Incomplete Contrastive Multi-View Clustering with High-Confidence Guiding

Guoqing Chao, Yi Jiang, Dianhui Chu

Harbin Institute of Technology,
2 West Culture Road, Weihai, Shandong 264209, China
guoqingchao10@gmail.com, jiangyijcx@163.com, chudh@hit.edu.cn

Abstract

Incomplete multi-view clustering becomes an important research problem, since multi-view data with missing values are ubiquitous in real-world applications. Although great efforts have been made for incomplete multi-view clustering, there are still some challenges: 1) most existing methods didn't make full use of multi-view information to deal with missing values; 2) most methods just employ the consistent information within multi-view data but ignore the complementary information; 3) For the existing incomplete multi-view clustering methods, incomplete multi-view representation learning and clustering are treated as independent processes, which leads to performance gap. In this work, we proposed a novel Incomplete Contrastive Multi-View Clustering method with high-confidence guiding (ICMVC). Firstly, we proposed a multi-view consistency relation transfer plus graph convolutional network to tackle missing values problem. Secondly, instance-level attention fusion and high-confidence guiding are proposed to exploit the complementary information while instance-level contrastive learning for latent representation is designed to employ the consistent information. Thirdly, an end-to-end framework is proposed to integrate multi-view missing values handling, multi-view representation learning and clustering assignment for joint optimization. Experiments compared with state-of-the-art approaches demonstrated the effectiveness and superiority of our method. Our code is publicly available at https://github.com/liunian-Jay/ICMVC. The version with supplementary material can be found at http://arxiv.org/abs/2312.08697.

Introduction

In practical applications, the majority of data can be considered as multi-view data (Chao, Sun, and Bi 2021; Yang and Wang 2018) and exist missing values due to some uncontrollable factors during data collection, transmission or storage. Incomplete Multi-View Clustering (IMVC) aims to learn a final clustering result based on the multi-view data with missing values.

Many IMVC algorithms have been proposed to solve the missing value problem within multi-view data (Wen et al. 2023). The existing IMVC methods are generally based on non-negative matrix decomposition (Hu and Chen 2018; Chao et al. 2022), subspace-based clustering methods (Wang et al. 2022; Chao et al. 2019), kernel learning (Liu et al. 2019; Ye et al. 2017), graph spectral clustering (Zhou, Wang, and Yang 2019) or deep learning (Lin et al. 2021; Wen et al. 2021; Ke et al. 2023a,b). The popular IMVC methods PVC (Li, Jiang, and Zhou 2014) and MIC (Shao, He, and Yu 2015) conduct multi-view clustering by ignoring the missing values, which leads to inferior performance. There are some other IMVC methods that adopted the missing value imputation methods to fill in the missing position in each view (Wen et al. 2023) and then conducted multi-view clustering. These methods just use the relation between features within each view but ignore the relation between multiple views.

Multi-view fusion plays a vital role in IMVC process, consistent information and complementary information should be exploited to finish this task. However, most of current IMVC methods such as (Xu, Tao, and Xu 2015) take good advantage of the consistent information within multi-view data but ignore the complementary information. Thus how to mine the complementary information within multi-view data and even make full use of both complementary and consistent information is worth further investigation.

Recently, deep IMVC methods have achieved great success due to their powerful representation learning capabilities. Based on auto-encoder (AE), generative adversarial network (GAN), graph neural network (GNN), they first use existing data part to infer and fill in missing data, and then use traditional MVC methods for clustering. For example, Completer (Lin et al. 2021) trains the model with paired instances, and then completes the missing data by dual prediction while SDIMC-net (Wen et al. 2021) uses GNN to tackle incomplete multi-view data problem. Due to the separate processes of missing data handling and multi-view clustering, their performance is not ideal.

To deal with the above problems, we proposed a novel end-to-end incomplete contrastive multi-view clustering method with high-confidence guiding. Firstly, we proposed a multi-view consistency relation transfer plus graph convolutional network (GCN) to handle missing values within multi-view data. Secondly, we designed an instance-level attention module and high-confidence guiding to exploit the complementary information while conducted instance-level contrastive learning for latent representation. Thirdly, we

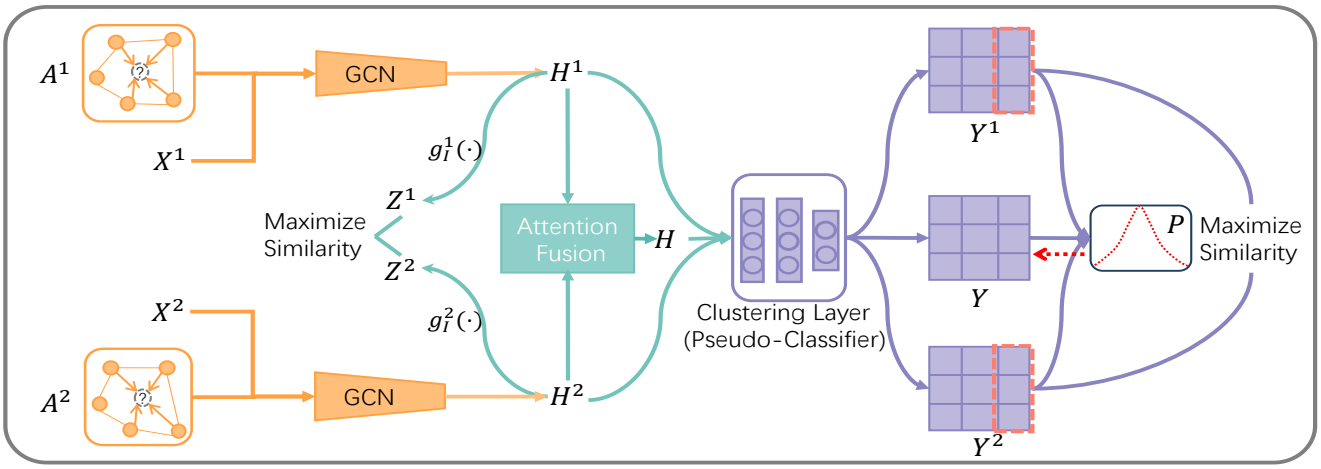

Figure 1: Schematic illustration of the proposed network architecture ICMVC (two views as an example). The three modules are represented with different colors. Firstly, we handle missing data via multi-view consistency relation transfer and adopt GCN to encode incomplete multi-view data, where the representation of missing data is gradually learned with GCN message passing. Secondly, a common representation for multi-view fusion is then obtained through an attention module, while view-specific hidden representations are projected into the embedding space for instance-level comparative learning. Finally, the clustering predictions are obtained through weight-sharing pseudo-classifier and target distributions are computed for high-confidence guidance.

unified multi-view missing data handling, multi-view representation learning and clustering assignment for joint optimization to obtain the clustering result directly. The novelty existing in three stages may motivate more systematic investigation of IMVC.

The Proposed Method

Notations

We define the multi-view data including N instances with V views as $\boldsymbol{X} = \{\boldsymbol{X}^1, \cdots, \boldsymbol{X}^v, \cdots, \boldsymbol{X}^V\}$, and $\boldsymbol{X}^v = \{\boldsymbol{x}_1^v, \cdots, \boldsymbol{x}_i^v, \cdots, \boldsymbol{x}_N^v\} \in \mathbf{R}^{N \times d_v}$ denotes the feature matrix of the v-th view, where $\boldsymbol{x}_i^v$ denotes the v-th view feature of the i-th instance and d_v is the feature dimension of the v-th view. The graph structure of the data is represented as an adjacency matrix $\boldsymbol{A} = \{\boldsymbol{A}^1, \cdots, \boldsymbol{A}^v, \cdots, \boldsymbol{A}^V\}$, and $\boldsymbol{A}_{ij}^v \in \mathbf{R}^{N \times N}$ indicates whether there is an edge between $\boldsymbol{x}_i^v$ and $\boldsymbol{x}_j^v$. For convenience, we will only discuss the two-view case, and the model can be easily extended to more than two views.

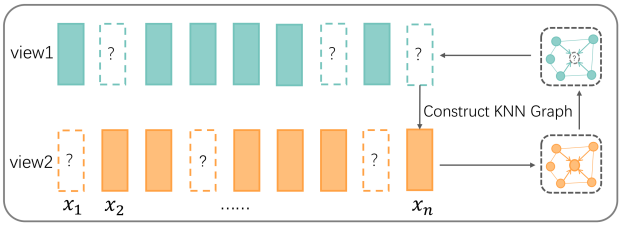

Figure 2: Illustration of the missing view case of multi-view data and how to obtain the adjacency matrix of the missing view via multi-view consistency relation transfer.

Multi-View Missing Handling

To build up the graph structure, we first compute the similarity matrix $\boldsymbol{S}^v \in \mathbf{R}^{n \times n}$ at each view according to the radial basis function: $\boldsymbol{S}_{ij}^v = e^{-\frac{\|\boldsymbol{x}_i^v - \boldsymbol{x}_i^v\|^2}{t}}$, where $\boldsymbol{S}_{ij}^v$ denotes the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from the v-th view. After that, we use the K-nearest neighbors to construct the graph structure $\boldsymbol{A}$ based on the similarity.

Multi-View Consistency Relation Transfer The missing view case of multi-view data and how to obtain the adjacency matrix of the missing view are shown in Figure 2. The dashed lines indicate the missing views, and thus the adjacency relation of the instance from the missing view cannot be obtained. However, based on the consistency assumption between views in multi-view data, the adjacency matrix obtained from the existing view of the same instance can be used as the adjacency matrix for the missing view.

For instance, if the graph structure of instance i is unavailable because its first view information is missing, we can transfer the graph structure of the second view to the first one, namely, $\boldsymbol{A}_i^1 = \boldsymbol{A}_i^2, i = 1, 2, \cdots, N$, where $\boldsymbol{A}_i^1$ and $\boldsymbol{A}_i^2$ denote the i-th row of $\boldsymbol{A}^1$ and $\boldsymbol{A}^2$ respectively.

GCN Missing Handling After obtaining the adjacency matrix, we feed $\boldsymbol{A}^v$ and $\boldsymbol{X}^v$ with missing values into the GCN encoder. During the encoding process, benefiting from the message passing mechanism of GCN, the missing latent representations will be filled in. Specifically, this is estimated using the aggregated information of the neighbors of the missing instance. It should be noted that the missing values in $\boldsymbol{X}^v$ is not involved in the graph convolution operation due to zero in A^v to be multiplied. The GCN encoder module will be explained in detail in the next subsection. For

extensions to the case of more than two views, the union or intersection of the graph structures of any existing views of an instance can be transferred to the missing view.

Network Architecture

As shown in Figure 1, our model architecture consists of three modules: graph encoder module, multi-view fusion module, clustering module. High-confidence guidance is designed to exploit the complementary information within multi-view data. In the following subsections, we will describe them in detail.

Graph Encoder Module The graph auto-encoder captures not only the attribute information of the nodes, but also the graph structure information (Scarselli et al. 2008; Salehi and Davulcu 2020; Wang et al. 2019). To learn the useful representation and deal with the missing data, we use stacked GCN layers as encoders. Specifically, this module consists of V view-specific encoders, which encode V views respectively. The mapping function for the v-th view can be expressed as $f_v(\boldsymbol{A}^v, \boldsymbol{X}^v; \theta^v) \rightarrow \boldsymbol{H}^v$, where $\boldsymbol{H}^v$ is the latent representation for the v-th view, and θ^v indicates the encoder parameter. The graph convolution operation of the m-th layer is represented as

$$\boldsymbol{H}^v_{(m)} = \phi(\tilde{\boldsymbol{D}}^{v^{-\frac{1}{2}}} \tilde{\boldsymbol{A}}^v \tilde{\boldsymbol{D}}^{v^{-\frac{1}{2}}} \boldsymbol{H}^v_{(m-1)} \boldsymbol{W}^v_{(m)}), \tag{1}$$

where $\tilde{\boldsymbol{A}}^v = \boldsymbol{A}^v + \boldsymbol{I}$ and $\tilde{\boldsymbol{D}}^v{}_{ii} = \sum_j \tilde{\boldsymbol{A}}^v{}_{ij}$. $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{W}^v_{(m)}$ indicates the trainable parameters in the m-th layer of the v-th view encoder and ϕ denotes the activation function, such as Relu (Nair and Hinton 2010), etc. Note that here $\boldsymbol{H}^v_{(0)} = \boldsymbol{X}^v$, $\boldsymbol{A}^v$ is the complete graph structure obtained from the v-th view after multi-view consistency relation transfer, and $\boldsymbol{X}^v$ is the feature matrix with missing values from the v-th view. To avoid learning a trivial solution, a skip connection (He et al. 2016) is introduced. Mathematically,

$$\boldsymbol{H}^v_{(m)} = \boldsymbol{H}^v_{(m)} + \boldsymbol{H}^v_{(m-1)}, \tag{2}$$

where $\boldsymbol{H}^v_{(m-1)}$ is the feature representation learned in the $m-1$ layer.

Multi-View Fusion Module Our multi-view fusion module consists of two parts: an instance-level attention module and a view alignment module utilizing instance-level contrastive learning.

Direct concatenation or average fusion of the multi-view features usually leads to sub-optimal clustering performance. In order to make full use of the complementary information of different views, inspired by (Vaswani et al. 2017), we propose a fine-grained instance-level attention approach for automatically perceiving view fusion weights, and its weights can be used to guide the modules in the network to reinforce each other. Specifically, we calculate the multi-view common representation $\boldsymbol{H}$ by the following instance-level attention fusion formula:

$$\boldsymbol{H} = \sum_{v=1}^{V} \boldsymbol{\Lambda}^v \odot \boldsymbol{H}^v, \tag{3}$$

where $\odot$ indicates the element-wise multiplication and $\boldsymbol{\Lambda}^v = [\boldsymbol{\lambda}^v, ..., \boldsymbol{\lambda}^v] \in \mathbf{R}^{N \times d}, v = 1, ...V$. $\boldsymbol{\lambda}^v \in \mathbf{R}^{N \times 1}$ is the attention coefficient computed by

$$\begin{aligned} \widetilde{\boldsymbol{H}} &= [\boldsymbol{H}^1, \boldsymbol{H}^2, ..., \boldsymbol{H}^V], \\ \widetilde{\boldsymbol{G}} &= f_u(\widetilde{\boldsymbol{H}}), \\ [\boldsymbol{\lambda}^1, ..., \boldsymbol{\lambda}^v, ..., \boldsymbol{\lambda}^V] &= \text{sotfmax}(\text{sigmod}(\widetilde{\boldsymbol{G}})/\tau), \end{aligned} \tag{4}$$

where f_u represents a nonlinear mapping like MLP, d is the feature dimension of $\boldsymbol{H}$ and $[\cdot, ..., \cdot]$ denotes the horizontal concatenation of a collection of matrices or vectors along row. Sigmoid function before Softmax operation is a trick to avoid assigning a score close to one to a particular view (Zhou and Shen 2020).

With the instance-level attention module, complementary information within multi-view data is adequately utilized. In order to fully explore the consistency information, view representation distribution alignment is introduced as an auxiliary task of regularizing the encoder to keep the local geometry structure (Trosten et al. 2021). To achieve this goal, we employ instance-level contrastive learning. Specifically, we project the potential representations of two views into the embedding space using two instance-level contrastive heads (ICH) consisting of two stacked nonlinear MLPs, $\boldsymbol{z}^v_i = g^v_I(\boldsymbol{h}^v_i)$.

In the embedding space, we use different views of the same instance to construct positive pairs and different instances to construct negative pairs. We achieve view alignment by maximizing the similarity between positive instance pairs while minimizing the similarity between negative instance pairs. To achieve this goal, the loss for the first view of the ith instance $\boldsymbol{x}^1_i$ is given as follows:

$$\ell^1_i = -\log \frac{\exp\left(s(\boldsymbol{z}^1_i, \boldsymbol{z}^2_i)/\tau_I\right)}{\sum_{j=1}^{N} \left[\exp\left(s(\boldsymbol{z}^1_i, \boldsymbol{z}^1_j)/\tau_I\right) + \exp\left(s(\boldsymbol{z}^1_i, \boldsymbol{z}^2_j)/\tau_I\right)\right]}, \tag{5}$$

where τ_I is the instance-level temperature parameter that controls the softness. For the whole dataset, the loss function is given as follows:

$$\mathcal{L}_{ins} = \frac{1}{2N} \sum_{i=1}^{N} (\ell^1_i + \ell^2_i). \tag{6}$$

Clustering Module To directly obtain the cluster labels and achieve the end-to-end joint optimization, we design a pseudo-classifier that shares the weights among different views for clustering. Instances can be passed through a pseudo-classifier to obtain the soft assignments to clusters, ie. $\boldsymbol{Y}^v = g_C(\boldsymbol{H}^v)$, where $\boldsymbol{Y}^v \in \mathbf{R}^{N \times C}$ and C is the numbers of the clusters. For the convenience of expression, we denote the j-th column of $\boldsymbol{Y}^v$ as $\boldsymbol{y}^v_j$:

$$y^v_j = \begin{bmatrix} \boldsymbol{Y}^v_{1j} \\ ... \\ \boldsymbol{Y}^v_{Nj} \end{bmatrix}, \tag{7}$$

$\boldsymbol{y}_j^v$ collects the probability values of all the instances assigned to the j-th cluster, which can represent the corresponding cluster. Same with (Huang, Gong, and Zhu 2020), we call it cluster-wise Assignment Statistics Vector (ASV).

ASVs from different clusters should be mutually exclusive (and ideally orthogonal), while ASVs from different views of the same cluster should be consistent. To achieve this goal, we extend contrastive clustering to multi-view data. Specifically, the contrastive loss for the i-th ASV in the first view is computed as follows:

$$\hat{\ell}_j^1 = -\log \frac{\exp(s(\boldsymbol{y}_j^1, \boldsymbol{y}_j^2)/\tau_C)}{\sum_{k=1}^{C} [\exp(s(\boldsymbol{y}_j^1, \boldsymbol{y}_k^1)/\tau_C) + \exp(s(\boldsymbol{y}_j^1, \boldsymbol{y}_k^2)/\tau_C)]}, \tag{8}$$

where τ_C is the cluster-level temperature parameter controlling the softness. For the whole data, the cluster-level contrastive loss is represented as

$$\mathcal{L}_{clu} = \frac{1}{2C} \sum_{j=1}^{C} (\hat{\ell}_j^1 + \hat{\ell}_j^2) - H(\boldsymbol{Y}^1) - H(\boldsymbol{Y}^2), \tag{9}$$

where $H(\boldsymbol{Y}^v) = -\sum_{j=1}^{C} [P(\boldsymbol{y}_j^v) \log P(\boldsymbol{y}_j^v)]$ is the information entropy and $P(\boldsymbol{y}_j^v) = \frac{1}{N} \sum_{t=1}^{N} \boldsymbol{Y}_{tj}^v$. Maximizing entropy is the strategy to avoid assigning all samples to the same cluster (Hu et al. 2017; Huang, Gong, and Zhu 2020).

High-Confidence Guiding For unsupervised learning task, some works introduce self-supervised auxiliary objectives, such as DEC (Xie, Girshick, and Farhadi 2016). For the probability assignment of the clustering module output, we hope to design an auxiliary objective to achieve three goals: 1) Instances with consistency and high confidence can be self-supervised to further enhance their own representation learning. 2) Instances with one view of a high-confidence probability assignment and other views of approximately uniform distribution assignment can be self-supervised to obtain a cluster assignment with a high-confidence probability. 3) For instances with all views of the approximately uniform distribution assignment, the auxiliary target can further blur them to position them at the cluster boundary.

Inspired by the above motivation, a target guidance that emphasizes the high-confidence instances is designed and introduced and it can take advantage of the complementary information. Specifically, each latent representation obtained from each view and their fused representation are fed into a weight-sharing pseudo-classifiers to obtain multiple cluster assignments. That is, $\boldsymbol{Y}^v = g_C(\boldsymbol{H}^v), v = 1, 2$ and $\boldsymbol{Y} = g_C(\boldsymbol{H})$. g_C indicates the weigh-sharing pseudo-classifier. Based on these cluster assignment results, we can obtain the target assignment:

$$\boldsymbol{q}_{ij} = \max\{\boldsymbol{Y}_{ij}^1, \boldsymbol{Y}_{ij}^2, \boldsymbol{Y}_{ij}\}, \tag{10}$$

which guarantees the high-confidence instance be emphasized. The confident target assignment with highest probability will be chosen as the target assignment. Highly confident assignments are enhanced and the instances at cluster boundary are further blurred by the following actions:

$$\boldsymbol{p}_{ij} = \frac{\boldsymbol{q}_{ij}^2}{\sum_{j=1}^{k} \boldsymbol{q}_{ij}^2}, \tag{11}$$

which will be used as the auxiliary target distribution to guide the clustering results, and the optimization objective is formulated as

$$\mathcal{L}_{hg} = KL(\boldsymbol{Y} \| \boldsymbol{P}) = \sum_i \sum_j \boldsymbol{p}_{ij} \log \frac{\boldsymbol{p}_{ij}}{\boldsymbol{y}_{ij}}. \tag{12}$$

Algorithm 1 Optimization of the proposed ICMVC

Input: Dataset $\boldsymbol{X} = \{\boldsymbol{X}^v, v = 1, ..., V\}$, numbers of cluster C, hyper-parameter K, temperature parameter τ_I, τ_C, total iteration numbers $epoches$

Output: Clustering predictions

```
Calculate the adjacency matrix A^1, A^2
Transfer adjacency matrix to the missing views.
for epoch = 1 → epoches do
    Compute the representations and soft assignment
      H^1 = f_1(X^1, A^1), Z^1 = g_I(H^1), Y^1 = g_C(H^1)
      H^2 = f_2(X^2, A^2), Z^2 = g_I(H^2), Y^2 = g_C(H^2)
    Compute fused representations H by equation (3)
    Compute the soft assignment Y = g_C(H)
    Calculate the loss L_ins by equation (6)
    Calculate the loss L_clu by equation (9)
    Calculate the target distribution P by(10)(11)
    Calculate the loss L_hg by equation (12)
    Calculate the overall loss L by equation (13)
    Update through gradient descent to minimize L
end for
return Y //output
```

Objective Function

Our model is an end-to-end clustering method that does not require prior pre-training to complete missing views (Lin et al. 2021), nor does it require pre-training to initialize cluster centroids (Xie, Girshick, and Farhadi 2016), nor does it require additional k means clustering to obtain the final clustering results. Thus we can simultaneously optimize the whole model, the total loss is represented as follows:

$$\mathcal{L} = \mathcal{L}_{ins} + \mathcal{L}_{clu} + \mathcal{L}_{hg}. \tag{13}$$

Although trade-off parameters can be added to balance the different losses, we just enforced the same weights and achieved good performance. Algorithm 1 summarizes the training process.

Experiments

Experimental Settings

We implement ICMVC in PyTorch 1.12.1 and conduct all the experiments on Ubuntu 20.04 with NVIDIA 2080Ti GPU. The Adam optimizer is adopted, and the learning rate is set to 0.001, the hyper-parameter K is set to 10. The instance-level temperature parameter τ_I is fixed at 1.0, and

η	Method	Scene-15			LandUse-21			MSRC-V1			Noisy MNIST		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
0	BSV	26.88	25.38	11.47	19.70	22.46	6.40	63.43	53.80	44.46	54.40	48.49	37.12
	$Concat$	28.93	28.11	12.85	19.24	24.11	6.91	67.33	58.72	49.49	44.56	46.64	31.83
	PVC	30.39	32.76	15.67	26.93	31.40	12.60	70.67	63.19	54.43	40.83	35.50	23.00
	MIC	33.09	32.75	16.51	22.95	28.86	9.41	74.57	65.29	57.88	32.79	31.91	16.02
	$DAIMC$	29.08	26.20	12.47	24.33	29.25	10.44	78.76	69.32	62.73	39.68	37.11	24.95
	$Completer$	33.55	31.84	18.18	25.32	30.28	10.32	74.19	62.55	53.40	81.82	82.44	74.76
	$DSIMVC$	19.95	17.63	7.95	19.70	21.39	6.67	47.14	39.53	25.09	85.59	80.10	76.51
	$DIMVC$	27.92	23.45	12.91	24.27	31.32	11.56	77.05	70.82	63.22	51.38	52.66	38.65
	$Ours$	**38.29**	**36.13**	**21.60**	**27.76**	**31.57**	**14.50**	**89.24**	**79.94**	**77.50**	**97.94**	**94.57**	**95.51**
0.3	BSV	26.47	23.54	9.30	18.53	20.89	5.17	60.19	50.10	37.58	49.99	46.35	34.08
	$Concat$	26.78	25.06	10.42	17.38	21.36	5.34	60.00	49.22	36.98	44.96	45.30	31.58
	PVC	29.51	27.90	13.62	24.80	28.04	10.64	50.10	44.44	29.06	45.32	35.19	24.03
	MIC	28.32	28.18	12.28	21.90	25.79	7.81	68.29	59.45	49.05	29.26	28.84	12.76
	$DAIMC$	26.49	22.25	10.40	22.84	25.93	8.47	76.76	68.51	60.82	39.67	33.79	22.60
	$Completer$	31.90	30.24	17.12	24.09	**31.24**	7.35	71.24	61.93	52.09	79.96	80.08	73.57
	$DSIMVC$	19.78	17.44	7.82	18.87	19.91	6.01	48.10	39.0	24.52	73.83	67.25	60.06
	$DIMVC$	28.26	22.73	12.82	23.70	29.14	10.05	73.43	63.38	55.24	57.30	55.89	44.11
	$Ours$	**36.20**	**34.21**	**19.69**	**26.94**	29.67	**12.92**	**87.72**	**76.97**	**74.03**	**96.82**	**91.96**	**93.13**

Table 1: The clustering results of nine methods on four complete datasets and incomplete datasets with missing rate $\eta = 0.3$, the 1^{st} and 2^{nd} best results are indicated in bold and underlined, respectively.

the cluster-level parameter τ_C is fixed at 0.5. We observe that it can fully converges after 500 epoches after training the network, thus 500 epoches is set to terminate.

To evaluate the proposed method on incomplete multi-view data, we randomly delete one view to construct missing data. The missing rate η is defined as $\eta = (n-m)/n$, where m is the number of complete instances and n is the number of all instances.

Datasets and Metrics

We used four commonly-used datasets in our experiments to evaluate our model. **Scene-15**: It consists of 4,485 images distributed in 15 scene categories with GIST and LBP features as two views. **LandUse-21**: It consists of 2100 satellite images from 21 categories with two views: PHOG and LBP. **MSRC-V1**: It is an image dataset consisting of 210 images in seven categories, including trees, buildings, airplanes, cows, faces, cars, and bicycles, with GIST and HOG features as two views. **Noisy MNIST**: the original images are used as view 1, and the sampled intra-class images with Gaussian white noise are used as view 2, and we use its subset containing 10k samples in the experiments.

To evaluate the performance, three commonly-used clustering metrics: accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI) are adopted. The larger these metrics are, the better the clustering performance.

Baselines and Experimental Results

In order to verify the effectiveness and superiority of our method, two commonly-used single-view clustering methods and six state-of-the-art IMVC methods are used to compare with our proposed method ICMVC. These methods are listed as follows:

- **BSV**: It performs k means clustering algorithm individually on each view and picks the best one as the final result.
- **Concat**: It concatenates all the views and then performs the k means clustering algorithm to obtain the result.
- **PVC** (Li, Jiang, and Zhou 2014): It is a method based on NMF to learn the latent representations of incomplete multi-view data in subspaces.
- **MIC** (Shao, He, and Yu 2015): It learns a latent representation for each view based on weighted NMF, and uses co-regularization to obtain a common representation.
- **DAIMC** (Hu and Chen 2018): It learns a consistent representation for all views using weighted NMF via the missing indicator matrix.
- **Completer** (Lin et al. 2021): It uses dual predictions to complete the latent representations for missing views, and then performs k means clustering algorithm on the latent representations.
- **DSIMVC** (Tang and Liu 2022): It achieves safe IMVC by dynamically filling in missing views from the learned semantic neighborhoods.
- **DIMVC** (Xu et al. 2022): It is an imputation-free and fusion-free IMVC method implemented via mining cluster complementarity.

Note that for the baseline methods BSV and Concat, we impute the missing values using the average feature value of that corresponding view of the existing instances. For the other compared methods, the hyper-parameters are set to the recommended values in their original papers.

We set the missing rate $\eta = 0.3$ to compare all the methods, and we also run them on the complete data. To avoid

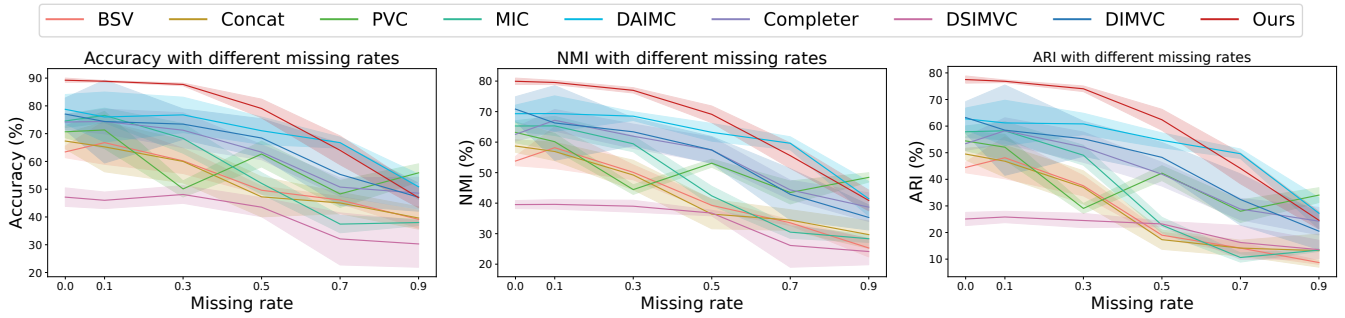

Figure 3: Error band plot of performance on MSRC-V1 as missing rate increases, with padding representing the standard deviation.

randomness, we run all the methods five times, and then reported the average value of the clustering results, as shown in Table 1.

From Table 1, we can find that compared with two single-view methods, our proposed method ICMVC performs better both in complete and incomplete settings. This demonstrates the effectiveness of multi-view fusion and missing value handling. Compared with other six state-of-the-art baselines, ICMVC outperforms them almost in every case. In particular, on the MSRC-V1 dataset with missing rate 0.3, our method outperforms the state-of-the-art baselines by 10.96%, 8.46%, and 13.21% in measures ACC, NMI, and ARI, respectively. This verifies the superiority of our proposed ICMVC.

Performance with Different Missing Rates

In order to further verify the effectiveness of our model, we vary the missing rate from 0.1 to 0.9 with an interval 0.2 on MSRC-V1, and take the results on complete dataset as the starting point.

The error band diagram is shown in Figure 3. It can be clearly seen that at low missing rates, our method outperforms other baseline methods by far and the performance does not drop significantly as the missing rate increases. However, the performance of our method degrades quickly when the missing rate is higher than 0.5. We guess that when the missing rate is too high, it is difficult to get the correct graph structure through the multi-view consistent relation transfer, thus the performance declines quickly.

Furthermore, we have observed that the standard deviation of the deep learning baseline methods is large, and we guess it may be easy to fall into a local optimal solution for deep learning methods due to some random reasons such as parameter initialization. In comparison, the standard deviation of our method is relatively small, because our model has less dependence on the randomness of the parameter initialization and training process, and thus is robust.

Ablation Study

To investigate and explore the effectiveness of each module of our model, we performed the ablation study experiments on MSRC-V1 with missing rate $\eta = 0.3$. Specifically, we separate each module and then retrain each one to conduct experiments. It should be noted that since we use a pseudo-classifier to predict cluster labels, if $\mathcal{L}_{clu}$ is removed, the clustering results will no longer be valid, and target guidance will lose its value. Thus we remove these two modules simultaneously, and then perform k means clustering on the latent representation. The experimental results are shown in Table 2. It can be clearly seen that each module plays an important role.

η	$\mathcal{L}_{ins}$	$\mathcal{L}_{hg}$	$\mathcal{L}_{clu}$	ACC	NMI	ARI
0.3	✓	✓	✓	**87.72**	**76.97**	**74.03**
	✗	✓	✓	80.38	70.30	64.35
	✓	✗	✓	85.62	76.27	72.56
	✓	✗	✗	70.67	63.94	46.83

Table 2: Ablation study on MSRC-V1. The guidance module is only valid if the clustering module exists, and if $\mathcal{L}_{clu}$ is removed we use k means to perform clustering.

Visualization Analysis

To compare with other competitive methods, we use t-SNE to visualize the latent representations of the other top method on MSRC-v1 with $\eta = 0.3$, as shown in Figure 5. We can find that the latent representation obtained from DAIMC looks good, but there are still some overlaps between different clusters. In comparison, our method ICMVC performs better, which maybe because it makes full use of the consistent and complementary information within multi-view data.

Based on the learned latent representations $\boldsymbol{H}, \boldsymbol{H^1}, \boldsymbol{H^2}$, we compute the similarity matrix using cosine distance and visualize them in Figure 4. From Figure 4, we can clearly see the block structure of the matrix. That is, the instance similarity within the same cluster is high, and the instance similarity between different clusters is low. We also observed that the similarity matrix obtained from fused view has better structure than that from each view. For example, there are some samples that are difficult to distinguish in the single view, such as the red box in view 1 and the yellow box in view 2. Instance-level attention fusion and high-confidence guiding take full advantage of this complementarity, and the resulting representations are all discriminative, as seen from the fused view.

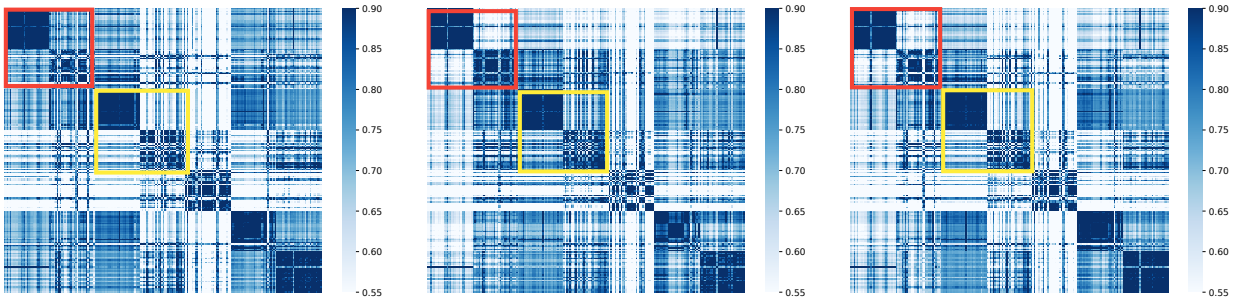

Figure 4: Visualization of the similarity matrix obtained from each view and fused view of MSRC-V1 with missing rate 0.3.

Parameter Sensitivity Analysis

In this study, there is only one hyper-parameter K when constructing the graph structure in multi-view missing data handling stage. We conduct parameter sensitivity analysis for K on MSRC-V1 with missing rate 0.3, as shown in Figure 6. Since there are 30 instances for each cluster on MSRC-V1, we set K from 1 to 30 as shown in x axis, and we demonstrate the clustering performance by varying K, as shown in y axis in Figure 6. It is easy to see that when K is too small or too large, the clustering performance is bad. This may be because when K is small, too few neighbors cannot capture the precise structural information and when K is too large, a wrong adjacency relation will be established. We think the K value will be in a reasonable range if it take values not too small and not more than half of the number of instances in a cluster. Within the reasonable range, the model performance fluctuates slightly with the K value, and the overall performance is stable, thus our model is insensitive to the K value within this suitable range.

Convergence Analysis

We show the convergence of ICMVC on MSRC-V1 with a missing rate 0.3 in Figure 7. As shown in Figure 7, we can clearly observe that in the first 100 epoches, the loss drops rapidly and ACC, NMI, and ARI rise steadily. After that, as the number of epoches increase, the loss decreases slowly with fluctuations, while ACC, NMI, and ARI rise steadily and eventually converge.

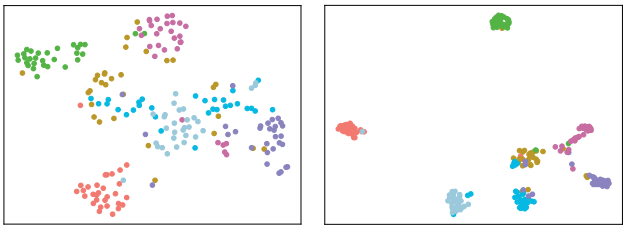

Figure 5: t-SNE visualization of the latent representations learned by DAIMC and ICMVC on MSRC-V1.

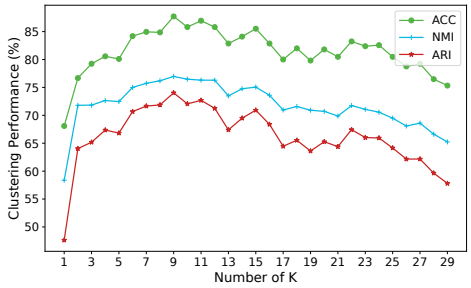

Figure 6: Illustration of the sensitivity analysis of K on MSRC-V1, x-axis represents the number of K, and y-axis represents the clustering performance.

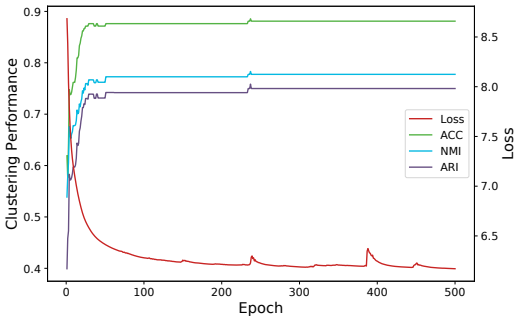

Figure 7: Illustration of the convergence analysis, x-axis represents number of training epoches, and left and right x-axes represent clustering performance and loss, respectively.

Conclusion

We propose an end-to-end incomplete multi-view clustering method with high-confidence guiding. We designed a multi-view consistency relation transfer plus GCN to handle the missing values and exploit the complementary information by designing instance-level attention and high-confidence guiding. Moreover, our method unites multi-view missing data handling, multi-view representation learning and clustering into a unified framework to obtain the clustering results without pre-training or post-processing. Experimentsaa verified the effectiveness and superiority of ICMVC.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62276079), Young Teacher Development Fund of Harbin Institute of Technology IDGA10002071, Research and Innovation Foundation of Harbin Institute of Technology IDGAZMZ00210325, Key Research and Development Plan of Shandong Province 2021SFGC0104 and the Special Funding Program of Shandong Taishan Scholars Project.

References

Chao, G.; Sun, J.; Lu, J.; Wang, A.-L.; Langleben, D. D.; Li, C.-S.; and Bi, J. 2019. Multi-view cluster analysis with incomplete data to understand treatment effects. *Information Sciences*, 494.

Chao, G.; Sun, S.; and Bi, J. 2021. A survey on multiview clustering. *IEEE Transactions on Artificial Intelligence*, 2(2): 146–168.

Chao, G.; Wang, S.; Yang, S.; Li, C.; and Chu, D. 2022. Incomplete multi-view clustering with multiple imputation and ensemble clustering. *Applied Intelligence*, 52(13): 14811–14821.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 770–778.

Hu, M.; and Chen, S. 2018. Doubly aligned incomplete multi-view clustering. In *International Joint Conference on Artificial Intelligence*, 2262–2268.

Hu, W.; Miyato, T.; Tokui, S.; Matsumoto, E.; and Sugiyama, M. 2017. Learning discrete representations via information maximizing self-augmented training. In *International Conference on Machine Learning*, 1558–1567.

Huang, J.; Gong, S.; and Zhu, X. 2020. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 8849–8858.

Ke, G.; Chao, G.; Wang, X.; Xu, C.; Zhu, Y.; and Yu, Y. 2023a. A Clustering-guided Contrastive Fusion for Multiview Representation Learning. *IEEE Transactions on Circuits and Systems for Video Technology*.

Ke, G.; Yu, Y.; Chao, G.; Wang, X.; Xu, C.; and He, S. 2023b. Disentangling Multi-view Representations Beyond Inductive Bias. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2582–2590.

Li, S.-Y.; Jiang, Y.; and Zhou, Z.-H. 2014. Partial multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1968–1974.

Lin, Y.; Gou, Y.; Liu, Z.; Li, B.; Lv, J.; and Peng, X. 2021. Completer: Incomplete multi-view clustering via contrastive prediction. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 11174–11183.

Liu, X.; Zhu, X.; Li, M.; Wang, L.; Zhu, E.; Liu, T.; Kloft, M.; Shen, D.; Yin, J.; and Gao, W. 2019. Multiple kernel k-means with incomplete kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(5): 1191–1204.

Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 807–814.

Salehi, A.; and Davulcu, H. 2020. Graph attention auto-encoders. In *International Conference on Tools with Artificial Intelligence (ICTAI)*, 989–996.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1): 61–80.

Shao, W.; He, L.; and Yu, P. S. 2015. Multiple Incomplete Views Clustering via Weighted Nonnegative Matrix Factorization with $L_{2,1}$. In *Lecture Notes in Computer Science*, 318–334.

Tang, H.; and Liu, Y. 2022. Deep safe incomplete multi-view clustering: Theorem and algorithm. In *International Conference on Machine Learning*, 21090–21110.

Trosten, D. J.; Lokse, S.; Jenssen, R.; and Kampffmeyer, M. 2021. Reconsidering representation alignment for multi-view clustering. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1255–1265.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Neural Information Processing Systems*, 6000–6010.

Wang, C.; Pan, S.; Hu, R.; Long, G.; Jiang, J.; and Zhang, C. 2019. Attributed graph clustering: A deep attentional embedding approach. In *International Joint Conference on Artificial Intelligence*, 3670–3676.

Wang, S.; Chen, Y.; Yi, S.; and Chao, G. 2022. Frobenius norm-regularized robust graph learning for multi-view subspace clustering. *Applied Intelligence*, 52(13): 14935–14948.

Wen, J.; Wu, Z.; Zhang, Z.; Fei, L.; Zhang, B.; and Xu, Y. 2021. Structural deep incomplete multi-view clustering network. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, 3538–3542.

Wen, J.; Zhang, Z.; Fei, L.; Zhang, B.; Xu, Y.; Zhang, Z.; and Li, J. 2023. A survey on incomplete multiview clustering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(2): 1136–1149.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, 478–487.

Xu, C.; Tao, D.; and Xu, C. 2015. Multi-view learning with incomplete views. *IEEE Transactions on Image Processing*, 24(12): 5812–5825.

Xu, J.; Li, C.; Ren, Y.; Peng, L.; Mo, Y.; Shi, X.; and Zhu, X. 2022. Deep incomplete multi-View clustering via mining cluster complementarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 8761–8769.

Yang, Y.; and Wang, H. 2018. Multi-view clustering: A survey. *Big Data Mining and Analytics*, 1(2): 83–107.

Ye, Y.; Liu, X.; Liu, Q.; and Yin, J. 2017. Consensus kernel-means clustering for incomplete multiview data. *Computational Intelligence and Neuroscience*, 2017: 1–11.

Zhou, R.; and Shen, Y.-D. 2020. End-to-end adversarial-attention network for multi-modal clustering. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 14619–14628.

Zhou, W.; Wang, H.; and Yang, Y. 2019. Consensus graph learning for incomplete multi-view clustering. In *Lecture Notes in Computer Science*, 529–540.