Fast Parameter-Free Multi-View Subspace Clustering With Consensus Anchor Guidance

Siwei Wang, Xinwang Liu, *Senior Member, IEEE*, Xinzhong Zhu, *Member, IEEE*, Pei Zhang, Yi Zhang, Feng Gao, *Member, IEEE*, and En Zhu

***Abstract*—Multi-view subspace clustering has attracted intensive attention to effectively fuse multi-view information by exploring appropriate graph structures. Although existing works have made impressive progress in clustering performance, most of them suffer from the cubic time complexity which could prevent them from being efficiently applied into large-scale applications. To improve the efficiency, anchor sampling mechanism has been proposed to select vital landmarks to represent the whole data. However, existing anchor selecting usually follows the heuristic sampling strategy, e.g. k-means or uniform sampling. As a result, the procedures of anchor selecting and subsequent subspace graph construction are separated from each other which may adversely affect clustering performance. Moreover, the involved hyper-parameters further limit the application of traditional algorithms. To address these issues, we propose a novel subspace clustering method termed Fast Parameter-free Multi-view Subspace Clustering with Consensus Anchor Guidance (FPMVS-CAG). Firstly, we jointly conduct anchor selection and subspace graph construction into a unified optimization formulation. By this way, the two processes can be negotiated with each other to promote clustering quality. Moreover, our proposed FPMVS-CAG is proved to have linear time complexity with respect to the sample number. In addition, FPMVS-CAG can automatically learn an optimal anchor subspace graph without any extra hyper-parameters. Extensive experimental results on various benchmark datasets demonstrate the effectiveness and efficiency of the proposed method against the existing state-of-the-art multi-view subspace clustering competitors. These merits make FPMVS-CAG more suitable for large-scale subspace clustering. The code of FPMVS-CAG is publicly available at https://github.com/wangsiwei2010/FPMVS-CAG.**

***Index Terms*—Large-scale clustering, multiple subspace clustering, multiple view clustering.**

I. Introduction

Clustering plays a vital role among unsupervised image processing, data mining and machine learning community. With the rapid growth of unlabelled collected data, many clustering algorithms have been proposed to automatically discover the intrinsic structure and group similar items into the same clusters, e.g. k-means, spectral clustering and kernel k-means clustering [1]–[9]. Among the several commonly-adopted clustering approaches, subspace clustering aims to explore topological pairwise relations of data with graph structure. As an important extension, self-expressive subspace clustering assumes that the data samples can be represented by linear combinations of themselves and therefore can be separated by independent subspaces. After obtaining the symmetrical self-representation matrix, the traditional spectral clustering is then adopted to get the final clustering result. However, real data are described in various forms or collected from different sources. For example, texts with the same semantic meanings can be written in different languages (English, Chinese and French so on). To better collect complementary information among the provided multiple information, multi-view subspace clustering (MVSC) is proposed in recent years and has attracted massive attention. Most MVSC methods adopt multi-view subspace regularization stereotype to capture the inter-view relationships for multi-view clustering [10]–[13], [13]–[20].

Although these aforementioned MVSC approaches are able to solve multi-view clustering from various aspects, most of the existing algorithms suffer from the high time complexity (normally $\mathcal{O}(n^3)$ or even higher) limiting their applications in real-world applications. The high time complexity consists of two major parts: (i) the graph construction stage. Normally, building the self-representation graph matrix needs $\mathcal{O}(n^4)$ (see section II-A for details).(ii) spectral clustering stage. After obtaining the similarity matrix, the clustering stage needs $\mathcal{O}(n^3)$ for Singular Value Decomposition (SVD). With the rapid growth of data, it is difficult to efficiently run the existing approaches with limited computational sources.

Recently, to improve the efficiency of subspace algorithms, anchor graph strategy has been proposed to select landmarks to represent the entire data items [8], [12], [21]–[29]. Then the size of individual view graph has been reduced from $n \times n$ to $n \times l$ while the l denotes the number of landmarks. However,

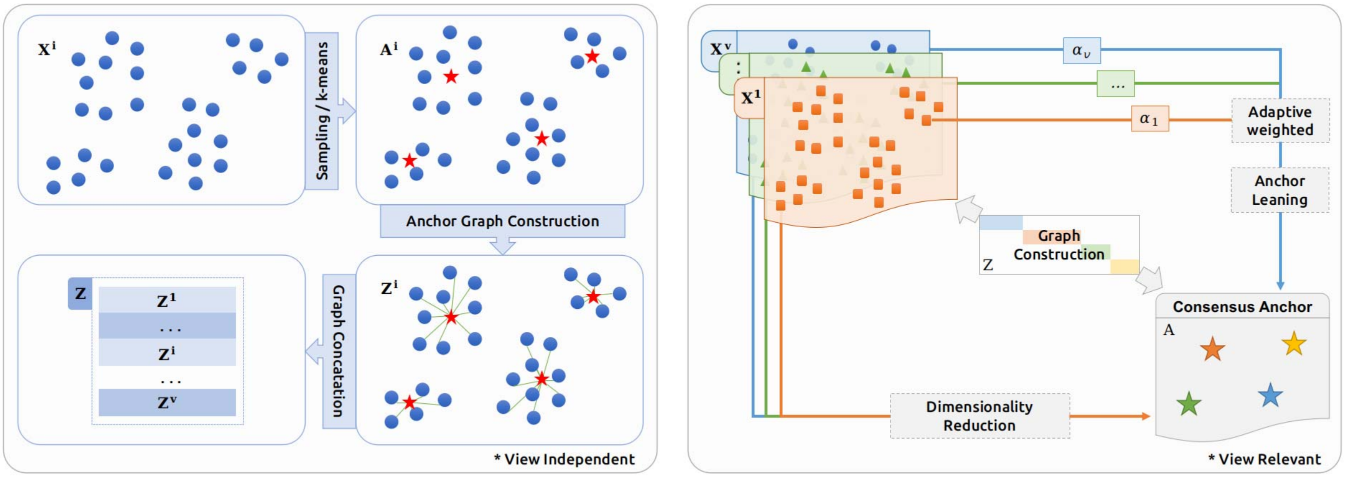
Manuscript received July 13, 2021; revised November 10, 2021; accepted November 23, 2021. Date of publication December 10, 2021; date of current version December 20, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2020AAA0107100 and in part by the Natural Science Foundation of China under Project 61922088, Project 61773392, Project 61872377, and Project 61976196. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kui Jia. *(Corresponding author: Xinwang Liu.)*

Siwei Wang, Xinwang Liu, Pei Zhang, Yi Zhang, and En Zhu are with the School of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: fwangsiwei13@nudt.edu.cn; xinwangliu@nudt.edu.cn; zhangpei@nudt.edu.cn; enzhug@nudt.edu.cn).

Xinzhong Zhu is with the College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua, Zhengjiang 321004, China (e-mail: zxz@zjnu.edu.cn).

Feng Gao is with the School of Arts, Peking University, Beijing 100871, China (e-mail: gaof@pku.edu.cn).

Digital Object Identifier 10.1109/TIP.2021.3131941



(a) Traditional Methods

(b) Our Method

Fig. 1. The framework comparison between traditional methods (left) and ours (right). Traditional methods separately select anchors by random sampling or k-means in each view, and then conduct the graph construction. On the contrary, our method adaptively **learns** the consensus low-dimension anchors with multi-view information and simultaneously construct the anchor graph. These two parts are jointly optimized and boosted each other.

existing anchor graph work adopts the heuristic sampling strategy, e.g. k-means or uniform sampling. Therefore, the anchor selecting and the followed subspace graph construction are separated from each other which may adversely affect clustering performance. Moreover, existing work contains massive involved hyper-parameters which further limits their application on large-scale modern data.

To further improve the effectiveness and efficiency, we propose a novel method termed Fast Parameter-free Multi-view Subspace Clustering with Consensus Anchor Guidance (FPMVS-CAG) in this paper. The framework of our proposed method is shown in Fig. 1. Different from existing large-scale work, we firstly jointly conduct anchor selection and the followed subspace graph construction into a unified optimization. The two processes can negotiate with each other to promote clustering quality. To solve the resultant optimization problem, we design a four-step alternate optimization algorithm with proved convergence. By the virtue of it, FPMVS-CAG is proved to have linear time complexity respecting to the sample number. More specially, FPMVS-CAG can automatically learn an optimal low-rank anchor subspace graph without additional hyper-parameters as previous methods do. The two factors contribute to FPMVS-CAG more suitable for large-scale subspace clustering. Extensive experimental results on various benchmark datasets demonstrate the effectiveness and efficiency of the proposed method when compared to the existing state-of-the-art multi-view clustering competitors.

In general, the main contributions of this paper are threefold:

1) Different from existing anchor sampling strategy, we firstly combine anchor learning and the graph construction into a unified framework. The two counterparts contribute to each other and are jointly optimized so that the learned consensus anchor graph can better utilize multi-view information.
2) FPMVS-CAG naturally constructs the graph satisfying low-rank property, which does not involve any hyperparameter as previous methods do. More importantly, our proposes method with linear time complexity is proved to be more efficient and effective to large-scale subspace clustering problems.
3) We design a four-step alternating optimization algorithm to solve the resultant optimization problem with theoretically proved convergence. Extensive experimental results demonstrate the superiority and efficiency of our proposed algorithm, on large-scale multi-view datasets (even more than 100000 samples).

The rest of this paper is organized as follows. Section II outlines the related work of multi-view subspace clustering. Section III presents the proposed formula and the four-step alternate algorithm. Further, we also provide analysis of the convergence and the computational complexity of our two proposed algorithm. Section IV shows the experiment results with evaluation. Section V concludes the paper.

II. Related Work

In this section, we introduce existing work most related to our study in this paper, including subspace clustering algorithm, multi-view clustering and large-scale multi-view clustering. Table I lists main notations used throughout the paper.

A. Self-Expressive Subspace Clustering

Given a single-view data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$ with n samples and d dimension, subspace clustering is classified two stages: graph construction and spectral embedding stages.

1) Graph Construction Stage: Firstly, we build subspace graph by minimizing the self-expressive reconstruction error as follow,

$$\min_{\mathbf{S}} ||\mathbf{X} - \mathbf{X}\mathbf{S}||_{\mathbf{F}}^2 + \Psi(\mathbf{S}), \quad \text{s.t. } \mathbf{S} \geq 0, \mathbf{S}^\top \mathbf{1} = \mathbf{1}. \tag{1}$$

TABLE I

MAIN NOTATIONS USED THROUGHOUT THE PAPER

Notation	Meaning
n	The number of samples
v	The number of views
k	The number of clusters
l	The number of selected anchors
$\boldsymbol{\alpha} \in \mathbb{R}^{v \times 1}$	The view coefficient vector
d_i	The dimension for the i-th view
h	$\sum_{i=1}^{v} d_i$
$\mathbf{X} \in \mathbb{R}^{d \times n}$	The single-view data matrix
$\mathbf{X}_i \in \mathbb{R}^{d_i \times n}$	The data matrix for the i-th view
$\mathbf{W}_i \in \mathbb{R}^{d \times k}$	The projection matrix for the i-th view
$\mathbf{S}_i$	The self-expressive representation matrix for the i-th view
$\mathbf{A} \in \mathbb{R}^{k \times k}$	The consensus anchors matrix
$\mathbf{H} \in \mathbb{R}^{n \times k}$	The spectral embedding matrix
$\mathbf{Z} \in \mathbb{R}^{k \times n}$	The consensus anchor graph

where Ψ denotes for the various forms of regularization terms on $\mathbf{S}$. For example, ℓ_1 norm ensures sparsity and ℓ_* (nuclear-norm) seeks for the low-rank property. The $\mathbf{S} \geq 0$ and $\mathbf{S}^\top \mathbf{1} = \mathbf{1}$ ensure the non-negativity and normalization of the obtained self-expressive representation matrix $\mathbf{S}$.

Suppose the given regularization norm is Frobenius norm, then Eq. (1) is introduced with the balanced hyperparameter λ as follows,

$$\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{X}\mathbf{S}\|_{\mathbf{F}}^2 + \lambda \|\mathbf{S}\|_{\mathbf{F}}^2, \quad \text{s.t. } \mathbf{S} \geq 0, \mathbf{S}^\top \mathbf{1} = \mathbf{1}. \tag{2}$$

Eq. (2) can be further divided into n subproblems,

$$\min \frac{1}{2} \mathbf{S}_{:,j}^\top \mathbf{M} \mathbf{S}_{:,j} + f^\top \mathbf{S}_{:,j}, \quad \text{s.t. } \mathbf{S}_{:,j}^\top \mathbf{1} = 1, \ \mathbf{S} \geq 0, \tag{3}$$

where $\mathbf{M} = 2(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})$, $f^\top = -2(\mathbf{X}^\top \mathbf{X})_{:,j}^\top$. Since $\mathbf{M}$ is absolutely positive definite matrix, each subproblem in Eq. (3) is a quadratic programming (QP) problem and can be solved in $\mathcal{O}(n^3)$ to get the optimal value. Therefore, the first graph construction stage in Eq. (2) needs $\mathcal{O}(n^4)$. Some recent methods accelerate the time complexity into $\mathcal{O}(n^3)$ per iteration [3], [15], [30]–[35].

2) Spectral Embedding Stage: After obtaining the self-expressive representation matrix $\mathbf{S}$, the symmetric similarity matrix are regarded as $\frac{\mathbf{S}+\mathbf{S}^\top}{2}$. Then spectral clustering is then applied to get the spectral embedding $\mathbf{H} \in \mathbf{R}^{n \times k}$,

$$\max_{\mathbf{H}} \operatorname{Tr}(\mathbf{H}^\top \frac{\mathbf{S} + \mathbf{S}^\top}{2} \mathbf{H}), s.t. \ \mathbf{H} \in \mathbb{R}^{n \times k}, \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k, \tag{4}$$

where $\mathbf{H}$ is regarded as the spectral embedding of data matrix and is input to k-means algorithm to get the final clustering result. To solve Eq. (4), we need to do singular value decomposition (SVD) on $\mathbf{S}$ which needs $\mathcal{O}(n^3)$.

B. Multi-View Subspace Clustering

As an extension of traditional single-view subspace clustering, multi-view subspace clustering assumes that data usually lie in underlying low-dimension subspaces rather than distribute uniformly in the entire space.

It is natural to expand Eq. (1) into multi-view setting. Mathematically, given the multi-view data $\{\mathbf{X}_i\}_{i=1}^{v} \in \mathbb{R}^{d_i \times n}$ with d_i dimension feature in the i-th view, n the number of samples, the fundamental equation of multi-view subspace clustering can be expressed as follows,

$$\min_{\mathbf{S}} \sum_{i=1}^{v} \underbrace{\|\mathbf{X}_i - \mathbf{X}_i \mathbf{S}_i\|_{\mathbf{F}}^2}_{Graph\ Construction} + \underbrace{\mathbf{\Omega}(\mathbf{S}\ ,\ \mathbf{S}_i)}_{Fusion},$$
$$\text{s.t. } diag(\mathbf{S}_i) = 0, \ diag(\mathbf{S}) = 0, \ \mathbf{S}_i^\top \mathbf{1} = \mathbf{1}, \tag{5}$$

where $\mathbf{\Omega}$ refers to the consensus regularization term that could co-train a global graph among different views. After obtaining the fused global graph $\mathbf{S}$, the final clustering result can be reached by performing spectral clustering on $\mathbf{S}$.

Many multi-view subspace clustering methods are proposed along with this framework due to their ability to capture the global structure. However, Gao *et al.* [36] think it is unreasonable to align multi-view graphs into a unified one directly since the magnitude of $\mathbf{S}_i$ is different. They incorporate spectral clustering into their objective by using graphs of all views to obtain a uniform partition matrix. Considering that learning representation independently cannot ensure the complementary information, Cao *et al.* [37] propose to induce Hilbert-Schmidt Independence Criterion (HSIC) to explore diverse subspace representations. Zhang *et al.* [9], [30] reconstruct the data points in a latent space which could make subspace representation more accurate and robust. Since most of the multi-view subspace methods only explore diversity or consistency information, Luo *et al.* [38] propose to learn the subspace representation with consistency and specificity jointly. Some methods aim at finding a proper constraint for the subspace, such as low-rank or sparse or both [39]. Another theme in this field is fusing multiple information in partition level [28], [29], [31], [33], [40]–[42].

The approaches mentioned above have promising results in terms of clustering performance. However, regardless of the level of fusion, multi-view subspace clustering cannot avoid performing graph construction, which requires a time complexity of $O(n^3)$ and a space complexity of at least $O(n^2)$. This dramatically limits the scalability of the multi-view subspace clustering approach.

C. Large-Scale Multi-View Clustering

Anchor graph has been widely regarded as an effective strategy to deal with large-scale datasets in multi-view spectral clustering [8] and multi-view subspace clustering [24], etc. The main proposal of anchor graph is to select/sample a relative small proportion of representative landmarks and explore the relationship between themselves and the original samples. Therefore, the size of anchor graph has been reduced from $\mathbf{S} \in \mathbb{R}^{n \times n}$ to $\mathbf{Z} \in \mathbb{R}^{l \times n}$ while the l denotes the number of landmarks. It is to see that we use $\mathbf{Z}^\top \Delta^{-1} \mathbf{Z}$ as the input in traditional spectral clustering stage. As demonstrated by former anchor subspace methods, the anchor graph can help reduce both storage and computational time while providing comparable clustering performance.

Recently, as a representative of multi-view anchor graph approaches, Kang *et al.* [24] propose a method termed as Large-scale Multi-view Subspace Clustering in Linear

Time (LMVSC). Firstly, they perform k-means in each view to obtain the view-specific anchors and then construct the respective anchor graphs between anchors and all samples on each view. Moreover, Li *et al.* propose an alternate anchor sampling strategy to build individual anchor graphs and then combine them into the consensus graph [12].

Although these methods have fulfilled anchor-based scalable multi-view clustering form several aspects, existing work can be improved with the following considerations: (i) The sampling procedure is isolated from the multi-view clustering process and the anchors selection is performed independently in each view without information connection with other views. (ii) Existing work contains massive involved hyper-parameters which further limits their application on large-scale modern data. In the next section, we propose a novel method termed as Fast Parameter-free Multi-view Subspace Clustering with Consensus Anchor Guidance (FPMVS-CAG) to solve this problem.

III. Methodology

In this section, we firstly describe the motivation and the formulation of our proposed FPMVS-CAG method with proved convergence. Then, we show the respective optimization process and analyze the difference of the proposed method with its competitors. Moreover, the time complexity and space complexity are introduced to illustrate the time and space efficiency of our method.

A. Motivation and Proposed Formula

Researchers utilize all the original data points to represent each point in the self-representation strategy, which is widely used in multi-view subspace clustering. Although a global relationship is well explored, the optimization time and storage cost related to the global graph restrict the scalability of multi-view subspace clustering. Besides, depicting one point with all samples is unnecessary and redundant. Therefore, we adopt the anchor strategy [8], [24] to select a small set of data points called anchor points or landmarks to reconstruct the underlying subspace and capture the manifold structure.

It can be concluded from the existing multi-view anchor-based subspace methods that they adopt the heuristic anchor sampling strategy, e.g. k-means or uniform sampling. Therefore, the anchor selecting and latter subspace graph construction are separated from each other which may adversely affect clustering performance.

Different from traditional strategy, we decide to *learn* anchors automatically not based on sampling. Moreover, multi-view clustering holds the assumption that there shares a consensus latent data distribution. Therefore, it is reasonable that the anchors should be consistent in latent space. To accomplish this, we define the respective projection matrix $\{\mathbf{W}_i\}_{i=1}^{v}$ aiming at the consensus anchor guidance. With the consensus latent anchor $\mathbf{A}$, we define the optimization goal as follows,

$$\min_{\boldsymbol{\alpha},\mathbf{W}_i,\mathbf{A},\mathbf{Z}} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{Z}\|_{\mathbf{F}}^2,$$
$$\text{s.t. } \boldsymbol{\alpha}^\top\mathbf{1} = 1,\ \mathbf{W}_i^\top\mathbf{W}_i = \mathbf{I}_k, \mathbf{A}^\top\mathbf{A} = \mathbf{I}_k, \mathbf{Z} \geq 0, \mathbf{Z}^\top\mathbf{1} = \mathbf{1}. \tag{6}$$

In Eq. (6), $\mathbf{X}_i \in \mathbb{R}^{d_i \times n}$ is the i-th view of original data where d_i is the dimension of corresponding view, n is the number of samples. $\mathbf{A} \in \mathbb{R}^{k\times k}$ is the unified anchor matrix with k selected anchors and dimension.

Although Eq. (6) seems simple, we summarize the advantages of the newly-proposed model as follows:

- Naturally induced **low-rank** property: Since the whole similarity matrix can be formed as $\mathbf{S} = \mathbf{Z}^\top\Delta^{-1}\mathbf{Z}$, it is easy to obtain that $rank(\mathbf{S}) = rank(\mathbf{Z}^\top\Delta^{-1}\mathbf{Z}) \leq rank(\mathbf{Z}) = k$. Therefore comparing to existing nuclear-norm constraint, our model naturally output the low-rank similarity matrix.
- Different from existing anchor sampling strategy, the anchor selection and the graph construction are combined into a unified framework. The two counterparts contribute to each other and are jointly optimized together so that the **learned** consensus anchors can better capture multi-view information.
- **Parameter-free**: our model does not involve any hyper-parameters as previous methods do, which is more suitable for practical large-scale clustering problems.
- **Adaptive weighted**: comparing to fixed equally-weighted coefficients in existing multi-view anchor graph methods, FPMVS-CAG can adaptively learn the view coefficients due to their contributions to the consensus graph, which is more reasonable in applications

Once the anchor graph $\mathbf{Z}$ is constructed, the $\mathbf{Z}^\top\Delta^{-1}\mathbf{Z}$ is used as the input for the second spectral embedding stage where $\Delta_{ii} = \sum_{j=1}^{n}\mathbf{Z}_{ji}$. Unlike traditional solutions, we can quickly and easily obtain the spectral embedding $\mathbf{H}$ according to the following Theorem. The spectral embedding $\mathbf{H}$ is obtained as the eigenvectors of $\mathbf{Z}^\top\Delta^{-1}\mathbf{Z}$.

Theorem 1: The right singular vectors of $\mathbf{Z}$ *is the same as the eigenvectors of* $\mathbf{Z}^\top\Delta^{-1}\mathbf{Z}$.

Proof: Suppose the singular value decomposition (SVD) of $\mathbf{Z}$ is $\mathbf{Z} = \mathbf{U}\Sigma\mathbf{V}^\top$, we can easily see that $\mathbf{Z}^\top\Delta^{-1}\mathbf{Z} = \mathbf{V}\Sigma^\top\Delta^{-1}\Sigma\mathbf{V}^\top$. Therefore, the right singular vector of $\mathbf{Z}$ is the same as the eigenvectors of $\mathbf{Z}^\top\Delta^{-1}\mathbf{Z}$. This completes the proof. □

According to the Theorem 1, we can conclude that the spectral embedding $\mathbf{H}$ can be obtained by performing SVD on the anchor graph $\mathbf{Z}$ which only needs $\mathcal{O}(nk^2)$ instead of existing $\mathcal{O}(n^3)$.

B. Optimization

The optimization problem in Eq. (6) is not jointly convex when all variables are considered simultaneously. Therefore, we propose an alternating optimization algorithm to optimize each variable with the other variables been fixed. After that, we provide the total framework of optimization algorithm and time/space complexity analysis.

1) Update $\mathbf{W}_i$*:* When $\mathbf{A}$, $\mathbf{Z}$ and α_i are fixed, the objective function w.r.t. $\mathbf{W}_i$ can be formulated as

$$\min_{\mathbf{W}_i} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{Z}\|_{\mathbf{F}}^2,\ \text{s.t. } \mathbf{W}_i^\top\mathbf{W}_i = \mathbf{I}_k. \tag{7}$$

Since each $\mathbf{W}_i$ is separated from each other in terms of corresponding views, thus we can transform Eq. (7) into the following equivalent problem by expanding the Frobenius norm by trace and removing the items that are not related to $\mathbf{W}_i$.

$$\max_{\mathbf{W}_i} \operatorname{Tr}(\mathbf{W}_i^\top \mathbf{B}_i), \ \text{s.t. } \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I}_k, \tag{8}$$

where $\mathbf{B}_i = \mathbf{X}_i \mathbf{Z}^\top \mathbf{A}^\top$. Supposing the Singular value decomposition (SVD) result of $\mathbf{B}_i$ is $\mathbf{U}\Sigma\mathbf{V}^\top$, the optimal $\mathbf{W}_i$ can be easily obtained by calculating $\mathbf{U}\mathbf{V}^\top$ according to [43].

Calculating each $\mathbf{B}_i$ needs $\mathcal{O}(d_i nk + d_i k^2)$. Hence for the v sub-problems, calculating $\mathbf{B}$ needs $\mathcal{O}(hnk + hk^2)$. Moreover, solving Eq. (8) needs $\mathcal{O}(k^2 \sum_{i=1}^{v} d_i) = \mathcal{O}(k^2 h)$. Therefore, the total time complexity of updating $\mathbf{W}$ stage is $\mathcal{O}(hnk + hk^2)$.

2) Update $\mathbf{A}$*:* With $\mathbf{W}_i$, $\mathbf{Z}$ and α_i being fixed, the optimization for $\mathbf{A}$ can be transformed into solving the following problem

$$\min_{\mathbf{A}} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A}\mathbf{Z}\|_{\mathbf{F}}^2, \ \text{s.t. } \mathbf{A}^\top \mathbf{A} = \mathbf{I}_k. \tag{9}$$

Similar to the optimization of $\mathbf{W}_i$, the optimization of $\mathbf{A}$ in Eq. (9) equals to the following form

$$\max_{\mathbf{A}} \operatorname{Tr}(\mathbf{A}^\top \mathbf{C}), \ \text{s.t. } \mathbf{A}^\top \mathbf{A} = \mathbf{I}_k, \tag{10}$$

where $\mathbf{C} = \sum_{i=1}^{v} {\alpha_i}^2 \mathbf{W}_i^\top \mathbf{X}_i \mathbf{Z}^\top$. Similarly, the optimal solution of updating variable $\mathbf{A}$ can be attained the multiply of left singular matrix and the right singular matrix of $\mathbf{C}$. Similar to $\mathbf{B}$, to obtain the consensus anchor matrix $\mathbf{A}$ needs $\mathcal{O}(hnk + k^3)$.

3) Update $\mathbf{Z}$*:* Fixing other variables $\mathbf{W}_i$, $\mathbf{A}$ and α_i, the optimization problem for updating variable $\mathbf{Z}$ can be rewritten as,

$$\begin{aligned} &\min_{\mathbf{Z}} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A}\mathbf{Z}\|_{\mathbf{F}}^2, \\ &\text{s.t. } \mathbf{Z} \geq 0, \ \mathbf{Z}^\top \mathbf{1} = \mathbf{1}. \end{aligned} \tag{11}$$

The above optimization problem of $\mathbf{Z}$ can be easily formulated as the following Quadratic Programming (QP) problem as the former described,

$$\begin{aligned} &\min \frac{1}{2} \mathbf{Z}_{:,j}^\top \mathbf{G} \mathbf{Z}_{:,j} + f^\top \mathbf{Z}_{:,j}, \\ &\text{s.t. } \mathbf{Z}_{:,j}^\top \mathbf{1} = 1, \ \mathbf{Z} \geq 0, \end{aligned} \tag{12}$$

where $\mathbf{G} = 2(\sum_{i=1}^{v} \alpha_i^2)\mathbf{I}$, $f^\top = -2\sum_{i=1}^{v} \mathbf{X}_{i[:,j]}^\top \mathbf{W}_i \mathbf{A}$. Optimization can be performed by solving the QP problem for each row of $\mathbf{Z}$. Because the each row of $\mathbf{Z}$ is a k-dimensional vector, in this sub-problem the time complexity is $\mathcal{O}(nk^3)$.

4) Update α_i*:* Fixing the irrelevant variables, we can obtain the optimization problem for updating α_i.

$$\min_{\alpha_i} \sum_{i=1}^{v} \alpha_i^2 r_i^2, \ \text{s.t. } \boldsymbol{\alpha}^\top \mathbf{1} = 1, \boldsymbol{\alpha} \geq 0, \tag{13}$$

where $r_i = \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A}\mathbf{Z}\|_{\mathbf{F}}$. According to Cauchy-Schwarz inequality, the optimal α_i can be directly obtained by

$$\alpha_i = \frac{\frac{1}{r_i}}{\sum_{i=1}^{v} \frac{1}{r_i}}. \tag{14}$$

Algorithm 1 FPMVS-CAG

```
Input: Input $v$ views dataset $\{\mathbf{X}_i\}_{i=1}^{v}$ and the number of cluster $k$.
Initialize: Initialize $\mathbf{A}$, $\mathbf{Z}$, $\mathbf{W}$. Initialize $\alpha_i$ with $\frac{1}{v}$.
while not converged do
    Update $\mathbf{W}_i$ by solving the problem in Eq. (8).
    Update $\mathbf{A}$ by solving the problem in Eq. (10).
    Update $\mathbf{Z}$ by solving the problem in Eq. (12).
    Update $\alpha_i$ by calculating Eq. (14).
end while
Obtain $\mathbf{H}$ by performing SVD on $\mathbf{Z}$.
Output: Perform $k$-means on $\mathbf{H}$ to achieve the final
    clustering result.
```

We summarize our whole procedures of the above optimization are listed in the following Algorithm 1. After obtaining the spectral embedding $\mathbf{H}$, we perform k-means algorithm on $\mathbf{H}$ to get the final clustering result labels.

C. Complexity Analysis

Firstly, we will analyze the time complexity during the total optimization. Then a comparison is conducted between the compared method in terms of main space complexity.

1) Time Complexity: The computational complexity is composed of the cost of optimization of each variable. When updating $\mathbf{W}_i$, it costs $\mathcal{O}(d_i h^2)$ to perform SVD on $\mathbf{B}_i$ and $\mathcal{O}(d_i hk^2)$ to execute matrix multiplication to get the optimal $\mathbf{W}_i$. Similar to updating $\mathbf{W}_i$, updating $\mathbf{A}$ needs $\mathcal{O}(mh^2)$ and $\mathcal{O}(hk^3)$ for SVD and matrix multiplication. When solving the QP problem of updating $\mathbf{Z}$, it costs $\mathcal{O}(nk^3)$ for all columns. The time cost of calculating α_i is only $\mathcal{O}(1)$. Therefore, the total time cost of the optimization process is $\mathcal{O}(hk^2 + hk^3 + nk^3)$. Consequently, the computational complexity of our proposed optimization algorithm is linear complexity $\mathcal{O}(n)$ respecting to the number of samples.

After the optimization, we perform SVD on $\mathbf{Z}$ to obtain its left singular matrix $\mathbf{U}$ and get the final clustering result by k-means. In the post-process, the computational complexity is $\mathcal{O}(nk^2)$, which is also a linear complexity. Consequently, we achieve a linear-time algorithm in both optimization process and post-process. This enables our algorithm to efficiently handle large-scale clustering tasks. In contrast, as shown in Table II, most of the subspace-based multi-view clustering methods hold a $\mathcal{O}(n^3)$ time complexity in the processes mentioned above.

2) Space Complexity: In this paper, the major memory costs of our method are matrices $\mathbf{W}_i \in \mathbb{R}^{d_i \times k}$, $\mathbf{A} \in \mathbb{R}^{k \times k}$ and $\mathbf{Z} \in \mathbb{R}^{k \times n}$. Thus the space complexity of our FPMVS-CAG is $kn + (h + k)k$, where $h = \sum_{i=1}^{v} d_i$. In our algorithm, $k \ll n$. Therefore, the space complexity of FPMVS-CAG is $\mathcal{O}(n)$. We counted the major memory cost of the compared algorithms in the following Table II. It is easy to observe that the space complexity of most state-of-the-art algorithms is $\mathcal{O}(n^2)$, such as MVSC, AMGL, MLRSSC, FMR, etc. LMVSC method also performs $\mathcal{O}(n)$ space complexity, but they have to construct a graph for each view, which will cost more than our consensus graph. The high time and space complexities limit

TABLE II
COMPLEXITY ANALYSIS ON EXISTING MULTI-VIEW CLUSTERING METHODS AND OURS

Method	Memory Cost	Time Complexity	Max Reported
RMKM	$(n+h)k$	$\mathcal{O}(n)$	30,475
MVSC	$2vn^2+nk$	$\mathcal{O}(n^3)$	1,230
AMGL	vn^2+nk	$\mathcal{O}(n^3)$	12,613
MLRSSC	$(v+1)n^2$	$\mathcal{O}(n^3)$	2,000
FMR	n^2+nm	$\mathcal{O}(n^3)$	10,158
PMSC	$2vn^2+(v+1)nk$	$\mathcal{O}(n^3)$	2,386
MLES	vn^2	$\mathcal{O}(n^4)$	544
LMVSC	$vk(n+h)$	$\mathcal{O}(n)$	30,000
Ours	$kn+(h+k)k$	$\mathcal{O}(n)$	101,499

the scalability of many multi-view subspace clustering, making them only applicable to relatively small datasets. We show in Table II the largest dataset reported in the comparison algorithm, which can reflect the efficiency of the algorithm to some extent.

D. Convergence

Theorem 2: The proposed algorithm 1 is proved to be converged.

Proof: We first define the objective function of the proposed algorithm 1 as follows,

$$\mathcal{J}\left(\boldsymbol{\alpha}, \{\mathbf{W}_i\}_{i=1}^m, \mathbf{A}, \mathbf{Z}\right) = \min_{\boldsymbol{\alpha}, \mathbf{W}_i, \mathbf{A}, \mathbf{Z}} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A}\mathbf{Z}\|_{\mathbf{F}}^2. \tag{15}$$

As can be seen form Eq. (15), the whole function is not jointly convex when all variables are considered simultaneously. Instead, we propose an alternate optimization algorithm to optimize each variable with the other three variables been fixed. Let we define $\boldsymbol{\alpha}^{(t)}$, $\left\{\mathbf{W}_i^{(t)}\right\}_{i=1}^m$, $\mathbf{A}^{(t)}$, $\mathbf{Z}^{(t)}$ be the solution at the t-iteration.

i) Optimizing $\{\mathbf{W}_i\}_{i=1}^m$ with fixed $\mathbf{A}, \mathbf{Z}$ and $\boldsymbol{\alpha}$. Given $\boldsymbol{\alpha}^{(t)}, \mathbf{A}^{(t)}, \mathbf{Z}^{(t)}$, the optimization in Eq. (8) respect to $\{\mathbf{W}_i\}_{i=1}^m$ can be analytically obtained. The detailed derivation can be found in the manuscript. Suppose the obtained optimal solution be $\left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m$. We have

$$\mathcal{J}\left(\boldsymbol{\alpha}^t, \left\{\mathbf{W}_i^{(t)}\right\}_{i=1}^m, \mathbf{A}^t, \mathbf{Z}^t\right) \geq \mathcal{J}\left(\boldsymbol{\alpha}^t, \left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m, \mathbf{A}^t, \mathbf{Z}^t\right). \tag{16}$$

ii) Optimizing $\mathbf{A}$ with fixed $\{\mathbf{W}_i\}_{i=1}^m$, $\mathbf{Z}$ and $\boldsymbol{\alpha}$. Given $\boldsymbol{\alpha}^{(t)}$, $\left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m$, $\mathbf{Z}^{(t)}$, the optimization in Eq. (9) respect to $\mathbf{A}$ can be analytically obtained. Suppose the obtained optimal solution be $\mathbf{A}^{(t+1)}$. We have

$$\begin{aligned}&\mathcal{J}\left(\boldsymbol{\alpha}^t, \left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m, \mathbf{A}^t, \mathbf{Z}^t\right)\\ &\qquad \geq \mathcal{J}\left(\boldsymbol{\alpha}^t, \left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m, \mathbf{A}^{(t+1)}, \mathbf{Z}^t\right).\end{aligned} \tag{17}$$

iii) Optimizing $\mathbf{Z}$ with fixed $\mathbf{A}, \{\mathbf{W}_i\}_{i=1}^m$ and $\boldsymbol{\alpha}$. Given $\boldsymbol{\alpha}^{(t)}$, $\left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m$, $\mathbf{A}^{(t)}$, the optimization in Eq. (9) respect to $\mathbf{Z}$ can be optimally solved with n quadprog programs. Suppose the obtained optimal solution be $\mathbf{Z}^{(t+1)}$. We have

$$\begin{aligned}&\mathcal{J}\left(\boldsymbol{\alpha}^t, \left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m, \mathbf{A}^{(t+1)}, \mathbf{Z}^t\right)\\ &\qquad \geq \mathcal{J}\left(\boldsymbol{\alpha}^t, \left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m, \mathbf{A}^{(t+1)}, \mathbf{Z}^{(t+1)}\right).\end{aligned} \tag{18}$$

TABLE III
LARGE-SCALE MULTI-VIEW DATASETS USED IN OUR EXPERIMENTS

Dataset	#Samples	#View	#Class	#Feature
Caltech101-20	2386	6	20	48, 40, 254, 1984, 512, 928
CCV	6773	3	20	20, 20, 20
Caltech101-all	9144	5	102	48, 40, 254, 512, 928
SUNRGBD	10335	2	45	4096, 4096
NUSWIDEOBJ	30000	5	31	65, 226, 145, 74, 129
AwA	30475	6	50	2688, 2000, 252, 2000, 2000, 2000
MNIST	60000	3	10	342, 1024, 64
YoutubeFace_sel	101499	5	31	64, 512, 64, 647, 838

iv) Optimizing $\boldsymbol{\alpha}$ with fixed $\mathbf{A}, \mathbf{Z}$ and $\{\mathbf{W}_i\}_{i=1}^m$. Given $\mathbf{Z}^{(t+1)}$, $\left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m$, $\mathbf{A}^{(t)}$, the optimization in Eq. (14) respect to $\boldsymbol{\alpha}$ can be analytically obtained. The detailed derivation can be found in the manuscript. Suppose the obtained optimal solution be $\boldsymbol{\alpha}^{(t+1)}$. We have

$$\begin{aligned}&\mathcal{J}\left(\boldsymbol{\alpha}^t, \left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m, \mathbf{A}^{(t+1)}, \mathbf{Z}^{(t+1}\right)\\ &\qquad \geq \mathcal{J}\left(\boldsymbol{\alpha}^{(t+1)}, \left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m, \mathbf{A}^{(t+1)}, \mathbf{Z}^{(t+1)}\right).\end{aligned} \tag{19}$$

Together with Eq. (16), (17), (18) and (19), we have that

$$\begin{aligned}&\mathcal{J}\left(\boldsymbol{\alpha}^t, \left\{\mathbf{W}_i^{(t)}\right\}_{i=1}^m, \mathbf{A}^t, \mathbf{Z}^t\right)\\ &\qquad \geq \mathcal{J}\left(\boldsymbol{\alpha}^{(t+1)}, \left\{\mathbf{W}_i^{(t+1)}\right\}_{i=1}^m, \mathbf{A}^{(t+1)}, \mathbf{Z}^{(t+1)}\right).\end{aligned} \tag{20}$$

which indicates that the objective function of our algorithm in Eq. (6) monotonically decreases with the increase of iterations. Also, the objective function in Eq. (6) is lower bounded by zero. As a result, the proposed algorithm can be verified to converge to a local minimum. This completes the proof. □

IV. EXPERIMENTS

In this section, we evaluate the effectiveness and efficiency of the proposed FPMVS-CAG for eight widely used large-scale multi-view datasets from the aspects of clustering performance, computational efficiency and convergence.

A. Benchmark Datasets

The proposed algorithm is experimentally evaluated on eight widely used multi-view benchmark data sets shown in Table III. Caltech101-20 is a widely used subset of the image collection Caltech101 [44] which contains 101 categories. In Caltech101-20, there are 2386 instances in 20 categories. The multiple features of Caltech101-20 are extracted by following previous work [8]. Caltech101-all dataset has 9144 samples and five views. CCV is a rich database of YouTube videos containing 20 semantic categories. SUNRGBD [45] contains 10335 indoor scene images in 45 classes. NUSWIDEOBJ [46] is an object image datasets, composed of 30,000 image over 31 classes. The

TABLE IV

THE CLUSTERING PERFORMANCE (ACC, NMI, PURITY, FSCORE) OF THE COMPARED MULTI-VIEW CLUSTERING METHODS. THE BEST RESULT IS HIGHLIGHTED IN BOLDFACE. UNDERLINES MEAN THE SECOND BEST COMPETITORS AND N/A MEANS THE CORRESPONDING METHOD SUFFERS OUT-OF-MEMORY DUE TO THE SIZE OF THE DATASET

Datasets	MVSC	PMSC	MLES	FMR	mPAC	MLRSSC	AMGL	SFMC	RMKM	BMVC	LMVSC	Ours
Number of hyper-parameters	3	2	3	2	3	2	0	0	1	4	1	0
Caltech101-20	0.5080	0.5981	0.3495	0.3873	0.4983	0.3600	0.1876	0.5947	0.3961	0.1769	0.4304	**0.6547**
CCV	N/A	N/A	N/A	0.1671	0.2311	0.1259	0.1102	0.1156	0.1584	0.1326	0.2014	**0.2399**
Caltech101-all	N/A	N/A	N/A	N/A	0.2031	0.1365	0.0359	0.1777	0.1650	0.2123	0.2005	**0.3015**
SUNRGBD	N/A	N/A	N/A	N/A	0.1906	0.1741	0.0643	0.1113	0.1836	0.1669	0.1858	**0.2392**
NUSWIDEOBJ	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.1221	0.1193	0.1299	0.1583	**0.1946**
AwA	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0390	0.0656	0.0867	0.0770	**0.0919**
MNIST	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.8621	0.4595	0.9852	**0.9884**
YoutubeFace_sel	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0897	0.1479	**0.2414**
						NMI						
Caltech101-20	0.5271	0.5244	0.3158	0.5276	0.5855	0.2008	0.1101	0.5641	0.5034	0.1708	0.5553	**0.6326**
CCV	N/A	N/A	N/A	0.1326	0.1744	0.0471	0.0758	0.0346	0.1136	0.0763	0.1657	**0.1760**
Caltech101-all	N/A	N/A	N/A	N/A	0.3809	0.1066	0.0187	0.2613	0.3494	**0.4246**	0.4155	0.3549
SUNRGBD	N/A	N/A	N/A	N/A	0.1335	0.1108	0.0371	0.0202	**0.2612**	0.1954	0.2607	0.2418
NUSWIDEOBJ	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0095	0.0926	0.1290	0.1337	**0.1351**
AwA	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0032	0.0738	**0.1195**	0.0879	0.1083
MNIST	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.9209	0.3959	0.9576	**0.9651**
YoutubeFace_sel	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0593	0.1327	**0.2433**
						Purity						
Caltech101-20	0.7125	0.6480	0.5268	0.7163	0.6622	0.4476	0.6313	0.7045	0.3150	0.4166	0.7125	**0.7368**
CCV	N/A	N/A	N/A	0.2110	**0.2917**	0.1307	0.2021	0.1194	0.1902	0.1652	0.2396	0.2605
Caltech101-all	N/A	N/A	N/A	N/A	0.2914	0.1371	**0.4311**	0.2430	0.0875	0.4124	0.3975	0.3460
SUNRGBD	N/A	N/A	N/A	N/A	0.1992	0.1741	0.2411	0.1144	0.3771	0.3357	**0.3818**	0.3400
NUSWIDEOBJ	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.1227	0.2062	0.2333	**0.2488**	0.2382
AwA	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0399	0.0849	**0.1094**	0.0957	0.0961
MNIST	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.8988	0.4766	0.9852	**0.9884**
YoutubeFace_sel	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.2662	0.2816	**0.3279**
						Fscore						
Caltech101-20	0.4329	0.5474	0.2972	0.3521	0.4806	0.3069	0.4661	0.4303	0.2799	0.1197	0.3414	**0.6905**
CCV	N/A	N/A	N/A	0.1018	0.1346	0.1095	0.1215	0.1085	0.1084	0.0826	0.1194	**0.1419**
Caltech101-all	N/A	N/A	N/A	N/A	0.1254	0.0815	**0.3617**	0.0462	0.0548	0.1854	0.1586	0.2326
SUNRGBD	N/A	N/A	N/A	N/A	0.1298	0.1453	**0.1894**	0.1215	0.1168	0.1019	0.1201	0.1597
NUSWIDEOBJ	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.1140	0.0750	0.0881	0.0990	**0.1372**
AwA	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0457	0.0359	0.0472	0.0378	**0.0640**
MNIST	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.8728	0.3357	0.9704	**0.9768**
YoutubeFace_sel	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0579	0.0849	**0.1433**

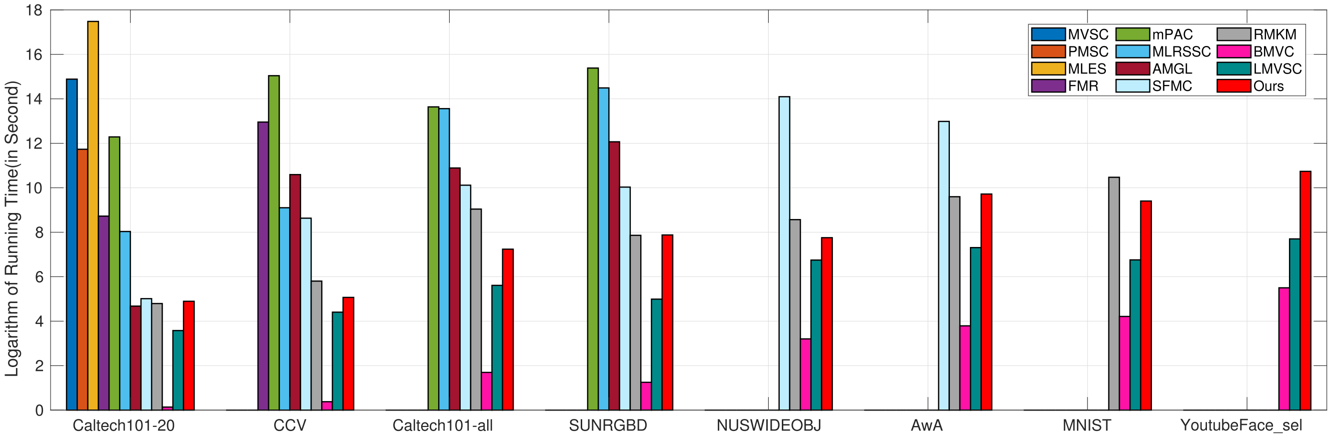
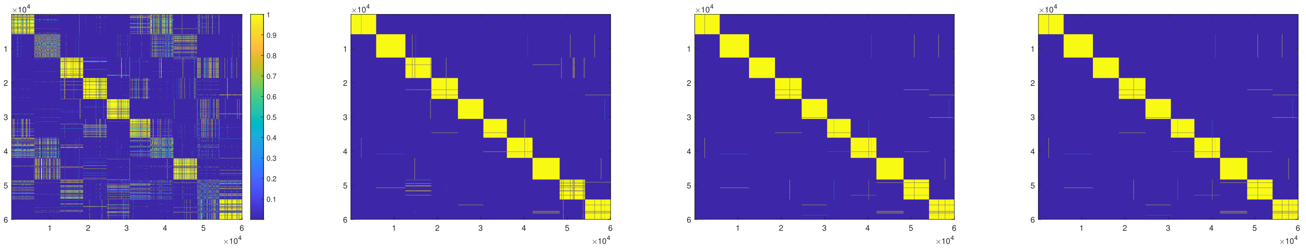


Fig. 2. The relative running time of the compared algorithms on the benchmark datasets. The empty bar means the respective method is out of memory on the dataset.

five views used in this paper are Color Histogram, Color Moments, Color Correlation, Edge Distribution and wavelet texture. The animal dataset with attributes is called AwA. It contains 50 kinds of animals depicted in six features. YouTubeFace_sel is a face video dataset obtained from YouTube.



(a) 1-st iteration (b) 3-rd iteration (c) 5-th iteration (d) 10-th iteration

Fig. 3. An illustration of the complete similarity matrices **S** on MNIST dataset.

B. Compared Multi-View Clustering Algorithms and Experimental Setting

The following state-of-the-art multi-view clustering methods are compared with our proposed algorithm in the experiment.

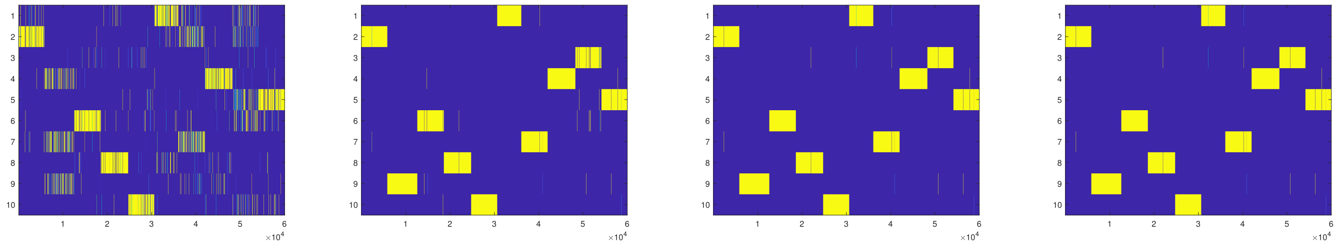
- **Multi-view Subspace Clustering (MVSC) [36]**. In this paper, an efficient multi-view subspace clustering method is proposed and the effectiveness of the algorithm is verified.
- **Partition Level Multiview Subspace Clustering (PMSC) [40]**. This work proposes a unified multi-view subspace clustering model and verifies the effectiveness of the algorithm.
- **Multi-view Clustering in Latent Embedding Space (MLES) [47]**. The algorithm can simultaneously learn the global structure and the clustering indicator matrix and then cluster multi-view data in the potential embedding space.
- **Flexible Multi-View Representation Learning for Subspace Clustering (FMR) [48]**. This work flexibly encodes complementary information from different views, thus avoiding the use of partial information for data reconstruction.
- **Multiple Partitions Aligned Clustering (mPAC)** [49]. This algorithm jointly learns basic partitions, weights and consensus clustering in a unified framework.
- **Multi-view Low-rank Sparse Subspace Clustering (MLRSSC) [39]**. This work learns subspace representations by constructing affinity matrices shared among all views and solves the associated low-rank and sparse constrained optimization problems.
- **Parameter-free Auto-weighted Multiple Graph Learning (AMGL) [32]**. This work proposes a framework that automatically learns the optimal weights for each graph and obtains globally optimal results.
- **Multi-view clustering: a Scalable and Parameter-free Bipartite Graph Fusion Method (SFMC) [12]**. SFMC provides an initialization-independent anchor selection strategy to fulfill scalable multi-view graph clustering algorithm.
- **Multi-view *k*-means Clustering on Big Data (RMKM) [50]**. This work is a robust large-scale multi-view clustering method that integrates heterogeneous representations of large-scale data.
- **Binary Multi-view Clustering BMVC [51]**. This paper proposes to concurrently learn the multi-view binary representation and consensus clustering result to reduce computational complexity and storage cost.
- **Large-scale Multi-view Subspace Clustering in linear time (LMVSC) [24]**. The algorithm is designed to handle large-scale data and has linear complexity.

For a fair comparison, we download the released code of comparison algorithms from their original websites. Since all methods need to utilize k-means to get the final clustering results, we run 50 times k-means to eliminate the randomness in k-means initialization for all compared methods. Then the clustering performance is evaluated by the widely used metrics accuracy (ACC), normalized mutual information (NMI), purity, and Fscore. By the way, the experimental environment is implemented on a desktop computer with an Intel Core i7-7820X CPU and 64GB RAM, MATLAB 2020b (64-bit).

C. Experiments Results

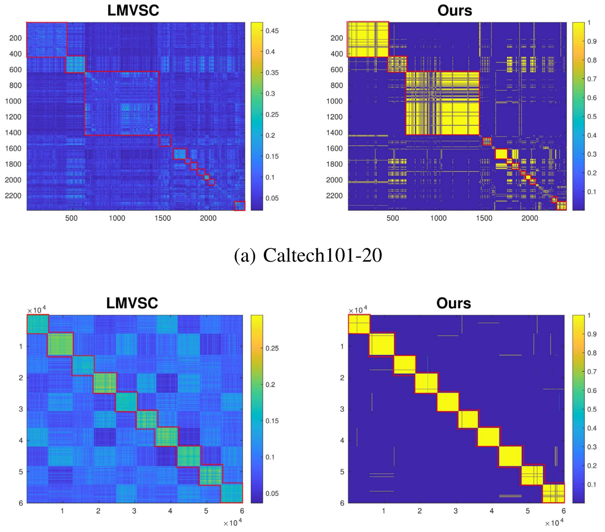
Table IV presents the clustering performance of ours and eleven compared algorithms on eight benchmark datasets. The best results are marked in bold and underlining indicates the second best. Moreover, 'N/A' means that the method encounters an out-of-memory error on the corresponding dataset on our device. Based on the results, we have the following observations:

- FPMVS-CAG shows clear advantages over other multi-view clustering baselines. Especially in term of ACC, our proposed algorithm outperforms than all of the compared methods. Our method improves 5.66%, 0.89%, 8.92%, 4.86%, 3.63%, 0.52%, 0.32%, 9.35% over the second best method on eight datasets, respectively. Moreover, FPMVS-CAG also achieves comparable performance on the other metrics.
- Compared with the traditional subspace-based multi-view clustering algorithms (MVSC, PMSC, MLES, FMR, mPAC, MLRSSC), the anchor subspace methods (LMVSC and ours) are more suitable for large-scale dataset and could achieve the best performance in most cases, which illustrates the effectiveness of anchor graph.
- Compared with some methods that use bipartite graphs, k-means, or binarization for large-scale multi-view clustering (SFMC, RMKM, BMVC), our FPMVS-CAG also shows better performance. As for LMVSC, which is also



(a) 1-st iteration (b) 3-rd iteration (c) 5-th iteration (d) 10-th iteration

Fig. 4. An illustration of the learned **Z** on MNIST dataset in different iterations.



(a) Caltech101-20

(b) MNIST

Fig. 5. The visualization of the complete graphs between LMVSC and ours on dataset Caltech101-20 and MNIST.

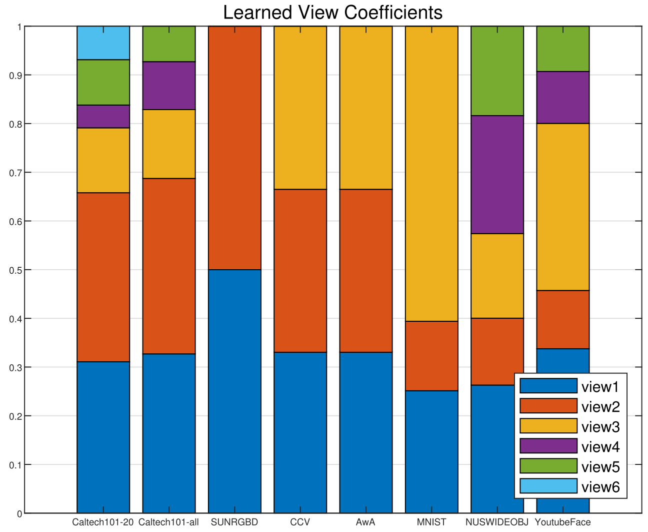


Fig. 6. The learned view coefficients on eight benchmark datasets.

an anchor graph based algorithm, our method outperforms it by 22.43%, 3.85%, 10.10%, 5.34%, 3.63%, 1.49%, 0.32%, 9.35% on the eight datasets, respectively. This proves the effectiveness of our strategy of learning anchor from multiple views with the guidance of consensus anchor graph.

In summary, the above experimental results have well demonstrated the effectiveness of our proposed FPMVS-CAG comparing to other state-of-the-art methods. We attribute the superiority of proposed algorithm as two aspects: i) The anchor selection and latter subspace graph construction are jointly into a unified optimization. Then the two processes can be negotiated with each other to promote clustering quality. ii) To solve the resultant optimization problem, we design a four-step alternate optimization algorithm with proved convergence. By the virtue of property, FPMVS-CAG is proved to only have the linear time complexity respecting to the sample number. More specially, FPMVS-CAG can automatically learn an optimal low-rank anchor subspace graph without additional hyper-parameters as previous methods do. The two factors contribute to FPMVS-CAG more suitable for large-scale subspace clustering.

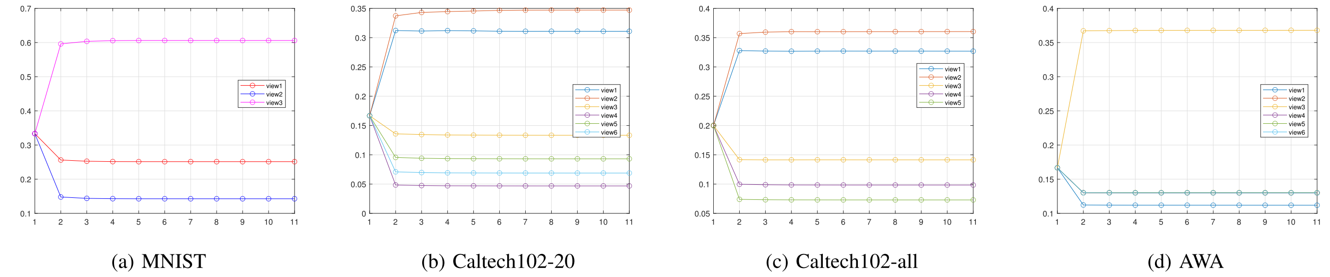
D. Running Time Comparison

To compare the computational efficiency of the proposed algorithms, we record the running time of various algorithms on these benchmark datasets and report them in Figure 2. As can be seen, FPMVS-CAG has much shorter running time on tested datasets comparing to the-state-of-art multi-view methods (MVSC, PMSC, FMR, mPAC, MLRSSC), demonstrating the computational efficiency of the proposed method. As theoretically demonstrated, our algorithm's time complexity is proved to be linear to n. Although BMVC and LMVSC have less running time, their heuristic and simple procedures do not sufficiently utilize multi-view information, leading to much poorer clustering performance.

In sum, both the theoretical and the experimental results have well demonstrated the computational advantage of proposed algorithm, making FPMVS-CAG efficient to handle with multi-view clustering. FPMVS-CAG enjoys comparable complexity with existing large-scale multi-view clustering methods. Besides, the clustering performance has been significantly improved due to utilizing multi-view information and determining consensus anchors.

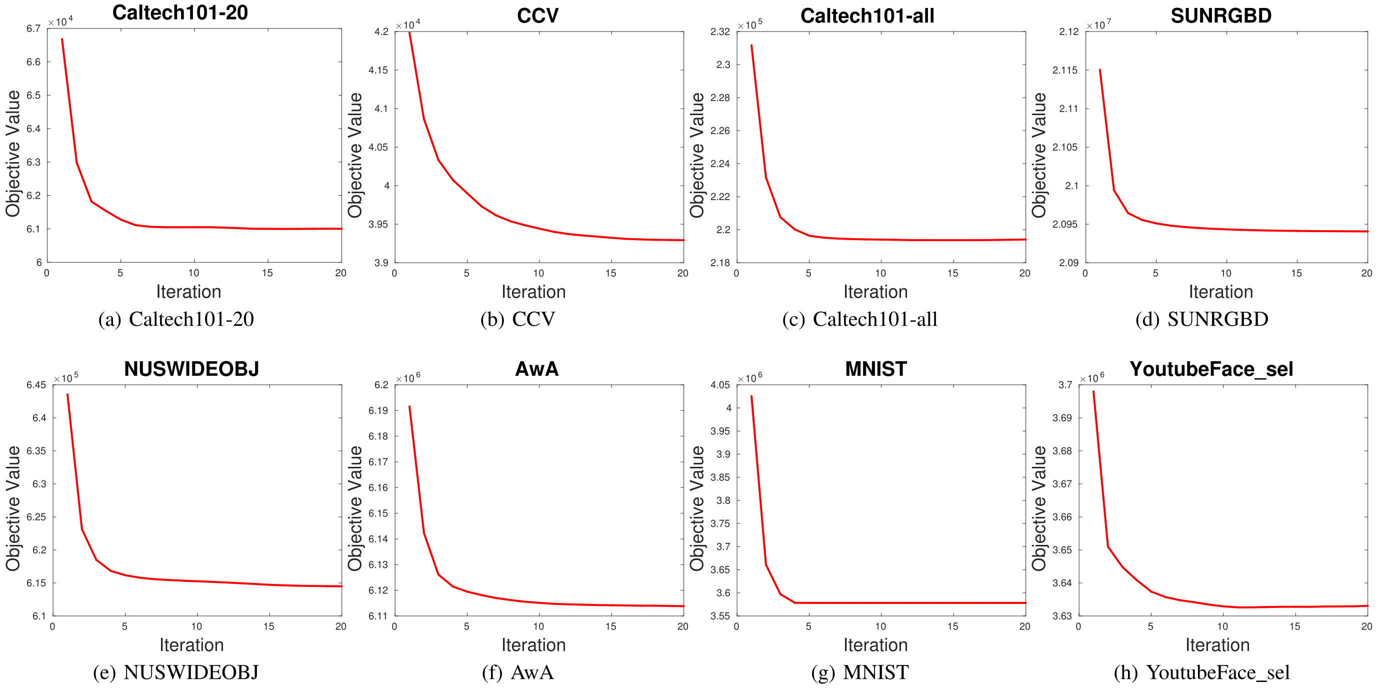
E. Handling With Large-Scale Datasets More Than Large-Scale Samples

To further demonstrate the effectiveness and efficiency of our proposed method, we evaluate the clustering performance when facing with large-scale datasets. Specially, we compare our algorithm with the widely-used large-scale



(a) MNIST (b) Caltech102-20 (c) Caltech102-all (d) AWA

Fig. 7. An illustration of the learned view coefficients on benchmark datasets.



(a) Caltech101-20 (b) CCV (c) Caltech101-all (d) SUNRGBD

(e) NUSWIDEOBJ (f) AwA (g) MNIST (h) YoutubeFace_sel

Fig. 8. The objective of our proposed method on eight benchmark datasets.

multi-view clustering algorithms on datasets with more 30000 samples (including NUSWIDEIBJ, AWA, MNIST and YoutubeFace_sel). The experimental results are shown in Table V.

It can be observed from our proposed FPMVS-CAG still maintains excellent clustering performance on these large datasets especially on the largest dataset YouTubeFace (more than 100000 samples). These have well demonstrated the superiority of our method in terms of scalablity and clustering performance.

TABLE V

COMPARISON RESULTS FOR LARGE-SCALE ORIENTED ALGORITHMS. N/A MEANS THE CORRESPONDING METHOD SUFFERS OUT-OF-MEMORY DUE TO THE SIZE OF THE DATASET

Dataset	Metric	SFMC	RMKM	BMVC	LMVSC	Ours
NUSWIDEOBJ	ACC	0.1221	0.1193	0.1299	0.1583	**0.1946**
	NMI	0.0095	0.0926	0.1290	0.1337	**0.1351**
	Purity	0.1227	0.2062	0.2333	**0.2488**	0.2382
	Fscore	0.1140	0.0750	0.0881	0.0990	**0.1372**
AwA	ACC	0.0390	0.0656	0.0867	0.0770	**0.0919**
	NMI	0.0032	0.0738	**0.1195**	0.0879	0.1083
	Purity	0.0399	0.0849	**0.1094**	0.0957	0.0961
	Fscore	0.0457	0.0359	0.0472	0.0378	**0.0640**
MNIST	ACC	N/A	0.8621	0.4595	0.9852	**0.9884**
	NMI	N/A	0.9209	0.3959	0.9576	**0.9651**
	Purity	N/A	0.8988	0.4766	0.9852	**0.9884**
	Fscore	N/A	0.8728	0.3357	0.9704	**0.9768**
YoutubeFace_sel	ACC	N/A	N/A	0.0897	0.1479	**0.2414**
	NMI	N/A	N/A	0.0593	0.1327	**0.2433**
	Purity	N/A	N/A	0.2662	0.2816	**0.3279**
	Fscore	N/A	N/A	0.0579	0.0849	**0.1433**

F. Qualified Study

To further illustrate the effectiveness of the obtained anchor graph $\mathbf{Z}$, we have also plotted the learned $\mathbf{Z}$ in Figure 4. From the Figure. 4, we observe that after iterations, the learned anchor graph $\mathbf{Z}$ shows high confident belongs to the anchor point. while the 1-st iteration shows scatters for 1 sample to several anchor points, the 10-th iteration shows clear belongings with sample to the respective anchor point. Further, we have plotted the learned affinity matrix at different iterations on MNIST dataset in Figure 3. As can be seen in Figure 3, the clustering block structure becomes more clear and the noises have been eliminated. Comparison the affinity graph obtained by LMVSC and ours can also be found in Fig. 5. These results clearly demonstrate the effectiveness of our proposed algorithm for large-scale clustering.

G. The Learned View Coefficients

In this subsection, we conduct experiments on the learned and the evolution of view coefficients. As mentioned in the former section, our method adaptively learns the view coefficients respecting to their contributions. We plot the learned coefficients in testes benchmark datasets in Fig 6 and Figure 7. From the figure, we can conclude that our proposed method jointly optimizes the view coefficients to connect with comprehensive information while previous methods [12], [24] equally treat each view. The self-adaptive weights seem more practical and reasonable in application.

H. Convergence of the Proposed Algorithm

According to Theorem 2 and [52], our proposed algorithm is theoretically guaranteed to converge to a local minimum. Moreover, we also conduct experiments on benchmark datasets to demonstrate the convergence of the proposed algorithm. The examples of the evolution of the objective value on the experimental results are shown in Figure 8. In the above experiments, we observe that the objective values of our algorithm monotonically decrease at each iteration and they can quickly converge in less than 10 iterations. These results clearly verify our proposed algorithm's convergence.

V. Conclusion

In this paper, we propose a novel method termed as Fast Parameter-free Multi-view Subspace Clustering with Consensus Anchor Guidance (FPMVS-CAG). Different from existing large-scale work, we firstly jointly conduct anchor selection and latter subspace graph construction into a unified optimization. Then the two processes can be negotiated with each other to promote clustering quality. To solve the resultant optimization problem, we design a four-step alternate optimization algorithm with proved convergence. By the virtue of property, FPMVS-CAG is proved to only have the linear time complexity respecting to the sample number. More specially, FPMVS-CAG can automatically learn an optimal low-rank anchor subspace graph without additional hyper-parameters as previous methods do. The two factors contribute to FPMVS-CAG more suitable for large-scale subspace clustering. In the future, we will explore the influence of various anchor selection strategy on the clustering quality.

References

[1] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, and X. Huang, "Robust subspace clustering for multi-view data by exploiting correlation consensus," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3939–3949, Nov. 2015.

[2] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[3] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1, pp. 2408–2414.

[4] Z. Zhang, F. Li, M. Zhao, L. Zhang, and S. Yan, "Robust neighborhood preserving projection by nuclear/L2,1-norm regularization for image feature extraction," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1607–1622, Apr. 2017.

[5] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "COMIC: Multi-view clustering without parameter selection," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5092–5101.

[6] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2410–2423, Oct. 2019.

[7] Q. Gao, W. Xia, Z. Wan, D. Xie, and P. Zhang, "Tensor-SVD based graph learning for multi-view subspace clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 3930–3937.

[8] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. 2nd AAAI Conf. Artif. Intell.*, 2015, pp. 2750–2756.

[9] C. Zhang *et al.*, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.

[10] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2149–2155.

[11] W. Liang *et al.*, "Multi-view spectral clustering with high-order optimal neighborhood Laplacian matrix," *IEEE Trans. Knowl. Data Eng.*, early access, Sep. 18, 2020, doi: 10.1109/TKDE.2020.3025100.

[12] X. Li, H. Zhang, R. Wang, and F. Nie, "Multi-view clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 22, 2020, doi: 10.1109/TPAMI.2020.3011148.

[13] M. Sun *et al.*, "Projective multiple kernel subspace clustering," *IEEE Trans. Multimedia*, early access, Jun. 4, 2021, doi: 10.1109/TMM.2021.3086727.

[14] Q. Ou, S. Wang, S. Zhou, M. Li, X. Guo, and E. Zhu, "Anchor-based multiview subspace clustering with diversity regularization," *IEEE MultimediaMag.*, vol. 27, no. 4, pp. 91–101, Oct. 2020.

[15] 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. (ICCV)*, Dec. 2015, pp. 1582–1590.

[16] X. Wang, X. Guo, Z. Lei, C. Zhang, and S. Z. Li, "Exclusivity-consistency regularized multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 923–931.

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

[18] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for image clustering and semi-supervised classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1501–1511, Mar. 2018.

[19] 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.

[20] C. Zhang *et al.*, "Multi-view clustering via deep matrix factorization and partition alignment," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 4156–4164.

[21] X. Yao, J. Han, D. Zhang, and F. Nie, "Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3196–3209, Jul. 2017.

[22] J. Xu, J. Han, F. Nie, and X. Li, "Re-weighted discriminatively embedded K-means for multi-view clustering," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 3016–3027, Jun. 2017.

[23] 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.

[24] 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.

[25] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

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

[27] X. Liu *et al.*, "One pass late fusion multi-view clustering," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 6850–6859.

[28] X. Liu *et al.*, "Absent multiple kernel learning algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1303–1316, Jun. 2020.

[29] X. Liu *et al.*, "Efficient and effective regularized incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2634–2646, Aug. 2021.

[30] 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.

[31] P. Zhang *et al.*, "Consensus one-step multi-view subspace clustering," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 18, 2020, doi: 10.1109/TKDE.2020.3045770.

[32] 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.

[33] X. Wang, Z. Lei, X. Guo, C. Zhang, H. Shi, and S. Z. Li, "Multi-view subspace clustering with intactness-aware similarity," *Pattern Recognit.*, vol. 88, pp. 50–63, Apr. 2019.

[34] Y. Wang and L. Wu, "Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering," *Neural Netw.*, vol. 103, pp. 1–8, Jul. 2018.

[35] H. Wang, F. Nie, and H. Huang, "Multi-view clustering and feature learning via structured sparsity," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 352–360.

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

[37] 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.

[38] S. Luo, C. Zhang, W. Zhang, and X. Cao, "Consistent and specific multi-view subspace clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 3731–3737.

[39] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern Recognit.*, vol. 73, pp. 247–258, Jan. 2018.

[40] Z. Kang *et al.*, "Partition level multiview subspace clustering," *Neural Netw.*, vol. 122, pp. 279–288, Feb. 2020.

[41] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, "A unified framework for representation-based subspace clustering of out-of-sample and large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2499–2512, Dec. 2016.

[42] X. Liu *et al.*, "Multiple kernel KK-means with incomplete kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1191–1204, May 2020.

[43] S. Wang *et al.*, "Multi-view clustering via late fusion alignment maximization," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3778–3784.

[44] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2004, p. 178.

[45] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 567–576.

[46] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "NUS-WIDE: A real-world web image database from National University of Singapore," in *Proc. ACM Int. Conf. Image Video Retr. (CIVR)*, Santorini, Greece, Jul. 2009, pp. 1–9.

[47] M.-S. Chen, L. Huang, C.-D. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 3513–3520.

[48] R. Li, C. Zhang, Q. Hu, P. Zhu, and Z. Wang, "Flexible multi-view representation learning for subspace clustering," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2916–2922.

[49] Z. Kang *et al.*, "Multiple partitions aligned clustering," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2701–2707.

[50] 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. 2598–2604.

[51] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2018.

[52] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural, Parallel Sci. Comput.*, vol. 11, no. 4, pp. 351–368, 2003.

Siwei Wang is currently pursuing the Ph.D. degree with the National University of Defense Technology (NUDT), China. He has published several papers and served as a PC Member/Reviewer in top journals and conferences, such as IEEE Transactions on Knowledge and Data Engineering (TKDE), IEEE Transactions on Neural Networks and Learning Systems (TNNLS), IEEE Transactions on Image Processing (TIP), IEEE Transactions on Cybernetics (TCYB), IEEE Transactions on Multimedia (TMM), ICML, CVPR, ECCV, ICCV, AAAI, and IJCAI. His current research interests include kernel learning, unsupervised multiple-view learning, scalable clustering, and deep unsupervised learning.

Xinwang Liu (Senior Member, IEEE) received the Ph.D. degree from the National University of Defense Technology (NUDT), China. He is currently a Professor with the School of Computer, NUDT. He has published more than 60 peer-reviewed papers, including those in highly regarded journals and conferences, such as IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on Knowledge and Data Engineering (TKDE), IEEE Transactions on Image Processing (TIP), IEEE Transactions on Neural Networks and Learning Systems (TNNLS), IEEE Transactions on Multimedia (TMM), IEEE Transactions on Information Forensics and Security (TIFS), ICML, NeurIPS, ICCV, CVPR, AAAI, and IJCAI. His current research interests include kernel learning and unsupervised feature learning. He serves as an Associated Editor for *Information Fusion* journal. More information can be found at https://xinwangliu.github.io/

Xinzhong Zhu (Member, IEEE) received the Ph.D. degree from Xidian University, China. He is currently a Professor with the College of Mathematics and Computer Science, Zhejiang Normal University, and the President of Research Institute of Ningbo Cixing Company Ltd., China. He has published more than 30 peer-reviewed papers, including those in highly regarded journals and conferences, such as IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on Multimedia (TMM), IEEE Transactions on Knowledge and Data Engineering (TKDE), AAAI, and IJCAI. His research interests include machine learning, computer vision, manufacturing informatization, robotics and system integration, and intelligent manufacturing. He is a member of ACM and certified as a CCF Senior Member.

Pei Zhang received the bachelor's degree from Yunnan University in 2018 and the master's degree in computer science from the National University of Defense Technology (NUDT) in 2020, where she is currently pursuing the Ph.D. degree. Her current research interests include multi-view learning, incomplete multi-view clustering, and deep clustering.

Yi Zhang is currently pursuing the Ph.D. degree with the National University of Defense Technology (NUDT), China. He has published several peer-reviewed papers in journals and conferences, such as *ACM TOMM* and ICML. His current research interests include kernel learning and unsupervised multiple-view learning.

Feng Gao (Member, IEEE) received the B.S. degree in computer science from University College London in 2007 and the Ph.D. degree in computer science from Peking University in 2018. He was a Postdoctoral Research Fellow at the Future Laboratory, Tsinghua University, from 2018 to 2020. He has been an Assistant Professor at Peking University since 2020. His research interests include the intersection of computer science and art, including but not limit on artificial intelligence and painting art, deep learning, and painting robot.

En Zhu received the Ph.D. degree from the National University of Defense Technology (NUDT), China. He is currently a Professor with the School of Computer Science, NUDT. He has published more than 60 peer-reviewed papers, including IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), *PR*, AAAI, and IJCAI. His main research interests include pattern recognition, image processing, machine vision, and machine learning. He was awarded the China National Excellence Doctoral Dissertation.