Neurocomputing 515 (2023) 79–88

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Cross-view graph matching for incomplete multi-view clustering

Jing-Hua Yang [a], Le-Le Fu [b], Chuan Chen [c,*], Hong-Ning Dai [d], Zibin Zheng [c]

[a] The Faculty of Information Technology, Macau University of Science and Technology, Macau, China
[b] The School of Systems Science and Engineering, Sun Yat-Sen University, Guangzhou, China
[c] The School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China
[d] The Department of Computer Science, Hong Kong Baptist University, Hong Kong, China

ARTICLE INFO

*Article history:*
Received 13 February 2022
Revised 12 June 2022
Accepted 1 October 2022
Available online 18 October 2022
Communicated by Zidong Wang

*Keywords:*
Incomplete multi-view clustering
Consensus learning
Graph learning
Cross-view graph matching

ABSTRACT

Multi-view clustering (MVC) focuses on adaptively partitioning data from diverse sources into the respective groups and has been widely studied under the assumption of complete data. However, real-world applications often encounter a more realistic *incomplete multi-view clustering* (IMVC) problem, where data samples are missing in certain views. There are two challenges in IMVC: 1) how to reduce the impact of the missing instances; 2) how to effectively extract the consistent information to cluster the multi-view data. To address the challenges, we propose an adaptive graph learning framework for IMVC, which optimizes the missing information to fit the intrinsic structure of each view and clusters the multi-view data by cross-view graph matching. The proposed method mainly consists of three steps. Firstly, owing to the outstanding performance of the intrinsic structure of data, we adapt it to complete the missing data of each view. Secondly, the connection graph of each view from a projection space is adaptively constructed wherein the data points are connected if and only if they belong to the same cluster. Thirdly, we further introduce a cross-view graph matching strategy to appropriately utilize complementary multi-views information and preserve view-specific semantic information. We develop an iterative algorithm for solving the proposed model. Numerical experiments on several standard datasets demonstrate the effectiveness of the proposed method.



## 1. Introduction

In the fields of machine learning, multi-view clustering (MVC) has received increasing attention and achieved success in various applications, such as web page classification [1,2], speech recognition [3], and disease diagnosis [4]. The main idea of MVC is to cluster the data coming from multiple channels or with multiple modalities [5–7]. Many methods have been researched for solving MVC, for example, the multi-view *k*-means clustering [8–10], the co-regularized multi-view spectral clustering [11–14], and the matrix factorization-based clustering [15–17]. These studies on multi-view learning mainly focus on the clustering of complete multi-view data.

However, due to temporary failures of data collectors or human errors, many practical applications often encounter the absence of partial views among samples, thereby forming incomplete multi-view data [18–22]. For example, a text description of an item to be clustered may be missing. Due to the lack of data for some views, some rows or columns of the data matrix are missing. As a result, the traditional matrix completion methods cannot effectively fill the missing entries. Moreover, most existing MVC algorithms that are designed for complete data thus cannot directly and efficiently handle the incomplete cases. Therefore, incomplete multi-view clustering (IMVC) came into being with an aim of reducing the impact of missing samples and effectively exploiting multi-view information to achieve an applicable clustering.

In recent years, various efforts are devoted to solving the IMVC problem. We roughly divide these methods into two categories. The one category is called the "two-stage" methods [23–25]. In these methods, incomplete multi-view data is first filled with missing values through some data preprocessing strategies. Then, some standard MVC methods are applied to the preprocessed data to obtain the final clustering results. The common data preprocessing strategies include *k*-nearest-neighbor completion, mean value completion, and zero-filling. However, with the increase of the missing views, these simple filling methods may cause a large deviation and also lose the semantic information implied by the missing data [26,20]. To simultaneously infer the missing data and obtain the consistent representation, the second category of "one-stage" methods is formed. A number of IMVC methods based

* Corresponding author.
*E-mail addresses:* chenchuan@mail.sysu.edu.cn (C. Chen), hndai@ieee.org (H.-N. Dai), zhzibin@mail.sysu.edu.cn (Z. Zheng).

https://doi.org/10.1016/j.neucom.2022.10.007
0925-2312/

on the "one-stage" framework have emerged, such as matrix decomposition-based methods [4,27,28], spectral clustering-based methods [29,30,26,31], and neural network-based methods [32–34]. These methods aim to learn the consensus graph/representation to find the common structure of multiple views. However, the value of the learning representation may be different in each view. Meanwhile, some useful information may be lost when a common representation is forced. In summary, the prevalent IMVC methods still suffer from the following limitations: (1) the missing view information has not been well restored to meet the intrinsic structure of data; (2) the missing view information has not been fully utilized to guide the consensus representation learning; (3) the common structure learning of multiple views with the highest probability consequently incurs the loss of semantic information.

To address the above issues, in this work, we propose a novel IMVC method called *Cross-view Graph Matching for Incomplete Multi-view Clustering (CGMIMC)*. In Fig. 1, we illustrate the overall framework of our CGMIMC. On the whole, CGMIMC puts the learning of missing data into an optimization process so that the missing instances meet the intrinsic structure of each view. Meanwhile, the graph learning strategy is used to convert the representations of recovered views into the compact graph connection representations that can capture the pairwise relationship. Therefore, both non-missing views and missing views can jointly guide the consensus representation learning. Then, we minimize the inconsistency between the pair of connected graphs rather than that among all graphs to reach the view consensus and preserve view-specific semantic information. The main contributions are as follows:

- We optimize the missing views to meet the intrinsic structure of each view, which can reduce the impact of missing data and guide the consensus learning on clustering.
- We use cross-graph matching to achieve view consensus and preserve view-specific semantic information, where the view representation of possible changes is converted to a stable graph connection representation.
- We design an efficient iterative algorithm to solve the proposed model. Numerical experiments on multiple datasets show the effectiveness of the proposed CGMIMC.

The rest of this paper has the following structure: Section 2 reviews the related representative clustering methods. Section 3 presents the proposed CGMIMC, including missing data completion and cross-view graph matching. Section 4 proposes an iterative algorithm to solve CGMIMC. The analysis and discussions of numerical experiments are presented in Section 5. Finally, Section 6 summarizes this work.

## 2. Related work

In this second, we review some representative studies related to single-view clustering, complete multi-view clustering, and incomplete multi-view clustering methods.

### 2.1. Single-view clustering

The single-view clustering method only considers collecting data from a single source [35–37]. In order to cluster high-dimensional data, sparse representation-based [38] and low-rank representation-based [39] subspace clustering methods were studied. Moreover, Peng et al. used graph regularization [40] and structured autoencoders [41] for subspace clustering. With the development of deep learning, the recurrent graph neural networks clustering method [34] and the deep fusion clustering network [42] were also proposed. Although these methods are effective for single-view data, the observed data usually come from diverse domains, each of which denotes a specific perspective. The above single-view clustering methods do not apply the complementary multi-view information of the observed data. Therefore, the single-view clustering algorithm is less effective than the MVC algorithm in handling the multi-view data.

### 2.2. Complete multi-view clustering

MVC has been intensively investigated [9,43,12,44,45]. The complete MVC methods require that all views of the observed samples are available. To find the common latent structure among all views, Wang et al. [46] used a views-agreement constraint to keep the consistency of all views. Meanwhile, Xia et al. [47] learned a common low-rank graph representation from multiple similarity graphs based on all views. Wang et al. [48,49] applied the divergence constraint term to coordinate all views to be agreeable by using the low-rank matrix factorization. Similarly, to minimize disagreement between different views, Zhan et al. [50] proposed a consensus graph clustering method. Wang et al. [51] fused the data

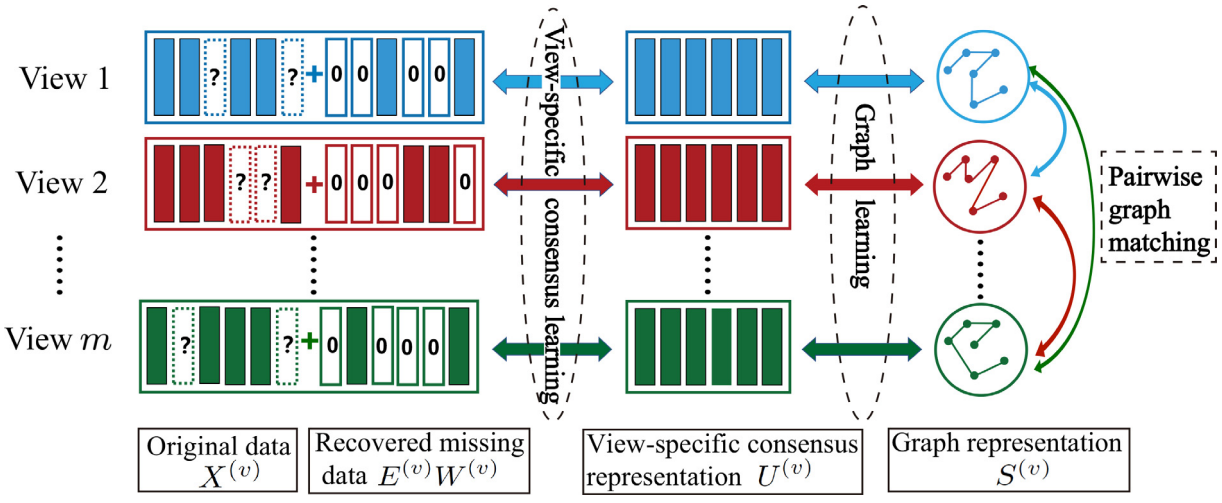


**Fig. 1.** The overall framework of the proposed method CGMIMC. Given a collection of data points with multiple views, $\{X^{(1)}, X^{(2)}, \ldots, X^{(m)}\}$, the proposed method optimizes the missing data to form the complete data $X^{(v)} + E^{(v)}W^{(v)}$, so that non-missing information and missing information can simultaneously guide the consensus learning of each view. Then, the connection graph $S^{(v)}$ of each view is adaptively constructed to transform the view representation and preserve the pairwise relationship of each view. Finally, we use the cross-view graph matching strategy to minimize the difference between any two connection graphs to properly achieve view consensus.

graph matrices of all views into a unified matrix. Tang et al. [7] applied the cross-view graph diffusion way to obtain a unified graph for data clustering. However, real-world applications often fail to meet the assumptions that each example of data appears in all views. In fact, every view may suffer from data missing. Therefore, traditional complete MVC methods are limited in implementing practical applications.

### 2.3. Incomplete multi-view clustering

In recent years, IMVC methods have received extensive attention for processing the incomplete data with multiple features. For instance, Li et al. [4] presented a partial multi-view clustering (PVC) algorithm to handle incomplete multi-view data by learning the latent subspace. This approach focuses on the two incomplete views. Hu et al. [20] extended PVC to IMVC by exploring instance missing information and aligning the clustering centers among different views simultaneously. Moreover, by filling the missing entries with the average value in each view, Shao et al. [23] proposed a clustering approach for incomplete multi-view data. However, the simple filling method has poor performance with a large number of missing data samples. Based on the common structure among all views, Wen et al. [21] used graph regularized matrix factorization to solve IMVC by simultaneously exploiting the local and global information among views. Further, to solve the nonlinear relations between the unified representation and data of different views, Zhuge et al. [22] learned low-dimensional representations of existing instances and a shared label matrix. To explore the hidden information of missing instances, Liu et al. [52] proposed an efficient and effective IMVC method based on multi-kernel $k$-means clustering, where the absent views and the clustering result are iteratively updated until convergence. Wen et al. [53] jointly learned the graph completion and common representation for IMVC. In particular, they used the similarity information of other views to infer the missing views. Meanwhile, the consensus representation is learned by exploiting shared information across all views. Li et al. [54] applied the partition space and graph learning for IMVC. To be specific, they first learned the view-specific local partition with incomplete data, and then integrated the partitions of all views to produce the unified partition and obtain the consensus graph.

The differences between the proposed method and related works are as follows:

- The single-view clustering method is only suitable for processing data with one feature and cannot directly process multi-view data. Differently, the proposed method can fully exploit the complementary information of multi-view data for data clustering.
- The complete multi-view clustering methods require that all views of the observed samples are available. However, in practical applications, observation samples may be missing from some views. By contrast, our method can handle the challenging and realistic incomplete multi-view data.
- Most existing IMVC methods [4,20–23] either fill the missing entries with the average value, or directly explore the common structure of incomplete views. By contrast, the proposed method optimizes the missing views to meet the intrinsic structure of each view, thereby reducing the impact of missing samples for clustering. Compared with the state-of-the-art IMVC methods [52–54], instead of forcing all representations to be consistent, we adopt the cross-graph matching strategy to mine the view consensus while preserving view-specific semantic information.

## 3. Proposed method

In the following, we present the proposed method CGMIMC with three tasks, including the view-specific consensus learning, the graph learning, and the cross-view graph matching.

**Motivations**. Most existing MVC methods have an underlying assumption that each view contains all instances. However, the available multi-view data usually is not complete. For example, in document clustering (e.g., a document has multi-language versions), some documents may not have all translated versions. On the other hand, most MVC and IMVC methods focus on finding the common features with the highest probability. But for incomplete views, each view may be very different from other views. In this way, some semantic information may be lost. Therefore, incomplete multi-view data poses two challenges: *Challenge 1*) how to reduce the impact of missing instances; *Challenge 2*) how to effectively extract the complementary and consistent information to obtain better clustering results. The solution to Challenge 1 can essentially address the aforementioned limitations (1) and (2) of IMVC methods while the solution to Challenge 2 can tackle limitation (3).

**Problem formulation**. We consider the problem of clustering a set of $n$ data points. For MVC, we have $m$ data views denoted by $\left\{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots, \boldsymbol{X}^{(m)}\right\}$, where $\boldsymbol{X}^{(v)} \in \mathbb{R}^{d_v \times n}$ and $d_v$ denotes the feature dimension of the $v$th view. For each $\boldsymbol{X}^{(v)}$, the values of missing entries are filled with 0. In the following, we use the bold upper case letter $\boldsymbol{X}$, the bold lower case letter $\mathbf{x}$, and the lower case letter $x$ to denote the matrix, the vector, and the scalar, respectively. We use the $\mathbf{x}_{i,:}$ and $\mathbf{x}_{:,j}$ to denote the $i$th row and the $j$th column of $\boldsymbol{X}$, respectively. The $(i,j)$th element of $\boldsymbol{X}$ is denoted as $\boldsymbol{X}_{i,j}$ or $x_{i,j}$. For convenience, we summarize the important notations used in the proposed method in Table 1.

### 3.1. View-specific consensus learning

To address *Challenge 1*, some IMVC methods only apply the complete views to learn the common structure. Another strategy is to complete the missing samples by an average and a median

**Table 1**
The notations and their meanings.

| Notations | Meanings | Notations | Meanings |
|---|---|---|---|
| $X, \mathbf{x}$, and $x$ | matrix, vector, and scalar | $d_v$ | the feature dimension of the $v$th view |
| $\mathbf{x}_{i,:}$ | the $i$th row of $X$ | $n_v$ | the number of missing instances in the $v$th view |
| $\mathbf{x}_{:,j}$ | the $j$th column of $X$ | $\mathbf{1}$ | the column vector of all ones |
| $X_{i,j}$ or $x_{i,j}$ | the $(i,j)$th element of $X$ | $X^{(v)} \in \mathbb{R}^{d_v \times n}$ | the observed data of the $v$th view |
| $X^\top$ | the transpose of $X$ | $E^{(v)} \in \mathbb{R}^{d_v \times n_v}$ | the missing data of the $v$th view |
| $\|\cdot\|_F, \|\cdot\|_2$ | Frobenius norm, $l_2$-norm | $U^{(v)} \in \mathbb{R}^{d_v \times n}$ | the consensus representation of the $v$th view |
| $\mathrm{Tr}(\cdot)$ | the trace of the matrix | $G^{(v)} \in \mathbb{R}^{d_v \times d_v}$ | the feature similarity matrix of the $v$th view |
| $n, m$ | the number of samples, views | $S^{(v)} \in \mathbb{R}^{n \times n}$ | the connection graph of the $v$th view |

to satisfy the data structure. Such a simple completion strategy not only loses the semantic information of each view, but also leads to a large deviation in the cases with a high proportion of missing values. Thus, for incomplete multi-view data, how to use the missing information to learn the consensus representation remains critical and challenging. In this work, we complete the missing instances by using local feature structures of each view and adequately leveraging underlying information of missing entries. Specifically, we introduce the error matrix $\boldsymbol{E}^{(v)} \in \mathbb{R}^{d_v \times n_v}$ to model the missing data of $n_v$ missing instances. To characterize the relation of the observed data $\boldsymbol{X}^{(v)}$ and the missing data $\boldsymbol{E}^{(v)}$ of the $v$th view, an index matrix $\boldsymbol{W}^{(v)} \in \mathbb{R}^{n_v \times n}$ [26] is introduced as follows:

$$\boldsymbol{W}_{i,j} = \begin{cases} 1, & \text{if the } i \text{ th missing point is the } j \text{ th point of } \boldsymbol{X}^{(v)}, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Therefore, we measure the missing data in every single view by the following strategy:

$$\begin{aligned} \min_{\boldsymbol{E}^{(v)}, \boldsymbol{U}^{(v)}} \quad & \|\boldsymbol{X}^{(v)} + \boldsymbol{E}^{(v)}\boldsymbol{W}^{(v)} - \boldsymbol{U}^{(v)}\|_F^2 \\ & + \frac{\lambda_1}{2} \sum_{i,j=1}^{d_v} \|\mathbf{e}_{i,:}^{(v)} - \mathbf{e}_{j,:}^{(v)}\|_2^2 \boldsymbol{G}_{i,j}^{(v)}, \end{aligned} \tag{2}$$

where $\lambda_1 > 0$ is a parameter, $\boldsymbol{E}^{(v)}\boldsymbol{W}^{(v)}$ just corresponds to the missing data of each view, as shown in the left of Fig. 1. The term $\boldsymbol{X}^{(v)} + \boldsymbol{E}^{(v)}\boldsymbol{W}^{(v)}$denotes the complete view after filling and $\mathbf{e}_{i,:}^{(v)}$is the $i$th row of the matrix $E$, i.e., the $i$th feature. $\boldsymbol{U}^{(v)}$ is the consensus representation of each view. $\boldsymbol{G}_{i,j}^{(v)}$ denotes the feature similarity between features $\mathbf{e}_{i,:}^{(v)}$ and $\mathbf{e}_{j,:}^{(v)}$. The second term in (2) is to constrain the similar features, since two features are similar to each other in the missing instances with high similarity under the original instances. Feature similarity matrix $\boldsymbol{G}^{(v)} \in \mathbb{R}^{d_v \times d_v}$is constructed by mutual $k$nn graph if and only if the $i$th and the $j$th features are mutually the $k$ closest neighbors, i.e., $\boldsymbol{G}_{i,j}^{(v)} = 1$.

According to the properties of normalized graph Laplacians [55,56], we can easily deduce that $1/2\sum_{i,j=1}^{d_v} \|\mathbf{e}_{i,:}^{(v)} - \mathbf{e}_{j,:}^{(v)}\|_2^2 \boldsymbol{G}_{i,j}^{(v)}$has the following equivalent form:

$$\mathrm{Tr}\left(\boldsymbol{E}^{(v)\top} \boldsymbol{L}_{\boldsymbol{G}}^{(v)} \boldsymbol{E}^{(v)}\right), \tag{3}$$

where $\mathrm{Tr}(\cdot)$ is the trace of the matrix, $\boldsymbol{L}_{\boldsymbol{G}}^{(v)}$ is a Laplacian matrix of $\boldsymbol{G}^{(v)}$, which is calculated by $\boldsymbol{L}_{\boldsymbol{G}}^{(v)} = \boldsymbol{D}^{(v)} - \boldsymbol{G}^{(v)}$ with $\boldsymbol{D}^{(v)} = \mathrm{Diag}\left\{\sum_{j=1}^{d_v} \boldsymbol{G}_{i,j}^{(v)} | i = 1, 2, \ldots, d_v\right\}$ is the degree matrix of $\boldsymbol{G}^{(v)}$.

It is worth noting that the objective function (2) optimizes jointly the missing view and view-specific consensus representation so that they can gain from the updates of each other.

### 3.2. Graph learning

Let $\left\{\boldsymbol{U}^{(v)} \in \mathbb{R}^{d_v \times n},\ v = 1, 2, \ldots, m\right\}$denotes the projection space (also known as the representation), which takes the original features as an important basis for learning. So $\boldsymbol{U}^{(v)}$ should be similar to $\boldsymbol{X}^{(v)}$, otherwise it will destroy the topology. In addition, the representation learning takes into account the similarity between instances. If two instances have high similarity in the $v$th view, then their representations $\mathbf{u}_{:,i}^{(v)}$ and $\mathbf{u}_{:,j}^{(v)}$ are also similar. Similarly, graph learning will need to consider the relationship between the instances. If two instances representation $\mathbf{u}_{:,i}^{(v)}$ and $\mathbf{u}_{:,j}^{(v)}$ are similar in $v$ views, then $s_{i,j}^{(v)}$ should be relatively large. Thus, representation learning and graph learning are essentially processes of mutual influence. According to the above discussion, we set the loss of each view as

$$\begin{aligned} L_1^{(v)} = & \min_{\boldsymbol{E}^{(v)}, \boldsymbol{U}^{(v)}, \boldsymbol{S}^{(v)}} \|\boldsymbol{X}^{(v)} + \boldsymbol{E}^{(v)}\boldsymbol{W}^{(v)} - \boldsymbol{U}^{(v)}\|_F^2 \\ & + \lambda_1 \mathrm{Tr}\left(\boldsymbol{E}^{(v)\top} \boldsymbol{L}_{\boldsymbol{G}}^{(v)} \boldsymbol{E}^{(v)}\right) + \frac{\lambda_2}{2} \sum_{i,j=1}^{n} \|\mathbf{u}_{:,i}^{(v)} - \mathbf{u}_{:,j}^{(v)}\|_2^2 s_{i,j}^{(v)}, \\ & \text{s.t. } \forall v,\ s_{i,j}^{(v)} > 0, \mathbf{s}_{i,:}^{(v)} \mathbf{1} = 1, \end{aligned} \tag{4}$$

where $\mathbf{1}$ is a vector and its all entries are one. By (4), non-missing information and missing information can simultaneously guide the consensus representation. The first two steps mainly solve *Challenge 1* on handling missing views. Using the above strategy, the proposed method can reduce the impact of missing samples for clustering.

### 3.3. Cross-view graph matching

To cluster the above completed multi-view data, we match the graphs constructed from the projection space, i.e., minimize the difference between any two connection graphs to achieve view consensus. The cross-view loss is set as

$$L_2 = \min_{\boldsymbol{S}^{(v)}} \frac{1}{2} \sum_{v=1}^{m} \sum_{w=1, w \neq v}^{m} \|\boldsymbol{S}^{(v)} - \boldsymbol{S}^{(w)}\|_F^2. \tag{5}$$

Different from most of the traditional MVC methods, our method matches *pairwise connection graphs rather than all graphs* to achieve the purpose of consistency, with the following advantages: the representation $\boldsymbol{U}$ may be significantly different in each view. Once we force all representations $\boldsymbol{U}^{(v)} (v = 1, 2, \ldots, m)$ to be consistent, the optimization may be distorted and some useful information may be lost, thereby resulting in poor clustering performance. Instead, our method facilitates the relationship between $\boldsymbol{U}^{(v)}$ as invariant connection $\boldsymbol{S}^{(v)}$ and makes them as close as possible to each other, thereby largely avoiding distorted representations. By designing the cross-view graph matching, the proposed method solves *Challenge 2* and can effectively mine the complementary and consistent information and preserve view-specific semantic information to achieve excellent clustering results.

Finally, by combining the missing view filling model (4) and the cross-view graph matching strategy (5), we propose the final optimization function of CGMIMC as follows:

$$L = \sum_{v=1}^{m} L_1^{(v)} + L_2. \tag{6}$$

We present the overall framework of CGMIMC in Fig. 1.

## 4. Optimization algorithm

In this section, we develop an iterative algorithm for solving the proposed optimization problem (6). Theoretically, the problem (6) can be decomposed into three sub-problems: 1) optimizing the missing matrix $\boldsymbol{E}$, 2) solving the representations $\boldsymbol{U}$, and 3) updating the connection graphs $\boldsymbol{S}$. We conduct these three sub-problems until convergence. After the final connection graphs are obtained, we calculate the average graph and apply the spectral clustering algorithm to obtain the final data partition. Finally, we present the proposed iterative algorithm in Algorithm 1.

### 4.1. Initialize $\boldsymbol{S}^{(v)}$

For initializing $\boldsymbol{S}^{(v)}$, the similar graph initialization method in CLR [56] is adopted. To reduce the impact of missing value on com-

position, we use the actual instances information $\overline{\boldsymbol{X}}^{(v)} \in \mathbb{R}^{d_v \times (n-n_v)}$ and connection graph $\widehat{\boldsymbol{S}}^{(v)} \in \mathbb{R}^{(n-n_v)\times(n-n_v)}$. The initialization objective function is

$$\begin{aligned} &\min_{\boldsymbol{s}^{(v)}} \sum_{i,j=1}^{(n-n_v)} \|\bar{\mathbf{x}}_{:,i}^{(v)} - \bar{\mathbf{x}}_{:,j}^{(v)}\|_2^2 \hat{s}_{i,j}^{(v)} + \gamma \sum_{i,j=1}^{(n-n_v)} \hat{s}_{i,j}^2, \\ &\text{s.t.} \quad \forall i, \hat{s}_{i,j}^{(v)} \geqslant 0, \hat{\mathbf{s}}_{i,:}^{(v)} \mathbf{1} = 1, \end{aligned} \tag{7}$$

where $\hat{s}_{i,j}^{(v)}$ denotes the $(i,j)$th element of $\widehat{\boldsymbol{S}}^{(v)}$ and $\hat{\mathbf{s}}_{i,:}^{(v)}$ is the $i$th row vector of $\widehat{\boldsymbol{S}}^{(v)}$. For any two different instances $\bar{\mathbf{x}}_{:,i}^{(v)}$ and $\bar{\mathbf{x}}_{:,j}^{(v)}$, once their distance $c_{i,j}^{(v)} = \|\bar{\boldsymbol{x}}_i^{(v)} - \bar{\boldsymbol{x}}_j^{(v)}\|_2^2$ tend to be smaller, the connection strength $\hat{s}_{i,j}^{(v)}$ become larger. The second regular term on $\widehat{\boldsymbol{S}}^{(v)}$ makes connection graph $\widehat{\boldsymbol{S}}^{(v)}$ sparse. Let $k$ be the number of nearest neighbors. We initialize $\widehat{\boldsymbol{S}}^{(v)}$ as

$$\hat{s}_{i,j}^{(v)} = \begin{cases} \frac{c_{i,k+1}^{(v)} - c_{i,j}^{(v)}}{k}, & j \leqslant k, \\ kc_{i,k+1}^{(v)} - \sum_{h=1}^{k} c_{i,h}^{(v)} & \\ 0, & j > k. \end{cases} \tag{8}$$

In order to get the complete graph $\boldsymbol{S}^{(v)}$, we perform a conversion operation on $\widehat{\boldsymbol{S}}^{(v)}$

$$\boldsymbol{S}^{(v)} = \boldsymbol{R}^{(v)T} \widehat{\boldsymbol{S}}^{(v)} \boldsymbol{R}^{(v)}, \tag{9}$$

where $\boldsymbol{R}^{(v)} \in \mathbb{R}^{(n-n_v)\times n}$ denotes the relation matrix as follows:

$$R_{i,j}^{(v)} = \begin{cases} 1, & \text{if the } i\text{th instance of } \widehat{X}^{(v)} \text{ is the } j\text{th one of } X^{(v)}, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

**Algorithm 1:** The iterative algorithm for solving (6).

**Input**: observed data $X^{(v)}(v = 1, 2, \ldots, m)$ with $m$ views, parameters $\lambda_1, \lambda_2$, number of clusters $c$.

**Output**: for $v = 1, 2, \ldots, m$, missing matrix $E^{(v)}$, representation $U^{(v)}$, connection graph $S^{(v)}$, and final clustering results.

**Initialization**: $E^{(v)}, U^{(v)}$, and $S^{(v)}$ for $v = 1, 2, \ldots, m$.

1: **for** iterator = 1 to $T$ **do**

2: Optimize $\left\{E^{(v)}\right\}_{v=1}^{m}$ according to Eq. (11).

3: Optimize $\{U^{(v)}\}_{v=1}^{m}$ according to Eq. (13).

4: Optimize $\left\{S^{(v)}\right\}_{v=1}^{m}$ according to Eq. (15).

5: **end for**

6: $S = \frac{1}{m}\sum_{v=1}^{m} S^{(v)}$, applying spectral clustering on $S$ to obtain optimal clustering results.

### 4.2. Update $E^{(v)}$

The objective function $L_E^{(v)}$ is as follows:

$$\lambda_1 \mathrm{Tr}\left(E^{(v)\top} L_G^{(v)} E^{(v)}\right) + \min_{E^{(v)}} \|E^{(v)} - U^{(v)} W^{(v)\top}\|_F^2.$$

Using the optimal condition $\partial L_E^{(v)} / \partial E^{(v)} = 0$, we obtain the following closed-form solution:

$$E^{(v)} = \left(I + \lambda_1 L_G^{(v)}\right)^{-1} U^{(v)} W^{(v)\top}. \tag{11}$$

### 4.3. Update $U^{(v)}$

We fix the other irrelevant variables and update $U^{(v)}$ as

$$\begin{aligned} &\min_{U^{(v)}} \|X^{(v)} + E^{(v)} W^{(v)} - U^{(v)}\|_F^2 + \frac{\lambda_2}{2} \sum_{i,j=1}^{n_v} \|U_{:,i} - U_{:,j}\|_2^2 s_{i,j} \\ &= \min_{U^{(v)}} \|X^{(v)} + E^{(v)} W^{(v)} - U^{(v)}\|_F^2 + \lambda_2 \mathrm{Tr}\left(U^{(v)} L_S^{(v)} U^{(v)\top}\right). \end{aligned} \tag{12}$$

Similar to the update of $E^{(v)}$, we calculate $U^{(v)}$ by

$$U^{(v)} = \left(X^{(v)} + E^{(v)} W^{(v)}\right)\left(I + \lambda_2 L_S^{(v)}\right)^{-1}, \tag{13}$$

where $L_S^{(v)} = D_{S^{(v)}} - S^{(v)}$.

### 4.4. Update $S^{(v)}$

Fixing variables $E^{(v)}$ and $U^{(v)}$, the optimization problem of $S^{(v)}$ is

$$\begin{aligned} &\min_{S^{(v)}} \frac{\lambda_2}{2} \sum_{i,j=1}^{n} \|U_{:,i}^{(v)} - U_{:,j}^{(v)}\|_2^2 s_{i,j}^{(v)} + \frac{1}{2} \sum_{w=1}^{m} \|S^{(v)} - S^{(w)}\|_F^2, \\ &\text{s.t.} \quad s_{i,j}^{(v)} > 0, s_{i,:}^{(v)} \mathbf{1} = 1. \end{aligned} \tag{14}$$

It can be decoupled into multiple sub-problems to solve. Let $y_{i,j} = \|U_{:,i}^{(v)} - U_{:,j}^{(v)}\|_2^2$, and $\mathbf{y}_{i,:}$ denotes the vector that the $j$th element is $y_{i,j}$. Then, the above formula is equivalent to

$$\begin{aligned} &\min_{s_{i,:}^{(v)}} \left\| s_{i,:}^{(v)} - \left( \frac{\sum_{w=1, w\neq v} s_{i,:}^{(w)}}{m-1} - \frac{\lambda_2 \mathbf{y}_{i,:}^{(v)}}{2(m-1)} \right) \right\|_2^2, \\ &\text{s.t.} \quad s_{i,:}^{(v)} \mathbf{1} = 1, s_{i,j}^{(v)} > 0, \end{aligned} \tag{15}$$

which can be effectively solved by the iterative algorithm in [57].

### 4.5. Time complexity analysis

The main time cost of Algorithm 1 depends on the calculation of subproblems $E$ (11), $U$ (13), and $S$ (15). Next, we analyze the time complexity of three sub-problems. In Algorithm 1, the complexity of Steps 2 and 3 are $\mathcal{O}\left(md_v n\hat{n} + md_v^2 \hat{n}\right)$ and $\mathcal{O}\left(md_v n\hat{n} + mn^3 + md_v n^2\right)$, respectively, where $\hat{n} = \max(n_1, n_2, \ldots, n_m)$. According to [57], the complexity of Step 4 is $\mathcal{O}(mn^2 \log n)$. Therefore, the total computational complexity is $\mathcal{O}\left(Tm\left(d_v n\hat{n} + n^3 + d_v n^2 + d_v^2 \hat{n} + n^2 \log n\right)\right)$, where $T$ denotes the number of iterations in Algorithm 1. Since the dimension $d_v$ of the feature is generally much smaller than the number $n$ of data instances, the total time complexity is $\mathcal{O}\left(Tmn^3\right)$.

## 5. Numerical experiments

We test the proposed CGMIMC for IMVC on four standard multi-view datasets. We also evaluate the impact of the parameters and analyze the convergence of the proposed algorithm.

### 5.1. Experimental settings

**Datasets.** Following the conventional study in multi-view learning, in Table 2, we show the common multi-view datasets to test the proposed method. The dataset **Caltech101-7** contains 1,474 samples from 7 categories and the dataset **Caltech101-20** contains 2,386 samples from 20 categories, where each sample is

**Table 2**
Characteristics of the used multi-view datasets.

| Datasets | instances | views | clusters |
|---|---|---|---|
| Caltech101-7 | 1,474 | 6 | 7 |
| Caltech101-20 | 2,386 | 6 | 20 |
| Flower17 | 1,360 | 7 | 17 |
| NUS | 2,000 | 5 | 31 |

represented by 6 views. The dataset **Flower17**[1] contains 17 classes of flower image data sets, each consisting of 80 images. Meanwhile, seven different features are extracted as views. The dataset **NUS**[2] is an object recognition database. It includes 31 classes with 2,000 images, and each image has 5 views.

**Compared algorithms**. To compare with our proposed CGMIMC, we consider one single view baseline, one multi-view graph learning baseline, and four IMVC baselines as follows.

- BSV [58]: Best single view (BSV) uses the average instance of each view to add the missing entries, and independently conducts $k$-means for each view and displays the highest clustering index.
- GMC [51]: The graph-based multi-view clustering (GMC) fuses connection graphs of all views together to build a unified connection graph. Again, missing data is populated with the average value of each view.
- MIC [23]: Multiple incomplete views clustering (MIC) via weighted NMF is a feasible method for incomplete data clustering, which completes missing examples with the view-specific average feature and then learns a common consensus matrix with the co-regularization approach.
- DAIMC [20]: Based on weighted semi-nonnegative matrix factorization, doubly aligned incomplete multi-view clustering method (DAIMC) applies the given instance alignment information to obtain the shared latent feature matrix of all views.
- IMSC [30]: Adaptively learning multiple incomplete graphs using subspace learning, incomplete multi-view spectral clustering (IMSC) approach applies the spectral clustering on the generate a complete graph from whole incomplete graphs.
- PIC [59]: Spectral perturbation based incomplete multi-view clustering (PIC) approach transfers the missing data problem to similarity-value missing problem. It uses a similarity matrix completion scheme to fill the missing similarity entries and then performs clustering on the consensus matrix.
- EE-IMVC [52]: Based on multi-kernel $k$-means clustering, the efficient and effective incomplete multi-view clustering (EE-IMVC) approach iteratively updates the absent views and the clustering result until convergence.
- AGC-IMC [53]: The adaptive graph completion-based incomplete multi-view clustering (AGC-IMC) method uses the similarity information of other views to infer the missing view. In particular, the consensus representation is learned by exploiting shared information across all views.

**Evaluation metrics**. We adopt two evaluation indicators: the Normalized Mutual Information (NMI) and the clustering Accuracy (ACC) [26] for testing the clustering performance of our method. ACC is the accuracy metric computed for the best matching permutation between clustered labels and true labels. Based on the information theoretic, NMI can balance the quality of the clustering against the number of clusters. All indicators are distributed within the range $[0, 1]$. The higher the NMI and ACC values, the better the clustering performance. We run each experiment 10 times and take the average value and the variance as the final clustering result.

**Parameters setting**. In the following experiments, to construct the incomplete multi-view test datasets, we randomly remove the same probability samples of each view. Missing probability $R$ represents the probability of missing samples taking up all the samples and is set to $0.1, 0.2, 0.3$, and 0.5, respectively. When constructing the feature similarity matrix of the view $G$, We use the cosine similarity to calculate the similarity between the features and set $k$ as 8 for the closest neighbors. When initializing the connection diagram $S$, we set $k$ as 10 for each of its closest neighbors. For all comparison methods, the parameters are well-tuned to obtain the best performance, which is consistent with the values recommended in their corresponding papers.

### 5.2. Experimental results

Tables 3 and 4 represent the average NMI and ACC values of each comparison method and the proposed CGMIMC on tasted datasets when the miss probability $R$ is $0.1, 0.2$, and 0.3, respectively. According to these results, we have the following observations:

(1) From Tables 3 and 4, we observe that the single view clustering method BSV shows poor performance in most cases because it only used single view data without complementary information. As a comparison, by effectively using the complementary information of multi-view data, other methods obtain the better clustering results.
(2) Compared with the graph-based MVC method GMC and NMF-based incomplete multi-views clustering method MIC, the methods DAIMC, IMSC, PIC, and the proposed CGMIMC achieve a great improvement in terms of NMI and ACC values. The main reason is that missing entries are completed with the average value of each view so that some important features are destroyed. Thus, GMC and MIC are not effective in test datasets by simply filling the missing value with the mean value. The proposed method CGMIMC infers the missing information to satisfy the intrinsic structure of the view and effectively reduces the impact of missing samples.
(3) Compared with the incomplete multi-view clustering methods DAIMC, IMSC, PIC, EE-IMVC, and AGC-IMC, the proposed CGMIMC achieves a better performance in terms of NMI and ACC values in most cases. The reason may lie in the fact that these compared methods force a common representation of multiple non-identical views, thereby leading to some view-specific semantic information being lost. This demonstrates that our CGMIMC can adequately utilize the complementary multi-view information and preserve view-specific semantic information to improve the clustering results.
(4) To test the robustness of CGMIMC, in Fig. 2, we show the clustering performance of all compared methods on Caltech101-7 dataset with the large missing rate $R = 0.5$. Clearly, the proposed CGMIMC achieves the best results in both clustering metrics. Specifically, when the missing rate $R = 0.5$, compared with the second-best PIC method, our CGIMC improves NMI and ACC by 7.84% and 3.53%, respectively. In addition, although the GMC method is not designed for the IMVC problem, the average strategy is better than many clustering methods for incomplete data, which also shows that the other comparison methods cannot solve the problem of the missing views well. In summary, our CGMIMC still has significant advantages over other methods with the increase of missing data.

[1] http://www.robots.ox.ac.uk/vgg/data/flowers/.
[2] https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html.

**Table 3**
The average NMIs (mean ± variance) of different methods on four test datasets. The number in bold indicates the best result.

| Datasets | *R* | BSV | GMC | MIC | DAIMC | IMSC | PIC | EE-IMVC | AGC-IMC | CGMIMC |
|---|---|---|---|---|---|---|---|---|---|---|
| Caltech101-7 | 0.1 | 0.3839 ± 0.0427 | 0.5492 ± 0.0000 | 0.4877 ± 0.0340 | 0.4346 ± 0.0279 | 0.4569 ± 0.0150 | 0.5242 ± 0.0142 | 0.3932 ± 0.0000 | 0.5040 ± 0.0196 | **0.6239 ± 0.0000** |
| | 0.2 | 0.3302 ± 0.0352 | 0.2159 ± 0.0000 | 0.4831 ± 0.0372 | 0.4330 ± 0.0198 | 0.4334 ± 0.0008 | 0.5228 ± 0.0022 | 0.4013 ± 0.0000 | 0.5299 ± 0.0000 | **0.6156 ± 0.0000** |
| | 0.3 | 0.2933 ± 0.0314 | 0.5325 ± 0.0000 | 0.4781 ± 0.0371 | 0.4413 ± 0.0223 | 0.4235 ± 0.0044 | 0.5429 ± 0.0130 | 0.3460 ± 0.0000 | 0.5246 ± 0.0000 | **0.6113 ± 0.0000** |
| Caltech101-20 | 0.1 | 0.4990 ± 0.0194 | 0.3945 ± 0.0000 | 0.5842 ± 0.0140 | 0.5492 ± 0.0144 | 0.5448 ± 0.0121 | 0.5665 ± 0.0065 | 0.5331 ± 0.0000 | 0.5677 ± 0.0001 | **0.6316 ± 0.0000** |
| | 0.2 | 0.4604 ± 0.0128 | 0.5412 ± 0.0000 | 0.5843 ± 0.0141 | 0.5530 ± 0.0158 | 0.5279 ± 0.0055 | 0.5620 ± 0.0080 | 0.5193 ± 0.0000 | 0.4295 ± 0.0004 | **0.6194 ± 0.0000** |
| | 0.3 | 0.4166 ± 0.0032 | 0.4073 ± 0.0000 | 0.5836 ± 0.0178 | 0.5593 ± 0.0131 | 0.5213 ± 0.0088 | 0.5614 ± 0.0069 | 0.4958 ± 0.0000 | 0.5346 ± 0.0000 | **0.6125 ± 0.0000** |
| Flower17 | 0.1 | 0.3745 ± 0.0072 | 0.2644 ± 0.0000 | 0.3299 ± 0.0087 | 0.4884 ± 0.0094 | 0.5268 ± 0.0084 | 0.5175 ± 0.012 | **0.5369 ± 0.0000** | 0.4049 ± 0.0000 | 0.5346 ± 0.0000 |
| | 0.2 | 0.3328 ± 0.0071 | 0.2751 ± 0.0000 | 0.3250 ± 0.0038 | 0.4866 ± 0.0146 | 0.4932 ± 0.0021 | 0.5172 ± 0.0059 | 0.5150 ± 0.0000 | 0.3994 ± 0.0000 | **0.5259 ± 0.0000** |
| | 0.3 | 0.2959 ± 0.0057 | 0.1182 ± 0.0000 | 0.3245 ± 0.0054 | 0.4773 ± 0.0100 | 0.4749 ± 0.0039 | 0.4948 ± 0.0042 | 0.4931 ± 0.0000 | 0.3540 ± 0.0000 | **0.5068 ± 0.0000** |
| NUS | 0.1 | 0.1536 ± 0.0045 | 0.0657 ± 0.0000 | 0.0300 ± 0.0090 | 0.1885 ± 0.0050 | 0.1711 ± 0.0034 | 0.1861 ± 0.0035 | 0.1571 ± 0.0000 | 0.1230 ± 0.0005 | **0.1922 ± 0.0000** |
| | 0.2 | 0.1418 ± 0.0019 | 0.0628 ± 0.0000 | 0.0300 ± 0.0043 | 0.1862 ± 0.0059 | 0.1775 ± 0.0030 | 0.1731 ± 0.0029 | 0.1661 ± 0.0000 | 0.0805 ± 0.0000 | **0.1889 ± 0.0000** |
| | 0.3 | 0.1274 ± 0.1194 | 0.0664 ± 0.0000 | 0.0300 ± 0.0070 | **0.1823 ± 0.0055** | 0.1738 ± 0.0046 | 0.1773 ± 0.0028 | 0.1554 ± 0.0000 | 0.0705 ± 0.0000 | 0.1778 ± 0.0000 |



**Table 4**
The average ACCs (mean ± variance) of different methods on four test datasets. The number in bold indicates the best result.

| Datasets | *R* | BSV | GMC | MIC | DAIMC | IMSC | PIC | EE-IMVC | AGC-IMC | CGMIMC |
|---|---|---|---|---|---|---|---|---|---|---|
| Caltech101-7 | 0.1 | 0.4904 ± 0.0951 | 0.6791 ± 0.0000 | 0.4259 ± 0.0374 | 0.4242 ± 0.0471 | 0.5505 ± 0.0006 | 0.6513 ± 0.0057 | 0.3548 ± 0.0000 | 0.6583 ± 0.0004 | **0.6839 ± 0.0000** |
| | 0.2 | 0.4282 ± 0.0780 | 0.4294 ± 0.0000 | 0.4079 ± 0.0405 | 0.4303 ± 0.0361 | 0.5516 ± 0.0000 | 0.6443 ± 0.0116 | 0.3514 ± 0.0000 | 0.6628 ± 0.0000 | **0.6858 ± 0.0000** |
| | 0.3 | 0.4019 ± 0.0486 | 0.6757 ± 0.0000 | 0.4024 ± 0.0381 | 0.4256 ± 0.0451 | 0.5464 ± 0.0036 | 0.6526 ± 0.0031 | 0.3602 ± 0.0000 | 0.6546 ± 0.0000 | **0.6850 ± 0.0000** |
| Caltech101-20 | 0.1 | 0.4046 ± 0.0326 | 0.4384 ± 0.0000 | 0.4156 ± 0.0222 | 0.4451 ± 0.0237 | 0.4590 ± 0.0288 | 0.5380 ± 0.0173 | 0.4434 ± 0.0000 | 0.5165 ± 0.0000 | **0.5578 ± 0.0000** |
| | 0.2 | 0.3460 ± 0.0290 | 0.5591 ± 0.0000 | 0.4065 ± 0.0267 | 0.4539 ± 0.0378 | 0.4520 ± 0.0193 | 0.5333 ± 0.0091 | 0.4464 ± 0.0000 | 0.4844 ± 0.0000 | **0.5595 ± 0.0000** |
| | 0.3 | 0.3247 ± 0.0017 | 0.4317 ± 0.0000 | 0.4047 ± 0.0253 | 0.4672 ± 0.0271 | 0.4559 ± 0.0130 | 0.5436 ± 0.0194 | 0.3982 ± 0.0000 | 0.4812 ± 0.0000 | **0.5523 ± 0.0000** |
| Flower17 | 0.1 | 0.3515 ± 0.0070 | 0.2537 ± 0.0000 | 0.2772 ± 0.0086 | 0.4738 ± 0.0118 | 0.5060 ± 0.0131 | 0.5219 ± 0.0256 | **0.5515 ± 0.0000** | 0.3860 ± 0.0000 | 0.5458 ± 0.0000 |
| | 0.2 | 0.3169 ± 0.0080 | 0.2566 ± 0.0000 | 0.2685 ± 0.0104 | 0.4909 ± 0.0146 | 0.4901 ± 0.0020 | 0.5229 ± 0.0250 | 0.5015 ± 0.0000 | 0.3801 ± 0.0000 | **0.5389 ± 0.0000** |
| | 0.3 | 0.2885 ± 0.0075 | 0.1544 ± 0.0000 | 0.2707 ± 0.0055 | 0.4622 ± 0.0118 | 0.4712 ± 0.0037 | 0.5066 ± 0.0160 | 0.5059 ± 0.0000 | 0.3382 ± 0.0000 | **0.5279 ± 0.0000** |
| NUS | 0.1 | 0.1521 ± 0.0029 | 0.1440 ± 0.0000 | 0.1245 ± 0.0066 | 0.1619 ± 0.0050 | 0.1489 ± 0.0036 | 0.1549 ± 0.0045 | 0.1475 ± 0.0000 | 0.1535 ± 0.0000 | **0.1680 ± 0.0000** |
| | 0.2 | 0.1550 ± 0.0040 | 0.1430 ± 0.0000 | 0.1245 ± 0.0080 | 0.1610 ± 0.0039 | 0.1600 ± 0.0025 | 0.1566 ± 0.0050 | 0.1575 ± 0.0000 | 0.1480 ± 0.0000 | **0.1650 ± 0.0000** |
| | 0.3 | 0.1565 ± 0.1419 | 0.1525 ± 0.0000 | 0.1245 ± 0.0073 | **0.1601 ± 0.0052** | 0.1536 ± 0.0059 | 0.1582 ± 0.0038 | 0.1490 ± 0.0000 | 0.1475 ± 0.0000 | 0.1575 ± 0.0000 |

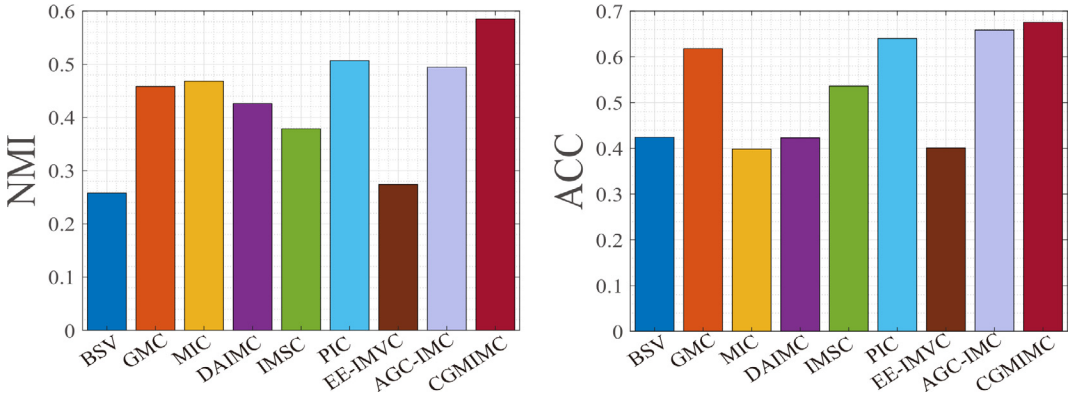


**Fig. 2.** The clustering comparison of different methods on the dataset Caltech101-7 with $R = 0.5$.

### 5.3. Convergence analysis

In theory, the original model (6) is not a joint convex problem about $E, U$, and $S$. And the global optimal solution cannot be guaranteed. For the first and second sub-problems, the solutions of $E$ and $U$ are closed-form solutions. The third sub-problem $S$ is solved by an efficient iterative algorithm. Therefore, in theory, the proposed alternating iteration algorithm for solving CGMIMC will converge to a locally optimal solution. Next, we study the numerical convergence of our Algorithm 1.

To show the convergence of the proposed algorithm numerically, taking Caltech101-7, Caltech101-20, and flower17 datasets as examples, we show the variable loss changes curves in Fig. 3 (a) ($R = 0.1$) and Fig. 3(b) ($R = 0.5$). We observed that the loss curves decrease rapidly (less than 10 iterations), then remain steady with the increase of the iterations. This demonstrates that the proposed algorithm is convergent numerically.

### 5.4. Parameters analysis

We test the influence of different parameter settings on the proposed method. The trade-off parameters $\lambda_1$ and $\lambda_2$ are selected from the range of $\{0.001, 0.01, 0.1, 1, 10, 100\}$. We test the influences of the parameters $\lambda_1$ and $\lambda_2$ on Caltech101-7 dataset with

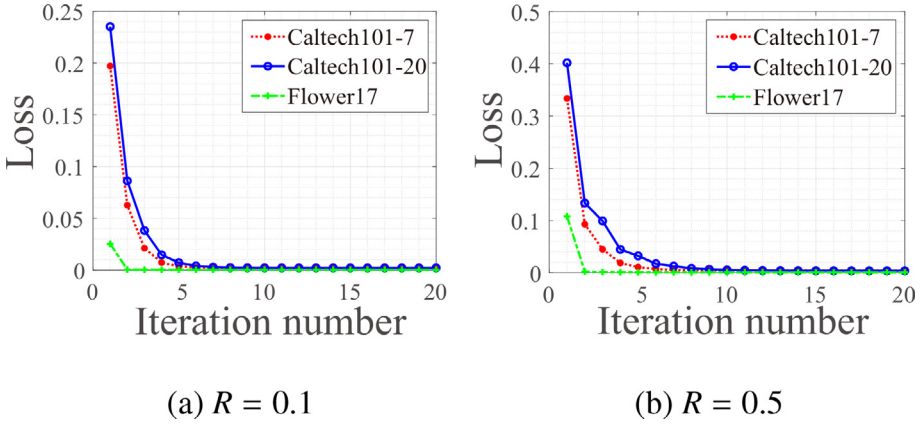


**Fig. 3.** The curves of variable loss changes versus iterations on datasets Caltech101-7, Caltech101-20, and Flower17.

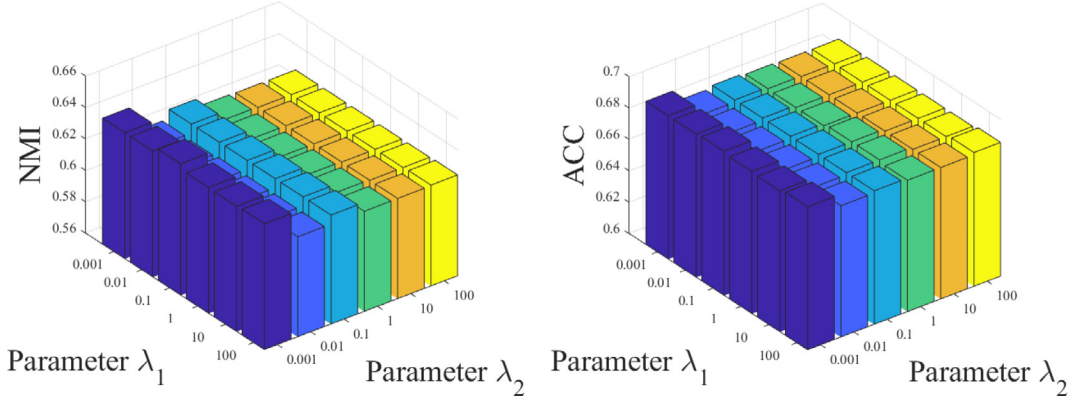


**Fig. 4.** The influences of the parameters $\lambda_1$ and $\lambda_2$ for the proposed method.

the missing rate $R$ as 0.1. As shown in Fig. 4, we observe that the proposed method CGMIMC is robust to the choice of $\lambda_1$. Although the parameter $\lambda_2$ has a small impact on the clustering metrics NMI and ACC, our method CGMIMC is also robust to the different $\lambda_2$.

## 6. Conclusion

We propose an incomplete multi-view clustering method called CGMIMC based on cross-view graph matching. Compared with existing IMVC methods, CGMIMC has the following advantages: 1) CGMIMC optimizes the missing value to meet the intrinsic structure of each view, thereby reducing the impact of missing data on clustering. 2) CGMIMC innovatively transforms the view representation into the graph connection representation while maintaining invariability. 3) CGMIMC realizes the consensus of multiple views through the cross-view graph matching strategy with the preservation of view-specific semantic information. The effectiveness and superiority of CGMIMC have been evaluated by numerical comparison experiments on representative datasets. In the future, more efficient and robust algorithms will be considered for large-scale datasets. Other learning tasks including semi-supervised and supervised learning are also within the scope of the next research.

## CRediT authorship contribution statement

**Jing-Hua Yang:** Conceptualization, Formal analysis, Methodology, Visualization, Writing – original draft. **Le-Le Fu:** Formal analysis, Data curation, Writing – review & editing. **Chuan Chen:** Supervision, Validation, Writing – review & editing. **Hong-Ning Dai:** Supervision, Validation, Writing – review & editing. **Zibin Zheng:** Supervision, Validation.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (62176269, 11801595), the Guang dong Basic and Applied Basic Research Foundation (2019A1515011043), and Macao Science and Technology Development Fund under Macao Funding Scheme for Key R & D Projects (0025/2019/AKP).

## References

[1] Z. Zhang, L. Liu, J. Qin, F. Zhu, F. Shen, Y. Xu, L. Shao, H.T. Shen, Highly-economized multi-view binary compression for scalable image clustering, In Proceedings of the European Conference on Computer Vision (2018) 731–748.
[2] P. Karczmarek, A. Kiersztyn, W. Pedrycz, E. Al, K-means-based isolation forest, Knowl. Based Syst. 195 (2020).
[3] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Y. Ng, Multimodal deep learning, In Proceedings of the International Conference on Machine Learning (2011) 689–696.
[4] S. Li, Y. Jiang, Z. Zhou, Partial multi-view clustering, In Proceedings of the AAAI Conference on Artificial Intelligence (2014) 1968–1974.
[5] Y.Y. Chen, X.L. Xiao, Y.C. Zhou, Multi-view subspace clustering via simultaneously learning the representation tensor and affinity matrix, Pattern Recognt. 106 (2020).
[6] M. Brbić, I. Kopriva, Multi-view low-rank sparse subspace clustering, Pattern Recognit. 73 (2018) 247–258.
[7] C. Tang, X.W. Liu, X.Z. Zhu, E. Zhu, L.Z. Luo, Z.G. and Wang, W. Gao, Cgd: Multi-view clustering via cross-view graph diffusion, In Proceedings of the AAAI Conference on Artificial Intelligence (2020) 5924–5931.
[8] C. Chen, Y. Wang, W.B. Hu, Z.B. Zheng, Robust multi-view k-means clustering with outlier removal, Knowl. Based Syst. 210 (2020).
[9] X. Cai, F. Nie, H. Huang, Multi-view k-means clustering on big data, In Proceedings of the International Joint Conference on Artificial Intelligence (2013) 2598–2604.
[10] H. Wang, Y. Yang, B. Liu, H. Fujita, A study of graph-based system for multi-view clustering, Knowl. Based Syst. 163 (2019) 1009–1019.
[11] J.P. Tan, Z.J. Yang, Y.Q. Cheng, J.L. Ye, B. Wang, Q.Y. Dai, Sragl-awcl: A two-step multi-view clustering via sparse representation and adaptive weighted cooperative learning, Pattern Recognit. 117 (2021).
[12] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, In Proceedings of the Neural Information Processing Systems 24 (NIPS 2011) (2011) 1413–1421.
[13] C. Chen, Y.Z. Li, H. Qian, Z.B. Zheng, Y.Q. Hu, Multi-view semi-supervised learning for classification on dynamic networks, Knowl. Based Syst. 195 (2020).
[14] X. Tao, R. Wang, R. Chang, C. Li, R. Liu, J. Zhou, Spectral clustering algorithm using density-sensitive distance measure with global and local consistencies, Knowl. Based Syst. 170 (2019) 26–42.
[15] J. Yin, S.L. Sun, Incomplete multi-view clustering with cosine similarity, Pattern Recognit. 123 (2022).
[16] H. Zhao, Z. Ding, Y. Fu, Multi-view clustering via deep matrix factorization, In Proceedings of the AAAI Conference on Artificial Intelligence (2017) 2921–2927.
[17] S. Huang, Z. Kang, Z. Xu, Self-weighted multi-view clustering with soft capped norm, Knowl. Based Syst. 158 (2018) 1–8.
[18] J.L. Liu, S.H. Teng, L.K. Fei, W. Zhang, X.Z. Fang, Z.X. Zhang, N.Q. Wu, A novel consensus learning approach to incomplete multi-view clustering, Pattern Recognit. 115 (2021).
[19] X.W. Liu, M.M. Li, C. Tang, J.Y. Xia, J. Xiong, L. Liu, M. Kloft, E. Zhu, Efficient and effective regularized incomplete multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. 43 (8) (2021) 2634–2646.
[20] M. Hu, S. Chen, Doubly aligned incomplete multi-view clustering, In Proceedings of the International Joint Conference on Artificial Intelligence (2018) 2262–2268.
[21] J. Wen, Z. Zhang, Y. Xu, Z.F. Zhong, Incomplete multi-view clustering via graph regularized matrix factorization, In Proceedings of the European Conference on Computer Vision Workshop (2018) 593–608.
[22] W.Z. Zhuge, C.P. Hou, X.W. Liu, H. Tao, D.Y. Yi, Simultaneous representation learning and clustering for incomplete multi-view data, In Proceedings of the International Joint Conference on Artificial Intelligence (2019) 4482–4488.
[23] W. Shao, L. He, S.Y. Philip, Multiple incomplete views clustering via weighted nonnegative matrix factorization with l_2,1) regularization, Joint European Conference on Machine Learning and Knowledge Discovery in Databases (2015) 318–334.
[24] S. Bhadra, S. Kaski, J. Rousu, Multi-view kernel completion, Mach. Learn. 106 (5) (2017) 713–739.
[25] Q. Yin, S. Wu, L. Wang, Incomplete multi-view clustering via subspace learning, In Proceedings of the ACM International on Conference on Information and Knowledge Management (2015) 383–392.
[26] J. Wen, Z. Zhang, Y. Xu, B. Zhang, L. Fei, H. Liu, Unified embedding alignment with missing views inferring for incomplete multi-view clustering, In Proceedings of the AAAI Conference on Artificial Intelligence (2019) 5393–5400.
[27] H. Zhao, H. Liu, Y. Fu, Incomplete multi-modal visual data grouping, In Proceedings of the International Joint Conference on Artificial Intelligence (2016) 2392–2398.
[28] M.L. Hu, S.C. Chen, One-pass incomplete multi-view clustering, In Proceedings of the AAAI Conference on Artificial Intelligence (2019) 1–9.
[29] N. Rai, S. Negi, S. Chaudhury, O. Deshmukh, Partial multi-view clustering using graph regularized NMF, In Proceedings of the International Conference on Pattern Recognition (2016) 2192–2197.
[30] J. Wen, Y. Xu, H. Liu, Incomplete multiview spectral clustering with adaptive graph learning, IEEE Trans. Cybern. 50 (4) (2018) 1418–1429.
[31] Y.H. Jia, H. Liu, J.H. Hou, Q.F. Zhang, Clustering ensemble meets low-rank tensor approximation, In Proceedings of the AAAI Conference on Artificial Intelligence (2021) 7970–7978.
[32] Y.B.Y. Jiang, Q.Q. Xu, Z.Y. Yang, X.C. Cao, Q.M. Huang, Dm2c: Deep mixed-modal clustering, In Proceedings of the Neural Information Processing Systems (2019) 5888–5892.
[33] Y.J. Lin, Y.B. Gou, Z.T. Liu, B.Y. Li, J.C. Lv, X. Peng, Completer: Incomplete multi-view clustering via contrastive prediction, In Proceedings of the Conference on Computer Vision and Pattern Recognition (2021) 11174–11183.
[34] Y.H. Yao, C.J. Wong, Interpretable clustering on dynamic graphs with recurrent graph neural networks, In Proceedings of the AAAI Conference on Artificial Intelligence (2021) 4608–4616.
[35] T. Hocking, J.P. Vert, F.R. Bach, A. Joulin, Clusterpath-an algorithm for clustering using convex fusion penalties, In Proceedings of the International Conference on International Conference on Machine Learning (2011) 745–752.
[36] S.A. Shah, V. Koltun, Robust continuous clustering, In Proceedings of the National Academy of Sciences 114 (37) (2017) 9814–9819.
[37] D.D. Sun, D.S. Li, Z.L. Ding, X.Y. Zhang, J. Tang, Dual-decoder graph autoencoder for unsupervised graph representation learning, Knowl. Based Syst. 234 (2021).
[38] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
[39] G. Liu, H. Xu, J. Tang, Q. Liu, S. Yan, A deterministic analysis for LRR, IEEE Trans. Pattern Anal. Mach. Intell. 38 (3) (2016) 417–430.

[40] X. Peng, Z. Yu, Z. Yi, H. Tang, Constructing the l_2)graph for robust subspace learning and subspace clustering, IEEE Trans. Cybern. 47 (4) (2017) 1053–1066.
[41] X. Peng, J. Feng, S. Xiao, W.Y. Yau, J.T. Zhou, S. Yang, Structured autoencoders for subspace clustering, IEEE Trans. Image Process. 27 (10) (2018) 5076–5086.
[42] W.X. Tu, S.H. Zhou, X.W. Liu, X.F. Guo, Z.P. Cai, E. Zhu, J.R. Cheng, Deep fusion clustering network, In Proceedings of the AAAI Conference on Artificial Intelligence (2021) 9978–9987.
[43] K. Chaudhuri, S.M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, In Proceedings of the Annual International Conference on Machine Learning (2009) 129–136.
[44] C. Zhang, H. Fu, S. Liu, G. Liu, X. Gao, Low-rank tensor constrained multiview subspace clustering, In Proceedings of the International Conference on Computer Vision (2015) 1582–1590.
[45] Z. Kang, G. Shi, S. Huang, W. Chen, X. Pu, J.T. Zhou, Z. Xu, Multi-graph fusion for multi-view spectral clustering, Knowl. Based Syst. 189 (2020).
[46] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, S. Pan, Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering, In Proceedings of the International Joint Conference on Artificial Intelligence (2016) 2153–2159.
[47] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, In Proceedings of the AAAI Conference on Artificial Intelligence (2014) 2149–2155.
[48] Y. Wang, L. Wu, X. Lin, J. Gao, Multiview spectral clustering via structured low-rank matrix factorization, IEEE Trans. Neur. Net. Lear. 29 (10) (2018) 4833–4843.
[49] Y. Wang, L. Wu, Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multiview spectral clustering, Neural Networks 103 (2018) 1–8.
[50] K. Zhan, F. Nie, J. Wang, Y. Yang, Multiview consensus graph clustering, IEEE Trans. Image Process. 28 (3) (2019) 1261–1270.
[51] H. Wang, Y. Yang, B. Liu, GMC: Graph-based multi-view clustering, IEEE Trans. Knowl. Data Eng. 32 (6) (2020) 1116–1129.
[52] X.W. Liu, X.Z. Zhu, M.M. Li, C. Tang, E. Zhu, J.P. Yin, W. Gao, Efficient and effective incomplete multi-view clustering, In Proceedings of the AAAI Conference on Artificial Intelligence (2019) 4392–4399.
[53] J. Wen, K. Yan, Z. Zhang, Y. Xu, J.Q. Wang, L.K. Fei, B. Zhang, Adaptive graph completion based incomplete multi-view clustering, IEEE Trans. MultiMedia 23 (2021) 2493–2504.
[54] L.S. Li, Z.Q. Wan, H.B. He, Incomplete multi-view clustering with joint partition and graph learning, IEEE Trans. Knowl. Data En. (2021), https://doi.org/10.1109/TKDE.2021.3082470.
[55] U.V. Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[56] F. Nie, X. Wang, M. Jordan, H. Huang, The constrained Laplacian rank algorithm for graph-based clustering, In Proceedings of the AAAI Conference on Artificial Intelligence (2016) 1969–1976.
[57] W. Wang, M. Carreira-Perpinán, Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application, arxiv preprint arXiv:1309.1541.
[58] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, In Proceedings of the International Conference on Neural Information Processing Systems (2001) 849–856.
[59] H. Wang, L. Zong, B. Liu, Y. Yang, W. Zhou, Spectral perturbation meets incomplete multi-view data, In Proceedings of the International Joint Conference on Artificial Intelligence (2019) 3677–3683.

**Jing-Hua Yang** received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2016. He is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Macau University of Science and Technology. Her current research interests include data mining, image processing, and artificial intelligence.

**Lele Fu** received his B.S. degree and M.E. degree from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China in 2019 and 2022, respectively. He is currently pursuing the Ph.D. degree with the School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou, China. His current research interests include machine learning and multi-view learning.

**Chuan Chen** received the Ph.D. degree from the Hong Kong Baptist University in 2016 and worked as a post-doc researcher in the Department of Electrical Engineering, KU Leuven. He is currently an Associate Professor with the School of Computer Science and Engineering, Sun Yat-Sen University. He published over 50 international journal and conference papers. His research interests primarily centered around numerical linear algebra, optimization and their applications in machine learning.

**Hong-Ning Dai** is currently with Department of Computing and Decision Sciences at Lingnan University, Hong Kong as an associate professor. He obtained the Ph.D. degree in Computer Science and Engineering from Department of Computer Science and Engineering at the Chinese University of Hong Kong. His current research interests include the Internet of Things, big data, and blockchain technology. He has served as associate editors/editors for IEEE Transactions on Industrial Informatics, IEEE Systems Journal, IEEE Access, Ad Hoc Networks, and Connection Science. He is also a senior member of Association for Computing Machinery (ACM).

**Zibin Zheng** received the Ph.D. degree from the Chinese University of Hong Kong, in 2011. He is currently a Professor at School of Data and Computer Science with Sun Yat-sen University, China. He serves as Chairman of the Software Engineering Department. He published over 120 international journal and conference papers, including 3 ESI highly-cited papers. According to Google Scholar, his papers have more than 7000 citations, with an H-index of 42. His research interests include blockchain, services computing, software engineering, and financial big data. He was a recipient of several awards, including the Top 50 Influential Papers in Blockchain of 2018, the ACM SIGSOFT Distinguished Paper Award at ICSE2010, the Best Student Paper Award at ICWS 2010. He served as BlockSysa'r19 and CollaborateComa'r16 General Co-Chair, SC2a'r19, ICIOTa'r18 and IoVa'r14 PC Co-Chair.