Efficient and Effective One-Step Multiview Clustering

Jun Wang, Chang Tang, *Senior Member, IEEE*, Zhiguo Wan, *Member, IEEE*, Wei Zhang, *Member, IEEE*, Kun Sun, *Member, IEEE*, and Albert Y. Zomaya, *Fellow, IEEE*

***Abstract*— Multiview clustering algorithms have attracted intensive attention and achieved superior performance in various fields recently. Despite the great success of multiview clustering methods in realistic applications, we observe that most of them are difficult to apply to large-scale datasets due to their cubic complexity. Moreover, they usually use a two-stage scheme to obtain the discrete clustering labels, which inevitably causes a suboptimal solution. In light of this, an efficient and effective one-step multiview clustering (E^2OMVC) method is proposed to directly obtain clustering indicators with a small-time burden. Specifically, according to the anchor graphs, the smaller similarity graph of each view is constructed, from which the low-dimensional latent features are generated to form the latent partition representation. By introducing a label discretization mechanism, the binary indicator matrix can be directly obtained from the unified partition representation which is formed by fusing all latent partition representations from different views. In addition, by coupling the fusion of all latent information and the clustering task into a joint framework, the two processes can help each other and obtain a better clustering result. Extensive experimental results demonstrate that the proposed method can achieve comparable or better performance than the state-of-the-art methods. The demo code of this work is publicly available at https://github.com/WangJun2023/EEOMVC.**

***Index Terms*— Anchor graph, data representation, feature fusion, multiview clustering.**

I. Introduction

With the development of sensor and data processing technologies, each object can be captured from diverse sources or represented by various types of features. For example, an image can be visually described by HOG features, LBP features, and SIFT features in image processing, respectively. A document can be translated into different languages in document analysis. In general, all these kinds of data are regarded as multiview data, and they often have complex properties. For instance, different views contribute to unbalanced information in data mining. Furthermore, each view also contains the consensus information and the view-specific information of multiview datasets. In the era of big data, these characteristics pose a challenge on how to effectively extract the critical information from the original large-scale datasets.

Clustering, as an efficient technique to discover the meaningful patterns of data in unsupervised learning, has been widely used in various fields, such as pattern recognition and data mining. To preprocess the above multiview data, a naive way is concatenating them as a single-view data from which the existing advanced single-view clustering algorithms are adopted to extract the critical information. However, this manner cannot achieve satisfying performance in most scenarios. Thus, to fully capture the consensus and view-specific information from the original multiview data, many multiview clustering methods have been developed, e.g., [1], [2], [3], [4].

Among the most existing multiview clustering methods, the latent representation learning-based multiview clustering methods are popular since they can dexterously extract the underlying data structure. These kinds of methods aim at learning a consensus data representation from the original heterogeneous features, and the representative manner of learning the consensus partition representation is based on the similarity graph learning. To be specific, after obtaining a unified similarity graph via graph construction and fusion of all the views, the consensus partition representation is generated by adopting eigendecomposition on the Laplacian matrix corresponding to the unified similarity graph. Benefiting from the similarity graph can capture the underlying data structure information, and many advanced graph-based multiview clustering methods have been proposed and have achieved satisfying performance, such as [5], [6], [7], [8], [9]. In general, these methods contain two critical steps, i.e., graph construction and graph eigendecomposition. However, when the number of samples n is large enough, the graph eigendecomposition needs a relatively high computational overhead due to its time complexity reaching $\mathcal{O}(n^2)$ or $\mathcal{O}(n^3)$. In this case, these methods are difficult to apply to large-scale data in many real applications. In addition,

Manuscript received 16 August 2022; revised 6 December 2022; accepted 31 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62076228 and Grant 62176242, in part by the Natural Science Foundation of Shandong Province under Grant ZR2021LZH001, and in part by the Key Research Project of Zhejiang Lab under Grant K2022PD1BB01. *(Corresponding author: Chang Tang.)*

Jun Wang, Chang Tang, and Kun Sun are with the School of Computer Science, China University of Geosciences, Wuhan 430079, China (e-mail: wang_jun@cug.edu.cn; tangchang@cug.edu.cn; sunkun@cug.edu.cn).

Zhiguo Wan is with the Zhejiang Lab, Hangzhou 311121, China (e-mail: wanzhiguo@zhejianglab.com).

Wei Zhang is with the Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputing Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250000, China (e-mail: wzhang@qlu.edu.cn).

Albert Y. Zomaya is with the School of Information Technologies, The University of Sydney, Camperdown, NSW 2006, Australia (e-mail: albert.zomaya@sydney.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3253246.

Digital Object Identifier 10.1109/TNNLS.2023.3253246

many previous methods require an additional postprocessing step (e.g., k-means) to generate the discrete cluster labels, and this manner would cause the suboptimal solution and result in unsatisfying performance, e.g., [10]. Finally, to distinguish the different importance of all the views, an extra parameter is needed in many existing multiview clustering methods.

To address the above limitations, an efficient and effective one-step multiview clustering (E^2OMVC) method is proposed in this article. Considering that the performance of many existing clustering methods highly depends on the quality of original features which are usually inevitably corrupted by noise, the low-dimensional latent features are generated from the similarity graph of each view to form the latent partition representation. In addition, to improve the computational efficiency, anchor graphs are used to construct the similarity graph. Based on the assumption that multiple views are derived from a unified latent representation [11], [12], [13], the proposed method is adopted to simultaneously learn a common latent representation from these weighted partition matrices, and the binary indicator matrix is directly obtained with introducing a clustering discrete mechanism on the common latent representation. In such a manner, the time complexity and optimization procedure are simplified by fusing latent information at the partition level. Moreover, E^2OMVC couples the fusion of all the latent features and the clustering task into a joint framework to guarantee that the two isolated processes can help each other naturally and obtain a better clustering result.

In summary, the main contributions of this article include the following.

1) We propose to generate the low-dimensional latent features from the anchor graph of each view, which can significantly reduce the time complexity from $\mathcal{O}(n^2)$ or $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$ with n the number of samples. In addition, with fusing all the latent information at partition level instead of similarity level, the feature redundancy and noise influence can be mitigated.
2) A unified one-step multiview clustering framework is proposed to simultaneously generate a common latent representation and a clustering indicator matrix for multiview data. By jointly optimizing the fusion of all the latent features and the generation of the clustering indicator matrix, our method can directly obtain a better clustering result to avoid the risk of additional discrete clustering.
3) An alternating iterative optimization algorithm is designed to solve the formulated model. Extensive experimental results verify the effectiveness of the proposed method on six benchmark multiview datasets.

II. Related Work

Based on the way of learning the consensus clustering partition for multiview data, the existing multiview clustering methods can be roughly divided into three categories, including co-training learning, latent representation learning, and multiple kernel learning. In this section, we will give a brief introduction to them.

A. Co-Training-Based Multiview Clustering

The co-training learning-based multiview clustering methods aim at using the prior information of one view to guide the clustering of other views until the consensus clustering result is obtained across all the views. Assuming that all similar samples should be grouped into the same cluster, Kumar et al. [14] proposed a co-training multiview clustering method for generating a consensus clustering result by exploiting the view-specific information. Considering the partial mapping for some incomplete views, a pairwise constraints' propagation is introduced in [15] to constrain multiview clustering. In addition, to consistently keep the clustering results for all the views, a co-regularized multiview clustering method with introducing graph Laplacian regularization is proposed in [14].

Although these methods can facilitate improvement between different views, their performance highly depends on the assumption that each view can achieve the clustering task independently. In such a manner, their applications would be limited in some complex multiview scenarios.

B. Latent-Representation-Based Multiview Clustering

The latent representation learning-based multiview clustering methods mainly learn a common representation across all the views. For example, Cao et al. [16] proposed a multiview subspace clustering method to learn a unified latent representation that can well capture the complementary information for all the views. Assuming that all the views can be mapped into a common latent subspace, the work in Xie et al. [17] adopted linear transformation to learn the unified latent representation of data from the latent subspace. Except learning the consensus representation in latent space, another popular manner of learning the consensus representation is based on the similarity graph. To obtain high-quality similarity graphs across all the views, Nie et al. [18] adopted the adaptive neighbor graph learning to construct a Laplacian rank constrained similarity graph for generating the unified latent representation. Based on the intrinsic graph structure obtained from multiple graphs, Zhan et al. [19] adaptively tuned each similarity graph of each view to capture the underlying data geometric property. Apart from them, many advanced methods have been proposed, such as [20], [21], [22], [23], [24]. However, these methods require a time complexity of $\mathcal{O}(n^2)$ or $\mathcal{O}(n^3)$, which means that they are difficult to directly apply to large-scale multiview data. To this end, Cai and Chen [25] proposed to select a few representative samples to compute the similarity graphs of the whole dataset, and eigendecomposition can be conducted on the small-size similarity graph. To improve computational efficiency, Li et al. [26] used bipartite graphs to approximate the similarity graphs, and heterogeneous features are integrated in a local manifold fusion manner. Although previous methods can significantly reduce the time complexity of multiview clustering, they still require an additional postprocessing step to produce the clustering results. In this way, the additional clustering step would deviate from solving the original problem. Thus, Chen et al. [27] proposed to jointly recover latent space, construct a similarity matrix, and generate a clustering indicator matrix into a unified framework. Unfortunately, this

method has cubic complexity, which is difficult to address the large-scale clustering problem. Furthermore, benefiting from the powerful feature representation ability of the deep neural network, many advanced deep multiview clustering methods have been developed, such as [28], [29], [30].

In a word, despite the great success in multiview clustering for these methods, the high time complexity limits their applications on large-scale multiview data. Furthermore, most existing multiview clustering methods tend to fuse latent information at the similarity graph level, which is usually affected by the noise.

C. Multiple Kernel-Learning-Based Multiview Clustering

Multiple kernel-learning-based multiview clustering methods aim at learning a unified kernel by linearly or nonlinearly combining different predefined kernels of all the views. Therefore, these kinds of methods mainly contain two critical steps, i.e., kernel construction and kernel fusion. For example, Yu et al. [31] mapped the original data into a Hilbert space and used the kernel matrices for data fusion. To reduce the redundancy of initial kernels, Liu et al. [32] proposed a multiple kernel k-means clustering method. Considering the different importance between diverse kernels, Tzortzis and Likas [33] proposed a weighted kernel multiview clustering method, where the weight is determined via the quality of information for each kernel.

In summary, although these kinds of methods can achieve promising performance, we can observe that most of them suffer from intensive time complexity and overcomplicated optimization procedure. Thus, E^{2}OMVC is proposed to address these issues.

III. Preliminaries

A. Spectral Clustering

Given a symmetric and doubly stochastic similarity graph $\mathbf{W}$ about the single-view data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ with d features and n samples, the objective function of spectral clustering with using graph normalized-cut manner can be formulated as [34], [35], [36], [37]

$$\max_{\mathbf{H}} \mathrm{Tr}(\mathbf{H}^\top \mathbf{W}\mathbf{H}), \quad \text{s.t. } \mathbf{H} = \mathbf{Y}(\mathbf{Y}^\top\mathbf{Y})^{-\frac{1}{2}},\ \mathbf{Y} \in \mathrm{Ind} \tag{1}$$

where $\mathbf{Y}$ denotes the clustering indicator matrix. For (1), it is difficult to be directly solved because $\mathbf{Y}$ is discrete. Thus, a common way is relaxing partition matrix $\mathbf{H}$ into the continuous value. Mathematically, the problem can be rewritten as

$$\max_{\mathbf{H}} \mathrm{Tr}(\mathbf{H}^\top \mathbf{W}\mathbf{H}), \quad \text{s.t. } \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k. \tag{2}$$

After obtaining the partition matrix $\mathbf{H}$, an additional clustering step is adopted to obtain the discrete clustering indicator matrix $\mathbf{Y}$, such as k-means and spectral rotation. It can be observed from (1) that $\mathbf{Y}(\mathbf{Y}^\top\mathbf{Y})^{-1/2}$ is the solution of (2). Thus, given one feasible solution $\mathbf{HQ}$ of (2), where $\mathbf{Q}$ denotes an arbitrarily orthogonal matrix, the spectral rotation problem can be formulated as follows:

$$\min_{\mathbf{Y},\mathbf{Q}} ||\mathbf{HQ} - \mathbf{Y}(\mathbf{Y}^\top\mathbf{Y})^{-\frac{1}{2}}||_F^2, \quad \text{s.t. } \mathbf{Y} \in \mathrm{Ind},\ \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_k. \tag{3}$$

B. Anchor Graph

Constructing anchor graph is an effective way to address the large-scale clustering problem, and it has been successfully used in some advanced multiview clustering methods, such as [38], [39]. Assuming that m ($m \ll n$) representative samples $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m] \in \mathbb{R}^{d\times m}$ are chosen as anchors to represent the whole dataset $\mathbf{X}$ based on k-means clustering, then a sparse affinity matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$ can be constructed as follows:

$$B_{ij} = \begin{cases} \dfrac{K_\sigma(x_i, a_j)}{\sum_{j'} K_\sigma(x_i, a_{j'})}, & j, j' \in \langle i \rangle \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\langle i \rangle$ denotes a set of indexes of r closest anchors to x_i, and $K_\sigma(\cdot)$ denotes a Gaussian kernel function with parameter σ. Based on the sparse affinity matrix $\mathbf{B}$, the similarity matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ can be generated, i.e.,

$$\mathbf{W} = \overline{\mathbf{B}}\,\overline{\mathbf{B}}^\top, \quad \text{s.t. } \overline{\mathbf{B}} = \mathbf{B}\mathbf{D}^{-\frac{1}{2}} \tag{5}$$

where $\mathbf{D}$ represents the diagonal matrix with each element $\mathbf{D}_{ii} = \sum_{j=1}^{n} B_{ji}$. According to the above descriptions, we can find that the time complexity can be significantly reduced because only mn samples are considered to construct the similarity graph.

IV. Proposed Method

A. Latent Partition Representation Learning

Considering that the data are often corrupted by the noise in the process of data acquisition, most existing multiview clustering methods propose to learn the latent representation to mitigate the noise influence and capture the relationships among samples. However, some methods adopt a linear combination manner to obtain the latent representation from the original features. In this way, their performance will be limited about processing nonlinear structured data. It is well-known that the similarity graph can well capture the underlying geometric structure of the original data, which motivates us to use it for latent representation learning. For large-scale multiview datasets, a small number of representative samples are sufficient to depict their global data distribution and maintain the geometric structure due to the existence of much redundant features. Thus, a set of representative anchors are generated based on the k-means clustering first, and the corresponding similarity graph $\mathbf{W}$ can be constructed according to (4) and (5). Subsequently, the low-dimensional latent features are extracted from the similarity graph to form the latent partition representation $\mathbf{H}$. Mathematically, the problem can be reformulated as follows:

$$\max_{\mathbf{H}} \mathrm{Tr}(\mathbf{H}^\top\overline{\mathbf{B}}\,\overline{\mathbf{B}}^\top\mathbf{H}), \quad \text{s.t. } \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k \tag{6}$$

where $\mathbf{H}$ denotes the latent partition representation. A trick is that the similarity graph $\mathbf{W}$ can be replaced by $\overline{\mathbf{B}}\,\overline{\mathbf{B}}^\top$ to reduce the time complexity of solving (6). To be specific, the optimal solution $\mathbf{H}$ can be directly obtained by adopting singular value decomposition (SVD) on $\overline{\mathbf{B}} \in \mathbb{R}^{n\times m}$ instead of $\mathbf{W} \in \mathbb{R}^{n\times n}$.

B. Proposed Multiview Clustering Method

As mentioned above that the multiview datasets can be described from various views, most existing multiview clustering methods tend to learn a consensus similarity graph across all the views, i.e., the unified similarity graph is generated by fusing all the similarity graphs. In this way, the final clustering results would be affected by the quality of each similarity graph, especially under the condition that one specific similarity graph is grossly corrupted by noise. Therefore, we propose to learn a consensus representation at the partition level rather than at the similarity graph level. To be specific, we extend (6) into multiview version and generate all the latent partition representations $\{\mathbf{H}^{(i)}\}_{i=1}^{v}$ for all the views, and then the consensus representation is constructed as follows:

$$\max_{\mathbf{H}^*,\mathbf{R},\boldsymbol{\gamma}} \mathrm{Tr}\left(\mathbf{H}^{*\top}\sum_{i=1}^{v}\gamma_i\mathbf{H}^{(i)}\mathbf{R}^{(i)}\right)$$
$$\text{s.t. } \mathbf{H}^{*\top}\mathbf{H}^* = \mathbf{I}_k, \quad \mathbf{R}^{(i)\top}\mathbf{R}^{(i)} = \mathbf{I}_k, \quad \sum_{i=1}^{v}\gamma_i^2 = 1, \ \gamma_i \geq 0 \tag{7}$$

where $\mathbf{H}^*$ denotes the consensus partition representation, $\mathbf{R}^{(i)}$ denotes the rotation matrix of the ith view, and γ is a weight coefficient of all the views.

After obtaining the consensus representation $\mathbf{H}^*$, similar to other graph-based multiview clustering methods, a natural way to obtain the final discrete clustering indicator matrix is by applying an additional postprocessing step on $\mathbf{H}^*$. However, this manner isolates the processes of consensus representation learning and the clustering task, which may cause insufficient utilization of latent information and lead to unsatisfying performance. To address this issue, a clustering discrete mechanism is introduced. Specifically, we couple the consensus representation learning and clustering into a joint framework so that the two steps can help each other and produce a better clustering result. In addition, based on the fact that all the partition representations should share the common clustering structure, an orthogonal matrix $\mathbf{Q}$ is introduced to guide $\mathbf{H}^*$ to approximate $\mathbf{Y}(\mathbf{Y}^\top\mathbf{Y})^{-(1/2)}$. Therefore, the problem can be formulated as follows:

$$\max_{\mathbf{H}^*,\mathbf{R},\boldsymbol{\gamma},\mathbf{Q},\mathbf{Y}} \mathrm{Tr}(\mathbf{H}^{*\top}\mathbf{G}) - \beta||\mathbf{H}^*\mathbf{Q} - \mathbf{Y}(\mathbf{Y}^\top\mathbf{Y})^{-\frac{1}{2}}||_F^2$$
$$\text{s.t. } \mathbf{H}^{*\top}\mathbf{H}^* = \mathbf{I}_k, \quad \mathbf{G} = \sum_{i=1}^{v}\gamma_i\mathbf{H}^{(i)}\mathbf{R}^{(i)}, \ \mathbf{R}^{(i)\top}\mathbf{R}^{(i)} = \mathbf{I}_k$$
$$\sum_{i=1}^{v}\gamma_i^2 = 1, \quad \gamma_i \geq 0, \ \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_k, \ \mathbf{Y} \in \mathrm{Ind} \tag{8}$$

where β is a tradeoff parameter, and the domain of it is set to $[0.01, 0.05, 0.1, 0.5, 1]$ in our proposed method.

For the anchor-based methods, the quality of selected anchors would largely determine the final clustering performance. Apart from the specific graph structure of each view, all the distinct views also contain the common underlying clustering structure, which should be exploited on constructing the anchor graphs. Thus, the concatenating features of all the distinct views are used to construct the global anchor graph, and then the corresponding latent features derived from the global similarity graph are used to promote the fusion of all the latent features. Mathematically, the overall objective function of the proposed method E^{2}OMVC is modeled as

$$\max_{\mathbf{H}^*,\mathbf{R},\boldsymbol{\gamma},\mathbf{Q},\mathbf{Y}} \mathrm{Tr}(\mathbf{H}^{*\top}(\mathbf{G} + \lambda\overline{\mathbf{H}})) - \beta||\mathbf{H}^*\mathbf{Q} - \mathbf{Y}(\mathbf{Y}^\top\mathbf{Y})^{-\frac{1}{2}}||_F^2$$
$$\text{s.t. } \mathbf{H}^{*\top}\mathbf{H}^* = \mathbf{I}_k, \quad \mathbf{G} = \sum_{i=1}^{v}\gamma_i\mathbf{H}^{(i)}\mathbf{R}^{(i)}, \ \mathbf{R}^{(i)\top}\mathbf{R}^{(i)} = \mathbf{I}_k$$
$$\sum_{i=1}^{v}\gamma_i^2 = 1, \quad \gamma_i \geq 0, \ \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_k, \ \mathbf{Y} \in \mathrm{Ind} \tag{9}$$

where λ is a parameter. Similar to β, the domain of it is also set to $[0.01, 0.05, 0.1, 0.5, 1]$ in our proposed method. $\overline{\mathbf{H}} \in \mathbb{R}^{n\times k}$ denotes the global latent feature matrix, which is generated from the global similarity graph of all the concatenating features, i.e.,

$$\max_{\mathbf{H}} \mathrm{Tr}(\overline{\mathbf{H}}\mathbf{S}\mathbf{D}^{-1}\mathbf{S}^\top\overline{\mathbf{H}}^\top), \quad \text{s.t. } \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k \tag{10}$$

where $\mathbf{S}$ denotes the global similarity graph of all the concatenating features.

From (9), it is observed that E^{2}OMVC integrates the latent information at the partition level rather than the similarity graph level, which can reduce the feature redundancy and noise at the information fusion stage. Furthermore, by coupling the fusion of all the latent features and the clustering task into a unified framework, two steps can help each other and a better clustering indicator matrix can be directly obtained. Finally, a remarkable advantage is that the optimization problem of the proposed method on each variable has smaller time complexity when compared with the previous methods, which will be analyzed in the next section.

V. Optimization

In this section, an effective iterative optimization algorithm is designed to solve (9). To be specific, the whole optimization problem can be divided into the following five subproblems.

A. Optimizing $\mathbf{H}^*$

When variables $\{\mathbf{R}^{(i)}\}_{i=1}^{v}$, $\mathbf{Q}$, $\mathbf{Y}$ and $\boldsymbol{\gamma}$ are fixed, (9) can be reformulated as

$$\max_{\mathbf{H}^*} \mathrm{Tr}\left(\mathbf{H}^{*\top}(\mathbf{G} + \lambda\overline{\mathbf{H}} + 2\beta\mathbf{Y}(\mathbf{Y}^\top\mathbf{Y})^{-\frac{1}{2}}\mathbf{Q}^\top)\right)$$
$$\text{s.t. } \mathbf{H}^{*\top}\mathbf{H}^* = \mathbf{I}_k, \mathbf{G} = \sum_{i=1}^{v}\gamma_i\mathbf{H}^{(i)}\mathbf{R}^{(i)}. \tag{11}$$

B. Optimizing $\{\mathbf{R}^{(i)}\}_{i=1}^{v}$

When variables $\mathbf{H}^*$, $\mathbf{Q}$, $\mathbf{Y}$, and $\boldsymbol{\gamma}$ are fixed, (9) can be turned into

$$\max_{\mathbf{R}^{(i)}} \mathrm{Tr}\left(\mathbf{R}^{(i)\top}\gamma_i\mathbf{H}^{(i)\top}\mathbf{H}^*\right), \quad \text{s.t. } \mathbf{R}^{(i)\top}\mathbf{R}^{(i)} = \mathbf{I}_k. \tag{12}$$

C. Optimizing $\mathbf{Q}$

When variables $\mathbf{H}^*$, $\{\mathbf{R}^{(i)}\}_{i=1}^v$, $\mathbf{Y}$, and $\boldsymbol{\gamma}$ are fixed, (9) can be rewritten as

$$\max_{\mathbf{Q}} \mathrm{Tr}\left(\mathbf{Q}^\top \beta \mathbf{H}^{*\top} \mathbf{Y} (\mathbf{Y}^\top \mathbf{Y})^{-\frac{1}{2}}\right), \quad \text{s.t. } \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_k. \tag{13}$$

For the above optimization problems, from (11) to (13), they all belong to the same optimization problem by conducting some mathematical transformations, i.e.,

$$\max_{\mathbf{F}} \mathrm{Tr}(\mathbf{F}^\top \mathbf{Z}), \quad \text{s.t. } \mathbf{F}^\top \mathbf{F} = \mathbf{I}_k. \tag{14}$$

That is to say, the optimal solutions to problem (13) are also their optimal solutions. Therefore, we can use (13) to solve (10)–(12) uniformly. Suppose the matrix $\mathbf{Z}$ is expressed as $\mathbf{Z} = \mathbf{U}\sum\mathbf{V}^\top$ via SVD, then the optimal solution $\mathbf{F}^* = \mathbf{U}\mathbf{V}^\top$ [40].

D. Optimizing $\mathbf{Y}$

When variables $\mathbf{H}^*$, $\{\mathbf{R}^{(i)}\}_{i=1}^v$, $\mathbf{Q}$, and $\boldsymbol{\gamma}$ are fixed, (9) can be simplified to

$$\max_{\mathbf{Y}} \mathrm{Tr}\left(\left(\mathbf{Y}\left(\mathbf{Y}^\top \mathbf{Y}\right)^{-\frac{1}{2}}\right)^\top \mathbf{H}^* \mathbf{Q}\right), \quad \text{s.t. } \mathbf{Y} \in \mathrm{Ind}. \tag{15}$$

Given auxiliary variable $\mathbf{T} = \mathbf{H}^*\mathbf{Q}$, (15) can be turned into

$$\max_{\mathbf{Y}} \mathbf{Tr}\left((\mathbf{Y}(\mathbf{Y}^\top \mathbf{Y})^{-\frac{1}{2}})^\top \mathbf{T}\right) = \sum_{c=1}^{k} \frac{\sum_{i=1}^n y_{ic} t_{ic}}{\sqrt{\mathbf{y}_c^\top \mathbf{y}_c}}$$
$$\text{s.t. } \mathbf{Y} \in \{0,1\}^{n\times k}. \tag{16}$$

As seen from (16), the term $\sqrt{\mathbf{y}_c^\top \mathbf{y}_c}$ only involves all the rows of Y, and thus, we can sequentially solve Y row-wise. To be specific, we need to take the increment of objective function value into account when $\mathbf{y}_c$ goes from $[1, 0, \ldots, 0]$ to $[0, 0, \ldots, 1]$, i.e.,

$$\delta_{ic} = \frac{\sum_{j=1}^n y_{jc} t_{jc} + t_{ic}(1 - y_{ic})}{\sqrt{\mathbf{y}_c^\top \mathbf{y}_c + (1 - y_{ic})}} - \frac{\sum_{j=1}^n y_{jc} t_{jc} - t_{ic} y_{ic}}{\sqrt{\mathbf{y}_c^\top \mathbf{y}_c - y_{ic}}}. \tag{17}$$

By the virtue of (17), the ith row optimal solution is generated from

$$y_{ic} = \left\langle c = \arg\max_{c\in[1,k]} \delta_{ic} \right\rangle \tag{18}$$

where $\langle . \rangle$ is 0 if the argument is false, or 1 otherwise.

E. Optimizing $\boldsymbol{\gamma}$

When variables $\mathbf{H}^*$, $\{\mathbf{R}^{(i)}\}_{i=1}^v$, $\mathbf{Q}$, and $\mathbf{Y}$ are fixed, the optimization problem of (9) is equivalent to

$$\max_{\boldsymbol{\gamma}} \sum_{i=1}^{v} \gamma_i \theta_i, \quad \text{s.t. } \sum_{i=1}^{v} {\gamma_i}^2 = 1 \tag{19}$$

where $\theta_i = \mathrm{Tr}(\mathbf{H}^{*\top}\mathbf{H}^{(i)}\mathbf{R}^{(i)})$. The optimal solution of (19) is given as follows:

$$\gamma_i = \frac{\theta_i}{\sqrt{\sum_{i=1}^v {\theta_i}^2}}. \tag{20}$$

In summary, the details for solving the proposed method are outlined in Algorithm 1.

Algorithm 1 E^{2}OMVC

Input: Multiview data $\{\mathbf{X}^{(i)}\}_{i=1}^v$, the number of clusters k, parameters λ and β.
1: Initialize $\{\mathbf{R}^{(i)}\}_{i=1}^v = \mathbf{Q} = \mathbf{I}_k$, $\{\gamma_i\}_{i=1}^v = \frac{1}{\sqrt{v}}$, $\mathbf{Y}$, $t = 1$.
2: Compute anchor graph $\{\mathbf{W}^{(i)}\}_{i=1}^v$, latent representation $\{\mathbf{H}^{(i)}\}_{i=1}^v$ and $\overline{\mathbf{H}}$.
3: **while not converged do**
4: Update $\mathbf{H}^*$ via (14).
5: Update $\{\mathbf{R}^{(i)}\}_{i=1}^v$ via (14).
6: Update $\mathbf{Q}$ via (14).
7: Update $\mathbf{Y}$ via (18).
8: Update $\{\gamma\}_{i=1}^v$ via (20).
9: $t = t + 1$.
10: **end**
Output: Clustering indicator matrix $\mathbf{Y}$.

F. Convergence Analysis

As aforementioned discussed, an alternating iterative algorithm is designed to solve the proposed method. Next, we will prove the convergence property of E^{2}OMVC.

Theorem 1: Algorithm 1 can reach convergence in a finite number of iterations.

Proof: Given $\forall\ i,\ j$, the following inequality can be derived.

$\mathrm{Tr}((\gamma_i\mathbf{H}^{(i)}\mathbf{R}^{(i)})^\top \gamma_j \mathbf{H}^{(j)}\mathbf{R}^{(j)}) \le \mathrm{Tr}((\mathbf{H}^{(i)}\mathbf{R}^{(i)})^\top \mathbf{H}^{(j)}\mathbf{R}^{(j)}) \le (1/2)\mathrm{Tr}((\mathbf{H}^{(i)}\mathbf{R}^{(i)})^\top \mathbf{H}^{(i)}\mathbf{R}^{(i)} + (\mathbf{H}^{(j)}\mathbf{R}^{(j)})^\top \mathbf{H}^{(j)}\mathbf{R}^{(j)}) = k$. As a result, the upper bound of (9) can be obtained. To be specific, with respect to the first term of objective function, we have $\mathrm{Tr}(\mathbf{H}^*\mathbf{G}) \le (1/2)\mathrm{Tr}(\mathbf{H}^{*\top}\mathbf{H}^* + \mathbf{G}^\top\mathbf{G}) \le (1/2)(d + dv^2)$ and $\lambda\mathrm{Tr}(\mathbf{H}^*\overline{\mathbf{H}}) \le (\lambda/2)\mathrm{Tr}(\mathbf{H}^{*\top}\mathbf{H}^* + \overline{\mathbf{H}}^\top\overline{\mathbf{H}}) = \lambda k$. Similarly, the inequality holds in the second term of objective function. Furthermore, since each subproblem is convex when optimizing one variable with others fixed, the objective function value is monotonically increasing. Thus, Algorithm 1 converges theoretically. □

G. Time Complexity Analysis

The time complexity of E^{2}OMVC mainly lies in the solution of $\{\mathbf{H}^{(i)}\}_{i=1}^v$, $\overline{\mathbf{H}}$, and the optimization of $\mathbf{H}^*$, $\{\mathbf{R}^{(i)}\}_{i=1}^v$, $\mathbf{Q}$, $\mathbf{Y}$, and $\boldsymbol{\gamma}$. For the solution of $\{\mathbf{H}^{(i)}\}_{i=1}^v$ and $\overline{\mathbf{H}}$, it costs $\mathcal{O}(nm^2)$, respectively, where m represents the number of anchors. As for updating $\mathbf{H}^* \in \mathbb{R}^{n\times k}$, it costs $\mathcal{O}(nk^2)$ in each iteration, where n and k denote the number of samples and clusters, respectively. For updating $\{\mathbf{R}^{(i)}\}_{i=1}^v \in \mathbb{R}^{k\times k}$ and $\mathbf{Q} \in \mathbb{R}^{k\times k}$, it costs $\mathcal{O}(vk^3)$ and $\mathcal{O}(k^3)$ in each iteration, respectively, where v denotes the number of views of multiview data. For updating $\mathbf{Y}$, it costs $\mathcal{O}(nk)$ in each iteration, and the time complexity of updating $\{\gamma\}_{i=1}^v$ is $\mathcal{O}(v)$. Accordingly, the total time complexity of the proposed method is $\mathcal{O}(2nm^2 + nk^2 + (v+1)k^3 + nk + v)$ which is linear with n.

VI. Experiments

In this section, we conduct extensive experiments to verify the effectiveness of E^{2}OMVC on different sample size datasets.

TABLE I
SUMMARY OF SIX BENCHMARK DATASETS

Dataset	#Sample	#View	#Cluster	# Feature
MSRCV1	210	5	7	24,576,512,256,254
100leaves	1600	3	100	64,64,64
Handwritten	2000	6	10	216,76,64,6,240,47
Caltech102	9144	4	102	1000,1000,1000,2048
Cifar10	10000	4	10	1000,1000,1000,2048
ALOI-100	10800	4	100	77,13,64,125

A. Datasets

In the experiment, six benchmark multiview datasets are selected to demonstrate the effectiveness of E^{2}OMVC, including MSRCV1, 100leaves, Handwritten, Caltech102, Cifar10, and ALOI-100 datasets.

1) *MSRCV1*[1]*:* This dataset contains 210 images with seven classes, and each image can be depicted by five different kinds of features, namely, LBP, GIST, SIFT, CENTRIST, and HOG.
2) *100leaves*[2]*:* The dataset consists of 1600 samples with 100 plant species. In addition, each sample can be represented by three different feature sets, including fine-scale margin, texture histogram, and shape descriptor.
3) *Handwritten*[3]*:* The dataset contains 2000 images with ten clusters, and it is described by six different features, i.e., 216-D FAC, 76-D FOU, 64-D KAR, 6-D MOR, 240-D pixel averages with 2×3 windows, and 47-D ZER.
4) *Caltech102*[4]*:* It is an object image dataset with 9144 samples. To obtain satisfactory clustering performance on this dataset, we adopt four deep neural network methods to extract the features of each image. To be specific, ResNet101, GoogLeNet, DenseNet, and Inception V3 are used to extract 1000-D, 1000-D, 1000-D, and 2048-D features of each image, respectively.
5) *Cifar10*[5]*:* This dataset contains 10 000 images with ten classes. We also adopt ResNet101, GoogleLeNet, DenseNet, and Inception V3 to extract the features of each image for the same reason as for the Caltech102 dataset.
6) *ALOI-100*[6]*:* This dataset contains 10 800 images with 100 objects. Four types of features are extracted to depict each image, including RGB, color similarity, HSV, and Haralick features.

The detailed information of these datasets is summarized in Table I.

[1]https://www.microsoft.com/en-us/research/project/
[2]https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set
[3]https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[4]http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[5]http://www.cs.toronto.edu/kriz/cifar.html
[6]http://elki.dbs.ifi.lmu.de/wiki/DataSets/MultiView

B. Compared Methods

With the above benchmark datasets, we compare the proposed method E^{2}OMVC with the following state-of-the-art algorithms.

1) *AASC [41]:* Affinity aggregation for spectral clustering.
2) *MKC [42]:* Multiview K-means clustering.
3) *MVSC [26]:* Large-scale multiview spectral clustering via bipartite graph.
4) *AMGL [43]:* Parameter-free autoweighted multiple graph learning.
5) *MVGL [2]:* Graph learning for multiview clustering.
6) *AWP [44]:* Adaptively weighted procrustes multiview clustering.
7) *GMC [45]:* Graph-based multiview clustering.
8) *SMSC [3]:* Multiview spectral clustering via integrating nonnegative embedding and spectral embedding.
9) *CoMVSC [46]:* Consensus one-step multiview subspace clustering.
10) *CGD [7]:* Multiview clustering via cross-view graph diffusion.
11) *MVRSE [47]:* Multiview clustering based on generalized low-rank approximation.
12) *LMVSC [48]:* Large-scale multiview subspace clustering in linear time.
13) *PFSC [49]:* Multiview subspace clustering via partition fusion.
14) *SMVSC [38]:* Scalable multiview subspace clustering with unified anchors.
15) *CDMGC [10]:* Measuring diversity in graph learning: a unified framework for structured multiview clustering.
16) *PMSC [50]:* Partition-level multiview subspace clustering.
17) *OPMC [51]:* One-pass multiview clustering for large-scale data.

C. Experimental Settings

In our experiments, all the datasets are normalized using the Z-score first. For those datasets with sample sizes 9000 and below, the similarity graphs of them are uniformly constructed via the k-nearest neighbor graph with 15 nearest neighbors. With respect to the datasets with more than 9000 samples, we conduct the k-means clustering algorithm in each view to partition the samples into their correspondence cluster, and the closest sample to the clustering center is selected as the anchors to construct the similarity graph. To enable all the competitors to work on all the datasets with the given number of anchors, the minimum value of anchors is set to 2% of the total number of samples and increased in the range of 2%–8% from which the best results are reported in the experiments. For the proposed method, the parameters λ and β are set to $[0.01, 0.05, 0.1, 0.5, 1]$ and $[0.01, 0.05, 0.1, 0.5, 1]$ with a grid search scheme, respectively. For all the compared methods, only six methods require additional parameter tuning, including MVSC, CoMVSC, LMVSC, PFSC, CDMGC, and PMSC. To ensure fairness of comparisons, the optimal results of these methods are reported in the experiments. Specifically, for MVSC, the one key parameter is tuned in the range of

TABLE II
CLUSTERING RESULTS ON THREE BENCHMARK DATASETS WITH SAMPLE SIZE 9000 AND BELOW (%)

Datasets	Methods	F-score	Precision	NMI	ARI	ACC	Purity	RunTime (s)
MSRCV1	AASC	**74.95±0.35**	**74.16±0.35**	75.00±0.34	**70.87±0.41**	**85.69±0.19**	**85.69±0.19**	1.46
	MKC	51.88±0.00	45.00±0.00	58.06±0.00	42.72±0.00	58.57±0.00	62.38±0.00	0.48
	MVGL	71.52±0.00	70.12±0.00	73.27±0.00	66.82±0.00	82.86±0.00	82.86±0.00	6.39
	AWP	59.36±0.00	52.54±0.00	63.68±0.00	51.80±0.00	66.67±0.00	67.14±0.00	0.39
	GMC	69.26±0.00	61.05±0.00	76.23±0.00	63.52±0.00	74.76±0.00	79.05±0.00	0.31
	SMSC	73.40±0.00	71.76±0.00	74.76±0.00	69.00±0.00	81.90±0.00	81.90±0.00	0.85
	CoMVSC	70.89±0.00	67.62±0.00	72.93±0.00	65.93±0.00	77.14±0.00	81.43±0.00	24.29
	MVRSE	74.15±0.30	73.32±0.32	74.63±0.33	69.92±0.35	84.79±0.11	84.79±0.11	0.69
	CDMGC	72.68±0.00	67.12±0.00	**77.37±0.00**	67.85±0.00	78.57±0.00	81.43±0.00	0.15
	Ours	**78.83±0.00**	**77.47±0.00**	**79.36±0.00**	**75.34±0.00**	**88.57±0.00**	**88.57±0.00**	0.29
100leaves	AASC	76.01±3.93	68.45±5.74	**93.69±0.64**	75.75±3.98	83.20±1.74	85.95±1.37	11.26
	MKC	1.86±0.00	0.94±0.00	00.00±0.00	00.00±0.00	1.00±0.00	1.00±0.00	936.24
	MVGL	17.98±0.00	10.23±0.00	82.18±0.00	16.60±0.00	65.94±0.00	69.50±0.00	255.54
	AWP	68.88±0.00	64.31±0.00	89.03±0.00	68.56±0.00	75.44±0.00	77.75±0.00	2.96
	GMC	29.75±0.00	18.52±0.00	86.23±0.00	28.67±0.00	71.88±0.00	75.62±0.00	4.88
	SMSC	78.85±0.00	76.47±0.00	92.14±0.00	78.65±0.00	**86.00±0.00**	87.25±0.00	6.73
	CoMVSC	32.07±0.00	25.21±0.00	71.59±0.00	31.25±0.00	45.12±0.00	48.31±0.00	24.29
	MVRSE	**79.69±1.47**	**76.41±1.46**	93.26±0.59	**79.48±1.48**	84.73±1.37	**86.55±1.18**	23.78
	CDMGC	61.06±0.00	49.77±0.00	92.22±0.00	60.61±0.00	75.88±0.00	85.50±0.00	10.91
	Ours	**82.71±0.00**	**77.04±0.00**	**95.64±0.00**	**82.54±0.00**	**92.00±0.00**	**92.87±0.00**	6.54
Handwritten	AASC	82.01±0.07	77.27±0.04	87.76±0.07	79.88±0.07	82.03±0.07	85.78±0.07	3.39
	MKC	75.00±0.00	71.63±0.00	80.58±0.00	72.09±0.00	79.00±0.00	83.80±0.00	36.09
	MVGL	85.36±0.00	79.34±0.00	90.83±0.00	83.61±0.00	85.75±0.00	88.05±0.00	115.48
	AWP	83.81±0.00	76.81±0.00	88.21±0.00	81.84±0.00	86.80±0.00	86.85±0.00	2.19
	GMC	86.40±0.00	83.35±0.00	89.93±0.00	84.84±0.00	88.05±0.00	88.05±0.00	19.40
	SMSC	84.26±0.00	79.25±0.00	89.57±0.00	82.40±0.00	82.85±0.00	87.65±0.00	11.85
	CoMVSC	80.21±0.00	79.37±0.00	85.68±0.00	77.99±0.00	**88.55±0.00**	**88.55±0.00**	832.48
	MVRSE	84.99±0.00	**84.93±0.00**	88.19±0.00	83.33±0.00	87.50±0.00	87.50±0.00	28.16
	CDMGC	**86.77±0.00**	80.30±0.00	**91.44±0.00**	**85.17±0.00**	87.40±0.00	87.70±0.00	22.87
	Ours	**94.11±0.00**	**94.07±0.00**	**93.18±0.00**	**93.46±0.00**	**97.00±0.00**	**97.00±0.00**	3.34

[0.1:0.2:2]. CoMVSC contains two parameters, and we tune them in the range of $[2^3{:}2^2{:}2^{13}]$ and [1.3:0.2:2.7], respectively. For LMVSC, the only parameter is tuned in the range of [0.001, 0.01, 0.1, 1, 10]. As for PFSC, the two parameters are tuned in the range of $[10^{-4}{:}10^2{:}10^4]$. The two parameters of CDMGC are tuned in the range of $[10^{-5}{:}10^1{:}10^5]$. In addition, the compared method PMSC involves three parameters, and we tune them in the range of [50:50:250], [0.01, 0.1, 1], and [0.001, 0.01] respectively.

To effectively evaluate the clustering performance of all the competitors on these multiview datasets, six metrics are selected to measure the experimental results, namely, F1 measure (F-score), precision, normalized mutual information (NMI), adjusted rand index (ARI), accuracy (ACC), and purity. All the methods are repeatedly run ten times, and the average results and standard deviations are reported in the experiments. Note that all the methods use MATLAB, and we conduct all the experiments on PC with Intel Core i7-7700 CPU and 24-GB RAM, MATLAB R2020a.

D. Experimental Results

1) Clustering Results on Datasets With Sample Size 9000 and Below: As mentioned above, for these kinds of datasets with small samples, the similarity graphs of these kinds of datasets are constructed with using k-nearest neighbor graph for all the samples. Note that some competitors are only applicable to anchor graphs in 17 multiview clustering methods, nine methods are selected as the competitors in the experiments, and their experimental results on three benchmark datasets are reported in Table II. To clearly show the clustering performance of all the competitors, two better clustering results are given in bold in Table II.

From the above results, the following observations can be obtained.

In most cases, the proposed method consistently achieves satisfying performance in terms of six metrics on three benchmark datasets. For example, E^{2}OMVC exceeds the second performer (CoMVSC) by 8.45% in terms of ACC on the Handwritten dataset in Table II. Nevertheless, a similar improvement can also be observed in the other datasets. Thus, these results clearly show that the proposed method is a promising method for multiview clustering.

The compared multiview clustering methods such as AASC, MVGL, and MVRSE need an additional postprocessing step to obtain the final clustering results on the consensus feature representation, and a large gap between them and the proposed

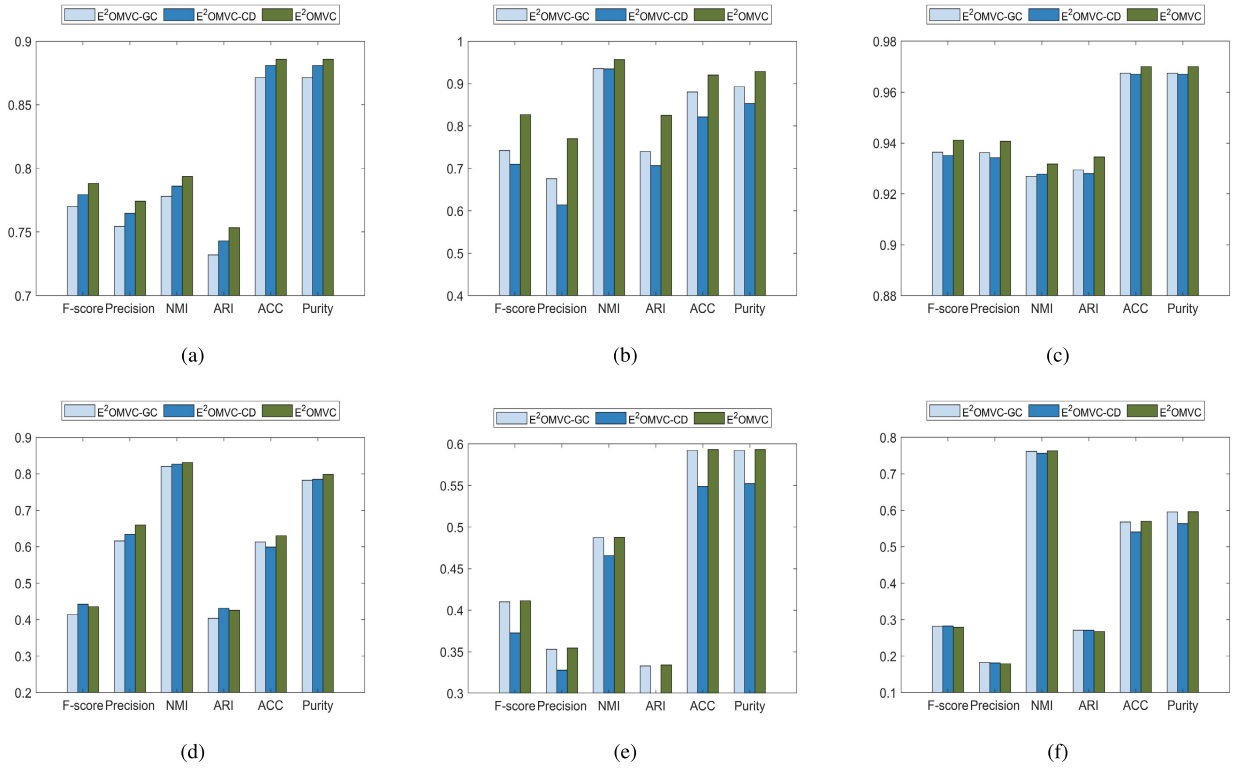

Fig. 1. Clustering performance of variant methods on all the datasets. (a) MSRCV1. (b) 100leaves. (c) Handwritten. (d) Caltech102. (e) Cifar10. (f) ALOI-100.

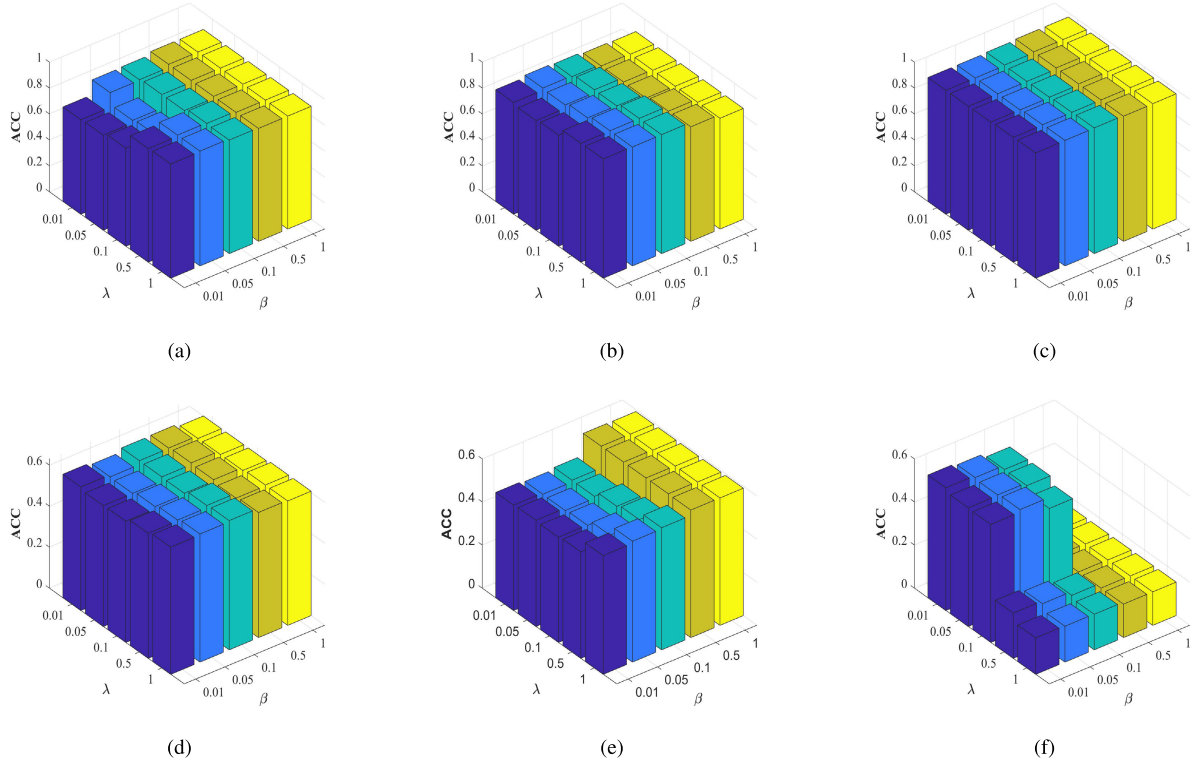

Fig. 2. Parameter sensitivity of the proposed method in terms of ACC. (a) MSRCV1. (b) 100leaves. (c) Handwritten. (d) Caltech102. (e) Cifar10. (f) ALOI-100.

method on three benchmark datasets is produced. The main reason probably can be attributed to the separation of consensus representation learning and clustering process, thereby the suboptimal solution is obtained.

2) Clustering Results on Datasets With More Than 9000 Samples: In this experiment, the similarity graph of each dataset is constructed using anchor graphs uniformly. In addition, since some competitors are not

TABLE III
CLUSTERING RESULTS ON THREE BENCHMARK DATASETS WITH MORE THAN 9000 SAMPLES (%),"NA" DENOTES OUT-OF-MEMORY FAILURE OR RUNNING TIME IS LARGER THAN 12 h

Datasets	Methods	F-score	Precision	NMI	ARI	ACC	Purity	RunTime (s)
Caltech102	MVSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	CGD	47.91±1.77	**74.60±1.40**	**84.05±0.36**	46.95±1.78	**64.26±1.07**	**79.62±0.62**	5868.70
	LMVSC	24.02±0.00	26.18±0.00	71.78±0.00	21.85±0.00	46.52±0.00	67.20±0.00	507.27
	PFSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SMVSC	**63.35±3.60**	59.81±2.56	77.70±0.80	**62.23±3.69**	57.72±1.55	63.75±1.12	5130.27
	CDMGC	31.66±0.00	19.64±0.00	76.56±0.00	28.41±0.00	59.45±0.00	64.95±0.00	706.85
	PMSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	OPMC	**51.36±0.00**	72.46±0.00	**83.93±0.00**	**50.37±0.00**	61.57±0.00	77.14±0.00	34.87
	Ours	48.28±0.00	**73.27±0.00**	83.07±0.00	47.30±0.00	**63.07±0.00**	**80.17±0.00**	33.07
Cifar10	MVSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	CGD	**43.04±0.23**	**39.68±0.46**	**49.56±0.08**	**36.11±0.31**	55.95±0.31	**59.25±0.11**	8073.71
	LMVSC	32.32±0.00	27.13±0.00	42.59±0.00	23.18±0.00	50.10±0.00	52.18±0.00	2625.07
	PFSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SMVSC	37.97±0.91	35.00±1.42	40.11±1.06	30.42±1.14	50.99±1.66	52.18±1.57	6028.79
	CDMGC	18.52±0.00	10.46±0.00	15.50±0.00	1.01±0.00	18.87±0.00	19.54±0.00	1220.45
	PMSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	OPMC	**41.18±0.00**	**40.46±0.00**	45.86±0.00	**34.52±0.00**	**57.16±0.00**	57.16±0.00	31.67
	Ours	41.15±0.00	35.45±0.00	**48.77±0.00**	33.41±0.00	**59.33±0.00**	**59.33±0.00**	9.43
ALOI-100	MVSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	CGD	1.01±0.01	0.98±0.01	11.18±0.13	0.00±0.00	3.81±0.04	3.97±0.06	3872.27
	LMVSC	1.07±0.00	0.99±0.00	11.11±0.00	0.00±0.00	3.91±0.00	4.05±0.00	1292.41
	PFSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SMVSC	12.58±0.20	7.31±0.14	54.45±0.27	11.07±0.21	26.15±0.52	27.21±0.52	6696.35
	CDMGC	3.17±0.00	1.62±0.00	47.38±0.00	1.26±0.00	29.09±0.00	33.09±0.00	1000.41
	PMSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	OPMC	**36.69±0.00**	**27.02±0.00**	**72.73±0.00**	**35.82±0.00**	**51.06±0.00**	**53.79±0.00**	6.89
	Ours	**41.84±0.00**	**33.33±0.00**	**76.63±0.00**	**41.10±0.00**	**56.97±0.00**	**60.20±0.00**	42.15

applicable to anchor graphs, we choose the results with using all the samples to construct the similarity graph as their corresponding clustering performance. Other experimental settings remain the same as aforementioned discussed.

The results of eight competitors on three benchmark datasets are shown in Table III. Similarly, two better clustering results are also in bold in Table III. According to the above results, the following observations can be obtained.

On all the datasets, the proposed method can achieve comparable clustering performance on the three benchmark datasets when compared with the eight state-of-the-art multiview clustering methods. Taking the ACC metric as an example, we can find that the proposed method exceeds the second performer OPMC by 2.17% and 5.91% on the Cifar10 and ALOI-100 datasets, respectively.

For the proposed method, the latent information is fused at the partition level, which is different from similarity-fuse methods, such as LMVSC and SMVSC. According to the experimental results, we can observe that E^{2}OMVC is superior to them. This is because the partition-level information is less noisy and feature redundancy than the similarity-level information. By virtue of the partition-level information, the subsequent clustering task can be well-guided and more accurate results are generated.

3) Running Time: To further verify the efficiency of the proposed method on these datasets, the running time of all the competitors is reported in Tables II and III. As seen in the tables, the proposed method also has the advantage of running time on these benchmark datasets. Compared with LMVSC, which has linear time complexity, we can find that the speedup of E^{2}OMVC greatly outperforms that of it on the datasets with more than 9000 samples. Thus, the proposed method with linear time complexity is more readily applied to large-scale multiview datasets.

Overall, the proposed method E^{2}OMVC is efficient and effective for multiview datasets with various sample sizes.

E. Model Evaluation

To further discuss the impact of different parts on the proposed method, the ablation experiments are conducted in this section. Specifically, two variants of E^{2}OMVC are created in the following.

1) Without Global Latent Feature Constraint: When the global latent feature constraint is removed, i.e., the parameter λ is set to zero, we denote the corresponding variant method by E^{2}OMVC-GC. Compared with E^{2}OMVC in Fig. 1 on six benchmark datasets, we can find that the global latent feature

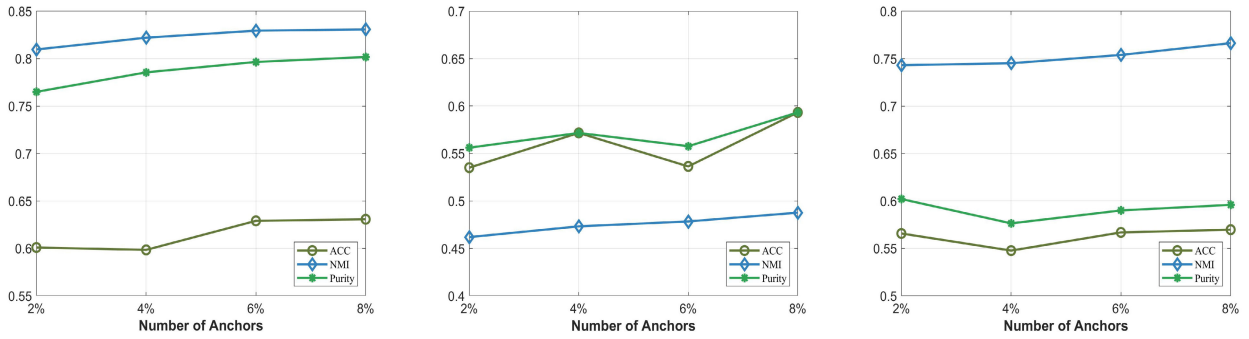

Fig. 3. Clustering performance of the proposed method with different numbers of anchors on (a) Caltech102, (b) Cifar10, and (c) ALOI-100 datasets, respectively.

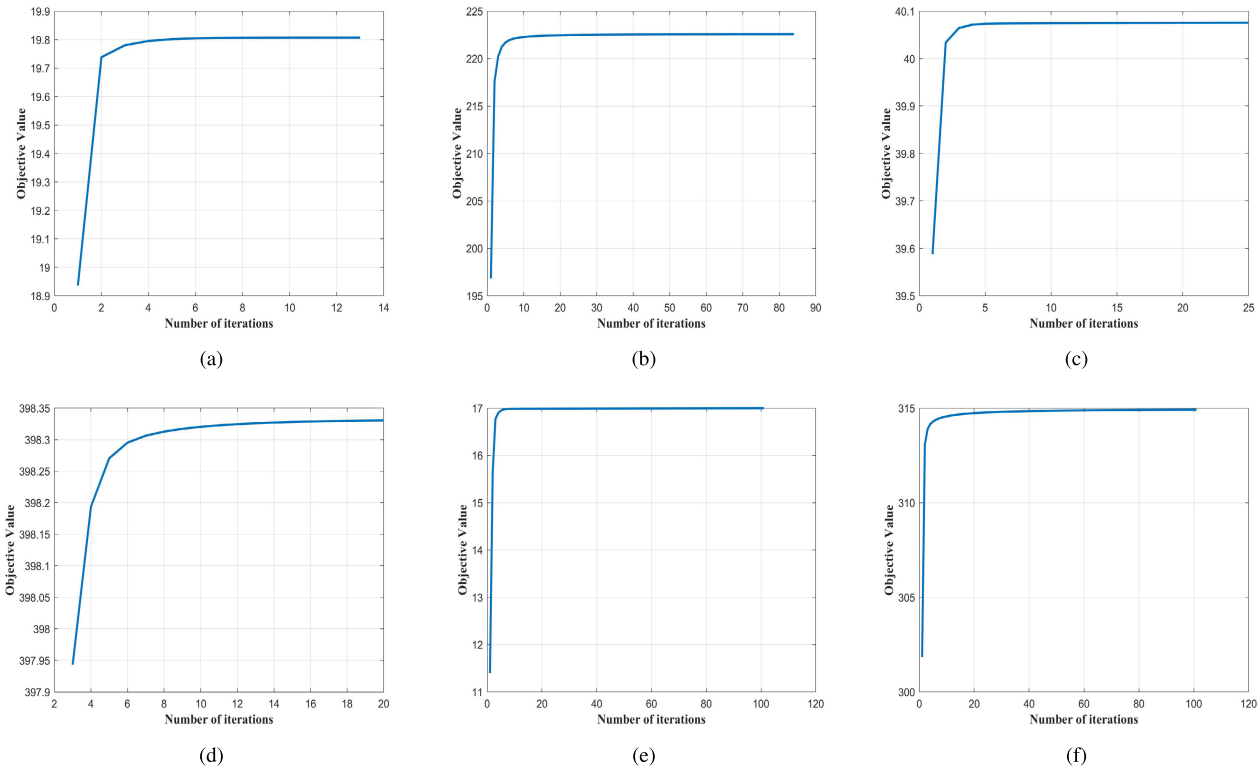

Fig. 4. Convergence curve of E^{2}OMVC on all the datasets. (a) MSRCV1. (b) 100leaves. (c) Handwritten. (d) Caltech102. (e) Cifar10. (f) ALOI-100.

constraint is helpful to improve the clustering performance on multiview datasets.

2) Without Clustering Indicators' Discretization: When the cluster discretization process is removed, i.e., the parameter β is set to zero, we denote the corresponding variant method by E^{2}OMVC-CD. After obtaining the unified latent partition representation $\mathbf{H}^*$ of the whole dataset, an additional postprocessing step is performed on it to generate the final clustering results. As seen in Fig. 1, the cluster discretization process can obtain a better clustering performance on these multiview datasets.

F. Parameter Sensitivity and Convergence Study

According to (9), we can find that the proposed method contains two parameters, i.e., λ and β. To further discuss the parameter sensitivity of the proposed method on all the datasets, the corresponding experiments are conducted in this section, as seen in Fig. 2. It can be clearly observed that some datasets are not sensitive to the parameters, such as 100leaves, Handwritten, and Caltech102. Although the clustering results of some datasets produce fluctuations within the given range of parameters, the proposed method can still achieve satisfying performance on all the datasets. Thus, it demonstrates that the utilization of global latent feature constraint and cluster discretization is effective to improve the clustering performance.

Furthermore, to study the impact of different numbers of anchors on the final clustering performance, the corresponding results are reported in Fig. 3. From these results, we can find that the clustering performance of E^{2}OMVC slightly fluctuates on three large-scale datasets. Therefore, it shows that the

proposed method can obtain satisfying performance with fewer anchors.

As aforementioned discussed, our method can achieve convergence theoretically. Here, the experiments are conducted to verify its convergence property. In Fig. 4, it is observed that the proposed method converges in several iterations, which demonstrates that the optimization algorithm is effective for solving each subproblem.

VII. Conclusion

In this article, an E^{2}OMVC method is presented. Specifically, based on anchor graphs, E^{2}OMVC constructs the similarity matrices of each view from which the low latent features are generated to reduce the time burden on addressing large-scale clustering problems. Then E^{2}OMVC takes the fusion of all the latent features and the generation of clustering indicators into a unified framework so that the two steps can reinforce each other. Extensive experimental results on six multiview datasets demonstrate the superior performance of E^{2}OMVC.

References

[1] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proc. ICDM*, vol. 4, Nov. 2004, pp. 19–26.

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

[3] 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, Mar. 2020.

[4] S. Wang, X. Liu, L. Liu, S. Zhou, and E. Zhu, "Late fusion multiple kernel clustering with proxy graph refinement," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 14, 2021, doi: 10.1109/TNNLS.2021.3117403.

[5] D. Wu, F. Nie, X. Dong, R. Wang, and X. Li, "Parameter-free consensus embedding learning for multiview graph-based clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7944–7950, Dec. 2022.

[6] Y. Chen, X. Xiao, Z. Hua, and Y. Zhou, "Adaptive transition probability matrix learning for multiview spectral clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4712–4726, Sep. 2022.

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

[8] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4833–4843, Oct. 2018.

[9] Z. Ren and Q. Sun, "Simultaneous global and local graph structure preserving for multiple kernel clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 1839–1851, May 2021.

[10] 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.

[11] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4279–4287.

[12] Y. Guo, "Convex subspace representation learning from multi-view data," in *Proc. AAAI Conf. Artif. Intell.*, 2013, vol. 27, no. 1, pp. 387–393.

[13] M. White, X. Zhang, D. Schuurmans, and Y.-L. Yu, "Convex multi-view subspace learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–14.

[14] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011, pp. 1413–1421.

[15] E. Eaton, M. des Jardins, and S. Jacob, "Multi-view clustering with constraint propagation for learning with an incomplete mapping between views," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2010, pp. 389–398.

[16] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 586–594.

[17] D. Xie, X. Zhang, Q. Gao, J. Han, S. Xiao, and X. Gao, "Multiview clustering by joint latent representation and similarity learning," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4848–4854, Nov. 2020.

[18] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[19] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, and Y. Yang, "Graph structure fusion for multiview clustering," *IEEE Trans. Know. Data Eng.*, vol. 31, no. 10, pp. 1984–1993, Oct. 2019.

[20] J. Wang et al., "Region-aware hierarchical latent feature representation learning-guided clustering for hyperspectral band selection," *IEEE Trans. Cybern.*, early access, Aug. 22, 2022, doi: 10.1109/TCYB.2022.3191121.

[21] C. Tang, X. Zheng, W. Zhang, X. Liu, X. Zhu, and E. Zhu, "Unsupervised feature selection via multiple graph fusion and feature weight learning," *Sci. China Inf. Sci.*, to be published.

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

[23] Z. Li, C. Tang, X. Liu, X. Zheng, W. Zhang, and E. Zhu, "Consensus graph learning for multi-view clustering," *IEEE Trans. Multimedia*, vol. 24, pp. 2461–2472, 2022.

[24] C. Tang et al., "Cross-view locality preserved diversity and consensus learning for multi-view unsupervised feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4705–4716, Oct. 2022.

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

[26] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp.1–7.

[27] 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, Apr. 2021.

[28] 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, 2020.

[29] J. Wen et al., "DIMC-Net: Deep incomplete multi-view clustering network," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 3753–3761.

[30] L. Huang, C.-D. Wang, and H.-Y. Chao, "Multi-view intact space clustering," *Pattern Recognit.*, vol. 86, pp. 344–353, Feb. 2018.

[31] S. Yu et al., "Optimized data fusion for kernel k-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, May 2012.

[32] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 1–7.

[33] G. Tzortzis and A. Likas, "Kernel-based weighted multi-view clustering," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 675–684.

[34] R. Zass and A. Shashua, "Doubly stochastic normalization for spectral clustering," in *Adv. Neural Inf. Process. Syst.*, vol. 19, 2006, pp. 1–8.

[35] X. Chen, W. Hong, F. Nie, D. He, M. Yang, and J. Z. Huang, "Spectral clustering of large-scale data by directly solving normalized cut," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1206–1215.

[36] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, and E. Zhu, "Unified one-step multi-view spectral clustering," *IEEE Trans. Knowl. Data Eng.*, early access, May 5, 2022, doi: 10.1109/TKDE.2022.3172687.

[37] X. Chen, F. Nie, J. Z. Huang, and M. Yang, "Scalable normalized cut with improved spectral rotation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1518–1524.

[38] M. Sun et al., "Scalable multi-view subspace clustering with unified anchors," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 3528–3536.

[39] J. Guo and J. Ye, "Anchors bring ease: An embarrassingly simple approach to partial multi-view clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 118–125.

[40] J. Huang, F. Nie, and H. Huang, "Spectral rotation versus k-means in spectral clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2013, vol. 27, no. 1, pp. 1–7.

[41] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Affinity aggregation for spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 773–780.
[42] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1–7.
[43] F. Nie et al., "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. IJCAI*, 2016, pp. 1881–1887.
[44] F. Nie, L. Tian, and X. Li, "Multiview clustering via adaptively weighted procrustes," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2022–2030.
[45] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, May 2019.
[46] P. Zhang et al., "Consensus one-step multi-view subspace clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4676–4689, Oct. 2022.
[47] Z. Li, Z. Hu, F. Nie, R. Wang, and X. Li, "Multi-view clustering based on generalized low rank approximation," *Neurocomputing*, vol. 471, pp. 251–259, Jan. 2022.
[48] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 4412–4419.
[49] J. Lv, Z. Kang, B. Wang, L. Ji, and Z. Xu, "Multi-view subspace clustering via partition fusion," *Inf. Sci.*, vol. 560, pp. 410–423, Jun. 2021.
[50] Z. Kang et al., "Partition level multiview subspace clustering," *Neural Netw.*, vol. 122, pp. 279–288, Feb. 2020.
[51] J. Liu et al., "One-pass multi-view clustering for large-scale data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 12344–12353.

Jun Wang received the B.E. degree in information management and information system from the Hubei University of Technology, Wuhan, China, in 2019. He is currently pursuing the master's degree with the School of Computer Science, China University of Geosciences, Wuhan.

His research interests are hyperspectral image processing and machine learning.

Chang Tang (Senior Member, IEEE) received the Ph.D. degree from Tianjin University, Tianjin, China, in 2016.

He was with AMRL, University of Wollongong, Wollongong, NSW, Australia, from September 2014 to September 2015. He is currently a Full Professor with the School of Computer Science, China University of Geosciences, Wuhan, China. He has published more than 50 peer-reviewed papers, including those in highly regarded journals and conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (T-PAMI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (T-KDE), IEEE TRANSACTIONS ON MULTIMEDIA (T-MM), IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS (T-HMS), IEEE SIGNAL PROCESSING LETTERS (SPL), AAAI, IJCAI, ICCV, CVPR, ACMM, and ICME. His current research interests include machine learning and computer vision.

Dr. Tang regularly served on the technical program committees of top conferences, such as NIPS, ICML, IJCAI, ICME, AAAI, ICCV, and CVPR.

Zhiguo Wan (Member, IEEE) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 2002, and the Ph.D. degree in information security from the National University of Singapore, Singapore, in 2007.

He was a Post-Doctoral Researcher with Katholieke Universiteit Leuven, Leuven, Belgium, and an Assistant Professor with the School of Software, Tsinghua University. He is currently a Principal Investigator with the Zhejiang Lab, Hangzhou, Zhejiang, China. His main research interests include security and privacy for cloud computing, the Internet-of-Things, and blockchain.

Wei Zhang (Member, IEEE) received the B.E. degree from Zhejiang University, Hangzhou, China, in 2004, the M.S. degree from Liaoning University, Shenyang, China, in 2008, and the Ph.D. degree from the Shandong University of Science and Technology, Qingdao, China, in 2018.

He is currently an Associate Professor with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. His research interests include future generation network architectures, edge computing, and edge intelligence.

Kun Sun (Member, IEEE) received the Ph.D. degree from the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology (HUST), Wuhan, China.

He is currently an Associate Professor with the School of Computer Science, China University of Geosciences (CUG), Wuhan, China. His research focuses on 3-D-related computer vision algorithms, such as multiview image matching, large-scale structure from motion (SfM), and 3-D point cloud processing.

Albert Y. Zomaya (Fellow, IEEE) is currently the Peter Nicol Russell Chair Professor of computer science with the School of Computer Science, The University of Sydney, Camperdown, NSW, Australia, and serves as the Director for the Center for Distributed and High-Performance Computing. He has published more than 700 scientific papers and articles and is the author, coauthor, or editor of more than 30 books. His research interests are in the areas of parallel and distributed computing, networking, and complex systems.

Prof. Zomaya is a Decorated Scholar with numerous accolades, including the Fellowship of AAAS and IET. He is also a fellow of the Australian Academy of Science and the Royal Society of New South Wales, a Foreign Member of the Academia Europaea, and a member of the European Academy of Sciences and Arts. He is the Editor-in-Chief of the *ACM Computing Surveys* and serves as an associate editor for several leading journals.