How to Construct Corresponding Anchors for Incomplete Multiview Clustering

Shengju Yu, Siwei Wang, Yi Wen, Ziming Wang, Zhigang Luo, En Zhu,
Xinwang Liu, *Senior Member, IEEE*

Abstract—Anchor based incomplete multiview clustering has grasped growing interest recently because of its great success in effectively partitioning multimodal data. However, due to the absence of label information, the constructed anchors could be mismatched. Such an Anchor Mismatching Problem (AMP) will cause the structure of generated bipartite graph to be chaotic, degrading the clustering performance. To tackle this issue, we design an algorithm termed Constructing Corresponding Anchors for Incomplete Multiview Clustering (CCA-IMC). Specifically, we first devise a permutation strategy to transform anchors on each view. Subsequently, we directly generate the consensus bipartite graph, which is shared for all incomplete views, by the transformed anchors rather than by fusing each view-specific bipartite graph. Afterwards, all anchors and permutation matrices as well as the consensus bipartite graph are jointly optimized in one common framework so as to promote each other. In such ways, anchors are rearranged towards correct matching relationship according to the consensus graph structure. In addition to these, our CCA-IMC has also been proven to be with linear time and memory overheads, which makes it able to scale up to work with large-scale tasks. Massive experiments implemented on ten popular datasets give evidence of our superiorities compared to current strong IMC competitors.

Index Terms—Incomplete multiview clustering, Corresponding anchors, Bipartite graph, Multiview learning.

I. INTRODUCTION

AS a representative of unsupervised learning, multiview clustering technology can automatically divide multiview data, which is typically collected from different views or modalities, into distinct sets and thereby discover the intrinsic structure between samples [1]–[3]. Compared to single-view clustering, multiview clustering achieves more encouraging results by taking advantage of complementary information between different views, and is widely applied in video processing [4]–[6], machine learning [7]–[9], intelligence applications [10]–[12], and so on. During the past few years, many prominent algorithms have been designed, such as spectral clustering [13]–[15], kernel clustering [16]–[18], hierarchical clustering [19]–[21]. Although these methods make obvious performance improvement from various aspects, all of them require each view to be complete [22]–[26]. In several real-world applications, however, this setting is unavoidably broken due to sensor breakdown or environment variation, leading to current multiview clustering algorithms failing to work. As a consequence, how to effectively group incomplete multiview data without any label information attracts increasing attention, which is generally referred to as incomplete multiview clustering (IMC) in literatures.

To alleviate IMC problem, a series of algorithms have been elaborately devised [27]–[32]. For example, Shao *et al.* [33] adopt weighted non-negative matrix factorization (NMF) and $\ell_{1,2}$ norm to derive the unified potential features of missing samples. Fang *et al.* [34] try to learn the unique and consistent information of incomplete views by subspace decomposition. Wen *et al.* [35] construct the locality-maintained reconstruction regularizer to infer partial views and utilize the reverse graph regularizer to align inferred views and partial views. Zhao *et al.* [36] couple incomplete views by a graph Laplacian term instead of projecting samples into a common space to preserve the global structure between data. Hu *et al.* [37] decrease the impact of incomplete instances by building a unified basis matrix and associate each view with a weight matrix so as to handle more than two views. Despite achieving impressive clustering results, these methods are generally with expensive computational and memory expenditures, which limits their further application on large-scale tasks. For increasing the efficiency, Wang *et al.* [38] introduce anchor technology into IMC for the first practice, and successfully generate anchor-instance similarities instead of pair-wise instance-instance similarities. Accordingly, the complexities are reduced to be linear w.r.t the number of samples.

Although anchor based IMC methods produce pleasing results under economical expenditures, the constructed anchors could be mismatched, which will to some extent hinder the clustering performance. Concretely, as shown in Fig. 1, since there is no any label information, the cluster order on each view could be different. For example, in view 1, the cluster order is $[1, 2, 3, 4]$ while in view 2 it is $[2, 4, 1, 3]$. Considering that the principle of anchor technology is to approximately represent the samples in clusters by using fewer anchors, we have that anchor 1 of view 1 will be mismatched to anchor 2 of view 2. Accordingly, the generated view-specific

S. Yu, Y. Wen, Z. Luo, E. Zhu and X. Liu are with the School of Computer, National University of Defense Technology, Changsha, 410073, China (e-mail: yu-shengju@foxmail.com, {wenyi21, zgluo, enzhu, xinwangliu}@nudt.edu.cn).
S. Wang is with Intelligent Game and Decision Lab, Beijing, P. R. China. (e-mail: wangsiwei13@nudt.edu.cn).
Z. Wang is with China Academy of Aerospace Science and Innovation, Beijing, P. R. China. (e-mail: wangziming@lsec.cc.ac.cn).

Fig. 1. The framework of current anchor-based incomplete multiview clustering. It first constructs anchors on respective view, and then generates the view-specific bipartite graph. Subsequently, it fuses all generated view-specific bipartite graphs to produce the consensus bipartite graph. After these three stages, the final discrete grouping results are obtained by performing spectral algorithm on the consensus bipartite graph. Since data does not have any label information, the cluster order on each view could be different, which will cause the constructed anchors to be mismatched. Mismatched anchors will further lead to the subsequent view-specific bipartite graphs and consensus bipartite graph being with chaotic structure, harming the clustering performance.

bipartite graphs are misaligned. For instance, the red first row of view 1 bipartite graph is misaligned to the purple first row of view 2 bipartite graph while it should be aligned to the red third row of view 2 bipartite graph. We also show the anchor mismatching phenomenon on dataset Handwritten in Fig. 2. Samples $400 \sim 600$ in view 2 correspond to anchor 3 while they in view 1 correspond to anchor 8. Such an Anchor Mismatching Problem (AMP) will lead to the structure of generated (view-specific and consensus) bipartite graph(s) being chaotic, impacting the clustering results. For this reason, how to construct corresponding anchors for IMC becomes an urgent problem that needs to be addressed.

One intuitive way is to calculate the distance between anchors on different views, and then rearrange anchors according to their similarity. Unfortunately, multiview data is typically with diverse dimensions, and naturally the anchors constructed on different views are also with different dimensions. It is hard to directly measure the distance for anchor sets under various dimension spaces. One solution is to project anchors into an unified space so that they share a common dimension. Nevertheless, the projection operation will bring about serious information loss. Besides, it also needs to employ extra efforts to tune the dimension hyper-parameter, which will limit the algorithm's scalability. For effectively alleviating AMP, in this paper we devise an algorithm named Constructing Corresponding Anchors for Incomplete Multiview Clustering (CCA-IMC), and the overall framework is demonstrated in Fig. 3. To be specific, we first introduce a permutation matrix for each view so as to transform anchors in their original dimension space. Then, we choose to skip the stage of generating view-specific bipartite graphs, and directly generate the consensus bipartite graph based on the transformed anchors. One advantage is that it can save the time and memory overheads caused by view-specific bipartite graphs. Another is that it can eliminate the impact of inappropriate fusion strategies on the quality of consensus bipartite graph. More importantly, the consensus structure can be regarded as a bridge that directly links anchors on different views, forcing them to rearrange towards the corresponding relationship. Additionally, existing algorithms usually adopt a separated three-stage mechanism,
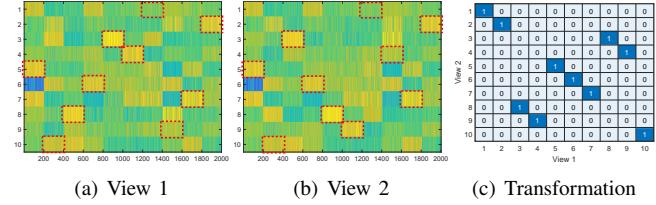

Fig. 2. The anchor mismatching phenomenon on dataset Handwritten. Anchors 3, 4, 8, 9 in view 2 are mismatched to anchors 8, 9, 3, 4 in view 1 respectively. (c) is the desired transformation relationship. After transforming (a) with (c), anchors on (a) will be correctly matched to these on (b).

i.e., firstly utilizing heuristic strategies to construct anchors, then generating view-specific bipartite graphs based on the constructed anchors, and subsequently fusing all view-specific bipartite graphs to generate the consensus bipartite graph. The separation of three stages could be not good for the clustering results since they can not communicate with each other. Different from them, we adopt a joint-optimization mechanism so that all anchors and permutation matrices as well as the consensus bipartite graph boost mutually. By such ways, anchors will be rearranged iteratively towards the correct matching relationship. Besides these, CCA-IMC is also proven to be with linear time and space complexities, and thus is suitable for coping with large-scale tasks. Substantial experiments conducted on ten popularly-used datasets demonstrate its superiorities against current SOTA methods. In short, our contributions are as follows:

1) We devise a CCA-IMC algorithm for incomplete multiview clustering to alleviate the Anchor Mismatching Problem (AMP). By introducing the permutation strategy and directly generating the consensus bipartite graph, CCA-IMC rearranges different anchor sets towards the corresponding relationship.

2) Unlike previous methods that utilize the separated three-stage mechanism, CCA-IMC adopts a joint-optimization mechanism, which makes anchors, permutation matrices and consensus bipartite graph able to boost each other.

3) CCA-IMC enjoys the time and space complexities linear to the number of samples, and is suitable for handling large-scale IMC tasks.
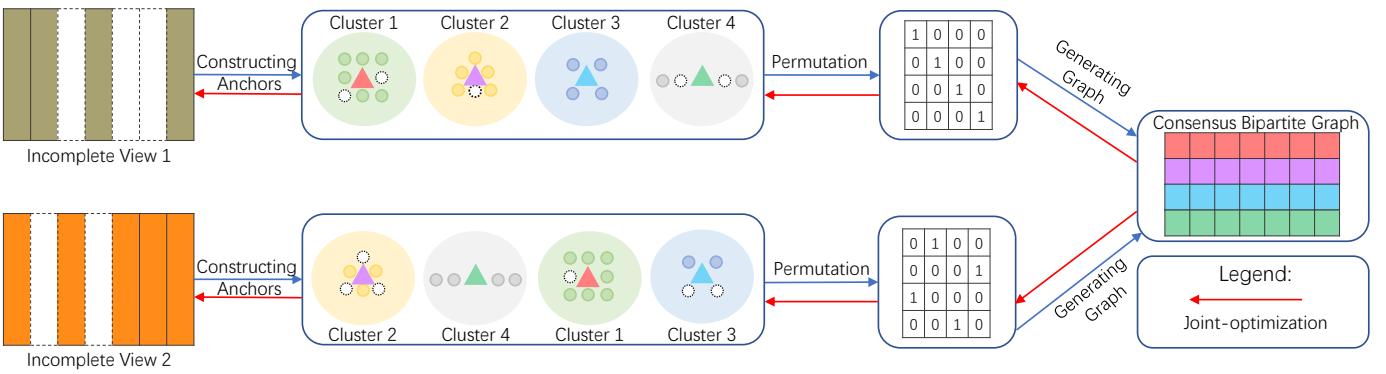
Fig. 3. The framework of proposed CCA-IMC algorithm. Each view is associated with a permutation matrix so as to transform anchors in their original dimension space. It skips the stage of generating view-specific bipartite graphs, and directly generates the consensus bipartite graph. In addition to saving the time and space overheads and getting rid of the negative impact caused by inappropriate fusion strategies, most importantly, the consensus graph structure plays a role in bridging different anchor sets, forcing anchors to rearrange towards the correct matching relationship. Different from existing algorithms that adopt a separated three-stage mechanism, it jointly optimizes all anchors and permutation matrices as well as the consensus bipartite graph in one shared framework such that they can promote mutually.

II. RELATED WORK

A. IMC

To effectively tackle IMC problem, in recent years extensive prominent algorithms have been designed. Existing IMC methods can be roughly divided into four kinds: NMF methods [33], [36], [39]–[41], kernel methods [42]–[46], graph methods [1], [47]–[53] and neural network methods [54]–[58]. NMF based IMC methods seek to generate the consensus latent partition space across incomplete views. For example, Zhao *et al.* [36] transform original samples to a complete representation in potential space, and then design a graph Laplacian regularizer to preserve compact global structure among all views. Wen *et al.* [40] alleviate the factorization errors by constructing the nearest neighbor graph, and employ orthogonal constraint on basic matrix to capture representation more discriminatively. Wen *et al.* [41] unify graph construction and representation learning together, and minimize the disagreement between representations by the co-regularization constraint. Kernel based IMC methods are devoted to producing the ideal similarity measure for all incomplete views. For instance, Li *et al.* [42] increase the selection probability of kernels with high diversity by building alignment criterion, and generate high-confidence similarity matrix by eliminating the unreliable evaluation between farther samples. Liu *et al.* [45] alternatively learn clustering matrix and impute based matrix in a shared framework, and generate consensus similarity by aligning them using an optimal permutation matrix. Zhang *et al.* [46] integrate partition matrix imputation and label generation together, and make consensus similarity and clustering task boost each other. Graph based IMC methods try to restore the affinity relationship between samples and construct a fused graph for all views. Cui *et al.* [59] adaptively explore the global structure by low-rank constraint, and unify related information between inter-views and intra-views to restore complete graph structure and affinity. Wen *et al.* [49] introduce tensor constraint to exploit the high-level correlations among views, and combine feature space and manifold space to capture hidden information within

incomplete views. Liang *et al.* [50] learn sample-level weight instead of directly weighting views to consider the affection of partial instances, and construct the fused graph by joint graph fusion scheme. Rather than learning the coefficient matrix for selected features, Tang *et al.* [51] choose to introduce the weighting matrix to directly learn the feature dimensions' weights, and map the captured features into a label space so as to regularize the underlying structure among multi-view data. Due to the original feature space unavoidably including the redundancy representation and noise, Li *et al.* [52] generate the affinity graph in the spectral embedding space so that the learned graph is sufficient to characterize the cluster structure. Tang *et al.* [53] skip the subsequent k-means procedure and directly output the discrete results via the one-step clustering strategy to alleviate the information loss caused by the separated two-stages strategy. Neural network based IMC methods intend to exploit high-level characteristics from incomplete samples by ingenious network architecture. Lin *et al.* [54] devise an innovative objective consisting of data recovery and consistency learning, and maximize the mutual information to capture informative characteristics. Xu *et al.* [55] design a network without requiring imputation and fusion, and discover linear separability by constructing embedding feature learning for each view. Zhang *et al.* [56] learn the consensus potential representation by jointly considering structure and completeness, and apply a clustering-like classification loss to increase the separability of representation.

B. Graph learning for IMC

Given incomplete multiview data $\mathbf{X}_q \in \mathbb{R}^{d_q \times n}$ and indicator vector $b_q \in \mathbb{R}^{n_q \times 1}$ where d_q and n_q denote the dimension and number of samples observed on the q-th view respectively, we define an index matrix $\mathbf{M}_q \in \mathbb{R}^{n \times n_q}$, $[\mathbf{M}_q]_{i,j} = 1 \; if \; [b_q]_j = i \quad \forall j = 1, 2, \cdots, n_q \quad else \quad 0$. Subsequently, we have that $\mathbf{X}_q \mathbf{M}_q \in \mathbb{R}^{d_q \times n_q}$ denotes the available data on the q-th view.

Due to missing samples, the full graph will contain the blanks in certain columns and rows. By positioning view-specific sub-graphs into corresponding parts, graph learning based IMC optimizes a group of sketched sub-graphs to obtain

the fused full graph. Specifically, it firstly generates view-specific sub-graph $\mathbf{G}_q \in \mathbb{R}^{n_q \times n_q}$ by the following framework:

$$\min_{\mathbf{G}_q} \|\mathbf{X}_q \mathbf{M}_q - \mathbf{X}_q \mathbf{M}_q \mathbf{G}_q\|_F^2 + \varphi(\mathbf{G}_q)$$
$$\text{s.t. } \mathbf{G}_q^\top \mathbf{1} = \mathbf{1}, \mathbf{G}_q \geq 0, \tag{1}$$

where $\varphi(\cdot)$ denotes some regularization strategies. Then, the full graph is restored by $\mathbf{M}_q \mathbf{G}_q \mathbf{M}_q^\top \in \mathbb{R}^{n \times n}$. Afterwards, the fused full graph $\mathbf{G} \in \mathbb{R}^{n \times n}$ is obtained by solving the problem:

$$\min_{\mathbf{G}} f\left(\sum_{q=1}^v \beta_q \mathbf{M}_q \mathbf{G}_q \mathbf{M}_q^\top, \mathbf{G}\right)$$
$$\text{s.t. } \boldsymbol{\beta}^\top \mathbf{1} = 1, \boldsymbol{\beta} \geq \mathbf{0}, \mathbf{G}^\top \mathbf{1} = \mathbf{1}, \mathbf{G} \geq 0, \tag{2}$$

where $f(\cdot, \cdot)$ denotes various fusion strategies [60]–[62], and β_q denotes the graph coefficient variable. The final clustering results can be acquired by running spectral algorithms on $\mathbf{G}$. However, due to the full graph with size $n \times n$, it usually needs $\mathcal{O}(n^3)$ computational expenditures to get the spectral embedding when running spectral algorithms. To increase the efficiency, anchor technology is introduced into IMC. It firstly constructs anchors $\mathbf{H}_q \in \mathbb{R}^{d_q \times m}$ by heuristic strategies [22], [63]–[65] where m is the number of anchors, and then generates view-specific bipartite graph $\mathbf{E}_q \in \mathbb{R}^{m \times n}$ by solving

$$\min_{\mathbf{E}_q} \sum_{q=1}^v \|\mathbf{X}_q \mathbf{M}_q - \mathbf{H}_q \mathbf{E}_q \mathbf{M}_q\|_F^2 + \varphi(\mathbf{E}_q)$$
$$\text{s.t. } \mathbf{E}_q^\top \mathbf{1} = \mathbf{1}, \mathbf{E}_q \geq 0. \tag{3}$$

Afterwards, fusing $\{\mathbf{E}_q\}_{q=1}^v$ generates the consensus bipartite graph $\mathbf{E} \in \mathbb{R}^{m \times n}$. Since the final graph is with size $m \times n$ and m is far less than the number of samples n, accordingly, the computational expenditures are reduced to $\mathcal{O}(mn)$. Despite improving the efficiency while providing desirable clustering results, it generally suffers from the AMP problem. To tackle AMP, in next section, we devise a named CCA-IMC algorithm.

III. FORMULATION

We elaborate CCA-IMC's principle, time complexity, space complexity and convergence respectively.

A. Framework

Firstly, we introduce a permutation matrix for each view. Besides being able to flexibly transform anchors, it also maintains the original dimension information. Afterwards, rather than generating view-specific bipartite graphs, we directly generates the consensus bipartite graph based on the transformed anchors, which makes all anchors share a common structure. At this point, the IMC framework is formulated as

$$\min_{\mathbf{T}_q, \mathbf{E}} \sum_{q=1}^v \|\mathbf{X}_q \mathbf{M}_q - \mathbf{H}_q \mathbf{T}_q \mathbf{E} \mathbf{M}_q\|_F^2 + \lambda \|\mathbf{E}\|_F^2$$
$$\text{s.t. } \mathbf{T}_q \mathbf{1} = \mathbf{1}, \mathbf{T}_q^\top \mathbf{1} = \mathbf{1}, [\mathbf{T}_q]_{i,j} \in \{0, 1\}, \tag{4}$$
$$\mathbf{E}^\top \mathbf{1} = \mathbf{1}, \mathbf{E} \geq 0,$$

where $\mathbf{T}_q \in \mathbb{R}^{m \times m}$ denotes the permutation matrix on the q-th view. In addition, the anchor construction procedure is separated from graph generation, which causes the graph quality to severely depend on the given anchors. In IMC, due to unpaired missing instances, the heuristic strategy could not provide high-quality anchors. To this end, we choose to make all anchors optimizable, which will make anchors, permutation matrices and the consensus bipartite graph able to communicate mutually, thus enhancing each other. Further, multiview data is typically with different dimensions. This could result in each view having different importance. Unlike previous methods that consider all views equally, we assign a coefficient for each view to automatically measure its contribution. Thus, our IMC framework is designed as

$$\min_{\mathbf{H}_q, \mathbf{T}_q, \mathbf{E}, \boldsymbol{\alpha}} \sum_{q=1}^v \alpha_q^2 \|\mathbf{X}_q \mathbf{M}_q - \mathbf{H}_q \mathbf{T}_q \mathbf{E} \mathbf{M}_q\|_F^2 + \lambda \|\mathbf{E}\|_F^2$$
$$\text{s.t. } \mathbf{H}_q^\top \mathbf{H}_q = \mathbf{I}_m, \mathbf{T}_q \mathbf{1} = \mathbf{1}, \mathbf{T}_q^\top \mathbf{1} = \mathbf{1}, [\mathbf{T}_q]_{i,j} \in \{0, 1\}, \tag{5}$$
$$\mathbf{E}^\top \mathbf{1} = \mathbf{1}, \mathbf{E} \geq 0, \boldsymbol{\alpha}^\top \mathbf{1} = 1, \boldsymbol{\alpha} \geq 0,$$

where the constraint $\mathbf{H}_q^\top \mathbf{H}_q = \mathbf{I}_m$ guarantees the optimized anchors to be more discriminative, and $\boldsymbol{\alpha} \in \mathbb{R}^v$ denotes the view weight vector.

B. Optimization

To address Eq. (5), we devise an alternating optimization scheme, which divides the entire problem into four parts so that the optimal value of each variable can be acquired when fixing the rest of variables.

1) $\mathbf{H}_q$ sub-problem : Fixing other variables, Eq. (5) with respect to $\mathbf{H}_q$ can be written as

$$\min_{\mathbf{H}_q} \sum_{q=1}^v \alpha_q^2 \|\mathbf{X}_q \mathbf{M}_q - \mathbf{H}_q \mathbf{T}_q \mathbf{E} \mathbf{M}_q\|_F^2$$
$$\text{s.t. } \mathbf{H}_q^\top \mathbf{H}_q = \mathbf{I}_m. \tag{6}$$

Considering that anchor matrices $\{\mathbf{H}_q\}_{q=1}^v$ are mutually independent, we can equivalently transform Eq. (6) as

$$\min_{\mathbf{H}_q} \|\mathbf{X}_q \mathbf{M}_q - \mathbf{H}_q \mathbf{T}_q \mathbf{E} \mathbf{M}_q\|_F^2 \text{ s.t. } \mathbf{H}_q^\top \mathbf{H}_q = \mathbf{I}_m. \tag{7}$$

Further, we have

$$\min_{\mathbf{H}_q} \|\mathbf{X}_q \mathbf{M}_q - \mathbf{H}_q \mathbf{T}_q \mathbf{E} \mathbf{M}_q\|_F^2$$
$$= \min_{\mathbf{H}_q} \text{Tr} \left(\mathbf{X}_q \mathbf{M}_q \mathbf{M}_q^\top \mathbf{X}_q^\top + \mathbf{H}_q \mathbf{T}_q \mathbf{E} \mathbf{M}_q \mathbf{M}_q^\top \mathbf{E}^\top \mathbf{T}_q^\top \mathbf{H}_q^\top \right.$$
$$\left. -2\mathbf{X}_q \mathbf{M}_q \mathbf{M}_q^\top \mathbf{E}^\top \mathbf{T}_q^\top \mathbf{H}_q^\top \right)$$
$$\Leftrightarrow \max_{\mathbf{H}_q} \text{Tr} \left(\mathbf{X}_q \mathbf{M}_q \mathbf{M}_q^\top \mathbf{E}^\top \mathbf{T}_q^\top \mathbf{H}_q^\top \right)$$
$$= \max_{\mathbf{H}_q} \text{Tr} \left[\mathbf{H}_q^\top (\mathbf{X}_q \odot \mathbf{W}_q) \mathbf{E}^\top \mathbf{T}_q^\top \right], \tag{8}$$

where $\mathbf{W}_q = \mathbf{1}_{d_q} \cdot \left[\sum_{j=1}^{n_q} [\mathbf{M}_q]_{1,j}, \ldots, \sum_{j=1}^{n_q} [\mathbf{M}_q]_{i,j} \ldots, \right.$ $\left. \sum_{j=1}^{n_q} [\mathbf{M}_q]_{n,j} \right]$. Subsequently, given the theorem in [66], the optimal $\mathbf{H}_q$ is the product of $\mathbf{U}$ and $\mathbf{V}^\top$ where $\mathbf{U}$ and $\mathbf{V}$ are the SVD results of $(\mathbf{X}_q \odot \mathbf{W}_q) \mathbf{E}^\top \mathbf{T}_q^\top$ respectively.

Hadamard product operation plays a role in decreasing the computational complexity. Specifically, due to $\mathbf{M}_q\mathbf{M}_q^\top$ being

$$
\begin{bmatrix}
\sum_{j=1}^{n_q}[\mathbf{M}_q]_{1,j} & 0 & \cdots & 0 \\
0 & \sum_{j=1}^{n_q}[\mathbf{M}_q]_{2,j} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & \sum_{j=1}^{n_q}[\mathbf{M}_q]_{n,j}
\end{bmatrix} \in \mathbb{R}^{n\times n}
\tag{9}
$$

and $\mathbf{X}_q \in \mathbb{R}^{d_q\times n}$, the construction of the term $\mathbf{X}_q\mathbf{M}_q\mathbf{M}_q^\top$ takes $\mathcal{O}(d_q n^2)$. Accordingly, performing SVD on $\mathbf{X}_q\mathbf{M}_q\mathbf{M}_q^\top\mathbf{E}^\top\mathbf{T}_q^\top$ to generate the optimal $\mathbf{H}_q$ takes $\mathcal{O}(n^2)$. In virtue of the Hadamard product operation, $\mathbf{X}_q\mathbf{M}_q\mathbf{M}_q^\top$ is equivalently transformed as $\mathbf{X}_q \odot \mathbf{W}_q$. The operation $\mathbf{X}_q \odot \mathbf{W}_q$ takes $\mathcal{O}(d_q n)$ complexity. Computing $(\mathbf{X}_q \odot \mathbf{W}_q)\mathbf{E}^\top\mathbf{T}_q^\top$ takes $\mathcal{O}(d_q n + d_q nm + d_q m^2)$ time cost and performing SVD on it takes $\mathcal{O}(d_q m^2)$ cost. Consequently, optimizing $\mathbf{H}_q$ takes $\mathcal{O}(d_q nm)$ totally per iteration.

2) $\mathbf{T}_q$ sub-problem : Fixing other variables, Eq. (5) with respect to $\mathbf{T}_q$ can be equivalently written as

$$
\min_{\mathbf{T}_q} \|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2
$$
$$
\text{s.t. } \mathbf{T}_q\mathbf{1} = \mathbf{1}, \mathbf{T}_q^\top\mathbf{1} = \mathbf{1}, [\mathbf{T}_q]_{i,j} \in \{0,1\}.
\tag{10}
$$

After deleting irrelevant items, Eq. (10) is equivalently converted as

$$
\max_{\mathbf{T}_q} \operatorname{Tr}(\mathbf{B}\mathbf{T}_q) \text{ s.t. } \mathbf{T}_q\mathbf{1} = \mathbf{1}, \mathbf{T}_q^\top\mathbf{1} = \mathbf{1}, [\mathbf{T}_q]_{i,j} \in \{0,1\},
\tag{11}
$$

where $\mathbf{B} = \mathbf{E}(\mathbf{X}_q \odot \mathbf{W}_q)^\top\mathbf{H}_q$. In conjunction with matrix vectorization operation, we further have

$$
\max_{\mathbf{T}_q} \left(\operatorname{Vec}(\mathbf{B}^\top)\right)^\top \operatorname{Vec}(\mathbf{T}_q)
$$
$$
\text{s.t. } \mathbf{T}_q\mathbf{1} = \mathbf{1}, \mathbf{T}_q^\top\mathbf{1} = \mathbf{1}, [\mathbf{T}_q]_{i,j} \in \{0,1\},
\tag{12}
$$

which is a combinatorial optimization problem and can be solved in $\mathcal{O}(m^3)$ complexity.

Computing $\mathbf{E}(\mathbf{X}_q \odot \mathbf{W}_q)^\top\mathbf{H}_q$ and performing combinatorial optimization solving on it cost $\mathcal{O}(d_q n + mnd_q + m^2 d_q)$ and $\mathcal{O}(m^3)$ time complexity respectively. Accordingly, optimizing $\mathbf{T}_q$ needs $\mathcal{O}(mnd_q)$ in total per iteration.

3) $\mathbf{E}$ sub-problem: Fixing other variables, Eq. (5) with respect to $\mathbf{E}$ becomes

$$
\min_{\mathbf{E}} \sum_{q=1}^v \alpha_q^2 \|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2 + \lambda \|\mathbf{E}\|_F^2
$$
$$
\text{s.t. } \mathbf{E}^\top\mathbf{1} = \mathbf{1}, \mathbf{E} \geq 0.
\tag{13}
$$

Expanding the objective of Eq. (13) yields

$$
\min_{\mathbf{E}} \sum_{q=1}^v \alpha_q^2 \|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2 + \lambda \|\mathbf{E}\|_F^2
$$
$$
\Leftrightarrow \min_{\mathbf{E}} \sum_{q=1}^v \alpha_q^2 \operatorname{Tr}\left(\mathbf{E}^\top\mathbf{E}\mathbf{M}_q\mathbf{M}_q^\top - 2\mathbf{E}^\top\mathbf{T}_q^\top\mathbf{H}_q^\top\mathbf{X}_q\mathbf{M}_q\mathbf{M}_q^\top\right)
$$
$$
+ \lambda \operatorname{Tr}\left(\mathbf{E}^\top\mathbf{E}\right)
$$
$$
= \min_{\mathbf{E}} \operatorname{Tr}\left(\mathbf{E}^\top\mathbf{E}\mathbf{C} - 2\mathbf{E}^\top\mathbf{D}\right),
\tag{14}
$$

where $\mathbf{C} = \sum_{q=1}^v \alpha_q^2\mathbf{M}_q\mathbf{M}_q^\top + \lambda\mathbf{I}_n$ and $\mathbf{D} = \sum_{q=1}^v \alpha_q^2\mathbf{T}_q^\top\mathbf{H}_q^\top\mathbf{X}_q\mathbf{M}_q\mathbf{M}_q^\top = \sum_{q=1}^v \alpha_q^2\mathbf{T}_q^\top\mathbf{H}_q^\top(\mathbf{X}_q \odot \mathbf{W}_q)$.

In conjunction with the facts that $\mathbf{C}$ is a diagonal matrix and that the constraint $\mathbf{E1} = \mathbf{1}$ is for each column of $\mathbf{E}$, we can transform Eq. (13) as:

$$
\min_{\mathbf{E}_{:,j}} \mathbf{E}_{:,j}^\top\mathbf{E}_{:,j}\mathbf{C}_{j,j} - 2\mathbf{E}_{:,j}^\top\mathbf{D}_{:,j} \text{ s.t. } \mathbf{E}_{:,j}^\top\mathbf{1} = 1, \mathbf{E}_{:,j} \geq 0.
\tag{15}
$$

Further, we have

$$
\min_{\mathbf{E}_{:,j}} \left\|\mathbf{E}_{:,j} - \frac{\mathbf{D}_{:,j}}{\mathbf{C}_{j,j}}\right\|_F^2 \text{ s.t. } \mathbf{E}_{:,j}^\top\mathbf{1} = 1, \mathbf{E}_{:,j} \geq 0.
\tag{16}
$$

The Lagrangian function of Eq. (16) is

$$
\mathcal{L}(\mathbf{E}_{:,j}, \beta, \boldsymbol{\gamma}) = \frac{1}{2}\left\|\mathbf{E}_{:,j} - \frac{\mathbf{D}_{:,j}}{\mathbf{C}_{j,j}}\right\|_F^2 - \beta\left(\mathbf{E}_{:,j}^\top\mathbf{1} - 1\right) - \boldsymbol{\gamma}^\top\mathbf{E}_{:,j}.
\tag{17}
$$

Based on KKT conditions, we have

$$
\mathbf{E}_{:,j} - \frac{\mathbf{D}_{:,j}}{\mathbf{C}_{j,j}} - \beta\mathbf{1} - \boldsymbol{\gamma} = 0, \quad \boldsymbol{\gamma} \odot \mathbf{E}_{:,j} = 0.
\tag{18}
$$

That is,

$$
\mathbf{E}_{:,j} = \frac{\mathbf{D}_{:,j}}{\mathbf{C}_{j,j}} + \beta\mathbf{1} + \boldsymbol{\gamma}, \quad \text{and} \quad \gamma_i[\mathbf{E}_{:,j}]_i = 0, \quad i = 1, 2, \ldots, m.
\tag{19}
$$

Given the constraints $\mathbf{E}_{:,j}^\top\mathbf{1} = 1$ and $\mathbf{E}_{:,j} \geq 0$, we further have

$$
\mathbf{E}_{:,j} = \left(\frac{\mathbf{D}_{:,j}}{\mathbf{C}_{j,j}} + \frac{1 - \frac{\mathbf{D}_{:,j}^\top\mathbf{1}}{\mathbf{C}_{j,j}}}{m}\right)_+
\tag{20}
$$

Then, stacking $\{\mathbf{E}_{:,j}\}_{j=1}^n$ by column generates the optimal $\mathbf{E}$.

Constructing $\mathbf{C}$ and $\mathbf{D}$ requires $\mathcal{O}(nd)$ and $\mathcal{O}(dn + m^2d + mdn)$ time overheads respectively where d is $\sum_{q=1}^v d_q$. Calculating the analytical solution of $\mathbf{E}$ takes $\mathcal{O}(mn)$. Thus, optimizing $\mathbf{E}$ costs $\mathcal{O}(mdn)$ time complexity per iteration.

4) $\boldsymbol{\alpha}$ sub-problem : Fixing other variables, Eq. (5) with respect to $\boldsymbol{\alpha}$ becomes

$$
\min_{\boldsymbol{\alpha}} \sum_{q=1}^v \alpha_q^2 \|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2 \text{ s.t. } \boldsymbol{\alpha}^\top\mathbf{1} = 1, \boldsymbol{\alpha} \geq 0.
\tag{21}
$$

Since the term $\|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2$ is irrelevant to α_q, we can acquire the optimal α_q by Cauchy-Schwarz inequality:

$$
\alpha_q = \frac{\frac{1}{\|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2}}{\sum_{q=1}^v \frac{1}{\|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2}}.
\tag{22}
$$

Note that

$$
\|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2 =
$$
$$
\|\mathbf{X}_q\mathbf{M}_q\mathbf{M}_q^\top - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\mathbf{M}_q^\top\|_F^2 =
$$
$$
\|\mathbf{X}_q \odot \mathbf{W}_q - \mathbf{H}_q\mathbf{T}_q(\mathbf{E} \odot \mathbf{G}_q)\|_F^2,
\tag{23}
$$

where $\mathbf{G}_q = \mathbf{1}_{m\times 1} \cdot \left[\sum_{j=1}^{n_q}[\mathbf{M}_q]_{1,j}, \ldots, \sum_{j=1}^{n_q}[\mathbf{M}_q]_{n,j}\right]_{1\times n}$. We can calculate the term $\|\mathbf{X}_q \odot \mathbf{W}_q - \mathbf{H}_q\mathbf{T}_q(\mathbf{E} \odot \mathbf{G}_q)\|_F^2$ instead of $\|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2$. The purpose of doing so is to reduce the computational complexity. To be specific, due to the computation of $\mathbf{X}_q\mathbf{M}_q$ taking $\mathcal{O}(d_q nn_q)$, calculating $\|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2$ will take $\mathcal{O}(n^2)$ when

TABLE I
CLUSTERING RESULTS COMPARISON. RANKING NO.1 AND NO.2 RESULTS ARE EMPHASIZED IN BLACK AND UNDERLINED RESPECTIVELY. MOE REPRESENTS THE MEMORY OVERFLOW ERROR.

Dataset	Metric	BSV [27]	MIC [33]	MKKM-IK [29]	DAIMC [37]	UEAF [35]	IK-MKC [43]	EEIMVC [44]	FLSD [39]	CBG [38]	PIMVC [67]	BGIMVSC [68]	Proposed
ORL	NMI	48.49±0.90	56.44±1.00	75.95±1.33	82.89±1.06	76.16±1.25	79.76±1.41	85.37±1.32	67.91±1.28	<u>87.58±1.31</u>	84.22±0.88	67.13±0.77	**88.78±1.29**
	ACC	24.32±0.89	37.56±1.66	59.80±2.44	68.03±2.32	60.25±2.50	64.95±2.62	73.24±2.54	48.09±1.85	<u>75.60±2.85</u>	70.64±2.00	45.75±1.07	**77.27±2.67**
	PUR	26.80±0.92	40.81±1.40	62.79±2.11	71.82±1.79	63.90±1.90	67.68±2.34	76.09±2.19	50.88±1.72	<u>78.54±2.35</u>	74.17±1.60	49.22±0.88	**80.12±2.24**
	FSC	9.01±0.69	17.30±1.18	46.32±2.50	56.84±2.87	42.53±2.74	53.30±2.91	63.67±2.85	31.17±2.00	<u>67.34±3.07</u>	57.65±2.88	29.26±1.87	**69.72±3.02**
NGs	NMI	19.53±0.96	2.39±0.59	63.16±0.11	62.39±5.28	66.98±0.03	32.78±0.15	56.50±1.66	64.45±1.87	<u>72.18±0.03</u>	71.16±0.10	61.90±0.02	**77.07±0.06**
	ACC	40.45±1.53	20.93±0.46	79.63±0.09	80.49±5.03	84.33±0.01	49.86±0.12	75.95±4.56	83.83±3.53	<u>88.47±0.01</u>	87.89±0.06	76.18±0.03	**90.73±0.00**
	PUR	43.12±0.85	21.29±0.42	79.63±0.09	80.55±4.93	84.33±0.01	50.56±0.10	76.50±3.09	84.09±2.95	<u>88.47±0.01</u>	87.89±0.06	76.71±0.02	**90.73±0.00**
	FSC	32.27±0.61	32.95±0.04	68.72±0.07	67.96±5.58	72.93±0.02	42.29±0.13	62.83±2.41	71.95±2.48	<u>78.64±0.02</u>	77.67±0.09	67.05±0.03	**82.43±0.01**
Prokaryotic	NMI	14.35±0.07	24.04±1.28	21.00±0.24	26.30±3.19	18.76±0.01	19.25±0.04	26.28±1.11	22.64±1.19	<u>28.26±0.17</u>	MOE	25.18±0.00	**31.20±0.21**
	ACC	48.65±0.12	52.25±1.69	45.91±0.12	50.37±3.95	45.57±0.03	46.12±0.07	55.98±2.30	<u>56.99±0.00</u>	55.97±0.12	MOE	54.93±0.00	**61.48±0.21**
	PUR	59.99±0.12	64.73±0.39	59.12±0.29	64.45±2.69	60.41±0.02	60.36±0.05	<u>67.88±0.82</u>	62.38±2.02	67.25±0.02	MOE	65.88±0.00	**71.61±0.17**
	FSC	46.55±0.17	47.27±0.89	38.90±0.12	43.67±1.79	39.72±0.01	40.50±0.05	44.40±1.28	**56.56±0.00**	45.42±0.08	MOE	48.54±0.01	<u>48.68±0.17</u>
WebKB	NMI	1.91±0.28	2.88±0.16	2.56±0.00	10.37±4.58	16.44±0.01	2.01±0.00	1.86±0.23	4.43±3.66	<u>18.61±0.08</u>	14.70±0.06	9.47±0.00	**42.15±0.00**
	ACC	56.33±0.47	61.57±0.84	66.14±0.00	72.26±6.05	76.99±0.02	50.53±0.00	63.26±2.74	78.21±0.00	<u>78.69±0.06</u>	75.60±0.11	56.50±0.00	**89.06±0.00**
	PUR	78.12±0.00	78.12±0.00	78.12±0.00	79.47±0.93	80.56±0.01	78.12±0.00	78.12±0.00	78.63±1.14	<u>80.66±0.06</u>	80.26±0.00	78.12±0.00	**89.06±0.00**
	FSC	59.53±0.15	60.02±0.63	63.52±0.00	70.30±5.75	72.77±0.02	57.47±0.00	63.06±2.07	<u>79.40±0.00</u>	73.41±0.07	72.22±0.10	59.35±0.00	**85.78±0.00**
HW	NMI	51.45±0.18	39.98±2.13	53.97±0.43	65.24±6.81	46.32±0.49	53.88±0.90	69.01±1.81	45.27±4.06	66.02±0.24	MOE	**73.43±0.72**	<u>69.73±0.44</u>
	ACC	49.99±0.48	44.31±2.69	64.93±0.66	73.09±8.86	50.19±0.62	54.64±1.15	<u>76.07±4.84</u>	39.79±5.02	74.31±0.23	MOE	63.68±2.02	**80.20±0.62**
	PUR	53.72±0.12	45.85±2.52	65.24±0.56	73.51±8.39	50.35±0.64	55.22±1.04	<u>77.39±3.75</u>	40.48±5.03	75.56±0.19	MOE	64.46±1.56	**80.20±0.62**
	FSC	38.27±0.40	32.97±1.90	50.92±0.50	62.15±8.01	41.00±0.44	47.09±0.82	<u>65.93±3.00</u>	36.76±3.70	62.62±0.26	MOE	63.43±1.20	**67.16±0.69**
Handwritten	NMI	50.75±0.22	39.95±1.99	64.90±0.68	65.34±4.59	45.86±0.63	54.67±0.69	<u>71.07±2.55</u>	43.85±4.29	66.21±0.53	MOE	**74.45±0.66**	70.87±0.41
	ACC	49.79±0.63	43.95±2.74	71.01±1.34	72.74±5.97	50.05±0.95	56.02±1.50	<u>77.62±6.16</u>	38.27±4.89	76.51±0.67	MOE	66.51±1.70	**80.90±0.41**
	PUR	53.12±0.15	45.49±2.34	72.43±1.00	73.03±5.74	50.36±0.91	57.11±1.41	<u>79.12±4.70</u>	38.98±4.36	77.13±0.53	MOE	67.48±1.31	**80.90±0.41**
	FSC	37.65±0.37	33.07±1.86	60.80±0.94	62.05±5.92	40.09±0.63	47.58±0.90	<u>68.20±4.12</u>	35.55±3.66	63.94±0.48	MOE	65.50±1.51	**68.73±0.64**
Hdigit	NMI	8.73±0.17	36.71±1.92	43.94±0.14	48.72±2.18	53.26±0.18	36.85±0.30	48.34±1.23	57.81±1.44	53.00±0.15	**82.52±0.03**	46.64±1.15	<u>61.85±0.08</u>
	ACC	14.27±0.19	43.46±3.08	53.56±0.17	58.87±5.34	64.15±0.36	44.39±0.56	59.10±2.69	64.37±4.06	66.51±0.19	**88.21±0.09**	49.40±1.41	<u>77.36±0.06</u>
	PUR	16.17±0.15	45.47±2.48	55.03±0.14	60.91±3.77	65.37±0.30	45.32±0.54	61.36±1.75	66.71±2.68	67.35±0.17	**88.57±0.03**	49.49±1.31	<u>77.36±0.06</u>
	FSC	17.87±0.02	29.60±1.50	40.34±0.15	46.32±2.88	51.04±0.28	32.91±0.35	46.14±1.64	53.38±2.49	51.72±0.20	**82.38±0.06**	38.38±0.76	<u>62.83±0.07</u>
NUSWIDEOBJ	NMI	MOE	MOE	MOE	**11.97±0.37**	MOE	MOE	MOE	MOE	9.91±0.07	MOE	MOE	<u>10.00±0.16</u>
	ACC	MOE	MOE	MOE	**13.83±0.41**	MOE	MOE	MOE	MOE	12.00±0.19	MOE	MOE	<u>12.66±0.14</u>
	PUR	MOE	MOE	MOE	**23.67±0.45**	MOE	MOE	MOE	MOE	21.50±0.10	MOE	MOE	<u>21.55±0.17</u>
	FSC	MOE	MOE	MOE	**8.59±0.16**	MOE	MOE	MOE	MOE	7.43±0.06	MOE	MOE	<u>7.70±0.06</u>
Cifar10	NMI	MOE	MOE	MOE	MOE	MOE	MOE	MOE	MOE	<u>91.04±0.00</u>	MOE	MOE	**91.35±0.00**
	ACC	MOE	MOE	MOE	MOE	MOE	MOE	MOE	MOE	<u>96.38±0.00</u>	MOE	MOE	**96.54±0.00**
	PUR	MOE	MOE	MOE	MOE	MOE	MOE	MOE	MOE	<u>96.38±0.00</u>	MOE	MOE	**96.54±0.00**
	FSC	MOE	MOE	MOE	MOE	MOE	MOE	MOE	MOE	<u>93.01±0.00</u>	MOE	MOE	**93.32±0.00**
MNIST	NMI	MOE	MOE	MOE	MOE	MOE	MOE	MOE	MOE	<u>96.03±0.00</u>	MOE	MOE	**96.16±0.00**
	ACC	MOE	MOE	MOE	MOE	MOE	MOE	MOE	MOE	<u>98.64±0.00</u>	MOE	MOE	**98.70±0.00**
	PUR	MOE	MOE	MOE	MOE	MOE	MOE	MOE	MOE	<u>98.64±0.00</u>	MOE	MOE	**98.70±0.00**
	FSC	MOE	MOE	MOE	MOE	MOE	MOE	MOE	MOE	<u>97.31±0.00</u>	MOE	MOE	**97.43±0.00**

n_q is close to n. In other words, it will take $\mathcal{O}(n^2)$ complexity when the incomplete data ratio is relatively low. To decrease the computational complexity to $\mathcal{O}(n)$ for any incomplete data ratios, we note that $\|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2 = \|\mathbf{X}_q\mathbf{M}_q\mathbf{M}_q^\top - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\mathbf{M}_q^\top\|_F^2$. Further, we have that $\mathbf{E}\mathbf{M}_q\mathbf{M}_q^\top$ equals to $\mathbf{E} \odot \mathbf{G}_q$. Therefore, we can transform the coefficient item $\|\mathbf{X}_q\mathbf{M}_q - \mathbf{H}_q\mathbf{T}_q\mathbf{E}\mathbf{M}_q\|_F^2$ as $\|\mathbf{X}_q \odot \mathbf{W}_q - \mathbf{H}_q\mathbf{T}_q(\mathbf{E} \odot \mathbf{G}_q)\|_F^2$. Since the calculation of $\mathbf{E} \odot \mathbf{G}_q$ takes $\mathcal{O}(mn)$ time complexity, computing the item $\|\mathbf{X}_q \odot \mathbf{W}_q - \mathbf{H}_q\mathbf{T}_q(\mathbf{E} \odot \mathbf{G}_q)\|_F^2$ will need $\mathcal{O}(d_q n + mn + d_q m^2 + d_q mn)$. Consequently, optimizing $\boldsymbol{\alpha}$ takes $\mathcal{O}(dmn)$ time overhead per iteration.

Our proposed CCA-IMC is summarized in Algorithm 1 where $obj^{(t)}$ is the objective value at the t-th iteration.

Algorithm 1 CCA-IMC

Input: Incomplete multiview data $\{\mathbf{X}_q \in \mathbb{R}^{d_q \times n}\}_{q=1}^v$, indicator vectors $\{b_q \in \mathbb{R}^{n_q \times 1}\}_{q=1}^v$, hyper-parameter λ;
Initialize: $\mathbf{H}_q$, $\mathbf{T}_q$, $\mathbf{E}$, $\boldsymbol{\alpha}$;

1: t=0;
2: **repeat**
3:     t=t+1;
4:     Update $\mathbf{H}_q$ using Eq. (8);
5:     Update $\mathbf{T}_q$ using Eq. (12);
6:     Update $\mathbf{E}$ using Eq. (20);
7:     Update $\boldsymbol{\alpha}$ using Eq. (22);
8: **until** $\frac{obj^{(t-1)}-obj^{(t)}}{obj^{(t)}} \leq 10^{-4}$;

Output: Clustering results by running spectral methods on $\mathbf{E}$;

C. Discussion

1) Time Complexity Analysis: The time expenditure of our devised CCA-IMC consists of optimizing all $\mathbf{H}_q$, $\mathbf{T}_q$, $\mathbf{E}$ and $\boldsymbol{\alpha}$. During optimizing $\{\mathbf{H}_q\}_{q=1}^v$, it takes $\mathcal{O}(dnm)$ complexity to acquire the optimal solution. During optimizing $\{\mathbf{T}_q\}_{q=1}^v$, it costs $\mathcal{O}(mnd)$. Optimizing $\mathbf{E}$ and $\boldsymbol{\alpha}$ costs $\mathcal{O}(mdn)$ and $\mathcal{O}(dmn)$ respectively. As a result, CCA-IMC's time complexity is proven to be $\mathcal{O}(dmn)$. Evidently, it is linear with respect to the number of samples n.
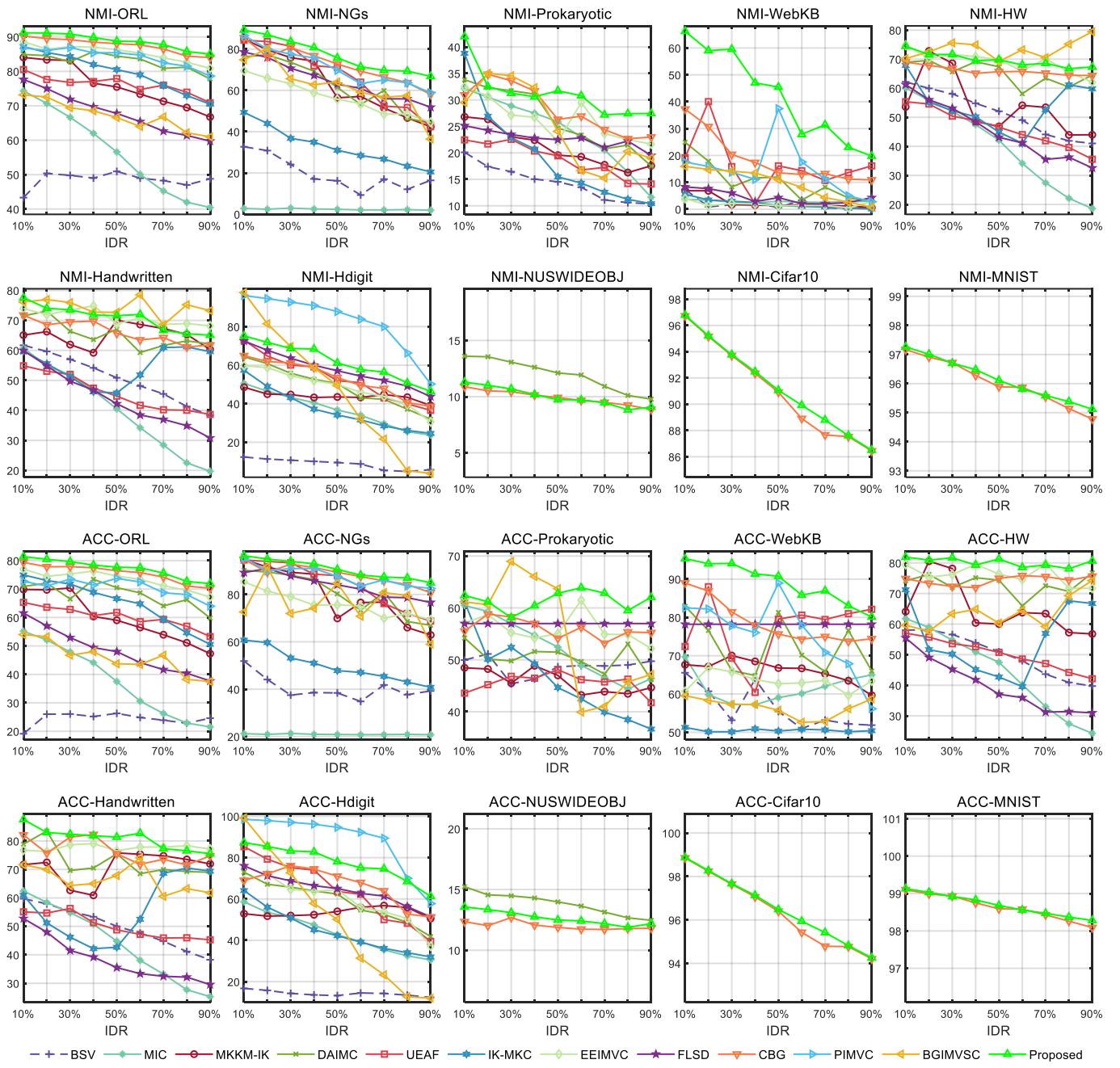
Fig. 4. The clustering results of twelve algorithms in terms of NMI and ACC under diverse IDRs.

IV. EXPERIMENT

A. Datasets and Compared Algorithms

2) Space Complexity Analysis: The space expenditure of our CCA-IMC mainly involves $\mathbf{H}_q \in \mathbb{R}^{d_q \times m}$, $\mathbf{T}_q \in \mathbb{R}^{m \times m}$, $\mathbf{E} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{\alpha} \in \mathbb{R}^v$. Hence, CCA-IMC's space complexity is $dm + vm^2 + mn + v$. Due to $d \ll n$, $v \ll n$ and $m \ll n$, CCA-IMC owns $\mathcal{O}(n)$ space complexity.

3) Convergence Analysis: During alternatively optimizing $\mathbf{H}_q$, $\mathbf{T}_q$, $\mathbf{E}$ and $\boldsymbol{\alpha}$, their optimal value can be analytically obtained by Eqs. (8), (12), (20) and (22) respectively. This indicates that the objective value of Eq. (5) decreases monotonically as iteration goes on. In addition to this, the objective function of Eq. (5) has the lower bound, for example 0. Given the alternating optimization theory [69], therefore, our CCA-IMC is convergent.

Datasets: We implement experiments on ten popularly used datasets, and their detailed descriptions are as following:

1) ORL[1]: This image dataset is taken from distinct lighting and facial expressions, and consists of 400 samples from 40 classes. The feature dimensions of 3 views are 4096, 3304, 6750 respectively.
2) NGs[2]: This news dataset is taken from different newsgroup documents, and consists of 500 samples from 5

---

[1]http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html
[2]http://qwone.com/jason/20Newsgroups/
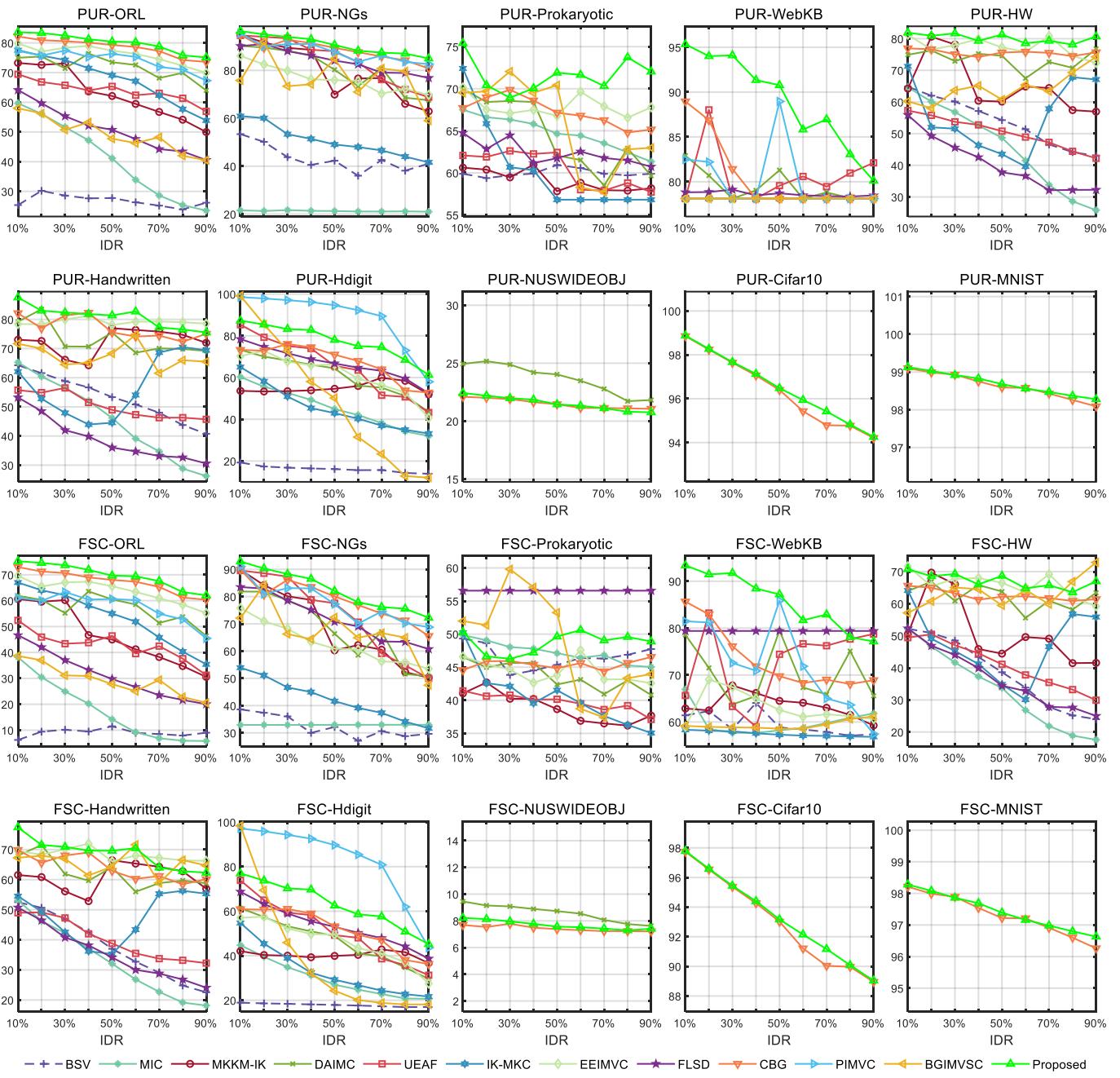
Fig. 5. The clustering results of twelve algorithms in terms of PUR and FSC under diverse IDRs.

classes. The feature dimensions of 3 views are all 2000.

3) Prokaryotic[3]: This specie dataset is collected from diverse prokaryotic cells, and consists of 551 samples from 4 classes. The feature dimensions of 3 views are 438, 3 and 393 respectively.

4) WebKB[4]: This webpage dataset is collected from distinct websites, and consists of 1051 samples from 2 classes. The feature dimensions of 2 views are 2949 and 334 respectively.

5) HW[5]: It is composed of 2000 image samples that are from 6 views and 10 classes. Their dimensions are 216, 76, 64, 6, 240 and 47 respectively.

6) Handwritten[6]: It is composed of 2000 handwritten numeral samples (i.e., "0" ∼ "9") that are from 6 views and 10 classes. Their dimensions are 240, 76, 216, 47, 64 and 6 respectively.

7) Hdigit[7]: There are 10000 image samples from 2 views and 10 classes. One view is with 784 dimensions while the other is with 256 dimensions.

---

[3] https://www.bacterio.net/
[4] https://lig-membres.imag.fr/grimal/data.html
[5] https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[6] https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[7] http://mkl.ucsd.edu/dataset/
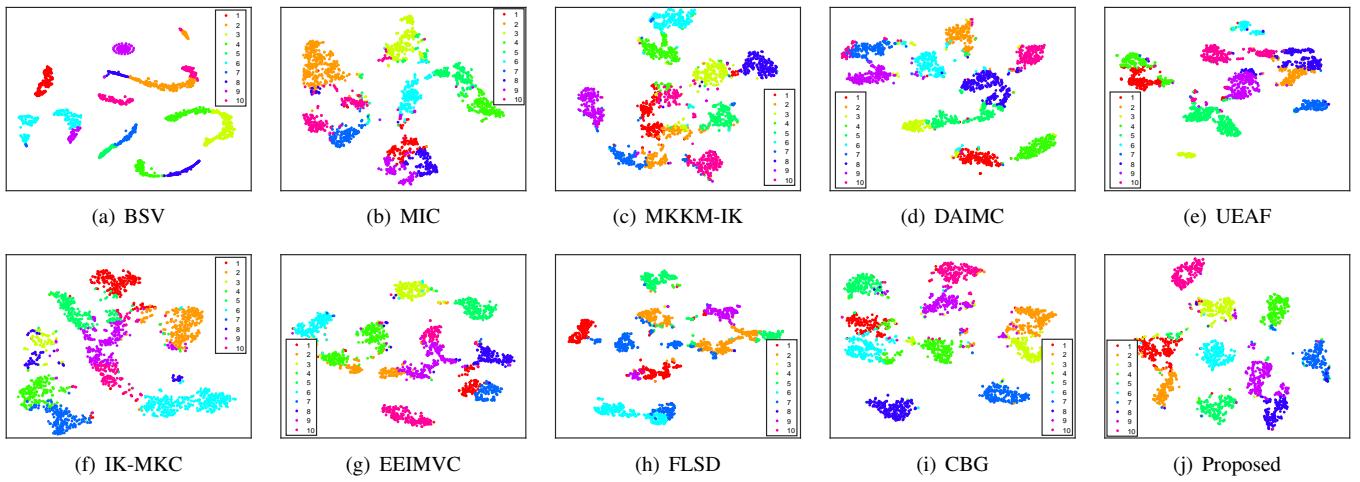
Fig. 6. Visualization of the representations learned by above mentioned IMC algorithms on dataset Handwritten.
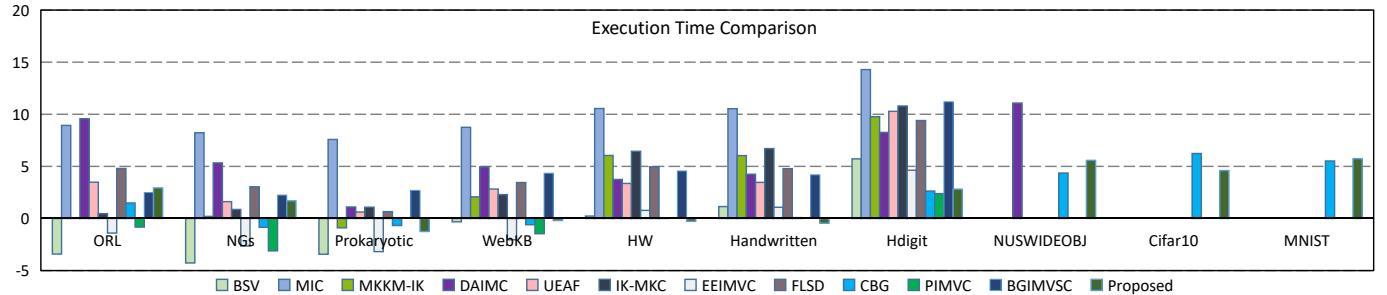
Fig. 7. Execution time comparison between twelve algorithms on ten benchmark datasets. All results are shown in the form of $\log_2$.

8) NUSWIDEOBJ[8]: There are 30000 web samples that are from 5 views and 31 classes. Their dimensions are 65, 226, 145, 74 and 129 respectively.
9) Cifar10[9]: This dataset contains 50000 color image samples totally over 10 classes. The feature dimensions of 3 views are 512, 2048 and 1024 respectively.
10) MNIST[10]: This dataset contains 60000 handwritten image samples totally over 10 classes. The feature dimensions of 3 views are 342, 1024 and 64 respectively.

Notice that the number of samples in the above datasets ranges from 400 to 60000. In current IMC, this size span is already quite large.

Compared Algorithms: To demonstrate the superiority of our CCA-IMC, we compare it with the following representative algorithms.

1) BSV [27]: This method imputes the missing instances with average value, and subsequently derives the class indicators by separately calculating the k way partitioning on the spectral graph of each incomplete view.
2) MIC [33]: This method minimizes the disagreement among view-specific features by the consensus matrix, and embeds $\ell_{1,2}$ regularization constraint into weighted

NMF to enhance the robustness of the learned potential representation to noises.
3) MKKM-IK [29]: This method alternatively performs filling and clustering operations in one unified framework, and guides partial kernels imputation with the clustering results from the previous iteration.
4) DAIMC [37]: This method decreases the impact of incomplete entries by building a common basis matrix for all views using $\ell_{1,2}$ regression, and employs semi-NMF scheme to increase the ability to cope with more than two views.
5) UEAF [35]: This method explicitly exploits the local structure between data by introducing the locality-conserved reconstruction item, and learns a shared graph by the reverse graph regularization to align views.
6) IK-MKC [43]: This method integrates kernel imputation and data grouping into a single-level optimization process, and fills absent kernel matrices mutually to further increase the clustering performance.
7) EEIMVC [44]: This method infers base clustering matrices rather than similarity ones, and utilizes the prior knowledge that consensus matrix is located in the neighborhood of predetermined one to improve the discrimination of consensus matrix.
8) FLSD [39]: This method explores local geometric information between views using the graph-regularized constraint, and employs the semantic consistency regularizer

(a) View 1   (b) View 2   (c) View 3   (d) View 4   (e) View 5   (f) View 6

(g) Permutation2-1   (h) Permutation3-2   (i) Permutation4-3   (j) Permutation5-4   (k) Permutation6-5   (l) Permutation1-6
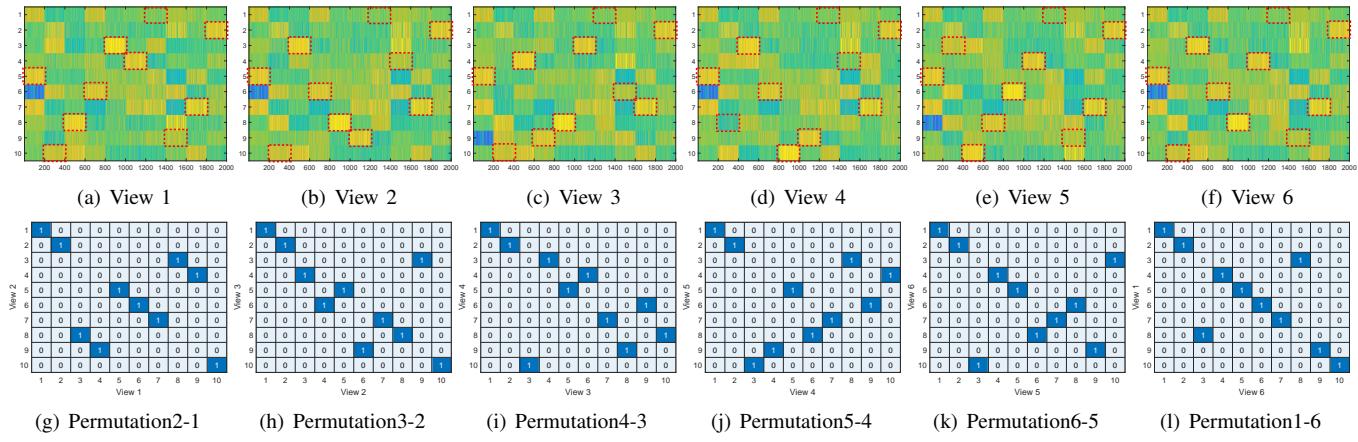
Fig. 8. Visualization of the restored view-specific bipartite graphs and permutation matrices between views on dataset Handwritten. Permutation2-1 is the permutation relationship between views 2 and 1. Taking (a) (b) (g) as an example, after multiplying (g), anchors on (a) are correctly matched to these on (b).
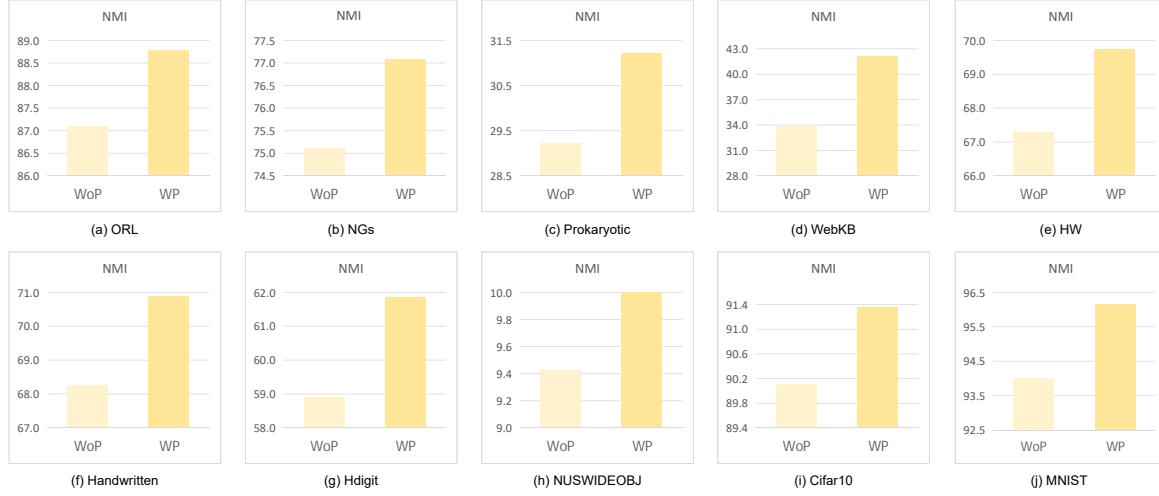
Fig. 9. Illustration of the effectiveness of our permutation strategy. "WoP" represents the results obtained **w**ith**o**ut using our **p**ermutation strategy while "WP" represents the results obtained **w**ith our **p**ermutation strategy.

to form the unified representation for all views.

9) CBG [38]: This method adopts the projection strategy to alleviate the problem of anchor dimension inconsistency, and provides a probability interpretation for the unified anchor graph from the perspective of random walk.

10) PIMVC [67]: This method tackles the new samples via a group of projections generated by a consensus graph regularizer, and alleviates the information imbalance among views by capturing the unified characteristics within a low-dimensional subspace.

11) BGIMVSC [68]: This method embeds the clustering indicators into a graph learning model to directly conduct the data partitioning operation, and relaxes the spectral framework to obtain the probability agreement representation so as to make the results stable.

B. Experimental Setting

We generate the incomplete versions of aforementioned datasets according to the strategy in [38]. The incomplete data ratio (IDR) is set from 10% to 90% with 10% as the interval. For these compared algorithms, we download the source code

from the authors' websites, and tune the hyper-parameters based on the guidelines in corresponding paper. For our CCA-IMC, we adjust λ in $[10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$ and the anchor number in $[1, 2, 3, 4, 5]k$ respectively. Additionally, we run each algorithm 20 times so as to alleviate the randomness, and count the mean values and corresponding standard deviations. We adopt four popular metrics, NMI, ACC, Purity (PUR), Fscore (FSC), to evaluate the algorithm performance. The definitions of NMI, ACC, PUR, FSC are respectively

$$\text{NMI} = \frac{M(p,q)}{(S(p)S(q))^{\frac{1}{2}}}, \quad \text{ACC} = \frac{1}{n}\sum_{i=1}^{n} f\left(p_i, g\left(q_i\right)\right),$$
$$\text{PUR} = \frac{1}{n}\sum_{i=1}^{k} n_i c_i, \quad \text{FSC} = \frac{\text{PR} \times \text{RE}}{\frac{1}{2}(\text{PR} + \text{RE})}, \quad (24)$$

where p and q represent the predicated label variable and true label variable. $M(p,q)$ is the mutual information with respect to p and q. $S(p)$ measures p's entropy. p_i is the predicated label for sample x_i while q_i is its true label. The function g plays a role in permuting the true labels by the KuhnMunkres method. The function f is that $f(a,b) = 0 \ \ if \ \ a \neq b \ \ else \ \ 1$. n_i is the
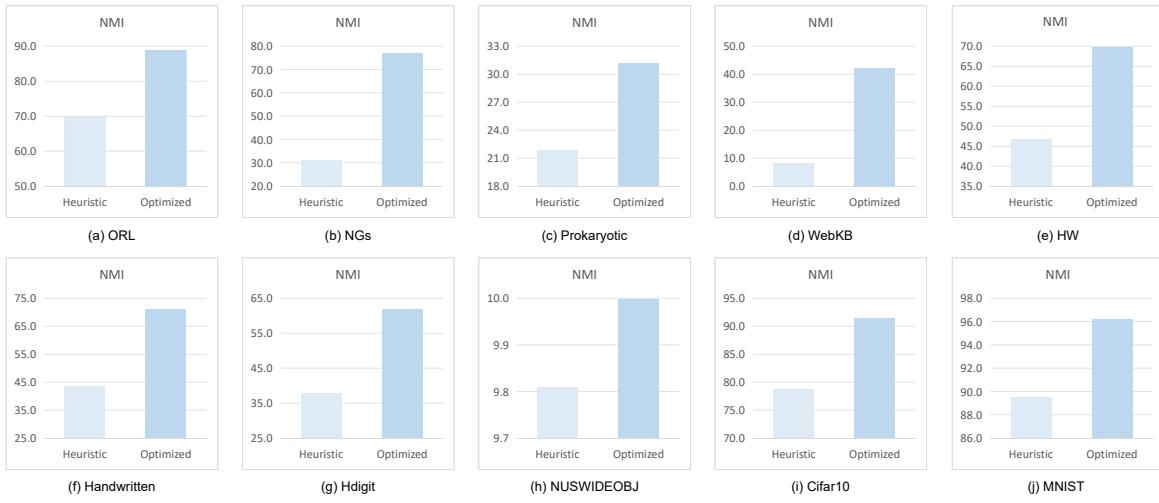
Fig. 10. Clustering performance comparison between traditional heuristic anchor strategy and our optimized anchor strategy.
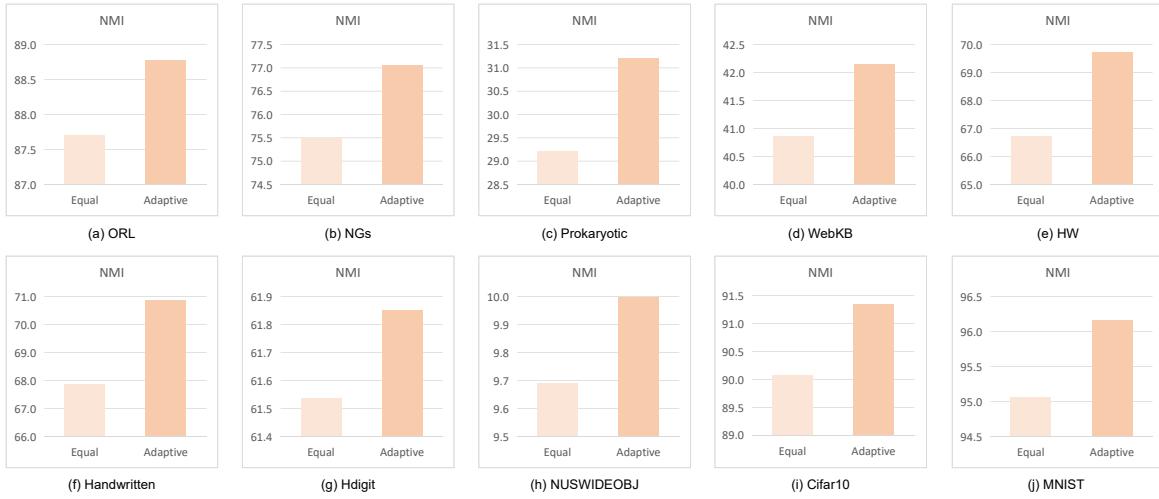

Fig. 11. Clustering performance comparison between traditional equal weighting and our adaptive weighting.

number of samples in the i-th cluster while $n_{i,j}$ is the number of samples, which falls into the j-th cluster, in the i-th cluster. $c_{i,j} = n_{i,j}/n_i$ and $c_i = \max\{c_{i,j}\}$. $\mathrm{RE} = \mathrm{tp}/(\mathrm{fn} + \mathrm{tp})$ and $\mathrm{PR} = \mathrm{tp}/(\mathrm{fp} + \mathrm{tp})$ where fn means false-negative, fp means false-positive, and tp means true-positive.

C. Results Analysis

We present the clustering results of these twelve IMC algorithms under IDR $10\% \sim 90\%$ in Figs. 4, 5, and the average results in Table I respectively. In Table I, the ranking No.1 results are emphasized in black while the ranking No.2 results are underlined. The notation MOE represents the memory overflow error. From Figs. 4, 5 and Table I, we have some observations:

1) Our devised CCA-IMC has the ability to produce the best clustering results in most situations. Taking algorithm CBG for example, which is with a superior performance against other IMC methods, our CCA-IMC makes the average improvements of 1.2%, 4.89%, 2.94%, 23.54%, 3.71%, 4.66%, 8.85%, 0.09%, 0.31%

and 0.13% than CBG in NMI on these incomplete datasets respectively. In metrics ACC and PUR as well as FSC, CCA-IMC can also exhibit clear superiorities. These demonstrate that our CCA-IMC is effective.

2) The standard deviation of the compared IMC algorithms is generally larger than ours in most cases. For instance, on datasets Handwritten and Hdigit, the deviation values of ACC obtained by DAIMC, EEIMVC and FLSD are 5.56%, 5.28%, 5.75%, 2.63%, 4.48% and 4.00% higher than ours, respectively. Among those techniques, CCA-IMC typically enjoys small standard deviation in terms of the four metrics. These results suggest that our CCA-IMC is relatively robust to diverse incomplete cases compared to other IMC algorithms.

3) With the increase of IDR, the clustering performance of almost all methods is gradually decreasing. When the incompleteness ratio of dataset is fairly high, all methods demonstrate relatively undesirable clustering results. This gives an empirical evidence that IMC is more intractable than (complete) multiview clustering.
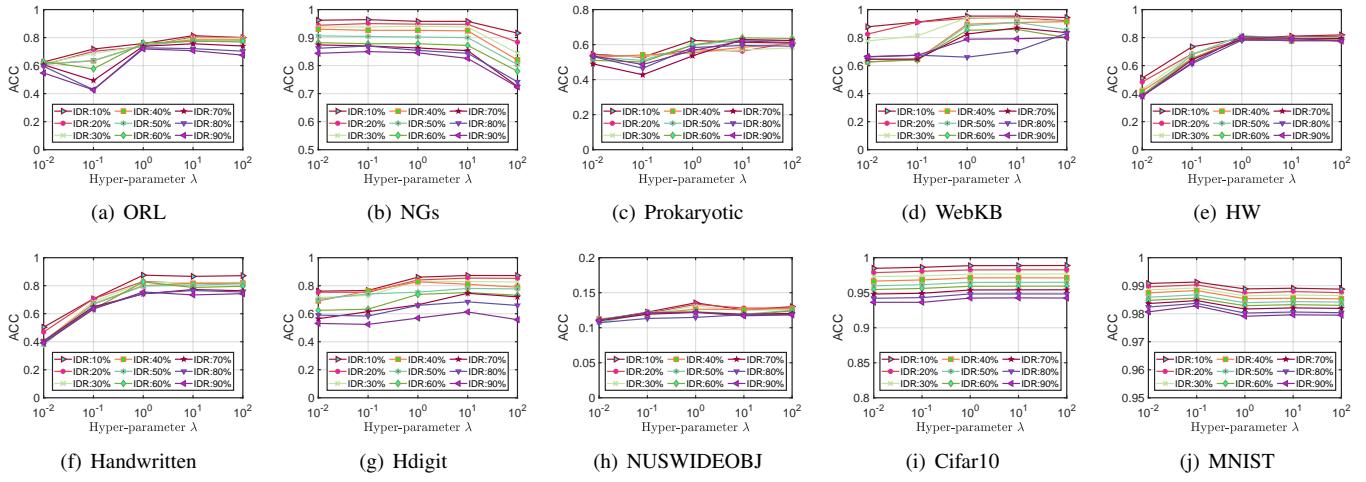
Fig. 12. Sensitivity of CCA-IMC to the hyper-parameter λ.

In addition, compared to these strong IMC competitors, our CCA-IMC still receives encouraging performance, even under quite large incomplete ratio. This illustrates that our proposed CCA-IMC is more suitable for IMC.

4) Compared to methods like MKKM-IK, UEAF, IK-MKC and EEIMVC which adopt subspace or kernel strategy to handle the IMC problem, our method based on bi-partite graph mechanism outperforms them consistently. CBG projects anchors into the same dimension and thereby produces the consensus bipartite graph, however, projection operation will lead to information loss and deteriorate the diversity of each view, bringing about sub-optimal performance. These experimental results state that our consensus bipartite graph constructed by view-specific anchors is more preferable.

5) Although PIMVC achieves superior clustering results to ours on Hdigit, it fails to properly work on datasets Prokaryotic, HW and Handwritten that are only with small size, which is due to the fact that it requires the data dimension to be greater than the number of clusters. This limitation causes it to suffer from relatively narrow applicability. DAIMC makes slightly better results on NUSWIDEOBJ than us, possibly because it introduces the $\ell_{2,1}$ regularizer to decrease the influence of partial samples, and enforces the basis matrices to be aligned so as to exploit the global information inside data.

6) On some large-scale datasets like Cifar10 and MNIST, methods DAIMC, EEIMVC, FLSD and etc encounter the memory overflow error due to the corresponding expensive space complexity. This highlights that they are powerless in face of large-scale IMC problems. As a comparison, our method can not only perform normally but generate impressive results on these datasets, which illustrates that our CCA-IMC is source-saving and more practical for IMC tasks.

In summary, our CCA-IMC wins the preferable performance and is more suitable for IMC tasks even on large-scale datasets or/and under diverse incomplete cases in comparison with other strong IMC algorithms. For the purpose of further validating the advantages of the devised CCA-IMC, we draw the learned representation of these IMC methods using t-SNE visualization algorithm, as presented in Fig. 6. One can observe that our CCA-IMC enjoys more distinguishable cluster structure, which means that CCA-IMC can effectively partition data into different groups where samples in the same one group have quite similar characteristics while in different groups having fairly distinct characteristics.

D. Execution Time

In Section III-C we have proven theoretically that CCA-IMC has linear computational complexity. For the sake of verifying its efficiency, we count the execution time of these twelve IMC algorithms on the mentioned-above datasets, and show the results in Fig. 7. We see that

1) As the number of samples increases, the execution time of these IMC algorithms is gradually increasing. Especially on dataset Hdigit, all algorithms take more time. When the sizes of datasets are close to each other, larger dimension dataset consumes more time.

2) MIC suffers from the slowest running speed, which is mainly due to its cubic time complexity. Other methods like MKKM-IK, EEIMVC, FLSD and etc report memory overflow error on datasets NUSWIDEOBJ, Cifar10 and MNIST. These demonstrate that they are not applicable to large-scale problems.

3) Our method is able to run on all datasets, and is faster on HW, Handwritten, Cifar10 and etc than the rest of IMC algorithms, which clearly illustrates our wide scalability and high efficiency. Method CBG requires slightly less time compared to ours on certain datasets, however, it can not take advantage of complementary features between views, causing relatively limited performance.

In short, both theory and experiments provide a great demonstration that our CCA-IMC enjoys fewer execution time, and is more appropriate for handling IMC tasks.

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3312979

13
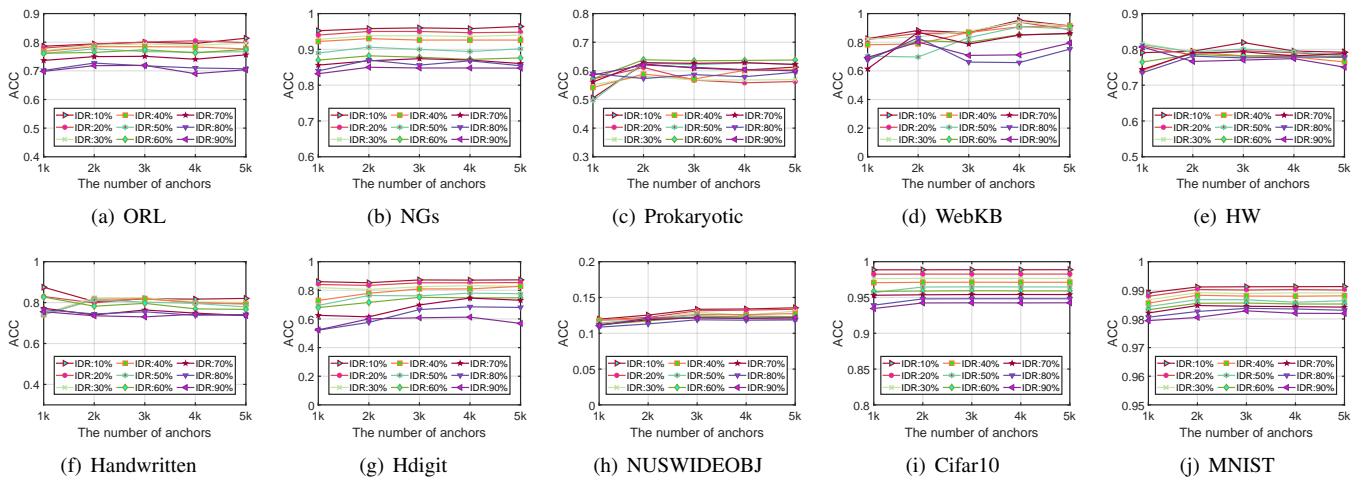
Fig. 13. Sensitivity of CCA-IMC to the number of anchors.
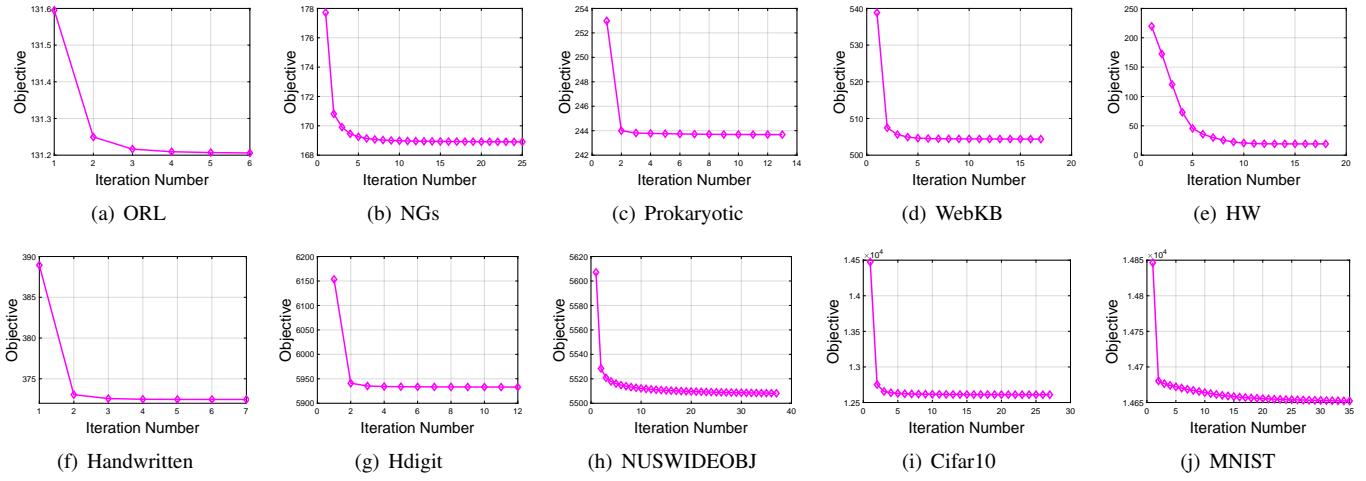
Fig. 14. The objective of CCA-IMC on ten datasets.

E. Ablation study for permutation strategy

For verifying that anchors are correctly matched through our permutation strategy, we restore the view-specific bipartite graphs and the permutation matrices between different views based on the learned consensus bipartite graph and view-specific permutation matrices, as presented in Fig. 8. One can observe that by the acquired permutation relationship, anchors on different views are correctly matched. For example, anchor 8 on view 6 is mismatched to anchor 3 on view 1. After multiplying view 6 with Permutation1-6, anchor 8 on view 6 will become anchor 3. Additionally, we also conduct a series of experiments without employing the permutation strategy while keeping all other conditions the same. The contrast results are reported in Fig. 9. It suggests that the permutation strategy indeed makes clear performance increase. Figs. 8 and 9 give a vivid illustration that our permutation strategy is effective.

F. Ablation study for optimizable anchor strategy

In this paper, we generate anchors by optimization rather than by the heuristic strategy like k-means. This makes anchors able to negotiate with the bipartite graph so as to promote each other. We organize some ablation experiments

to exhibit the superiority of optimizable anchor strategy. The experimental results are summarized in Fig. 10. From this figure, one can get that the results based on anchor optimization have obvious advantages. These results give an empirical evidence that our anchor optimization strategy is effective and conductive to bringing performance improvement.

G. Ablation study for adaptive view weighting strategy

Different from previous methods that treat each view equally, we associate each view with a weight variable to automatically measure its importance. This is primarily because the feature dimensions between views are generally diverse, and equal weighting could weaken the contribution of certain views, leading to limited performance. To validate its effectiveness, we design ablation experiments in terms of view weights, and present the results in Fig. 11. Evidently, adaptive weighting receives consistent improvement compared to equal weighting. This confirms that introducing adaptive view weighting strategy is beneficial for decreasing the impact of unbalanced features and improving the clustering performance.

H. Parameter Sensitivity

The hyper-parameter λ plays a role in balancing reconstruction error and bipartite graph generation. To investigate its effect, we count the clustering performance of CCA-IMC under different λ, as suggested in Fig. 12. We can observe that most of our results are superior to that of existing algorithms, and within a wide hyper-parameter scope, the results are still worth having and also do not fluctuate dramatically. These indicate that CCA-IMC to some extent is robust to the variations of hyper-parameter λ. Besides, CCA-IMC's performance could vary with the different number of anchors. To explore the sensitivity of CCA-IMC with respect to the number of anchors, we conduct some contrast experiments with diverse setting of the anchor number, as shown in Fig. 13. One can see that the performance curves in most situations are stable, which illustrates that the number of anchors do not bring dramatical influence on CCA-IMC's performance.

I. Algorithm Convergence

In addition to producing satisfactory results, an algorithm also needs to own the convergence property. We have demonstrated CCA-IMC's convergence from theory in Section III-C, and here we are devoted to validating its convergence from experiments. To this end, we record the objective values of CCA-IMC on the datasets mentioned earlier, and draw them in Fig. 14. One can see that as the iteration goes on, the objective value is monotonically decreasing and gradually reaching into a stable state. These results experimentally illustrate that the proposed CCA-IMC is with convergence characteristic.

V. Conclusion

In this paper, to address the anchor mismatching problem in incomplete multiview clustering, we design a named CCA-IMC algorithm, which transforms anchors directly in their original dimension space according to the shared graph structure. Different from previous separated mechanism, CCA-IMC owns a joint-optimization mechanism that alternatively updates anchors and permutation matrices as well as the consensus bipartite graph, boosting the clustering quality. Also, theoretical and experimental evidences illustrate that CCA-IMC has linear complexities and is applicable to large-scale tasks. In future work, we will attempt to combine binary clustering [70] and large-scale graph learning together so as to bring further improvements in clustering results and faster running speed for IMC problems.

References

[1] Y. Chen, X. Xiao, C. Peng, G. Lu, and Y. Zhou, "Low-rank tensor graph learning for multi-view subspace clustering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 1, pp. 92–104, 2021.
[2] M. Lan, M. Meng, J. Yu, and J. Wu, "Generalized multi-view collaborative subspace clustering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 3561–3574, 2021.
[3] S. Peng, J. Yin, Z. Yang, B. Chen, and Z. Lin, "Multiview clustering via hypergraph induced semi-supervised symmetric nonnegative matrix factorization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
[4] B. Peng, J. Lei, H. Fu, C. Zhang, T.-S. Chua, and X. Li, "Unsupervised video action clustering via motion-scene interaction constraint," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 131–144, 2018.
[5] X. Dong, L. Liu, L. Zhu, Z. Cheng, and H. Zhang, "Unsupervised deep k-means hashing for efficient image retrieval and clustering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3266–3277, 2020.
[6] J. Lei, X. Li, B. Peng, L. Fang, N. Ling, and Q. Huang, "Deep spatial-spectral subspace clustering for hyperspectral image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2686–2697, 2020.
[7] B. Yang, X. Zhang, Z. Lin, F. Nie, B. Chen, and F. Wang, "Efficient and robust multiview clustering with anchor graph regularization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6200–6213, 2022.
[8] Z. Zhu and Q. Gao, "Semi-supervised clustering via cannot link relationship for multiview data," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 12, pp. 8744–8755, 2022.
[9] G. Jiang, J. Peng, H. Wang, Z. Mi, and X. Fu, "Tensorial multi-view clustering via low-rank constrained high-order graph learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5307–5318, 2022.
[10] T. Wang, W. W. Ng, J. Li, Q. Wu, S. Zhang, C. Nugent, and C. Shewell, "A deep clustering via automatic feature embedded learning for human activity recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 1, pp. 210–223, 2021.
[11] Z. Chen, Z. Cui, C. Zhang, J. Zhou, and Y. Liu, "Dual clustering co-teaching with consistent sample mining for unsupervised person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
[12] S. Li, M. Yuan, J. Chen, and Z. Hu, "Adadc: Adaptive deep clustering for unsupervised domain adaptation in person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 3825–3838, 2021.
[13] Y. Jia, H. Liu, J. Hou, S. Kwong, and Q. Zhang, "Multi-view spectral clustering tailored tensor low-rank representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4784–4797, 2021.
[14] H. Yang, Q. Gao, W. Xia, M. Yang, and X. Gao, "Multiview spectral clustering with bipartite graph," *IEEE Transactions on Image Processing*, vol. 31, pp. 3591–3605, 2022.
[15] S. Zhou, X. Liu, J. Liu, X. Guo, Y. Zhao, E. Zhu, Y. Zhai, J. Yin, and W. Gao, "Multi-view spectral clustering with optimal neighborhood laplacian matrix," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 6965–6972.
[16] S. Zhou, X. Liu, M. Li, E. Zhu, L. Liu, C. Zhang, and J. Yin, "Multiple kernel clustering with neighbor-kernel subspace segmentation," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 4, pp. 1351–1362, 2020.
[17] X. Liu, "Simplemkkm: Simple multiple kernel k-means," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
[18] X. Li, Y. Sun, Q. Sun, and Z. Ren, "Consensus cluster center guided latent multi-kernel clustering," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.
[19] F. Nie, W. Zhu, and X. Li, "Unsupervised large graph embedding based on balanced and hierarchical k-means," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 2008–2019, 2020.
[20] K. Zeng, M. Ning, Y. Wang, and Y. Guo, "Hierarchical clustering with hard-batch triplet loss for person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13657–13665.
[21] J. Liu, X. Liu, S. Wang, S. Zhou, and Y. Yang, "Hierarchical multiple kernel clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 10, 2021, pp. 8671–8679.
[22] X. Li, H. Zhang, R. Wang, and F. Nie, "Multiview clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 330–344, 2022.
[23] X. Shu, X. Zhang, Q. Gao, M. Yang, R. Wang, and X. Gao, "Self-weighted anchor graph learning for multi-view clustering," *IEEE Transactions on Multimedia*, 2022.
[24] B. Yang, X. Zhang, F. Nie, F. Wang, W. Yu, and R. Wang, "Fast multi-view clustering via nonnegative and orthogonal factorization," *IEEE Transactions on Image Processing*, vol. 30, pp. 2575–2586, 2020.
[25] Y. Hu, Z. Song, B. Wang, J. Gao, Y. Sun, and B. Yin, "Akm 3 c: Adaptive k-multiple-means for multi-view clustering," *IEEE Transactions on*

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3312979

15

Circuits and Systems for Video Technology, vol. 31, no. 11, pp. 4214–4226, 2021.

[26] Y. Qin, H. Wu, X. Zhang, and G. Feng, "Semi-supervised structured subspace learning for multi-view clustering," *IEEE Transactions on Image Processing*, vol. 31, pp. 1–14, 2021.

[27] M. I. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, pp. 849–856, 2002.

[28] H. Wang, L. Zong, B. Liu, Y. Yang, and W. Zhou, "Spectral perturbation meets incomplete multi-view data," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 3677–3683.

[29] X. Liu, M. Li, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel k-means with incomplete kernels," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[30] S.-Y. Li, Y. Jiang, and Z.-H. Zhou, "Partial multi-view clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1, 2014.

[31] Y. Wang, D. Chang, Z. Fu, J. Wen, and Y. Zhao, "Incomplete multi-view clustering via cross-view relation transfer," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 1, pp. 367–378, 2022.

[32] J. Wen, Y. Xu, and H. Liu, "Incomplete multiview spectral clustering with adaptive graph learning," *IEEE transactions on cybernetics*, vol. 50, no. 4, pp. 1418–1429, 2020.

[33] W. Shao, L. He, and P. S. Yu, "Multiple incomplete views clustering via weighted nonnegative matrix factorization with $\ell_{\{2,1\}}$ regularization," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2015, pp. 318–334.

[34] X. Fang, Y. Hu, P. Zhou, and D. O. Wu, "V^3h: View variation and view heredity for incomplete multiview clustering," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 3, pp. 233–247, 2020.

[35] J. Wen, Z. Zhang, Y. Xu, B. Zhang, L. Fei, and H. Liu, "Unified embedding alignment with missing views inferring for incomplete multi-view clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 5393–5400.

[36] H. Zhao, H. Liu, and Y. Fu, "Incomplete multi-modal visual data grouping," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 2392–2398.

[37] M. Hu and S. Chen, "Doubly aligned incomplete multi-view clustering," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 2262–2268.

[38] S. Wang, X. Liu, L. Liu, W. Tu, X. Zhu, J. Liu, S. Zhou, and E. Zhu, "Highly-efficient incomplete large-scale multi-view clustering with consensus bipartite graph," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9776–9785.

[39] J. Wen, Z. Zhang, Z. Zhang, L. Fei, and M. Wang, "Generalized incomplete multiview clustering with flexible locality structure diffusion," *IEEE transactions on cybernetics*, vol. 51, no. 1, pp. 101–114, 2021.

[40] J. Wen, Z. Zhang, Y. Xu, and Z. Zhong, "Incomplete multi-view clustering via graph regularized matrix factorization," in *Proceedings of the European conference on computer vision workshops*, 2018.

[41] J. Wen, H. Sun, L. Fei, J. Li, Z. Zhang, and B. Zhang, "Consensus guided incomplete multi-view spectral clustering," *Neural Networks*, vol. 133, pp. 207–219, 2021.

[42] M. Li, J. Xia, H. Xu, Q. Liao, X. Zhu, and X. Liu, "Localized incomplete multiple kernel k-means with matrix-induced regularization," *IEEE Transactions on Cybernetics*, 2021.

[43] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, and W. Gao, "Multiple kernel k-means with incomplete kernels," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[44] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, and E. Zhu, "Efficient and effective regularized incomplete multi-view clustering," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 43, no. 08, pp. 2634–2646, 2021.

[45] 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 transactions on pattern analysis and machine intelligence*, vol. 41, no. 10, pp. 2410–2423, 2018.

[46] Y. Zhang, X. Liu, S. Wang, J. Liu, S. Dai, and E. Zhu, "One-stage incomplete multi-view clustering via late fusion," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2717–2725.

[47] L. Li, Z. Wan, and H. He, "Incomplete multi-view clustering with joint partition and graph learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[48] J. Wen, K. Yan, Z. Zhang, Y. Xu, J. Wang, L. Fei, and B. Zhang, "Adaptive graph completion based incomplete multi-view clustering," *IEEE Transactions on Multimedia*, vol. 23, pp. 2493–2504, 2021.

[49] J. Wen, Z. Zhang, Z. Zhang, L. Zhu, L. Fei, B. Zhang, and Y. Xu, "Unified tensor framework for incomplete multi-view clustering and missing-view inferring," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 11, 2021, pp. 10 273–10 281.

[50] N. Liang, Z. Yang, and S. Xie, "Incomplete multi-view clustering with sample-level auto-weighted graph fusion," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[51] C. Tang, X. Zheng, W. Zhang, X. Liu, X. Zhu, and E. Zhu, "Unsupervised feature selection via multiple graph fusion and feature weight learning," *Science China Information Sciences*, vol. 66, no. 5, pp. 1–17, 2023.

[52] Z. Li, C. Tang, X. Liu, X. Zheng, W. Zhang, and E. Zhu, "Consensus graph learning for multi-view clustering," *IEEE Transactions on Multimedia*, vol. 24, pp. 2461–2472, 2021.

[53] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, and E. Zhu, "Unified one-step multi-view spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 6449–6460, 2022.

[54] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "Completer: Incomplete multi-view clustering via contrastive prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 174–11 183.

[55] J. Xu, C. Li, Y. Ren, L. Peng, Y. Mo, X. Shi, and X. Zhu, "Deep incomplete multi-view clustering via mining cluster complementarity," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8761–8769.

[56] C. Zhang, Y. Cui, Z. Han, J. T. Zhou, H. Fu, and Q. Hu, "Deep partial multi-view learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 5, pp. 2402–2415, 2020.

[57] Q. Wang, Z. Ding, Z. Tao, Q. Gao, and Y. Fu, "Partial multi-view clustering via consistent gan," in *IEEE International Conference on Data Mining*. IEEE, 2018, pp. 1290–1295.

[58] J. Wen, Z. Zhang, Z. Zhang, Z. Wu, L. Fei, Y. Xu, and B. Zhang, "Dimcnet: Deep incomplete multi-view clustering network," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 3753–3761.

[59] J. Cui, Y. Fu, C. Huang, and J. Wen, "Low-rank graph completion-based incomplete multiview clustering," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[60] S. Wang, X. Liu, L. Liu, S. Zhou, and E. Zhu, "Late fusion multiple kernel clustering with proxy graph refinement," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[61] M. Yang, Y. Li, P. Hu, J. Bai, J. Lv, and X. Peng, "Robust multi-view clustering with incomplete information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1055–1069, 2022.

[62] X.-L. Li, M.-S. Chen, C.-D. Wang, and J.-H. Lai, "Refining graph structure for incomplete multi-view clustering," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[63] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 4412–4419.

[64] Q. Qiang, B. Zhang, F. Wang, and F. Nie, "Fast multi-view discrete clustering with anchor graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9360–9367.

[65] X. Lu and S. Feng, "Structure diversity-induced anchor graph fusion for multi-view clustering," *ACM Transactions on Knowledge Discovery from Data*, 2022.

[66] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, and J. Yin, "Multi-view clustering via late fusion alignment maximization," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 3778–3784.

[67] S. Deng, J. Wen, C. Liu, K. Yan, G. Xu, and Y. Xu, "Projective incomplete multi-view clustering," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[68] L. Sun, J. Wen, C. Liu, L. Fei, and L. Li, "Balance guided incomplete multi-view spectral clustering," *Neural Networks*, 2023.

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

[70] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1774–1782, 2018.

Shengju Yu is currently pursuing the Ph.D. degree in National University of Defense Technology (NUDT), China. He received the M.Sc. degree in information and communication engineering from Huazhong University of Science and Technology, Wuhan, China. His current research interests include neural architecture search, trustworthy artificial intelligence, unsupervised multi-view learning, and multi-kernel learning.

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 100 peer-reviewed papers, including TPAMI, TIP, TCSVT, TNNLS, TKDE, TCYB, ICML, NeurIPS, CVPR, AAAI, and IJCAI. His main research interests include pattern recognition, image processing, machine vision. He was awarded the China National Excellence Doctoral Dissertation.

Siwei Wang is currently assistant research professor with the Intelligent Game and Decision Lab, China. He has published several papers and served as a AC/SPC/Reviewer in top journals and conferences, such as TPAMI, TKDE, TNNLS, TIP, TCYB, TMM, ICML, NeurIPS, CVPR, ICCV, ECCV, AAAI, and IJCAI. His current research interests include multi-modal, unsupervised multi-view learning, scalable clustering, and deep unsupervised learning.

Xinwang Liu 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 100 peer-reviewed papers, including TPAMI, TIP, TKDE, TNNLS, TMM, TIFS, TCYB, 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 IEEE TNNLS, IEEE TCYB and Information Fusion journal. More information can be found at https://xinwangliu.github.io/

Yi Wen is recommended for admission to the National University of Defense Technology (NUDT) as a master's student with excellent grades and competition awards. He is working hard to pursue his master degree. He has authored or coauthored papers in journals and conferences, such as the TNNLS, TKDE, and ACM MM. His current research interests include multiple-view learning, scalable kernel k-means and graph representation learning.

Ziming Wang received his BSc in applied mathematics from University of Science and Technology of China, in 2018, and the PhD degree in computational mathematics at Academy of Mathematics and Systems Science, Chinese Academy of Science, Beijing, China, in 2023. He is currently a engineer at China Academy of Aerospace Science and Innovation. His research interests include machine learning, decision making and numerical methods to partial differential equations.

Zhigang Luo received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 1981, 1993, and 2000, respectively. He has published more than 100 peer-reviewed papers, including TIP, TCSVT, TSP, TNNLS, TKDE, PR, ACM MM, IJCAI, AAAI. He is currently a Professor with the College of Computer, National University of Defense Technology. His current research interests include machine learning, computer vision, and bioinformatics.