuses on reducing bias by leveraging the complementary information present in multiple views. Typically, these views are collected either from the same sensor with different parameters or from different modalities. The core of multi-view learning is to fully exploit the cross-view complementarity to enhance performance [1], [2], [3].

Due to the presence of noise and redundancy, directly using the observed multi-view data may result in suboptimal performance. To mitigate this issue, latent representation learning assumes that the observed views are generated from a common latent space [4], [5], [6], [7], [8]. The goal of these methods is to find transformations from individual views to a shared latent representation. This latent representation provides a compact interpretation of the observed data, helping to uncover the underlying data structures.

While latent representation learning has demonstrated strong empirical performance, most existing methods inherently assume that multi-view features are fully dependent [9]. However, this assumption is overly restrictive and may overlook valuable view-specific information. Recently, the complementarity of view-specific characteristics has garnered increasing research attention. For instance, Wan et al. proposed exploring latent representations under diverse dimensions to enhance model expressiveness and capture cross-view complementarity [10]. Zhou et al. learned shared and view-specific dictionaries to separately exploit the cross-view correlations and view-specific properties [11]. However, the dependence and independence between views are not accurately modeled, leading to inadequate representation capabilities.

To address the aforementioned issues, a pragmatic approach involves simultaneously modeling both cross-view dependence and independence. We demonstrate this with a simulation example that highlights the significance of jointly considering dependent and independent information. First, we generate a raw dataset from a 4-D space represented by $[a, b, c, d]$, where $\{(a, b)|a^2 + b^2 = 1\}$, and c and d are independently set to either 0 or 1. As shown in Table I, these samples are grouped into 16 classes $(L_1, \ldots, L_{16})$ based on the four quadrants of the $a - b$ plane and the values of c and d. In this case, a and b

TABLE I
The Raw Data is Grouped Into 16 Classes Based on the Values of Four Dimensions

$L_1 : \{(a,b,c,d)\|a \leq 0, b \leq 0, c = 0, d = 0\}$	$L_2 : \{(a,b,c,d)\|a \leq 0, b \leq 0, c = 0, d = 1\}$
$L_3 : \{(a,b,c,d)\|a \leq 0, b \leq 0, c = 1, d = 0\}$	$L_4 : \{(a,b,c,d)\|a \leq 0, b \leq 0, c = 1, d = 1\}$
$L_5 : \{(a,b,c,d)\|a \leq 0, b > 0, c = 0, d = 0\}$	$L_6 : \{(a,b,c,d)\|a \leq 0, b > 0, c = 0, d = 1\}$
$L_7 : \{(a,b,c,d)\|a \leq 0, b > 0, c = 1, d = 0\}$	$L_8 : \{(a,b,c,d)\|a \leq 0, b > 0, c = 1, d = 1\}$
$L_9 : \{(a,b,c,d)\|a > 0, b > 0, c = 0, d = 0\}$	$L_{10} : \{(a,b,c,d)\|a > 0, b > 0, c = 0, d = 1\}$
$L_{11} : \{(a,b,c,d)\|a > 0, b > 0, c = 1, d = 0\}$	$L_{12} : \{(a,b,c,d)\|a > 0, b > 0, c = 1, d = 1\}$
$L_{13} : \{(a,b,c,d)\|a > 0, b \leq 0, c = 0, d = 0\}$	$L_{14} : \{(a,b,c,d)\|a > 0, b \leq 0, c = 0, d = 1\}$
$L_{15} : \{(a,b,c,d)\|a > 0, b \leq 0, c = 1, d = 0\}$	$L_{16} : \{(a,b,c,d)\|a > 0, b \leq 0, c = 1, d = 1\}$

Received 2 October 2024; revised 30 December 2024 and 6 February 2025; accepted 22 February 2025. Date of publication 12 September 2025; date of current version 12 November 2025. This work was supported in part by the General Program of Guangdong Natural Science Foundation under Grant 2025A1515012267, in part by the Guangzhou Science and Technology Elite Talent Leading Program for Basic and Applied Basic Research under Grant SL2024A04J01361, in part by the Guangdong Natural Science Funds for Distinguished Young Scholars under Grant 2022B1515020049, in part by the National Natural Science Foundation of China under Grant 62276100, and in part by the Fundamental Research Funds for the Central Universities. The associate editor coordinating the review of this article and approving it for publicatio was Prof. Xiu-Shen Wei. *(Corresponding author: Yue-Jiao Gong.)*

Xiaolin Xiao is with the School of Computer Science, South China Normal University, Guangzhou 510631, China (e-mail: shellyxiaolin@gmail.com).

Yue-Jiao Gong is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: gongyuejiao@gmail.com).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/TMM.2025.3607704, provided by the authors.

Digital Object Identifier 10.1109/TMM.2025.3607704

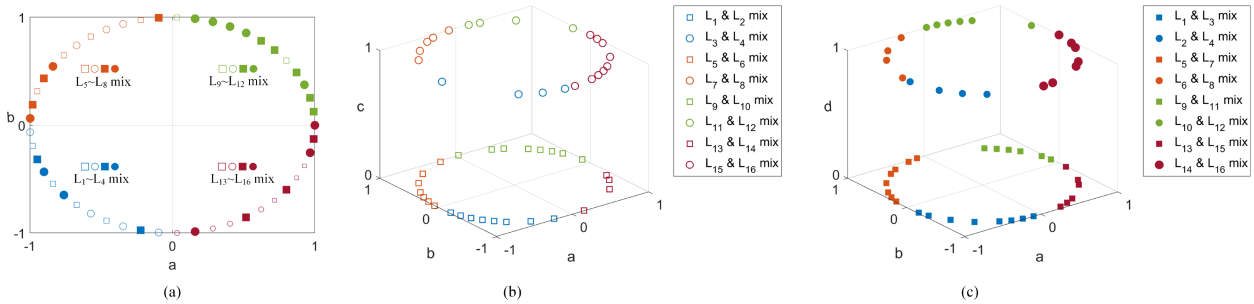

Fig. 1. Illustration of the importance of jointly modeling dependence $[a, b]$ and independence $[c, d]$: A two-view dataset with 16 classes is generated, where each view consists of two dependent dimensions and one independent dimension, namely, $x^{(1)} = [a, b, c]$ and $x^{(2)} = [a, b, d]$. (a) Only four classes are identified using the dependent information, i.e., $[a, b]$. (b) Eight classes are identified by incorporating the independent information in $x^{(1)}$, i.e., dimension c. (c) Eight classes are identified by incorporating the independent information in $x^{(2)}$, i.e., dimension d.

are dependent, whereas c and d are independent of each other and also independent from the $a - b$ plane. Next, we construct a two-view dataset from the raw data, where the first view is represented by $x^{(1)} = [a, b, c]$, and the second view is represented by $x^{(2)} = [a, b, d]$. As depicted in Fig. 1(a), if only the cross-view dependence (i.e., the $a - b$ plane) is utilized, four classes are mixed within each quadrant. By simultaneously leveraging both the cross-view dependence and the independence inherent in either x_1 or x_2, eight classes can be correctly identified (see Fig. 1(b) and 1(c)). Although visualizing the dataset in 4-D space is challenging, it is reasonable to expect that all 16 classes can be correctly identified by integrating both the cross-view dependence and the independence within both views.

Building on this insight, we propose Orthogonal Latent Representations for Multi-View Clustering (OLR-MVC), which jointly exploits cross-view dependence and independence. Specifically, the multi-view data are factorized into shared and private components using respective projection bases. The shared representation encodes dependent information across different views, while the private representations capture independent information specific to individual views. We introduce orthogonal constraints to separate the cross-view dependence and independence. Additionally, the self-expressive property of the latent representations is leveraged to reveal clear data structures with global constraints. The main novelty and contributions of OLR-MVC are summarized as follows.

1) We propose factorizing latent representations into orthogonal shared and private components to accurately capture cross-view dependence and independence. Compared to existing models that transform multiple views into a common latent representation, our framework is more flexible and comprehensive, opening up new avenues for designing latent representation learning approaches.
2) We explore orthogonal latent representations for multi-view clustering. To the best of our knowledge, OLR-MVC is the first model to accurately model both dependence and independence, resulting in significant performance improvements across a range of datasets.
3) We design an efficient optimization algorithm to solve the OLR-MVC model within the framework of the alternating direction method of multipliers.
4) Extensive experiments comparing OLR-MVC with state-of-the-art methods demonstrate its superiority.

The remainder of this article is organized as follows. Section II reviews existing multi-view clustering algorithms. Section III introduces the core idea of orthogonal latent representation learning and then proposes the OLR-MVC model. The optimization algorithm for OLR-MVC is also elaborated in this section. Section IV presents experimental results on real-world databases. Finally, Section V concludes the article.

II. Related Work

Drawing partially on [2], [3], we classify modern multi-view clustering methods into four technological approaches: (1) kernel learning methods, (2) graph learning methods, (3) subspace learning methods, and (4) deep learning methods.

As real-world data may not be linearly separable, kernel learning methods exploit the kernel trick to address data nonlinearity [12]. Typically, these methods apply different predefined kernels to process various views and then combine the results to obtain a unified kernel. The core challenge is selecting appropriate kernel functions and designing an optimal fusion strategy. For instance, Huang et al. [13] learned data similarity in kernel space to improve the quality of base kernels. Based on the kernel alignment criterion, Liu et al. [14], [15] and Zhang et al. [16] proposed multi-kernel clustering methods that guarantee good theoretical and empirical performance. Additionally, Huang et al. [17] enriched affinity matrix learning with iterative clustering in the kernel space. Recently, Liu et al. [18] introduced contrastive learning with kernel generation to better explore cross-view complementarity.

Graph learning is a popular technique in multi-view clustering [19], [20]. Typically, these methods either construct view-specific graphs from raw features and leverage graph fusion techniques to obtain a unified graph, or directly learn view-specific graphs during the optimization procedure. For example, Huang et al. [21] captured multi-view consistency and diversity in a unified framework and fused the consistent graphs, allocating cluster labels without post-processing. Li et al. [22] constructed initial graphs based on the inner products of normalized spectral

embedding matrices and enforced high-order cross-view relationships using a weighted tensor nuclear norm. Huang et al. [23] proposed learning a latent graph from view-specific graphs, considering both global and local data structures. Wang et al. [24] used the Hilbert-Schmidt independence criterion to learn a consensus graph without predefined similarity matrices. Bipartite graph-based methods have also been proposed to improve efficiency in multi-view clustering [25], [26], [27]. While graph learning methods have demonstrated strong empirical performance, they can struggle when the initial graph quality is poor.

Subspace learning-based methods assume that the underlying data structures can be uncovered in low-dimensional compact spaces. Most models either regularize view-specific representations using a common structure (i.e., self-expressiveness learning [28]) or directly find a shared latent space from multiple observations (i.e., latent representation learning [4]). In the first category, self-expressive matrices are generated for each view, and various techniques are used to align view-specific representations to achieve consensus. For example, tensor decomposition techniques have been used to regularize the stacked self-expressive matrices via third-order tensor ranks [29], [30], [31]. The second type of methods, latent representation learning, focuses on finding a common compact representation across all views. Commonly used techniques for learning low-dimensional representations include (non-negative) matrix factorization [32], [33], [34], projection learning [4], [6], [8], and mapping matrix learning [5], [7]. Note that projection learning methods enforce orthonormal constraints on the transformation matrix, while mapping matrix learning models replace these constraints with spherical constraints. Both approaches assume that the shared latent representation captures cross-view relationships by modeling cross-view dependencies [9]. However, this assumption may be too rigid, overlooking important view-specific information. Recently, mining cross-view complementarity has gained increasing research attention. For instance, Wan et al. [10] and Zhou et al. [11] explored the diverse and view-specific properties to uncover the underlying data structures. However, their models do not accurately capture the dependence and independence between views, resulting in limited representation capabilities.

Recently, the advancement of deep learning models has facilitated the development of multi-view learning [35]. Multi-view learning tasks have been integrated with autoencoders [36], generative adversarial networks [37], graph neural networks [38], deep belief nets [39], and contrastive learning [40], [41], leading to excellent results. While these deep learning methods often deliver enhanced performance, they require substantial computational resources. Therefore, in this work, we focus on mathematical modeling approaches.

III. Learning Orthogonal Latent Representations for Multi-View Clustering

A. Orthogonal Shared-Private Latent Representations

Since multi-view features represent data from different perspectives, assuming that they only share dependent information that projects to a common space is impractical. As illustrated in Fig. 1, independence also provides valuable insights for data clustering. Based on this insight, we introduce a more flexible and comprehensive method that accurately models cross-view dependence and independence in a unified framework. As shown in Fig. 2, each view is mapped into shared and private spaces using their respective projection bases. The shared latent representation captures cross-view dependence, while the private latent representations portray independent information. We then learn the self-expressive matrices from the latent representations to uncover the underlying data structures. The key challenge is accurately distinguishing between shared and private latent representations in an unsupervised manner. To address this, we enforce orthogonality between the shared and private latent representations. Additionally, the private latent representations are encouraged to be pairwise orthogonal to model independence.

TABLE II
Summary of Commonly Used Notations

Notation	Description
V	Number of views
N	Number of samples
d_i	Data dimension in the i-th view
k	Latent dimension over all view
$X^{(i)} \in \mathbb{R}^{d_i \times N}$	Observed data in the i-th view
$H^{(i)} \in \mathbb{R}^{k \times N}$	Latent representation for the i-th view
$H_s \in \mathbb{R}^{k \times N}$	Shared latent representation
$H_p^{(i)} \in \mathbb{R}^{k \times N}$	Private latent representation for the i-th view
$P_s^{(i)} \in \mathbb{R}^{d_i \times k}$	Projection to obtain H_s
$P_p^{(i)} \in \mathbb{R}^{d_i \times k}$	Projection to obtain $H_p^{(i)}$
$Z_s \in \mathbb{R}^{N \times N}$	Self-expressive matrix for H_s
$Z_p^{(i)} \in \mathbb{R}^{N \times N}$	Self-expressive matrix for $H_p^{(i)}$
$E_x^{(i)} \in \mathbb{R}^{d_i \times N}$	Reconstruction error in projection learning of the i-th view
$E_s \in \mathbb{R}^{k \times N}$	Reconstruction error in self-expressiveness learning of H_s
$E_p^{(i)} \in \mathbb{R}^{k \times N}$	Reconstruction error in self-expressiveness learning of $H_p^{(i)}$
$\mathcal{E}$	Reconstruction errors,$= \{E_x^{(1)}, \cdots, E_x^{(V)}, E_s, E_p^{(1)}, \cdots, E_p^{(V)}\}$
$\mathcal{Z}$	Set of self-expressive matrices, $= \{Z_s, Z_p^{(1)}, \cdots, Z_p^{(V)}\}$

The commonly used notations are presented in Table II. Formally, given multi-view features $\{X^{(i)} \in \mathbb{R}^{d_i \times N}\}_{i=1}^V$, where d_i is the observed dimension in the i-th view and N is the number of samples, let $\{H^{(i)} \in \mathbb{R}^{k \times N}\}_{i=1}^V$ be the set of multi-view latent representations, with k indicating the latent dimension. The j-th column of $H^{(i)}$ represents the latent representation of the j-th sample in $X^{(i)}$. To illustrate the main concept, we decompose the latent representations as $\{H^{(i)} = H_s + H_p^{(i)}\}_{i=1}^V$, where H_s and $\{H_p^{(i)}\}$ are the shared and private latent representations, respectively. By encouraging orthogonality, we factorize the shared and private latent representations as follows:

$$\sum_{i=1}^{V} \|H_s' H_p^{(i)}\|_F^2 + \sum_{i=1}^{V} \sum_{j \neq i}^{V} \|H_p^{(i)'} H_p^{(j)}\|_F^2,$$
$$\text{s.t. } \{H^{(i)} = H_s + H_p^{(i)}\}_{i=1}^V, \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Orthogonality is achieved by minimizing the squared Frobenius norm of the inner products between latent representations. Specifically, let $M = H_s' H_p^{(i)}$. If H_s and $H_p^{(i)}$ are orthogonal, each pair of columns in H_s and $H_p^{(i)}$ will be perpendicular, and hence all elements in M approach zero. To simplify the optimization process, we use the squared Frobenius norm of the inner products as the cost function.

By introducing orthogonal constraints, (1) can accurately model both dependence and independence across different

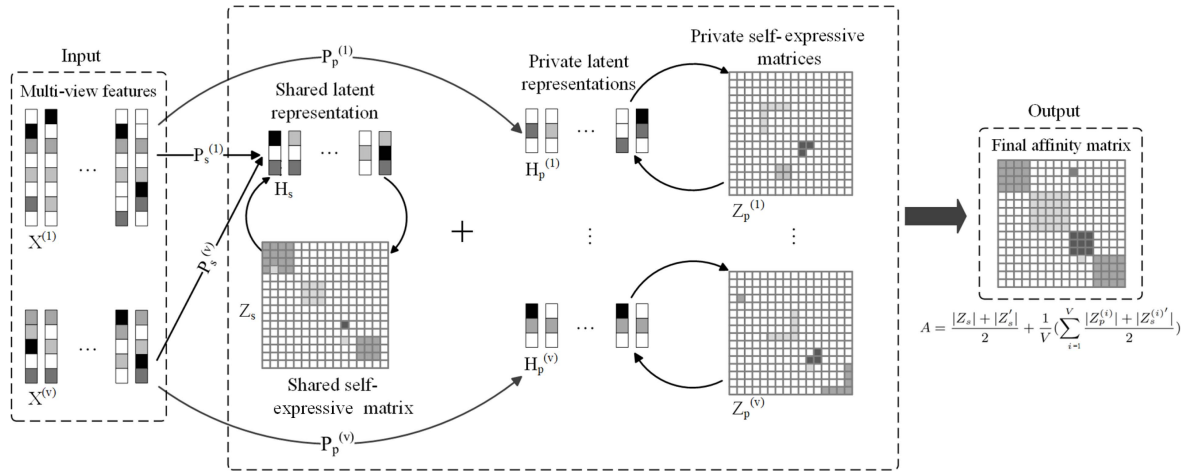

Fig. 2. The framework of the proposed OLR-MVC model. The multi-view data are mapped into a shared latent space and V private latent spaces using their respective projection bases, with self-expressive matrices learned from individual latent spaces. The latent representations are mutually orthogonal to model independence, while the self-expressive matrices are encouraged to be low-rank to capture global data structures. The final affinity matrix is obtained by integrating the shared and private self-expressive matrices.

views. This approach uncovers valuable cross-view complementarity, enabling more accurate modeling of data relationships. Moreover, the shared and private latent representations remain independent of each other, enhancing the interpretability.

However, directly solving (1) may lead to a trivial solution where one of the latent representations approaches zero. In the next section, we introduce a more realistic framework to avoid this issue.

$$
\begin{aligned}
&\mathcal{Z} = \{Z_s, Z_p^{(1)}, \ldots, Z_p^{(V)}\},\\
&\{\, P_s^{(i)\prime} P_s^{(i)} = I,\ \ P_p^{(i)\prime} P_p^{(i)} = I \,\}_{i=1}^{V},\\
&\{\, X^{(i)} = P_s^{(i)} H_s + P_p^{(i)} H_p^{(i)} + E_x^{(i)} \,\}_{i=1}^{V},\\
&H_s = H_s * Z_s + E_s,\\
&\{\, H_p^{(i)} = H_p^{(i)} Z_p^{(i)} + E_p^{(i)} \,\}_{i=1}^{V},
\end{aligned}
\tag{2}
$$

B. Learning Orthogonal Latent Representations for Multi-View Clustering

We explore orthogonal latent representations for multi-view clustering. Given the multi-view data $\{X^{(i)}\}_{i=1}^{V}$, we assume that each view is factorized into shared and private latent representations by projecting onto respective bases. Our method offers greater flexibility and comprehensiveness compared to previous latent representation learning models by jointly considering cross-view consistency and complementarity.

Specifically, we use two sets of bases, $\{P_s^{(i)}\}_{i=1}^{V}$ and $\{P_p^{(i)}\}_{i=1}^{V}$, to project the observed data into different latent spaces. We then apply the self-expressive property of latent representations to clearly capture data affinities. To avoid potential trivial solutions, we minimize the reconstruction errors associated with both projection learning and self-expressiveness learning. Formally, the Orthogonal Latent Representation-based Multi-View Clustering (OLR-MVC) model is formulated as follows:

$$
\begin{aligned}
&\min_{\substack{\mathcal{E},\mathcal{Z},H_s,\{P_s^{(i)},\\ P_p^{(i)},H_p^{(i)}\}}} \|\mathcal{E}\|_{2,1} + \alpha \|\mathcal{Z}\|_{*} + \beta \\
&\qquad \left(\sum_{i=1}^{V} \|H_s' H_p^{(i)}\|_F^2 + \sum_{i=1}^{V} \sum_{j \neq i}^{V} \|H_p^{(i)\prime} H_p^{(j)}\|_F^2 \right),\\
&s.t.\quad \mathcal{E} = \{E_x^{(1)}, \ldots, E_x^{(V)}, E_s, E_p^{(1)}, \ldots, E_p^{(V)}\},
\end{aligned}
$$

where α and β are hyper-parameters that balance the relative importance of different modules. For conciseness, the swash letters $\mathcal{E}$ and $\mathcal{Z}$ denote the sets of variables. The $l_{2,1}$-norm $\|\mathcal{E}\|_{2,1}$ and the nuclear norm $\|\mathcal{Z}\|_{*}$ are imposed on each element in $\mathcal{E}$ and $\mathcal{Z}$, respectively.

To improve understanding, we make the following remarks:

- The i-th view is factorized as $X^{(i)} = P_s^{(i)} H_s + P_p^{(i)} H_p^{(i)} + E_x^{(i)}$, where $P_s^{(i)}$ and $P_p^{(i)}$ are constrained by $P_s^{(i)\prime} P_s^{(i)} = I$ and $P_p^{(i)\prime} P_p^{(i)} = I$. Thus, the observed data are projected into shared and private latent spaces.
- By incorporating orthogonal constraints, we can effectively model both cross-view dependence and independence. The shared representation captures the dependent information shared by different views, whereas the private representations reflect the independent information unique to each view.
- We use the $l_{2,1}$ norm $\|\cdot\|_{2,1}$ to model sample-specific corruptions. Since sample-wise inconsistency may occur in both projection learning and self-expressiveness learning, the $l_{2,1}$ norm is utilized in both stages and is expressed as $\|\mathcal{E}\|_{2,1} = \sum_{i=1}^{V} \|E_x^{(i)}\|_{2,1} + \|E_s\|_{2,1} + \sum_{i=1}^{V} \|E_p^{(i)}\|_{2,1}$.
- The coefficient matrices are encouraged to be low-rank by minimizing the nuclear norm $\|\mathcal{Z}\|_{*} = \|Z_s\|_{*} + \sum_{i=1}^{V} \|Z_p^{(i)}\|_{*}$. This encourages the self-expressiveness of the latent representations, imposing global constraints on affinity learning and improving overall accuracy.

C. Solution

The variables in $\mathcal{Z}=\{Z_s,Z_p^{(1)},\ldots,Z_p^{(V)}\}$ are coupled in (2), making direct optimization challenging. To address this, we introduce an auxiliary variable set $\mathcal{W}=\mathcal{Z}$ to separate coupled variables as follows:

$$\begin{aligned}
&\min_{\substack{\mathcal{E},\mathcal{Z},\mathcal{W},H_s,\\ \{P_s^{(i)},P_p^{(i)},H_p^{(i)}\}}} \|\mathcal{E}\|_{2,1}+\alpha\|\mathcal{W}\|_*+\beta \\
&\qquad\left(\sum_{i=1}^{V}\|H_s'H_p^{(i)}\|_F^2+\sum_{i=1}^{V}\sum_{j\neq i}^{V}\|H_p^{(i)'}H_p^{(j)}\|_F^2\right), \\
&s.t.\quad \mathcal{E}=\{E_x^{(1)},\ldots,E_x^{(V)},E_s,E_p^{(1)},\ldots,E_p^{(V)}\}, \\
&\qquad \mathcal{Z}=\{Z_s,Z_p^{(1)},\ldots,Z_p^{(V)}\},\ \ \mathcal{W}=\mathcal{Z}, \\
&\qquad \{\ P_s^{(i)'}P_s^{(i)}=I,\ \ P_p^{(i)'}P_p^{(i)}=I\ \}_{i=1}^{V}, \\
&\qquad \{\ X^{(i)}=P_s^{(i)}H_s+P_p^{(i)}H_p^{(i)}+E_x^{(i)}\ \}_{i=1}^{V}, \\
&\qquad H_s=H_s*Z_s+E_s, \\
&\qquad \{\ H_p^{(i)}=H_p^{(i)}Z_p^{(i)}+E_p^{(i)}\ \}_{i=1}^{V}. \qquad (3)
\end{aligned}$$

Next, we apply the Alternating Direction Method of Multipliers (ADMM) [42], [43] to optimize (3). ADMM provides an efficient approach for solving constrained optimization problems by forming an unconstrained augmented Lagrangian function and iteratively updating one variable at a time while keeping the others fixed. The unconstrained augmented Lagrangian function of (3) is given by:

$$\begin{aligned}
&L(\mathcal{E},\mathcal{W},\mathcal{Z},H_s,\{P_s^{(i)},P_p^{(i)},H_p^{(i)}\})=\|\mathcal{E}\|_{2,1}+\alpha\|\mathcal{W}\|_* \\
&+\beta\left(\sum_{i=1}^{V}\|H_s'H_p^{(i)}\|_F^2+\sum_{i=1}^{V}\sum_{j\neq i}^{V}\|H_p^{(i)'}H_p^{(j)}\|_F^2\right) \\
&+\frac{\rho}{2}\sum_{i=1}^{V}\|X^{(i)}-P_s^{(i)}H_s-P_p^{(i)}H_p^{(i)}-E_x^{(i)}+\frac{Y_1^{(i)}}{\rho}\|_F^2 \\
&+\frac{\rho}{2}\|H_s-H_sZ_s-E_s+\frac{Y_2}{\rho}\|_F^2+\frac{\rho}{2}\sum_{i=1}^{V}\|H_p^{(i)}- \\
&H_p^{(i)}*Z_p^{(i)}-E_p^{(i)}+\frac{Y_3^{(i)}}{\rho}\|_F^2+\frac{\rho}{2}\|\mathcal{W}-\mathcal{Z}+\frac{\mathcal{Y}_4}{\rho}\|_F^2, \\
&s.t.\ \{P_s^{(i)'}P_s^{(i)}=I,\ \ P_p^{(i)'}P_p^{(i)}=I\}_{i=1}^{V}, \qquad (4)
\end{aligned}$$

where $\{Y_1^{(i)}\}$, Y_2, $\{Y_3^{(i)}\}$, and $\mathcal{Y}_4$ are the Lagrange multipliers, and $\rho>0$ is the penalty parameter. Within the ADMM framework, the iterative optimization scheme involves seven variables (or variable sets) and consists of the following steps:

$$\begin{aligned}
&\arg\min_{\mathcal{E}}\|\mathcal{E}\|_{2,1}+\frac{\rho}{2}\sum_{i=1}^{V}\|X^{(i)}-P_s^{(i)}H_s-P_p^{(i)}H_p^{(i)}-E_x^{(i)} \\
&+\frac{Y_1^{(i)}}{\rho}\|_F^2+\frac{\rho}{2}\|H_s-H_sZ_s-E_s+\frac{Y_2}{\rho}\|_F^2+\frac{\rho}{2}\sum_{i=1}^{V}\|H_p^{(i)}-H_p^{(i)}Z_p^{(i)} \\
&-E_p^{(i)}+\frac{Y_3^{(i)}}{\rho}\|_F^2; \qquad (5)
\end{aligned}$$

$$\arg\min_{\mathcal{W}}\ \alpha\|\mathcal{W}\|_*+\frac{\rho}{2}\|\mathcal{W}-\mathcal{Z}+\frac{\mathcal{Y}_4}{\rho}\|_F^2; \qquad (6)$$

$$\begin{aligned}
&\arg\min_{\mathcal{Z}}\ \|H_s-H_sZ_s-E_s+\frac{Y_2}{\rho}\|_F^2+\sum_{i=1}^{V}\|H_p^{(i)}-H_p^{(i)}Z_p^{(i)} \\
&-E_p^{(i)}+\frac{Y_3^{(i)}}{\rho}\|_F^2+\|\mathcal{W}-\mathcal{Z}+\frac{\mathcal{Y}_4}{\rho}\|_F^2; \qquad (7)
\end{aligned}$$

$$\begin{aligned}
&\arg\min_{\{P_s^{(i)}\}}\sum_{i=1}^{V}\|X^{(i)}-P_s^{(i)}H_s-P_p^{(i)}H_p^{(i)}-E_x^{(i)}+\frac{Y_1^{(i)}}{\rho}\|_F^2, \\
&\quad s.t.\{P_s^{(i)'}P_s^{(i)}=I\}_{i=1}^{V}; \qquad (8)
\end{aligned}$$

$$\begin{aligned}
&\arg\min_{\{P_p^{(i)}\}}\sum_{i}^{V}\|X^{(i)}-P_s^{(i)}H_s-P_p^{(i)}H_p^{(i)}-E_x^{(i)}+\frac{Y_1^{(i)}}{\rho}\|_F^2, \\
&\quad s.t.\{P_p^{(i)'}P_p^{(i)}=I\}_{i=1}^{V}; \qquad (9)
\end{aligned}$$

$$\begin{aligned}
&\arg\min_{H_s}\beta\sum_{i=1}^{V}\|H_s'H_p^{(i)}\|_F^2+\frac{\rho}{2}\sum_{i}^{V}\|X^{(i)}-P_s^{(i)}H_s-P_p^{(i)}H_p^{(i)}- \\
&E_x^{(i)}+\frac{Y_1^{(i)}}{\rho}\|_F^2+\frac{\rho}{2}\|H_s-H_sZ_s-E_s+\frac{Y_2}{\rho}\|_F^2; \qquad (10)
\end{aligned}$$

$$\begin{aligned}
&\arg\min_{\{H_p^{(i)}\}}\beta\sum_{i=1}^{V}\sum_{j\neq i}^{V}\|H_p^{(i)'}H_p^{(j)}\|_F^2+\frac{\rho}{2}\sum_{i=1}^{V}\|X^{(i)}-P_s^{(i)}H_s- \\
&P_p^{(i)}H_p^{(i)}-E_x^{(i)}+\frac{Y_1^{(i)}}{\rho}\|_F^2+\frac{\rho}{2}\sum_{i=1}^{V}\|H_p^{(i)}-H_p^{(i)}Z_p^{(i)}-E_p^{(i)} \\
&+\frac{Y_3^{(i)}}{\rho}\|_F^2; \qquad (11)
\end{aligned}$$

More specifically, one iteration of the optimization algorithm is updated as follows.

Step 1. $\mathcal{E}$–subproblem: (5) with respect to $\mathcal{E}=\{E_x^{(1)},\ldots,E_x^{(V)},E_s,E_p^{(1)},\ldots,E_p^{(V)}\}$ can be separated into $2V+1$ independent $l_{2,1}$-norm minimization problems, each corresponding to an element in the set $\mathcal{E}$. Taking the optimization of E_s as an example:

$$\arg\min_{E_s}\ \frac{1}{\rho}\|E_s\|_{2,1}+\frac{1}{2}\|E_s-\left(H_s-H_sZ_s+\frac{Y_2}{\rho}\right)\|_F^2. \qquad (12)$$

(12) is a typical group lasso problem, solvable by column-wise thresholding. Let $T_1=H_s-H_sZ_s+\frac{Y_2}{\rho}$ be a temporal matrix. The optimal E_s^* is achieved by setting

$$E_s^*(:,k)=\left(1-\frac{1}{\rho\|T_1(:,k)\|_F}\right)_+T_1(:,k), \qquad (13)$$

where $E_s(:,k)$ and $T_1(:,k)$ are the k-th columns of E_s and T_1 respectively, and $(\cdot)_+=\max(\cdot,0)$ denotes the positive part of $(\cdot)$.

Step 2. $\mathcal{W}$–subproblem: Let $\mathcal{W} = \{W_s, W_p^{(1)}, \ldots, W_p^{(V)}\}$. The optimization of (6) with respect to $\mathcal{W}$ consists of $V+1$ independent nuclear norm minimization problems. Here, we take the optimization of W_s as an example:

$$\arg\min_{W_s} \frac{\alpha}{\rho}\|W_s\|_* + \frac{1}{2}\|W_s - \left(Z_s - \frac{Y_4}{\rho}\right)\|_F^2, \tag{14}$$

where Y_4 is the element in $\mathcal{Y}_4$ corresponding to W_s. The nuclear norm minimization problem can be solved by singular value thresholding [44]. Specifically, if the Singular Value Decomposition (SVD) of $Z_s - \frac{Y_4}{\rho}$ is $U_W \Sigma_W V_W'$, the optimal solution to (14) is obtained as:

$$W_s^* = U_W \left(\Sigma_W - diag(\frac{\alpha}{\rho})\right)_+ V_W', \tag{15}$$

where $diag(\frac{\alpha}{\rho})$ denotes a diagonal matrix whose elements are filled with $\frac{\alpha}{\rho}$.

Step 3. $\mathcal{Z}$–subproblem: (7) with respect to $\mathcal{Z} = \{Z_s, Z_p^{(1)}, \ldots, Z_p^{(V)}\}$ can be separated into $V+1$ independent F-norm minimization problems. We take the optimization of Z_s as an example:

$$\arg\min_{Z_s} \|H_s Z_s - \left(H_s - E_s + \frac{Y_2}{\rho}\right)\|_F^2 + \|Z_s - \left(W_s + \frac{Y_4}{\rho}\right)\|_F^2. \tag{16}$$

(16) can be efficiently computed by setting the derivation with respect to Z_s to zero. Specifically, let the temporary matrices $T_2 = H_s - E_s + \frac{Y_2}{\rho}$ and $T_3 = W_s + \frac{Y_4}{\rho}$. The optimal Z_s^* is given by:

$$Z_s^* = (H_s' H_s + I)^{-1}(H_s' T_2 + T_3). \tag{17}$$

Step 4. $\{P_s^{(i)}\}$–subproblem: (8) with respect to $\{P_s^{(i)}\}$ can be decoupled into V independent orthogonal Procrustes problems. Here, we take the optimization of $P_s^{(i)}$ as an example. Eq (8) is thus reduced to

$$\arg\min_{P_s^{(i)}} \|P_s^{(i)} H_s - \left(X^{(i)} - P_p^{(i)} H_p^{(i)} - E_x^{(i)} + \frac{Y_1^{(i)}}{\rho}\right)\|_F^2,$$
$$\text{s.t. } P_s^{(i)'} P_s^{(i)} = I. \tag{18}$$

We solve the orthogonal Procrustes problem using the following Theorem 1 [45].

Theorem 1: Given two matrices A and B, the solution to equation

$$\arg\min_{\Omega} \|\Omega A - B\|_F^2, \quad s.t. \quad \Omega'\Omega = I, \tag{19}$$

is obtained as follows. Let $BA' = U_\Omega \Sigma_\Omega V_\Omega'$ be the SVD of BA'. The optimal solution is given by $\Omega^* = U_\Omega V_\Omega'$.

Step 5. $\{P_p^{(i)}\}$–subproblem: Similar to the optimization of $\{P_s^{(i)}\}$, the optimization of $\{P_p^{(i)}\}$ reduces to solving V orthogonal Procrustes problems. Specifically, the optimization of $P_p^{(i)}$ is formulated as follows:

$$\arg\min_{P_p^{(i)}} \|P_p^{(i)} H_p - \left(X^{(i)} - P_s^{(i)} H_s^{(i)} - E_x^{(i)} + \frac{Y_1^{(i)}}{\rho}\right)\|_F^2,$$
$$\text{s.t. } P_p^{(i)'} P_p^{(i)} = I. \tag{20}$$

Step 6. H_s–subproblem: The optimization of H_s reduces to an F-norm minimization problem. Let $\{T_4^{(i)} = X^{(i)} - P_p^{(i)} H_p^{(i)} - E_x^{(i)} + \frac{Y_1^{(i)}}{\rho}\}$, $T_5 = I - Z_s$, and $T_6 = E_s - \frac{Y_2}{\rho}$ be temporary variables. The derivation of (10) with respect to H_s is:

$$2\beta \sum_{i=1}^{V} H_p^{(i)} H_p^{(i)'} H_s + \rho \sum_{i=1}^{V} P_s^{(i)'}(P_s^{(i)} H_s - T_4^{(i)}) + \rho(H_s T_5 - T_6)T_5'. \tag{21}$$

By setting (21) to zero, the optimal H_s^* can be calculated using an off-the-shelf Lyapunov equation solver.

Step 7. $\{H_p^{(i)}\}$–subproblem: The optimization of $\{H_p^{(i)}\}$ can be solved separately by calculating the derivations. We take $H_p^{(i)}$ as an example. Fixing $\{H_p^{(j)}\}_{j\neq i}^{V}$, the derivation of (11) with respect to $H_p^{(i)}$ is:

$$2\beta \sum_{j\neq i}^{V} H_p^{(j)} H_p^{(j)'} H_p^{(i)} + \rho P_p^{(i)'}(P_p^{(i)} H_p^{(i)} - T_7) + \rho(H_p^{(i)} T_8 - T_9)T_8', \tag{22}$$

where $T_7 = X^{(i)} - P_s^{(i)} H_s - E_x^{(i)} + \frac{Y_1^{(i)}}{\rho}$, $T_8 = I - Z_p^{(i)}$, and $T_9 = E_s - \frac{Y_2}{\rho}$ are temporary variables used for conciseness. Similar to the optimization of H_s, the optimal $H_p^{(i)*}$ is obtained by setting (22) to zero and then solving it using a Lyapunov equation solver.

Step 8. update Lagrange multipliers and the penalty parameter After updating all variables/variable sets, the Lagrange multipliers and the penalty parameter are updated according to

$$\begin{cases} \{Y_1^{(i)} = Y_1^{(i)} + \rho(X^{(i)} - P_s^{(i)} H_s - P_p^{(i)} H_p^{(i)} - E_x^{(i)})\}, \\ Y_2 = Y_2 + \rho(H_s - H_s Z_s - E_s, \\ \{Y_3^{(i)} = Y_3^{(i)} + \rho(H_p^{(i)} - H_p^{(i)} Z_p^{(i)} - E_p^{(i)})\}, \\ \mathcal{Y}_4 = \mathcal{Y}_4 + \rho(\mathcal{W} - \mathcal{Z}), \\ \rho \ = \min(\varphi * \rho, \ \rho_{\max}). \end{cases} \tag{23}$$

In the context of ADMM, a small value of the penalty parameter ρ tends to minimize the objective function at the expense of increasing the residuals. Conversely, a large value of ρ places a strong penalty on violations of primal feasibility and hence produces small residuals. Theoretical analysis suggests initializing ρ to a small positive number and then increasing it by a positive factor φ. This approach enables superlinear convergence as ρ grows towards infinity during the iteration process [42], [43]. Furthermore, this strategy reduces the dependence of optimization performance on the specific choice of a fixed ρ value, which is advantageous for machine learning algorithms. The iterative scheme is repeated until the convergence condition is satisfied, which is defined by the residuals related to the four constraints:

$$\begin{cases} \gamma_1 = \max(\{\|X^{(i)} - P_s^{(i)} H_s - P_p^{(i)} H_p^{(i)} - E_x^{(i)})\|_{\max}\}) \leq \epsilon; \\ \gamma_2 = \|H_s - H_s Z_s - E_s\|_{\max} \leq \epsilon; \\ \gamma_3 = \max(\{\|H_p^{(i)} - H_p^{(i)} Z_p^{(i)} - E_p^{(i)})\|_{\max}\}) \leq \epsilon; \\ \gamma_4 = \|\mathcal{W} - \mathcal{Z}\|_{\max} \leq \epsilon, \end{cases} \tag{24}$$

Algorithm 1: OLR-MVC.

Input : Multi-view data: $\{X^{(i)}\}$; hyper-parameters: α, β; latent dimension k; optimization parameters $\rho = 10^{-1}$, $\rho^{max} = 10^{6}$, $\varphi = 2$, $\epsilon = 10^{-3}$.
Output: The affinity matrix.

repeat
- Update $\mathcal{E}$ by solving $2V + 1$ independent subtasks in the form of Eq. (12) according to Eq. (13);
- Update $\mathcal{W}$ by solving $V + 1$ independent subtasks in the form of Eq. (14) according to Eq. (15);
- Update $\mathcal{Z}$ by solving $V + 1$ independent subtasks in the form of Eq. (16) by setting the derivations to zero;
- Update $\{P_s^{(i)}\}$ by solving V independent subtasks in the form of Eq. (18) according to Theorem 1;
- Update $\{P_p^{(i)}\}$ by solving V independent subtasks in the form of Eq. (20) according to Theorem 1;
- Update H_s by setting Eq. (21) to zero and solving it using the Lyapunov solver;
- Update $\{H_p^{(i)}\}$ by solving V independent subtasks in the form of Eq. (22) using the Lyapunov solver;
- Update multipliers and penalty parameter by Eq. (23).

until *converged*;
Obtain the affinity matrix A with Eq. (25).

where $\|\cdot\|_{\max}$ denotes the maximum norm and $\max(\cdot)$ denotes the maximum value in current set. Once the algorithm converges, the affinity matrix is given by:

$$A = \frac{|Z_s| + |Z_s'|}{2} + \frac{1}{V}\left(\sum_{i=1}^{V} \frac{|Z_p^{(i)}| + |Z_s^{(i)'}|}{2}\right). \tag{25}$$

We then apply the spectral clustering algorithm [46] on A to obtain the final clustering results. The entire optimization procedure is summarized in Algorithm 1.

D. Complexity Analysis

Algorithm 1 exhibits an iterative behavior, comprising seven block variables. The main computational costs in each iteration are analyzed as follows: (1) $\mathcal{E}$ subproblem: This step consists of $2V + 1$ group Lasso problems, each involving column-wise thresholding. The computational cost of these operations is negligible compared with other steps; (2) $\mathcal{W}$ subproblem: This step involves $V + 1$ singular value shrinkage operations, with a computational cost of $\mathcal{O}(N^3)$ for each operation; (3) $\mathcal{Z}$ subproblem: This step requires matrix inversion and multiplication, with a cost of $\mathcal{O}(N^3)$ for each of the $V + 1$ subtasks; (4) $\{P_s^{(i)}\}$ and $\{P_p^{(i)}\}$ subproblems: This step involves solving $2V$ orthogonal Procrustes problems, requiring SVD operations with a computational cost of $\mathcal{O}(\max(D, N)^3)$, where $D = \max(\{d_i\})$ is the maximum dimension of observed features; (5) H_s and $\{H_p^{(i)}\}$ subproblems: These optimizations rely on solving Lyapunov equations, which require a cost of $\mathcal{O}(D^3 + N^3)$ for each subtask. Overall, the computational cost of Algorithm 1 is $\mathcal{O}(D^3 + N^3)$.

IV. Experiments

In this section, we evaluate the performance of OLR-MVC and present an in-depth analysis to enhance the understanding of our model.

A. Experimental Setup

We compare OLR-MVC against 12 state-of-the-art peer algorithms, as well as the standard spectral clustering algorithm. The experimental details are outlined as follows.

1) Dataset Overview:

- ORL:[1] It contains 400 face images from 40 persons. Three types of features are extracted: intensity (4096-D), LBP (3304-D), and Gabor (6750-D).
- Yale:[2] This dataset includes 165 face images from 15 individuals. It also includes intensity, LBP, and Gabor features.
- MSRC-v1 [47]: This database consists of 210 scene images from seven classes. Each image is represented by five different features: LBP (256-D), HOG (100-D), GIST (512-D), CENTRIST (1302-D), and SIFT (210-D).
- 3sources:[3] This dataset contains 169 news documents from three organizations. The multi-view features included term frequencies (3560-D), content-bearing terms (3631-D), and story identifiers (3068-D).
- BBC4views:[4] It collects 685 documents belonging to five classes. Four features are extracted, with dimensions 4659, 4633, 4665, and 4684.
- Flower17:[5] It consists of 17 flower categories, with 80 images per category. The χ^2 distance matrices of seven features constitute the multiple views, each with a dimension equal to the number of samples (1360-D).
- Scene15 [48]: It contains 4485 images from 15 categories, described by three features: pyramid histograms (1800-D), PRI-CoLBP (1180-D), and centrist features (1240-D).
- Caltech-101 [49]: It contains 9144 generic object images. Six features are extracted: Gabor (48-D), wavelet moments (40-D), census transform histogram (254-D), HOG (1984-D), Gist (512-D), and LBP (928-D).

2) Competitors: The competitors includes six latent representation-based methods and six algorithms from other categories. Additionally, the standard spectral clustering algorithm is included to establish a baseline. Specifically,

- SPC: standard spectral clustering algorithm [46]. It processes each individual view, and the best-performing results are reported.
- LMSC: latent multi-view subspace clustering [4]. It seeks the latent representation by concatenating all views and simultaneously performs self-expressiveness learning.

[1][Online]. Avilable: http://www.uk.research.att.com/facedatabase. html
[2][Online]. Avilable: http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html
[3][Online]. Avilable: http://mlg.ucd.ie/datasets/3sources.html
[4][Online]. Avilable: http://mlg.ucd.ie/datasets/segment.html
[5][Online]. Avilable: https://www.robots.ox.ac.uk/vgg/data/flowers/17/index.html

- MCLES: multi-view clustering in latent embedding space [5]. It clusters multi-view data in a latent embedding space while simultaneously learning the global data structure and the cluster indicator matrix.
- LCRSR: latent complete row space recovery for multi-view subspace clustering [6]. It aims to recover the row space of the common latent representation for clustering.
- RMCLES: relaxed multi-view clustering in latent embedding space [7]. It learns latent embedding, global similarity, and cluster indicator matrix in a unified framework.
- LRMVC: latent representation guided multi-view clustering [8]. It accomplishes three subtasks: latent representation extraction, similarity graph learning, and cluster allocation.
- OMVCDR: one-step multi-view clustering with diverse representation [10]. It projects data into latent spaces, and incorporates multi-view learning and k-means into a unified model.
- FSMSC: fast self-guided multi-view subspace clustering [50]. It integrates view-shared anchor and global-local self-guidance learning into a unified model.
- MLRR: multi-view low-rank representation [51]. It considers symmetric low-rank representations and uses the angular information of principal directions to construct the affinity matrix.
- UOMvSC: unified one-step multi-view spectral clustering [52]. It integrates spectral embedding and k-means into a unified framework to obtain the clustering results.
- SGF: similarity graph fusion [53]. It models multi-view consistency and inconsistency in a unified model and fuses the consistent parts for clustering.
- DGF: distance (dissimilarity) graph fusion [53]. It uses the same learning paradigm as SGF but generates initial graphs based on distance/dissimilarity.
- TSSR: tensorized scaled simplex representation for multi-view clustering [31]. It leverages a low-rank tensor constraint to capture the consensus and complementary information.

3) Evaluation Metrics: To comprehensively compare all competing algorithms, we employ six commonly used evaluation metrics to assess the clustering performance, including the normalized mutual information (NMI), accuracy (ACC), adjusted rand index (ARI), F-score, precision, and recall. Experiments were conducted using MATLAB R2022a on a server with an Intel Core i9-9920X 12-core CPU and a 128 GB RAM. Each algorithm was run ten times, and the average results are reported. For each run, the order of data samples was randomly shuffled, and all competitors used the same shuffling seed to avoid random disturbances. Since the NMI score provides a more comprehensive understanding of the clustering results [54], we report the clustering results with the highest NMI values when the optimal results do not align with a single parameter configuration.

4) Parameter Settings: For all competing algorithms, we obtained the source codes from the respective authors and used the default settings as recommended in their published papers. Our OLR-MVC model has three hyper-parameters: the trade-off parameters α, β, and the dimension of latent representations, $hdim$. Empirically, we tune α and β from $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$, and the optimal $hdim$ is chosen from [60,200] with a step size of 20.

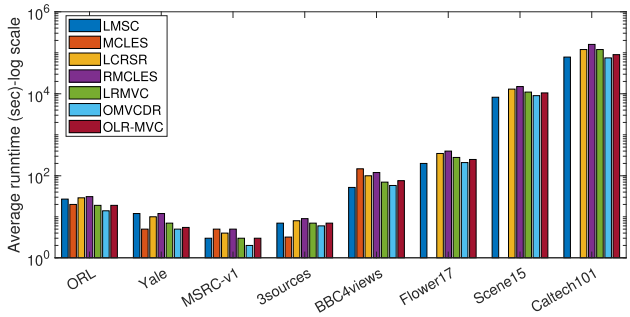

Fig. 3. Average runtime of different latent representation-based methods on real-world datasets. The empty bar indicates that MCLES encounters problems with the respective datasets.

B. Experimental Results

1) Numerical Comparison: We evaluate the performance of OLR-MVC compared to state-of-the-art methods. The clustering results, assessed using six evaluation metrics, are presented in Tables III and IV. Overall, our OLR-MVC model achieves the best or near-optimal performance in most cases. From the experimental results, we derive the following key findings:

- The proposed OLR-MVC model achieves the best results across all databases according to four out of six evaluation metrics: NMI, ACC, precision, and F-score. This demonstrates the robust performance of OLR-MVC. We attribute the success of OLR-MVC to its flexible latent representation learning framework.
- OLR-MVC performs best on seven out of eight databases based on the ARI score, securing third place on the Caltech101 database. The ARI score evaluates pairs of samples that are assigned to the same or different clusters in two clustering results. Variations in the ARI score can occur if categories are particularly small or large. The slightly lower ARI score on Caltech101 can be attributed to the imbalanced class sizes.
- OLR-MVC achieves the highest recall scores on ORL and Yale, and ranks second or third on MSRC-V1, 3Sources, BBC4Views, Flower17, and Caltech101. It ranks fourth in recall on Scene15. This may be due to imbalanced class distributions, which cause the spectral clustering algorithm to disproportionately prioritize the larger classes. As a result, samples from smaller classes may be incorrectly assigned to larger classes, potentially leading to a lower recall. Since there is an inherent trade-off between recall and precision, we focus on the F-score, which balances both metrics to provide a comprehensive evaluation. OLR-MVC consistently achieves the highest F-scores, demonstrating its

TABLE III
AVERAGE CLUSTERING RESULTS AND THE CORRESPONDING STANDARD DEVIATIONS FOR COMPETING METHODS ON ORL, YALE, MSRC-V1, AND 3SOURCES DATABASES. (THE BEST RESULTS AND THE SECOND BEST RESULTS ARE MARKED IN **BOLD** AND ***ITALIC BOLD***, RESPECTIVELY).

		NMI	ACC	ARI	F-score	Precision	Recall
ORL	SPC [46]	0.9106 ± 0.0075	0.8275 ± 0.0211	0.7564 ± 0.0203	0.7621 ± 0.0198	0.7363 ± 0.0225	0.7899 ± 0.0178
	LMSC [4]	0.9181 ± 0.0111	0.8265 ± 0.0173	0.7243 ± 0.0265	0.7502 ± 0.0251	0.6720 ± 0.0279	0.8496 ± 0.0284
	MCLES [5]	0.9015 ± 0.0154	0.8027 ± 0.0287	0.6778 ± 0.0446	0.7084 ± 0.0492	0.6239 ± 0.0647	0.8228 ± 0.0263
	LCRSR [6]	0.7755 ± 0.0072	0.6065 ± 0.0174	0.4482 ± 0.0234	0.4736 ± 0.0170	0.4021 ± 0.0186	0.5764 ± 0.0168
	RMCLES [7]	0.8564 ± 0.0091	0.7210 ± 0.0128	0.3012 ± 0.0289	0.3560 ± 0.0283	0.2245 ± 0.0222	0.8660 ± 0.0125
	LRMVC [8]	0.7559 ± 0.0000	0.6425 ± 0.0000	0.1910 ± 0.0000	0.4293 ± 0.0000	0.3597 ± 0.0000	0.5322 ± 0.0000
	OMVCDR [10]	0.8374 ± 0.0175	0.7180 ± 0.0369	0.5968 ± 0.0448	0.6041 ± 0.0409	0.5447 ± 0.0507	0.6798 ± 0.0319
	FSMSC [50]	0.8790 ± 0.0149	0.7578 ± 0.0302	0.6376 ± 0.0328	0.6645 ± 0.0329	0.5816 ± 0.0336	0.7753 ± 0.0351
	MLRR [51]	***0.9386 ± 0.0043***	0.8345 ± 0.0103	0.7588 ± 0.0127	0.7911 ± 0.0116	0.7028 ± 0.0157	***0.9052 ± 0.0164***
	UOMvSC [52]	0.9359 ± 0.0015	0.8618 ± 0.0024	0.7728 ± 0.0051	0.8025 ± 0.0042	0.7329 ± 0.0079	0.8869 ± 0.0037
	SGF [53]	0.9368 ± 0.0040	0.8725 ± 0.0118	***0.7919 ± 0.0131***	***0.8201 ± 0.0128***	***0.7658 ± 0.0168***	0.8828 ± 0.0103
	DGF [53]	0.9302 ± 0.0045	***0.8750 ± 0.0123***	0.7812 ± 0.0125	0.8061 ± 0.0122	0.7530 ± 0.0160	0.8672 ± 0.0109
	TSSR [31]	0.6826 ± 0.0147	0.4775 ± 0.0186	0.2907 ± 0.0102	0.3150 ± 0.0139	0.2315 ± 0.0127	0.4928 ± 0.0203
	OLR-MVC (ours)	**0.9470 ± 0.0050**	**0.8885 ± 0.0104**	**0.8206 ± 0.0152**	**0.8377 ± 0.0150**	**0.7781 ± 0.0196**	**0.9075 ± 0.0156**
Yale	SPC [46]	0.6895 ± 0.0100	***0.6733 ± 0.0205***	0.4875 ± 0.0224	0.5197 ± 0.0206	0.5048 ± 0.0265	0.5358 ± 0.0156
	LMSC [4]	0.6774 ± 0.011	0.6727 ± 0.0081	0.4668 ± 0.0188	0.4543 ± 0.0108	0.3700 ± 0.0140	0.5895 ± 0.0227
	MCLES [5]	***0.7029 ± 0.0232***	0.6527 ± 0.0398	0.4872 ± 0.0385	0.4153 ± 0.0577	0.3137 ± 0.0713	0.6296 ± 0.0217
	LCRSR [6]	0.6146 ± 0.0220	0.5473 ± 0.0164	0.3865 ± 0.0275	0.3843 ± 0.0414	0.3011 ± 0.0404	0.5347 ± 0.0471
	RMCLES [7]	0.7015 ± 0.0000	0.6848 ± 0.0000	0.4852 ± 0.0000	0.5010 ± 0.0000	0.4294 ± 0.0000	0.6012 ± 0.0000
	LRMVC [8]	0.5734 ± 0.0000	0.5636 ± 0.0000	0.3692 ± 0.0000	0.3768 ± 0.0000	0.3035 ± 0.0000	0.4970 ± 0.0000
	OMVCDR [10]	0.6331 ± 0.0200	0.6024 ± 0.0157	0.4341 ± 0.0257	0.3976 ± 0.0330	0.3213 ± 0.0467	0.5312 ± 0.0284
	FSMSC [50]	0.6687 ± 0.0391	0.6236 ± 0.0430	0.4552 ± 0.0488	0.4526 ± 0.0539	0.3722 ± 0.0617	0.5846 ± 0.0518
	MLRR [51]	0.6933 ± 0.0256	0.6570 ± 0.0354	***0.5060 ± 0.0374***	0.4479 ± 0.0464	0.3515 ± 0.0490	***0.6210 ± 0.0316***
	UOMvSC [52]	0.6903 ± 0.0080	0.6303 ± 0.0057	0.4803 ± 0.0117	0.3906 ± 0.0083	0.2859 ± 0.0067	0.6164 ± 0.0106
	SGF [53]	0.6918 ± 0.0050	0.6400 ± 0.0031	0.5014 ± 0.0070	***0.5327 ± 0.0065***	**0.5177 ± 0.0072**	0.5486 ± 0.0060
	DGF [53]	0.6868 ± 0.0018	0.6364 ± 0.0000	0.4941 ± 0.0017	0.5259 ± 0.0016	***0.5105 ± 0.0012***	0.5422 ± 0.0021
	TSSR [31]	0.6200 ± 0.0237	0.6158 ± 0.0286	0.3740 ± 0.0264	0.4097 ± 0.0313	0.3696 ± 0.0329	0.4607 ± 0.0353
	OLR-MVC (ours)	**0.7980 ± 0.0118**	**0.7642 ± 0.0226**	**0.6340 ± 0.0198**	**0.5917 ± 0.0272**	0.5004 ± 0.0395	**0.7268 ± 0.0187**
MSRC-v1	SPC [46]	0.6489 ± 0.0000	0.7000 ± 0.0000	0.5530 ± 0.0000	0.6161 ± 0.0000	0.6056 ± 0.0000	0.6269 ± 0.0000
	LMSC [4]	0.6589 ± 0.0676	0.7571 ± 0.0651	0.5693 ± 0.0863	0.6468 ± 0.0686	0.6006 ± 0.0859	0.7049 ± 0.0500
	MCLES [5]	0.7803 ± 0.0127	0.8729 ± 0.0087	0.7301 ± 0.0157	0.7680 ± 0.0135	0.7584 ± 0.0135	0.7779 ± 0.0137
	LCRSR [6]	0.7509 ± 0.0117	0.8448 ± 0.0154	0.6813 ± 0.0213	0.7268 ± 0.0180	0.7077 ± 0.0238	0.7471 ± 0.0147
	RMCLES [7]	0.6786 ± 0.0346	0.6562 ± 0.0294	0.4560 ± 0.0471	0.5663 ± 0.0401	0.4379 ± 0.0426	0.8049 ± 0.0353
	LRMVC [8]	0.6871 ± 0.0000	0.7286 ± 0.0000	0.3680 ± 0.0000	0.6402 ± 0.0000	0.5731 ± 0.0000	0.7251 ± 0.0000
	OMVCDR [10]	0.7679 ± 0.0197	0.8633 ± 0.0121	0.7158 ± 0.0214	0.7556 ± 0.0184	0.7471 ± 0.0182	0.7644 ± 0.0186
	FSMSC [50]	0.7783 ± 0.0381	0.8490 ± 0.0495	0.7103 ± 0.0750	0.7583 ± 0.0488	0.7395 ± 0.0717	0.7803 ± 0.0202
	MLRR [51]	0.7685 ± 0.0000	0.8571 ± 0.0000	0.7130 ± 0.0000	0.7537 ± 0.0000	0.7368 ± 0.0000	0.7714 ± 0.0000
	UOMvSC [52]	0.7877 ± 0.0061	0.8252 ± 0.0032	0.6857 ± 0.0069	0.7681 ± 0.0055	0.7011 ± 0.0053	0.8491 ± 0.0056
	SGF [53]	0.8172 ± 0.0061	0.8333 ± 0.0041	0.7283 ± 0.0044	0.7837 ± 0.0028	0.7147 ± 0.0043	**0.8673 ± 0.0026**
	DGF [53]	***0.8219 ± 0.0094***	***0.8857 ± 0.0106***	***0.7691 ± 0.0173***	***0.8016 ± 0.0112***	***0.7895 ± 0.0142***	0.8141 ± 0.0834
	TSSR [31]	0.3943 ± 0.0177	0.5033 ± 0.0160	0.2482 ± 0.0157	0.3820 ± 0.0170	0.3315 ± 0.0252	0.4595 ± 0.0617
	OLR-MVC (ours)	**0.8494 ± 0.0056**	**0.9205 ± 0.0032**	**0.8178 ± 0.0061**	**0.8433 ± 0.0052**	**0.8358 ± 0.0049**	***0.8510 ± 0.0056***
3sources	SPC [46]	0.6286 ± 0.0000	0.6391 ± 0.0000	0.4751 ± 0.0000	0.5770 ± 0.0000	0.6909 ± 0.0000	0.4953 ± 0.0000
	LMSC [4]	0.7044 ± 0.0166	0.8077 ± 0.0094	0.5481 ± 0.0201	0.7464 ± 0.0193	0.7579 ± 0.0164	0.7354 ± 0.0220
	MCLES [5]	0.6105 ± 0.0442	0.7172 ± 0.0334	0.4225 ± 0.0751	0.6692 ± 0.0390	0.5594 ± 0.0496	0.8400 ± 0.0660
	LCRSR [6]	0.6477 ± 0.0042	0.7385 ± 0.0025	0.3594 ± 0.0047	0.7502 ± 0.0024	0.6356 ± 0.0020	**0.9152 ± 0.0043**
	RMCLES [7]	0.4663 ± 0.0620	0.6201 ± 0.0481	0.2462 ± 0.1095	0.5275 ± 0.0690	0.4157 ± 0.0721	0.7403 ± 0.1127
	LRMVC [8]	0.6912 ± 0.0000	0.7929 ± 0.0000	0.5447 ± 0.0000	0.7198 ± 0.0000	0.6541 ± 0.0000	0.8002 ± 0.0000
	OMVCDR [10]	0.4110 ± 0.0176	0.6231 ± 0.0108	0.2682 ± 0.0187	0.5829 ± 0.0241	0.4994 ± 0.0544	0.7147 ± 0.0751
	FSMSC [50]	0.6724 ± 0.0328	0.7538 ± 0.0294	0.4449 ± 0.0686	0.7334 ± 0.0492	0.6826 ± 0.0574	0.7988 ± 0.0802
	MLRR [51]	***0.7859 ± 0.0000***	***0.8698 ± 0.0000***	***0.7317 ± 0.0000***	***0.8641 ± 0.0000***	***0.8308 ± 0.0000***	***0.9003 ± 0.0000***
	UOMvSC [52]	0.7433 ± 0.0031	0.8172 ± 0.0086	0.5597 ± 0.0226	0.7748 ± 0.0059	0.7646 ± 0.0473	0.7935 ± 0.0719
	SGF [53]	0.6906 ± 0.0000	0.6864 ± 0.0000	0.5481 ± 0.0000	0.6417 ± 0.0000	0.7201 ± 0.0000	0.5787 ± 0.0000
	DGF [53]	0.6860 ± 0.0046	0.6911 ± 0.0025	0.5587 ± 0.0061	0.6470 ± 0.0049	0.7512 ± 0.0054	0.5682 ± 0.0046
	TSSR [31]	0.6405 ± 0.0091	0.7615 ± 0.0029	0.3956 ± 0.0043	0.7243 ± 0.0054	0.6877 ± 0.0039	0.7651 ± 0.0074
	OLR-MVC (ours)	**0.8054 ± 0.0000**	**0.8817 ± 0.0000**	**0.7658 ± 0.0000**	**0.8818 ± 0.0000**	**0.8684 ± 0.0000**	0.8957 ± 0.0000

strong overall performance. However, in certain applications, false negatives have more severe consequences than false positives, making high recall preferable. This is evident in tasks such as cancer prediction and scene graph generation [55]. To improve recall by reducing false negatives, we can apply a smaller weight to the low-rank constrained term to capture more potential relationships. Additionally, post-processing techniques can be employed to prioritize the recall score.

- The competitors, MLRR, SGF, and DGF, exhibit suboptimal performance on most smaller databases. This is likely because these databases have relatively clear graph structures, making the predefined similarity graphs useful for uncovering underlying data relationships. Additionally,

TABLE IV
AVERAGE CLUSTERING RESULTS AND THE CORRESPONDING STANDARD DEVIATIONS FOR COMPETING METHODS ON BBC4VIEWS, FLOWER17, SCENE15, AND CALTECH101 DATABASES. (THE BEST RESULTS AND THE SECOND BEST RESULTS ARE MARKED IN **BOLD** AND ***ITALIC BOLD***, RESPECTIVELY) MCLES ENCOUNTERS PROBLEMS ON RELATIVELY LARGE DATABASES, I.E., FLOWER17, SCENE15, AND CALTECH101.

		NMI	ACC	ARI	F-score	Precision	Recall
BBC4views	SPC [46]	0.6414 ± 0.0000	0.8511 ± 0.0000	0.6854 ± 0.0000	0.7627 ± 0.0000	0.7282 ± 0.0000	0.8006 ± 0.0000
	LMSC [4]	0.6729 ± 0.0124	0.8695 ± 0.0076	0.7164 ± 0.0122	0.7830 ± 0.0096	0.7822 ± 0.0081	0.7838 ± 0.0128
	MCLES [5]	0.6288 ± 0.0889	0.7860 ± 0.0836	0.5950 ± 0.1830	0.6961 ± 0.1323	0.6708 ± 0.1485	0.7255 ± 0.1105
	LCRSR [6]	0.5690 ± 0.0013	0.6964 ± 0.0000	0.4319 ± 0.0008	0.6040 ± 0.0006	0.5618 ± 0.0009	0.6529 ± 0.0002
	RMCLES [7]	0.4202 ± 0.0387	0.5635 ± 0.0235	0.3163 ± 0.0545	0.5356 ± 0.0391	0.4001 ± 0.0275	0.8123 ± 0.0777
	LRMVC [8]	0.6181 ± 0.0000	0.7635 ± 0.0000	0.4923 ± 0.0000	0.7062 ± 0.0000	0.6766 ± 0.0000	0.7385 ± 0.0000
	OMVCDR [10]	0.2996 ± 0.1231	0.5507 ± 0.0730	0.2516 ± 0.1411	0.4818 ± 0.0790	0.3872 ± 0.0802	0.6541 ± 0.1214
	FSMSC [50]	0.6745 ± 0.0251	0.8067 ± 0.0358	0.6141 ± 0.0954	0.7563 ± 0.0346	0.7505 ± 0.0700	0.7682 ± 0.0440
	MLRR [51]	***0.7471 ± 0.0033***	***0.8933 ± 0.0020***	***0.7644 ± 0.0030***	***0.8217 ± 0.0023***	0.7926 ± 0.0023	***0.8529 ± 0.0027***
	UOMvSC [52]	0.7015 ± 0.0026	0.8810 ± 0.0010	0.7383 ± 0.0043	0.8037 ± 0.0031	0.7540 ± 0.0040	**0.8605 ± 0.0019**
	SGF [53]	0.7142 ± 0.0011	0.8701 ± 0.0006	0.7405 ± 0.0014	0.8004 ± 0.0011	***0.8126 ± 0.0010***	0.7886 ± 0.0011
	DGF [53]	0.7126 ± 0.0020	0.8686 ± 0.0006	0.7313 ± 0.0011	0.7937 ± 0.0008	0.8023 ± 0.0006	0.7852 ± 0.0010
	TSSR [31]	0.5887 ± 0.0031	0.7524 ± 0.0012	0.4682 ± 0.0023	0.7040 ± 0.0027	0.6747 ± 0.0024	0.7360 ± 0.0030
	OLR-MVC (ours)	**0.7614 ± 0.0091**	**0.9143 ± 0.0036**	**0.8089 ± 0.0089**	**0.8543 ± 0.0069**	**0.8561 ± 0.0054**	0.8525 ± 0.0088
Flower17	SPC [46]	0.4320 ± 0.0033	0.4001 ± 0.0020	0.2265 ± 0.0010	0.2734 ± 0.0011	0.2621 ± 0.0014	0.2858 ± 0.0025
	LMSC [4]	0.4544 ± 0.0132	0.4695 ± 0.0201	0.2756 ± 0.0143	0.3174 ± 0.0130	0.2781 ± 0.0113	0.3701 ± 0.0207
	MCLES [5]	-	-	-	-	-	-
	LCRSR [6]	0.4599 ± 0.0081	0.4790 ± 0.0127	0.2845 ± 0.0108	0.3297 ± 0.0093	0.2937 ± 0.0109	0.3765 ± 0.0164
	RMCLES [7]	0.5092 ± 0.0111	0.4364 ± 0.0237	0.1972 ± 0.0166	0.2733 ± 0.0133	0.1783 ± 0.0123	***0.5892 ± 0.0210***
	LRMVC [8]	0.4162 ± 0.0000	0.2632 ± 0.0000	0.1473 ± 0.0000	0.2314 ± 0.0000	0.1385 ± 0.0000	**0.7018 ± 0.0000**
	OMVCDR [10]	0.4279 ± 0.0093	0.4103 ± 0.0185	0.2331 ± 0.0122	0.2878 ± 0.0104	0.2453 ± 0.0133	0.3492 ± 0.0168
	FSMSC [50]	0.5255 ± 0.0184	0.5538 ± 0.0275	0.3616 ± 0.0226	0.3929 ± 0.0229	0.3606 ± 0.0292	0.4326 ± 0.0202
	MLRR [51]	0.4837 ± 0.0062	0.4806 ± 0.0095	0.2950 ± 0.0074	0.3378 ± 0.0067	0.2989 ± 0.0140	0.3894 ± 0.0130
	UOMvSC [52]	0.5553 ± 0.0117	0.5732 ± 0.0128	0.3191 ± 0.0130	0.3680 ± 0.0138	0.3118 ± 0.0192	0.4506 ± 0.0182
	SGF [53]	0.5607 ± 0.0011	0.6004 ± 0.0014	0.4091 ± 0.0011	0.4441 ± 0.0010	0.4362 ± 0.0021	0.4522 ± 0.0012
	DGF [53]	***0.5772 ± 0.0015***	***0.6057 ± 0.0020***	***0.4233 ± 0.0019***	***0.4576 ± 0.0018***	***0.4468 ± 0.0018***	0.4690 ± 0.0021
	TSSR [31]	0.3510 ± 0.0086	0.3774 ± 0.0108	0.2073 ± 0.0089	0.2524 ± 0.0054	0.2115 ± 0.0081	0.3141 ± 0.0179
	OLR-MVC (ours)	**0.5968 ± 0.0063**	**0.6147 ± 0.0092**	**0.4310 ± 0.0116**	**0.4842 ± 0.0141**	**0.4722 ± 0.0181**	0.4974 ± 0.0081
Scene15	SPC [46]	0.5494 ± 0.0030	0.5436 ± 0.0188	0.3927 ± 0.0134	0.4366 ± 0.0112	0.4178 ± 0.0236	0.4581 ± 0.0036
	LMSC [4]	0.5397 ± 0.0131	0.5962 ± 0.0147	0.3663 ± 0.0129	0.4414 ± 0.0093	0.3791 ± 0.0155	0.5281 ± 0.0387
	MCLES [5]	-	-	-	-	-	-
	LCRSR [6]	0.5389 ± 0.0061	0.5846 ± 0.0154	0.3685 ± 0.0150	0.4351 ± 0.0124	0.3852 ± 0.0180	0.4998 ± 0.0070
	RMCLES [7]	0.5203 ± 0.0151	0.5835 ± 0.0144	0.3675 ± 0.0079	0.4360 ± 0.0114	0.3851 ± 0.0115	0.5024 ± 0.0302
	LRMVC [8]	0.5954 ± 0.3055	0.5405 ± 0.2749	0.3549 ± 0.1076	0.4308 ± 0.2352	0.3218 ± 0.1668	***0.6516 ± 0.4092***
	OMVCDR [10]	0.3707 ± 0.0158	0.3423 ± 0.0238	0.1801 ± 0.0200	0.2714 ± 0.0131	0.2156 ± 0.0181	0.3730 ± 0.0437
	FSMSC [50]	0.5208 ± 0.0122	0.5864 ± 0.0207	0.3732 ± 0.0185	0.4353 ± 0.0171	0.3933 ± 0.0214	0.4880 ± 0.0190
	MLRR [51]	0.5947 ± 0.0119	0.6149 ± 0.0220	0.3990 ± 0.0237	0.4875 ± 0.0193	0.4020 ± 0.0255	0.6192 ± 0.0059
	UOMvSC [52]	***0.6467 ± 0.0123***	***0.6435 ± 0.0206***	***0.4275 ± 0.0253***	***0.5133 ± 0.0197***	***0.4218 ± 0.0263***	**0.6567 ± 0.0068**
	SGF [53]	0.5768 ± 0.0080	0.5487 ± 0.0163	0.3930 ± 0.0177	0.4378 ± 0.0138	0.4138 ± 0.0184	0.4651 ± 0.0027
	DGF [53]	0.5941 ± 0.0087	0.5371 ± 0.0207	0.4080 ± 0.0225	0.4535 ± 0.0165	0.4082 ± 0.0242	0.5106 ± 0.0079
	TSSR [31]	0.4310 ± 0.0174	0.3978 ± 0.0156	0.2166 ± 0.0162	0.3041 ± 0.0128	0.2296 ± 0.0163	0.4504 ± 0.0491
	OLR-MVC (ours)	**0.6583 ± 0.0123**	**0.6653 ± 0.0206**	**0.4562 ± 0.0253**	**0.5160 ± 0.0197**	**0.4509 ± 0.0263**	0.6030 ± 0.0068
Caltech101	SPC [46]	0.4857 ± 0.0056	0.2266 ± 0.0137	0.1558 ± 0.0139	0.1700 ± 0.0136	0.2861 ± 0.0128	0.1209 ± 0.0148
	LMSC [4]	0.5048 ± 0.0043	0.4767 ± 0.0051	***0.2013 ± 0.0213***	0.4573 ± 0.0188	0.3418 ± 0.0202	0.6906 ± 0.0127
	MCLES [5]	-	-	-	-	-	-
	LCRSR [6]	0.4138 ± 0.0046	0.4012 ± 0.0028	0.1657 ± 0.0076	0.4444 ± 0.0292	0.3412 ± 0.0232	0.6372 ± 0.0100
	RMCLES [7]	0.3413 ± 0.0146	0.3541 ± 0.0077	0.1344 ± 0.0081	0.3507 ± 0.0545	0.2437 ± 0.0443	0.6252 ± 0.0350
	LRMVC [8]	0.4542 ± 0.0054	0.4073 ± 0.0065	0.1549 ± 0.0280	0.4308 ± 0.0157	0.3218 ± 0.0211	0.6516 ± 0.0076
	OMVCDR [10]	0.3987 ± 0.0054	0.3947 ± 0.0043	0.1585 ± 0.0032	0.4331 ± 0.0209	0.3354 ± 0.0257	0.6131 ± 0.0128
	FSMSC [50]	***0.5246 ± 0.0056***	***0.4975 ± 0.0074***	**0.2176 ± 0.0214**	***0.5676 ± 0.0393***	***0.4731 ± 0.0552***	**0.7154 ± 0.0098**
	MLRR [51]	0.4243 ± 0.0094	0.4182 ± 0.0063	0.1717 ± 0.0070	0.4401 ± 0.0277	0.3361 ± 0.0207	0.6373 ± 0.0158
	UOMvSC [52]	0.4878 ± 0.0082	0.4419 ± 0.0046	0.0545 ± 0.0043	0.2228 ± 0.0356	0.1332 ± 0.0266	***0.7038 ± 0.0229***
	SGF [53]	0.4684 ± 0.0097	0.2662 ± 0.0079	0.1828 ± 0.0062	0.2011 ± 0.0060	0.2538 ± 0.0131	0.1667 ± 0.0039
	DGF [53]	0.4740 ± 0.0062	0.2697 ± 0.0028	0.1842 ± 0.0063	0.2033 ± 0.0415	0.2451 ± 0.0324	0.1740 ± 0.0304
	TSSR [31]	0.4804 ± 0.0160	0.4593 ± 0.0108	0.1904 ± 0.0119	0.4740 ± 0.0471	0.3852 ± 0.0352	0.6159 ± 0.0269
	OLR-MVC (ours)	**0.5464 ± 0.0030**	**0.5164 ± 0.0047**	0.1925 ± 0.0251	**0.5828 ± 0.0196**	**0.4978 ± 0.0242**	0.7011 ± 0.0098

MCLES encounters problems on relatively large databases, i.e., Flower17, Scene15, and Caltech101.

these graph learning-based methods experience slight performance declines when processing larger databases.

- The Scene15 and Caltech101 databases have relatively complex data structures. UOMvSC achieves the second-best results on Scene15, likely benefiting from its integrated use of graphs, embedding matrices, and a one-step learning strategy without post-processing. FSMSC shows the second-best results on Caltech101, accounting for noisy views with unclear clustering structures and cross-view diversity.

These findings highlight the effectiveness and robustness of OLR-MVC, outperforming or matching state-of-the-art peer algorithms. More experiments one the statistical significance and visualization results of the comparisons are provided in our supplementary materials.

2) Runtime Comparison: We also examine the empirical efficiency of all latent representation-based methods, namely, LMSC [4], MCLES [5], LCRSR [6], RMCLES [7], LRMVC [8], OMVCDR [10], and the proposed OLR-MVC. The average runtime of different algorithms is presented in Fig. 3. Generally

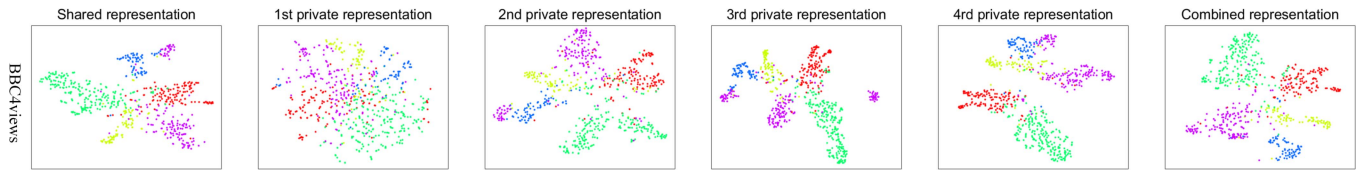

Fig. 4. Visualization of the learned latent representations via t-SNE.

TABLE V
ABLATION STUDY OF OLR-MVC

	Eq. (26)		*Eq.* (27)		OLR-MVC	
	NMI	ACC	NMI	ACC	NMI	ACC
ORL	**0.9482**	0.8825	0.9464	0.8800	0.9470	**0.8885**
Yale	0.7755	0.7515	**0.8050**	**0.7758**	0.7980	0.7642
MSRC-v1	0.8068	0.8905	0.7497	0.8381	**0.8494**	**0.8817**
3sources	0.6964	0.7929	0.6217	0.7633	**0.8054**	**0.8817**
BBC4views	0.7004	0.8540	**0.7702**	**0.9182**	0.7614	0.9143
Flower7	0.5092	0.5081	0.5617	0.5588	**0.5968**	**0.6147**
Scene15	0.5889	0.6076	0.6044	0.6482	**0.6583**	**0.6653**
Caltech101	0.4397	0.4309	0.5233	0.4987	**0.5464**	**0.5164**

speaking, (1) the proposed OLR-MVC method incurs a moderate runtime cost compared to other competing methods. (2) Existing latent representation-based methods do not scale well to large datasets, as the average runtime increases exponentially with dataset size. To improve the computation efficiency, we can apply dimension reduction methods to the multi-view data, as detailed in our supplementary materials.

C. Model Discussion

1) Ablation Study: Multi-view learning is inherently an ill-posed problem, and additional assumptions are necessary to distinguish between dependent and independent information in noisy data. In this work, we assume that the observed data can be divided into three components: the shared representation, which captures the dependent information; the private representations, which encode the independent information; and noise, which is eliminated through sparsity constraints. We introduce orthogonality constraints to eliminate the coupling between dependent and independent information, which is a simple yet efficient approach. For comparison, if we allow only shared information across views, our OLR-MVC model degenerates to:

$$\begin{aligned}&\min_{\mathcal{E},Z_s,H_s,\{P_s^{(i)}\}} \|\mathcal{E}\|_{2,1}+\alpha\|Z_s\|_*,\\ &\text{s.t. } \mathcal{E}=\{E_x^{(1)},\ldots,E_x^{(V)},E_s\},\ \{P_s^{(i)'}P_s^{(i)}=I\}_{i=1}^V,\\ &\{X^{(i)}=P_s^{(i)}H_s+E_x^{(i)}\}_{i=1}^V,\ H_s=H_s*Z_s+E_s. \end{aligned}\tag{26}$$

Conversely, if we only allow private information between the views, the model degenerates to:

$$\begin{aligned}&\min_{\mathcal{E},\mathcal{Z},\{P_p^{(i)}\},\{H_p^{(i)}\}} \|\mathcal{E}\|_{2,1}+\alpha\|\mathcal{Z}\|_*,\\ &\text{s.t. } \mathcal{E}=\{E_x^{(1)},\ldots,E_x^{(V)},E_p^{(1)},\ldots,E_p^{(V)}\},\\ &\mathcal{Z}=\{Z_p^{(1)},\ldots,Z_p^{(V)}\},\ \{P_p^{(i)'}P_p^{(i)}=I\}_{i=1}^V,\\ &\{X^{(i)}=P_p^{(i)}H_p^{(i)}+E_x^{(i)}\}_{i=1}^V,\ \{H_p^{(i)}=H_p^{(i)}Z_p^{(i)}+E_p^{(i)}\}_{i=1}^V. \end{aligned}\tag{27}$$

We compare the clustering performance of these two strategies with that of OLR-MVC in Table V. Experiments show that OLR-MVC achieves either much better or slightly worse performance than models that admit only shared or private information, thereby validating the effectiveness of our joint modeling strategy.

2) Visualization of the Learned Latent Representations: To better demonstrate the effectiveness of jointly modeling dependence and independence, we apply the t-distributed Stochastic Neighbor Embedding (t-SNE) method [56] for visualization. The t-SNE method uses the Kullback-Leibler divergence to measure the difference between the similarity distributions in the observed data and the learned representations. These low-dimensional representations are then visualized to reveal the clustering structure. We use BBC4views as a representative example and plot the t-SNE results for the shared latent representation, private latent representations, and their combinations in Fig. 4. It is evident that the clustering structures of different components vary significantly based on feature quality. The combined representations exhibit a better clustering structure, with the clustering boundaries being much clearer, validating the effectiveness of our OLR-MVC model. In addition, we visualize the t-SNE results on Scene15 in our supplementary materials for further comparison.

3) Block-Diagonal Structure of the Self-Expressive Matrices: In data clustering, a block-diagonal structure of the self-expressive matrix is highly desirable, as it indicates dense connections within each cluster and sparse connections between different clusters [57]. We provide visual results of self-expressive matrices learned by OLR-MVC on ORL, using the first five classes to enhance visibility. As shown in Fig. 5, the block-diagonal structures of self-expressive matrices learned from either shared or private latent representations are suboptimal. Specifically, in the first four columns of Fig. 5, it is evident that many within-cluster samples are poorly connected, and between-cluster samples are misconnected. However, by jointly exploring both shared and private latent representations, a clear block-diagonal structure emerges in the last column of Fig. 5, demonstrating the effectiveness of OLR-MVC. We also visualize the block-diagonal structure of the self-expressive matrices learned on Yale in our supplementary materials for further comparison.

4) Empirical Convergence Analysis: Ensuring the theoretical convergence of the ADMM method is generally infeasible when the number of optimization blocks exceeds two [58]. Instead, we demonstrate the empirical convergence of our optimization algorithm by plotting the convergence curves on real-world databases. As observed in Fig. 6, the four residuals (defined in (24)) approach relatively small values within dozens

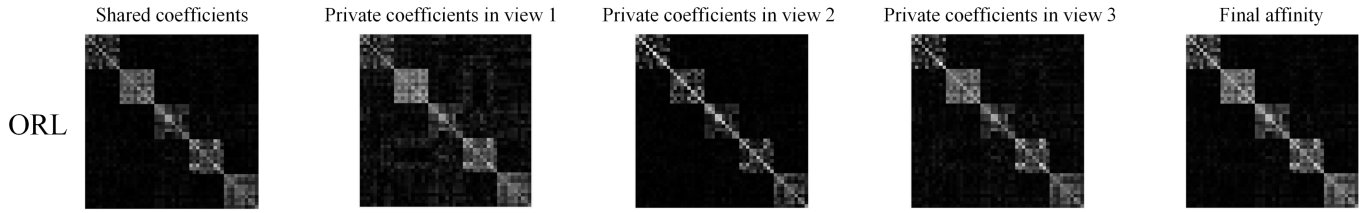

Fig. 5. Visualizations of the self-expressive matrices learned from the shared and private latent representations, as well as the final affinity matrix.

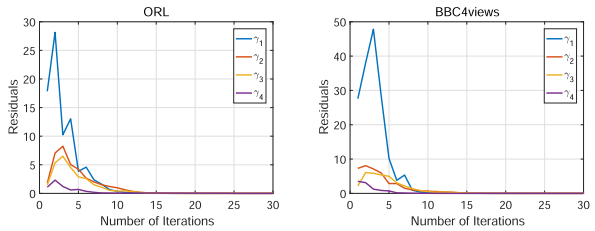

Fig. 6. Residuals of Algorithm 1 on representative databases.

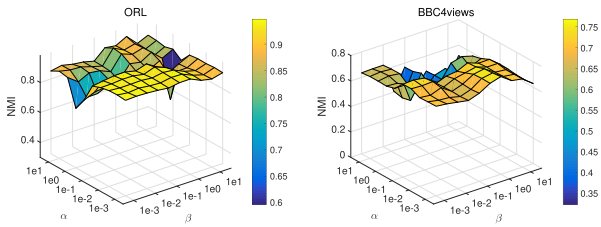

Fig. 7. NMI scores across different combinations of parameters α and β on representative databases.

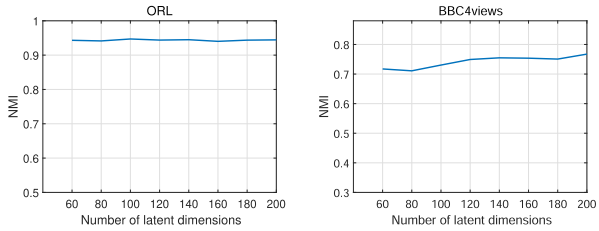

Fig. 8. NMI scores over different settings of the latent dimension $hdim$ on representative databases.

of iterations, highlighting the excellent empirical convergence of our optimization algorithm.

5) Parameter Sensitivity Analysis: Our OLR-MVC model includes three hyper-parameters: α, β, and $hdim$. α controls the weight of the low-rank term for learning the self-expressive coefficient matrices, β balances the importance of the orthogonal latent representation learning module, and $hdim$ determines the latent dimensions. We tune the parameters using grid search and present parameter sensitivity analysis on representative databases. The NMI scores obtained by OLR-MVC for different combinations of α and β are shown in Fig. 7. The relatively smooth NMI surfaces indicate that OLR-MVC achieves robust performance on these datasets. The NMI scores of OLR-MVC across different setting of $hdim$ are also provided in Fig. 8. OLR-MVC obtains consistently stable performance across large numerical intervals, demonstrating the robustness of our model. Additionally, we provide the parameter sensitivity analysis on all the other databases in our supplementary materials.

V. Conclusion

Latent representation learning is a powerful approach for multi-view clustering, but its performance is often limited by the rigid assumption that multiple views share only dependent information. To address this limitation, we propose to factorize the latent representations into shared and private components, and accordingly, introduce the Orthogonal Latent Representations for Multi-View Clustering (OLR-MVC). In contrast to traditional latent representation learning methods, our approach offers a more flexible and comprehensive framework for modeling complex relationships between different views, providing new insights into the design of advanced latent representation learning methods.

While OLR-MVC has shown promising results, its complexity is cubic with respect to the number of samples, which limits its performance when applied to large-scale or online databases. Inspired by the novel work in [59], our future work will incorporate prototype learning to efficiently handle large-scale and online databases.

References

[1] Y. Li, M. Yang, and Z. M. Zhang, "A survey of multi-view representation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1863–1883, Oct. 2019.
[2] Y. Yang and H. Wang, "Multi-view clustering: A survey," *Big Data Min. Anal.*, vol. 1, no. 2, pp. 83–107, 2018.
[3] U. Fang et al., "A comprehensive survey on multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12350–12368, Dec. 2023.
[4] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 4279–4287.
[5] M.-S. Chen, L. Huang, C.-D. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *Proc. Amer. Assoc. Artif. Intell.*, 2020, pp. 3513–3520.
[6] H. Tao, C. Hou, Y. Qian, J. Zhu, and D. Yi, "Latent complete row space recovery for multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 29, pp. 8083–8096, 2020.
[7] M.-S. Chen, L. Huang, C.-D. Wang, D. Huang, and J.-H. Lai, "Relaxed multi-view clustering in latent embedding space," *Inf. Fusion*, vol. 68, pp. 8–21, 2021.
[8] S. Huang, I. W. Tsang, Z. Xu, and J. Lv, "Latent representation guided multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 7082–7087, Jul. 2023.
[9] M. Salzmann, C. H. Ek, R. Urtasun, and T. Darrell, "Factorized orthogonal latent spaces," in *Proc. Mach. Learn. Res.*, 2010, pp. 701–708.

[10] 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.
[11] T. Zhou, C. Zhang, C. Gong, H. Bhaskar, and J. Yang, "Multiview latent space learning with feature redundancy minimization," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1655–1668, Apr. 2020.
[12] V. R. De Sa, P. W. Gallagher, J. M. Lewis, and V. L. Malave, "Multi-view kernel construction," *Mach. Learn.*, vol. 79, no. 1, pp. 47–71, 2010.
[13] S. Huang, Z. Kang, I. W. Tsang, and Z. Xu, "Auto-weighted multi-view clustering via kernelized graph learning," *Pattern Recognit.*, vol. 88, pp. 174–184, 2019.
[14] X. Liu, "SimpleMKKM: Simple multiple kernel K-means," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5174–5186, 2023.
[15] X. Liu et al., "Localized simple multiple kernel K-means," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9293–9301.
[16] T. Zhang et al., "Late fusion multiple kernel clustering with local kernel alignment maximization," *IEEE Trans. Multimedia*, vol. 25, pp. 993–1007, 2023.
[17] M. Zhang and K. Liu, "Enriched robust multi-view kernel subspace clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1993–2002.
[18] J. Liu, X. Liu, Y. Yang, Q. Liao, and Y. Xia, "Contrastive multi-view kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9552–9566, Aug. 2023.
[19] 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.
[20] 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.
[21] 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.
[22] Z. Li et al., "Consensus graph learning for multi-view clustering," *IEEE Trans. Multimedia*, vol. 24, pp. 2461–2472, 2022.
[23] A. Huang, W. Chen, T. Zhao, and C. W. Chen, "Joint learning of latent similarity and local embedding for multi-view clustering," *IEEE Trans. Image Process.*, vol. 30, pp. 6772–6784, 2021.
[24] H. Wang, G. Jiang, J. Peng, R. Deng, and X. Fu, "Towards adaptive consensus graph: Multi-view clustering via graph collaboration," *IEEE Trans. Multimedia*, vol. 25, pp. 6629–6641, 2023.
[25] X. Shu et al., "Self-weighted anchor graph learning for multi-view clustering," *IEEE Trans. Multimedia*, vol. 25, pp. 5485–5499, 2023.
[26] Y. Qin, N. Pu, and H. Wu, "EDMC: Efficient multi-view clustering via cluster and instance space learning," *IEEE Trans. Multimedia*, vol. 26, pp. 5273–5283, 2024.
[27] W. Zhao et al., "Anchor graph-based feature selection for one-step multi-view clustering," *IEEE Trans. Multimedia*, vol. 26, pp. 7413–7425, 2024.
[28] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1582–1590.
[29] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern Recognit.*, vol. 73, pp. 247–258, 2018.
[30] Y. Xie et al., "On unifying multi-view self-representations for clustering by tensor multi-rank minimization," *Int. J. Comput. Vis.*, vol. 126, no. 11, pp. 1157–1179, 2018.
[31] B. Cai, G.-F. Lu, H. Li, and W. Song, "Tensorized scaled simplex representation for multi-view clustering," *IEEE Trans. Multimedia*, vol. 26, pp. 6621–6631, 2024.
[32] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 252–260.
[33] J. Guo, Y. Sun, J. Gao, Y. Hu, and B. Yin, "Rank consistency induced multiview subspace clustering via low-rank matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 3157–3170, Jul. 2022.
[34] J. Li, Q. Gao, Q. Wang, M. Yang, and W. Xia, "Orthogonal non-negative tensor factorization based multi-view clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 18186–18202.
[35] X. Yan, S. Hu, Y. Mao, Y. Ye, and H. Yu, "Deep multi-view learning methods: A review," *Neurocomputing*, vol. 448, pp. 106–129, 2021.
[36] C. Zhang, Y. Liu, and H. Fu, "AE2-Nets: Autoencoder in autoencoder networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2577–2585.
[37] Z. Li et al., "Deep adversarial multi-view clustering network," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 2952–2958.
[38] S. Fan et al., "One2Multi graph autoencoder for multi-view graph clustering," in *Proc. Int. World Wide Web Conf.*, 2020, pp. 3070–3076.
[39] N. Zhang, S. Ding, H. Liao, and W. Jia, "Multimodal correlation deep belief networks for multi-view classification," *Appl. Intell.*, vol. 49, pp. 1925–1936, 2019.
[40] E. Pan and Z. Kang, "Multi-view contrastive graph clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 2148–2159.
[41] J. Xu et al., "Multi-level feature learning for contrastive multi-view clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 16051–16060.
[42] 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.
[43] 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.*, 2011, pp. 612–620.
[44] G. Liu et al., "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[45] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.
[46] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 849–856.
[47] J. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 756–763.
[48] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 524–531.
[49] F.-F. Li, F. Rob, and P. Pietro, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.
[50] Z. Chen, X.-J. Wu, T. Xu, and J. Kittler, "Fast self-guided multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 32, pp. 6514–6525, 2023.
[51] J. Chen, S. Yang, H. Mao, and C. Fahy, "Multiview subspace clustering using low-rank representation," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12364–12378, Nov. 2022.
[52] C. Tang et al., "Unified one-step multi-view spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6449–6460, Jun. 2023.
[53] Y. Liang, D. Huang, C.-D. Wang, and S. Y. Philip, "Multi-view graph learning by joint modeling of consistency and inconsistency," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 2848–2862, Feb. 2024.
[54] C. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," *Nat. Lang. Eng.*, vol. 16, no. 1, pp. 100–103, 2010.
[55] S. Sun, S. Zhi, Q. Liao, J. Heikkilä, and L. Liu, "Unbiased scene graph generation via two-stage causal modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12562–12580, Oct. 2023.
[56] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[57] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 487–501, Feb. 2019.
[58] 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. Prog.*, vol. 155, no. 1, pp. 57–79, 2016.
[59] C. Wen, H. Huang, Y. Ma, F. Yuan, and H. Zhu, "Dual-guided frequency prototype network for few-shot semantic segmentation," *IEEE Trans. Multimedia*, vol. 26, pp. 8874–8888, 2024.

Xiaolin Xiao (Member, IEEE) received the B.E. degree from Wuhan University, Wuhan, China, in 2013, and the Ph.D. degree from the University of Macau, Macau, China, in 2019. She is currently an Associate Professor with the School of Computer Science, South China Normal University, Guangzhou, China. Her research interests include multi-view learning and color image processing and understanding.

Yue-Jiao Gong (Senior Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2010 and 2014, respectively. She is currently a Full Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. Her research interests include optimization methods based on swarm intelligence, deep learning, reinforcement learning, and their applications in smart cities and intelligent transportation. She has authored or coauthored more than 100 papers, including more than 50 in ACM/IEEE Transactions and more than 50 with renowned conferences such as NeurIPS, ICLR, and GECCO. Dr. Gong was the recipient of the Pearl River Young Scholar by the Guangdong Education Department in 2017 and the Guangdong Natural Science Funds for Distinguished Young Scholars in 2022. She currently an Associate Editor for IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Yicong Zhou (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Tufts University, Medford, MA, USA. He is currently a Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security. Dr. Zhou is a Fellow of SPIE (the Society of Photo-Optical Instrumentation Engineers) and was recognized as one of "Highly Cited Researchers" in 2020, 2021, 2023 and 2024. He is a Senior Area Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.