Federated Multi-view Graph Clustering with Incomplete Attribute Imputation

Wei Feng[1] , **Zeyu Bi**[2*] , **Qianqian Wang**[3] and **Bo Dong**[4]
[1]College of Information Engineering, Northwest A&F University, Yangling, China
[2]School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
[3]School of Telecommunications Engineering, Xidian University, Xi'an, China
[4]School of Continuing Education, Xi'an Jiaotong University, Xi'an, China
wei.feng@nwafu.edu.cn, bizeyu@stu.xjtu.edu.cn, qqwang@xidian.edu.cn, dong.bo@xjtu.edu.cn

Abstract

Federated Multi-View Clustering (FedMVC) aims to uncover consistent clustering structures from distributed multi-view data for clustering while preserving data privacy. However, existing FedMVC methods under vertical settings either ignore the ubiquitous incomplete view issue or require uploading data features, which may lead to privacy leakage or induce high communication costs. To mitigate the view incompleteness issue and simultaneously maintain privacy and efficiency, we propose a novel **F**ederated **M**ulti-**v**iew **G**raph **C**lustering with **I**ncomplete **A**ttribute **I**mputation (FMVC-IAI). This method constructs a consensus graph structure through complementary multi-view data and then utilizes a non-parametric graph neural network (GNN) to impute missing features. Additionally, it utilizes the adjacency graph as the knowledge carrier to share and fuse the multi-view information. To alleviate the high communication cost due to graph sharing, we proposed to share the anchor graph for global adjacency graph construction, which reduces communication cost and also helps to reduce privacy leakage risk. Extensive experiments demonstrate the superiority of our method in FedMVC tasks with incomplete views.

1 Introduction

Multi-view data refers to data derived from different perspectives or sources to describe the same object. Each view provides a distinct aspect of information. Multi-view data have become ubiquitous across various real-world applications [Liang *et al.*, 2025; Yang *et al.*, 2022], such as remote sensing and social networking. In practice, it is expensive to obtain reliable labels for massive and complex multi-view data, which makes multi-view clustering (MVC) receive considerable attention from researchers in recent years as an unsupervised method [Huang *et al.*, 2023; Zhao *et al.*, 2025]. By exploiting consistent and complementary information from various views, MVC generally exhibits superior performance to single-view approaches.

There are numerous MVC methods developed. For instance, Xiao *et al.* [2023] employ graph neural networks (GNNs) as a feature fusion module to better integrate multi-view features, and Teng *et al.* [2024] use transformers to extract consensus features for MVC, both achieved good performance. However, in reality, multi-view data collected from different sources is often held by separate organizations, with each organization owning only one view of the data. This situation introduces two challenges: (1) Due to privacy concerns, data sharing between views is not permitted, and (2) the independence of data collection leads to instability in data quality, specifically with some views of certain samples being incomplete or missing. Traditional MVC methods, which rely on the assumption of centralized data distribution, are unable to effectively address these two real-world challenges [Chen *et al.*, 2023b].

To overcome these challenges, recent studies introduced federated learning into MVC and developed several Federated Multi-view Clustering (FedMVC) approaches. Huang *et al.* [2022] proposed an efficient FedMVC method based on matrix factorization and developed a federated optimization framework. Effective as it is, this method cannot obtain the deep features as a shallow model, resulting in suboptimal clustering performance. Therefore, Chen *et al.* [2023b] proposed a deep FedMVC method that employs deep learning to extract high-level features and introduced a missing data completion strategy based on prototypes, effectively improving the clustering performance. However, it requires each client to upload local embedded features to the server and may be vulnerable to model inversion attacks [Sun *et al.*, 2025] and might produce high communication overhead. To alleviate the privacy issues, Sun *et al.* [2025] proposed a structured graph learning framework for FedMVC, which avoided uploading embedding features and reduced privacy risks. However, the method does not consider the possible incomplete view problem and requires passing a $N \times N$ graph between the client and the server, resulting in a high communication overhead on large-scale datasets.

Therefore, we propose a novel deep FedMVC method based on graph convolutional network (GCN), *i.e.,*, **F**ederated **M**ulti-**v**iew **G**raph **C**lustering with **I**ncomplete **A**ttribute **I**mputation (FMVC-IAI). First, it utilizes the global struc-

*Corresponding Author

tural information to impute incomplete attributes based on GNN, thus mitigating the view incompleteness issue. Then, it uses Graph Autoencoder (GAE) for feature extraction to extract high-order features. To defend against model inversion attacks, inspired by [Zhang *et al.*, 2022], we extract view-specific anchor graphs instead of embedded features as knowledge carriers to support collaborative training. This method not only reduces privacy leakage risks but also significantly lowers communication overhead from $\mathcal{O}(N^2)$ to $\mathcal{O}(Nr)$, where r is the anchor number smaller than N. The main contributions of the work can be summarized as follows:

- We propose a novel FedMVC method that employs graph structures from other clients to impute the local missing attributes, which addresses the view-incompleteness problems under federated settings.
- We leverage the anchor graph for cross-client knowledge sharing, which helps to reduce the communication overhead and improve model performance.
- Our method effectively addresses data missing scenarios in federated environments, particularly under complex conditions with high missing rates. Extensive comparative experiments demonstrate its superiority.

2 Related Works

2.1 Multi-view Clustering

Multi-view clustering (MVC) is a method that utilizes multiple views of data to uncover consistent clustering structures. Its main challenge lies in extracting consistent information from multi-view features. Existing multi-view clustering methods can be broadly categorized into heuristic methods and deep learning-based methods [Chen *et al.*, 2023a]. Heuristic approaches include matrix factorization-based [Zhao *et al.*, 2017; Li *et al.*, 2023a], graph-based [Liang *et al.*, 2022; Zhao *et al.*, 2024], and kernel-based methods [Wang *et al.*, 2023; Wu *et al.*, 2024]. However, these methods typically extract shallow features and are not suitable for complex data.

Therefore, deep learning is introduced to MVC owing to its impressive deep feature extraction capacities. [Ren *et al.*, 2024b; Chen *et al.*, 2025a; Chen *et al.*, 2025b]. For example, Xu *et al.* [2021] proposed a deep MVC method based on a deep autoencoder, which leverages a feature fusion mechanism to extract consistent information across views. [Lin *et al.*, 2024] utilized a deep graph autoencoder to extract both structural and attribute features, enabling effective multi-view clustering for graph data. Some works take into account the incomplete multi-view clustering problem [Liu *et al.*, 2023], Pu *et al.* [2024] proposed an adaptive completion method. It first uses a deep autoencoder to extract embedding features, then estimates the embedding features of the missing data by leveraging cross-view soft clustering assignments and global cluster centroids, thereby achieving data completion. Although these methods have achieved excellent MVC performance, they are all based on the assumption of centralized data distribution. They do not consider the practical requirement where view data is distributed across different holders and cannot be accessed mutually.

2.2 Federated Multi-view Clustering

To address the above real-world challenges, Federated Multi-View Clustering methods [Huang *et al.*, 2022; Li *et al.*, 2023b; Qiao *et al.*, 2024] have been proposed. Inspired by vertical federated learning[Liu *et al.*, 2024], these methods aim to extract consensus information from distributed data while ensuring data privacy is preserved throughout the process. Huang *et al.* [2022] first combines multi-view learning with federated learning, proposing a federated multi-view clustering (FedMVC) model called FedMVL. To address challenges such as stragglers and fault tolerance in federated learning, it derives an iterative joint optimization algorithm that enables each node to flexibly handle subproblems. Chen *et al.* [2023b] addressed the issues of data heterogeneity and missing data in federated scenarios by proposing a federated multi-view clustering method called FedDMVC, which uploads autoencoder embedding features to the server to achieve feature completion. Building upon this, Ren *et al.* [2024a] further addressed the issue of sample misalignment in federated learning. While existing research has made significant progress, directly uploading embedding features poses a risk, as it is susceptible to model inversion attacks [Sun *et al.*, 2021] that could lead to privacy leakage.

3 Proposed Method

3.1 Problem Statement

A typical FedMVC under vertical settings can be defined as follows. Suppose there are M clients, with the m-th client identified by $\mathcal{C}_m$ holding the m-th view data $\mathbf{X}^{(m)}$. Most existing FedMVC methods assume that multi-view data $\mathbf{X}$ held by different entities are complete, *i.e.,* for a certain sample, all the clients hold its data with the corresponding views. However, it is almost impossible that all the samples have complete views, since the data are collected by different entities for different purposes. We introduce an mask vector $\mathbf{b}^{(m)} \in \mathbb{R}^{N\times 1}$ for $\mathcal{C}_m$, where $\mathbf{b}_i^{(m)} = 1$ indicates that $\mathcal{C}_m$ holds the data of the i-th sample. Besides, there exists a server $\mathcal{S}$ responsible for coordinating the collaborative training among various clients. Our design goal is to enable the clients to collaboratively train a FedMVC model that partitions data into K groups with the assistance of the server $\mathcal{S}$, ensuring low privacy risk and reasonable communication cost.

Overview: Each client preprocesses local data via a certain model according to the data characteristics (like Autoencoder, Convolutional Autoencoder, *etc*). Taking the autoencoder (AE) as an example, each client employs AE to process local data with a simple reconstruction loss for noise elimination and dimension reduction. The subsequent federated model training requires several rounds of interactions. In each round, it imputes the missing attributes via a simple feature diffusion process with a global adjacency graph. Subsequently, each client further extracts high-level embeddings with an Graph Autoencoder (GAE) network, and performs local clustering with the global supervision produced by $\mathcal{S}$ and produces local structural information and clustering information, which are sent to $\mathcal{S}$ to generate global adjacency graph

and global supervision. The general framework of FMVC-IAI is illustrated in Figure 1.

3.2 Client-side Design

Each client maintains a local model composed of four parts: incomplete-view imputation, GAE-based embedding, local clustering module with global supervision, and interaction with the server. whose details are as follows.

Incomplete-view Imputation

The missing views of some samples in different clients hinder the clustering. Chen *et al.* [2023b] addresses this issue by introducing global supervision modules and imputing the missing views. However, this method requires uploading the local data features, which may introduce privacy risks. Inspired by [Rossi *et al.*, 2022], we build a simple feature completion method for missing features based on GNN with globally fused graph $\mathbf{A}^g$ generated by $\mathcal{S}$. $\mathcal{C}_m$ leverages AE to map the raw data $\mathbf{X}^{(m)}$ into a latent feature space to obtain its low-level feature $\mathbf{Z}^{(m)}$ via a reconstruction loss before imputation. Then, the missing sample features in $\mathbf{Z}^{(m)}$ are filled with zero vectors, making the size of $\mathbf{Z}^{(m)}$ equal to $N \times d_z$. Finally, each client $\mathcal{C}_m$ leverages the global graph and local attributes for missing attribute completion. The process of feature completion is analogous to a multi-source heat diffusion process, where each existing sample acts as an independent heat source, diffusing its features along the edges of the global graph to complete all the missing data. First, we calculate the diffusion matrix $\tilde{\mathbf{A}}$ from $\mathbf{A}_g$ by:

$$\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}^g\mathbf{D}^{-\frac{1}{2}} \tag{1}$$

where $\mathbf{D}$ is the degree matrix of $\mathbf{A}^g$. Suppose $\mathbf{B}^{(m)} = diag(\mathbf{b}^{(m)})$, where $\mathbf{I}$ is an identity matrix. To impute the missing samples, we adopt the method in [Rossi *et al.*, 2022] to produce the missing attributes by minimizing the Dirichlet energy. By initializing the imputed feature matrix $\tilde{\mathbf{Z}}_0^{(m)} = \mathbf{Z}^{(m)}$, the iterative imputation process can be formalized as a feature propagation process below:

$$\tilde{\mathbf{Z}}_{t+1}^{(m)} = (\mathbf{I} - \mathbf{B}^{(m)})\tilde{\mathbf{A}}\tilde{\mathbf{Z}}_t^{(m)} + \mathbf{B}^{(m)}\tilde{\mathbf{Z}}_t^{(m)} \tag{2}$$

where $\tilde{\mathbf{Z}}^{(m)}$ is the imputed feature matrix. Note that a high-quality global graph structure plays a crucial role. Therefore, we update the graph structure $\mathbf{A}^g$ after each communication round to achieve better feature completion, which is illustrated in details in the subsequent part. At the beginning, $\mathbf{A}^g$ can be initialized by $\mathbf{A}^g = \sum_{m=1}^{M}\mathbf{A}^{(m)}$, where $\mathbf{A}^{(m)}$ is the local adjacency graph constructed from $\mathbf{Z}^{(m)}$.

GAE-based Embedding

This module incorporates GAE to further extract high-level embeddings from the imputed attributes and global structures. First, the graph attention network (GAT) is introduced to process $\hat{\mathbf{Z}}^{(m)}$ as follows:

$$\mathbf{H}^{(m)} = \text{GAT}(\tilde{\mathbf{Z}}^{(m)}, \mathbf{A}^g) \tag{3}$$

where $\mathbf{H}^{(m)} \in \mathbb{R}^{N\times d_h}$ represents the high-level features extracted by the GAE. Then, we utilize MLP as decoder to reconstruct the attributes and structure as follows:

$$\begin{aligned}\hat{\mathbf{Z}}^{(m)} &= \text{MLP}(\mathbf{H}^{(m)}) \\ \hat{\mathbf{A}}^{(m)} &= \sigma(\mathbf{H}^{(m)}{\mathbf{H}^{(m)}}^T)\end{aligned} \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid activation function. Finally, we use the MSE loss as the feature reconstruction loss term and the binary cross-entropy loss as the structural reconstruction loss. The full reconstruction loss is defined as:

$$\begin{aligned}\mathcal{L}_r = &\|\hat{\mathbf{Z}}^{(m)} - \tilde{\mathbf{Z}}^{(m)}\|_2^2 - \\ &\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\mathbf{A}_{ij}^g \log(\hat{\mathbf{A}}_{ij}^{(m)}) + (1-\mathbf{A}_{ij}^g)\log(1-\hat{\mathbf{A}}_{ij}^{(m)})\right]\end{aligned} \tag{5}$$

Local Clustering with Global Supervision

$\mathcal{C}_m$ leverages a clustering layer with learnable parameters $\mathbf{U}^{(m)} \in \mathbb{R}^{K\times d_h}$ to obtain the local clustering assignment matrix $\mathbf{Q}^{(m)} \in \mathbb{R}^{N\times K}$.

$$q_{ij}^{(m)} = \frac{(1 + \|\mathbf{h}_i^{(m)} - \mathbf{u}_j^{(m)}\|_2^2)^{-1}}{\sum_{j=1}^{K}(1 + \|\mathbf{h}_i^{(m)} - \mathbf{u}_j^{(m)}\|_2^2)^{-1}} \tag{6}$$

where he element $q_{ij}^{(m)}$ corresponds to the entry located at the i-th row and j-th column of the matrix $\mathbf{Q}^{(m)}$. Similarly, $\mathbf{h}_i^{(m)}$ and $\mathbf{u}_j^{(m)}$ denote the row vectors of matrices $\mathbf{H}^{(m)}$ and $\mathbf{U}^{(m)}$, respectively.

Then, each client $\mathcal{C}_m$ leverages the global cluster assignment matrix $\mathbf{P}$ sent by the server to compute the KL divergence between $\mathbf{Q}^{(m)}$ and $\mathbf{P}$ as the clustering loss, enabling global information to guide the local model training:

$$\mathcal{L}_c = D_{KL}(\mathbf{P}\|\mathbf{Q}^{(m)}) \tag{7}$$

Thus, the overall loss function on the client side is defined by **Eq.** (8).

$$\mathcal{L} = \mathcal{L}_r + \alpha\mathcal{L}_c \tag{8}$$

where α is a hyperparameter that balances the contribution of the clustering loss and the reconstruction loss.

3.3 Client-Server Interaction

To assist $\mathcal{S}$ to construct the global adjacency graph and supervision information, $\mathcal{C}_m$ is required to transfer local adjacency graph and clustering assignment to $\mathcal{S}$. However, directly transmitting the local graph constructed from $\mathbf{H}^{(m)}$ results in limited information sharing, while transmitting the distance matrix incurs an expensive communication overhead of $\mathcal{O}(N^2)$, which is not suitable for communication sensitive scenarios and large-scale data. To mitigate this problem and inspired by [Zhang *et al.*, 2022], we propose to construct the adjacency graph from anchor graphs and share the anchor graph instead of a full adjacency graph. To be specific, we adopt the method described in [Li *et al.*, 2020] to extract the

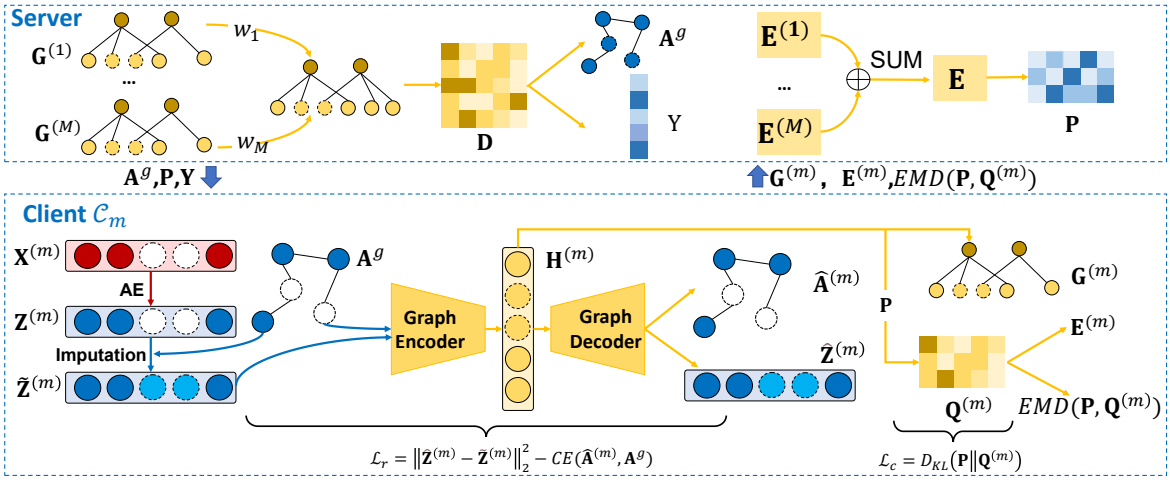

Figure 1: Framework of our method, which contains a global server and M local clients. At the beginning of each round, the server sends the global pseudo-labels $\mathbf{Y}$, the clustering assignment matrix $\mathbf{P}$, and the global graph $\mathbf{A}^g$ to each client. The clients then perform independent local feature completion and training, and upload their local anchor graphs $\mathbf{G}^{(m)}$, prototype distance matrices $\mathbf{E}^{(m)}$ and $EMD(\mathbf{P}, \mathbf{Q}^{(m)})$ to the server for global information updates.

anchor graph $\mathbf{G}^{(m)} \in \mathbb{R}^{N\times r}$ from $\mathbf{H}^{(m)}$ and transmit it to $\mathcal{S}$, reducing communication cost to $\mathcal{O}(Nr)$ from $\mathcal{O}(N^2)$.

In addition, to enable $\mathcal{S}$ to generate the global clustering assignment $\mathbf{P}$, $\mathcal{C}_m$ further computes the local prototype $\mathbf{F}^{(m)} \in \mathbb{R}^{K\times d_h}$ based on the pseudo-labels $\mathbf{Y}$ from the server and calculates matrix $\mathbf{E}^{(m)} \in \mathbb{R}^{N\times K}$ between local samples to the prototypes as in **Eq.** (9):

$$\begin{aligned} \mathbf{F}_k^{(m)} &= \frac{1}{|S_k^{(m)}|} \sum_{\mathbf{h}_i^{(m)} \in S_k^{(m)}} \mathbf{h}_i^{(m)} \\ e_{ij}^{(m)} &= \|\mathbf{h}_i^{(m)} - \mathbf{F}_j^{(m)}\|_2^2 \end{aligned} \tag{9}$$

where $S_k^{(m)} = \{\mathbf{h}_i^{(m)}, y_i | y_i = k\}_{i=1}^N$ is the set of samples with pseudo-label k on the m-th client $\mathcal{C}_m$, $e_{ij}^{(m)}$ is the element in the i-th row and j-th column of the matrix $\mathbf{E}^{(m)}$.

3.4 Server-side Clustering

Construction of Global Adjacency Graphs.

After receiving the anchor graphs $\mathbf{G}^{(m)}$ uploaded by the clients, $\mathcal{S}$ performs a weighted fusion as illustrated below:

$$\mathbf{G} = \sum_{m=1}^{M} w^{(m)} \mathbf{G}^{(m)} \tag{10}$$

where $w^{(m)}$ is the weight of $\mathcal{C}_m$, which measures the contribution of $\mathbf{G}^{(m)}$. We believe that $\mathbf{Q}^{(m)}$ closer to the global assignment $\mathbf{P}$ include more consistent information and should be assigned a larger weight. Therefore, we adopt Earth Mover's Distance (EMD) to measure the distance between the previous $\mathbf{P}$ and $\mathbf{Q}^{(m)}$ of $\mathcal{C}_m$ as the evidence to determine $w^{(m)}$:

$$w^{(m)} = \frac{e^{-EMD(\mathbf{P},\mathbf{Q}^{(m)})}}{\sum_{i=1}^{M} e^{-EMD(\mathbf{P},\mathbf{Q}^{(i)})}} \tag{11}$$

The global adjacency graph $\mathbf{A}^g$ by performing KNN on the distance matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ computed from the aggregated anchor graph $\mathbf{G}$ by $\mathbf{D}_{ij} = \|\mathbf{G}_i - \mathbf{G}_j\|_2^2$. In practice, $EMD(\mathbf{P}, \mathbf{Q}^{(m)})$ can be computed locally by $\mathcal{C}_m$ and then transmitted to $\mathcal{S}$ for lower communication cost.

Global Clustering

$\mathcal{S}$ is also responsible for generating the global clustering assignment and pseudo labels. $\mathcal{S}$ first aggregates all $\mathbf{E}^{(m)}$ from $\mathcal{S}_m$ into $\mathbf{E} = \sum_{m=1}^{M} \mathbf{E}^{(m)}$. We measure the similarity between global embedded features and global prototypes by converting their Euclidean distances into conditional probabilities using the Student's t-distribution:

$$t_{ij} = \frac{(1 + e_{ij})^{-1}}{\sum_j (1 + e_{ij})^{-1}} \tag{12}$$

where e_{ij} represents the element in the i-th row and j-th column of the matrix $\mathbf{E}$. Then, the global clustering assignment matrix $\mathbf{P}$ can be obtained by **Eq.** (13):

$$p_{ij} = \frac{(t_{ij}/\sum_j t_{ij})^2}{\sum_j (t_{ij}/\sum_j t_{ij})^2} \tag{13}$$

Finally, $\mathcal{S}$ performs spectral clustering on $\mathbf{D}$ to obtain pseudo label matrix $\mathbf{Y}$, and sends $\mathbf{A}^g$, $\mathbf{P}$ and $\mathbf{Y}$ to all the clients. It should be noted that $\mathbf{A}^g$ is a sparse (N, N) matrix, which can be compressed into a dense (N, k) matrix, so the data size to be transmitted for $\mathbf{A}^g$ is (N, k), where k is the number of neighbors in KNN. We summarize the workflow of our method in Algorithm 1.

3.5 Communication Overhead Analysis

We analyze the communication overhead of our method from two aspects. First, during the data upload phase from the client to the server, the data to be uploaded includes $\mathbf{E}^{(m)}$, $\mathbf{G}^{(m)}$ and $EMD(\mathbf{P}, \mathbf{Q}^{(m)})$, which respectively incurs

Algorithm 1 FMVC-IAI

Input: The incomplete multiview data $\{\mathbf{X}^{(m)}\}_{m=1}^{M}$ distributed in M clients, the number of clustering clusters K, the number of communication rounds R.
Output: Clustering result $\mathbf{Y}$

Each client $\mathcal{C}_m$ initializes KNN graph.
$\mathcal{S}$ initializes $\mathbf{P}$, $\mathbf{Y}$, $\mathbf{A}^g$
for Not reaching R rounds **do**
  ▷ **Local Training on** $\mathcal{C}_m$
  for Each client $\mathcal{C}_m$ in parallel **do**
    Generate $\mathbf{Z}^{(m)}$ from $\mathbf{X}^{(m)}$ with AE
    Impute features with global graph $\mathbf{A}^g$.
    Local training with loss in **Eq**. (8) to obtain $\mathbf{H}^{(m)}$.
    Extract the anchor graph $\mathbf{G}^{(m)}$ from $\mathbf{H}^{(m)}$.
    Update $\mathbf{E}^{(m)}$ using $\mathbf{Y}$ and $\mathbf{H}^{(m)}$.
    Upload $\mathbf{E}^{(m)}$ and $\mathbf{G}^{(m)}$ to $\mathcal{S}$.
  end for
  ▷ **Global Fusion on** $\mathcal{S}$
  $\mathcal{S}$ update $\mathbf{P}$ by **Eq.** (13).
  $\mathcal{S}$ update $\mathbf{Y}$
  $\mathcal{S}$ update $\mathbf{A}^g$ using the KNN and $\mathbf{G}^{(m)}$.
  $\mathcal{S}$ distribute $\mathbf{P}$, $\mathbf{A}^g$ and $\mathbf{Y}$ to each client $\mathcal{C}_m$.
end for

the communication cost of $\mathcal{O}(rN)$, $\mathcal{O}(NK)$, and $\mathcal{O}(1)$, resulting a total cost of $\mathcal{O}(NK + Nr)$. During the data transmission from the server to the clients, the data to be downloaded includes $\mathbf{P}, \mathbf{Y}$ and $\mathbf{A}^g$, which respectively produce a cost of $\mathcal{O}(NK)$, $\mathcal{O}(N)$, and $\mathcal{O}(Nk)$, leading to a total cost of $\mathcal{O}(NK + Nk + N)$. Therefore, the overall communication complexity is $\mathcal{O}(Nr+2NK+Nk+N)$, which is much smaller than $\mathcal{O}(N^2)$.

4 Experiments

4.1 Experimental Settings

Datasets and Metrics

We conducted experiments on three multi-view datasets:

- **HW** [Winn and Jojic, 2005] comprises multi-feature data for digits 0 through 9, with 200 samples for each class. For each binarized handwritten digit image, six different view features were extracted: 1. mfeat-fou, 2. mfeat-fac, 3. mfeat-kar, 4. mfeat-pix, 5. mfeat-zer, and 6. mfeat-mor.
- **OutdoorScene** [Monadjemi *et al.*, 2002] contains 15 scene categories with both indoor and outdoor environments, 4485 images in total. Following [Jiang *et al.*, 2024], we select 8 outdoor categories with total 2,688 images with four views.
- **NoisyMNIST** [Wang *et al.*, 2015] View 1 consists of the original MNIST images, whereas view 2 is comprised of intra-class images that have been sampled and embedded with Gaussian white noise. Following [Chao *et al.*, 2024], we utilize a subset that includes 10,000 samples.

Data Setting: Each view in the dataset is distributed across different clients, and clients cannot access each other's data. Additionally, following [Feng *et al.*, 2024], we set the missing rate of η, randomly select ηN missing samples, and delete half of the view data to simulate missing data with different missing rates in various federated scenarios.

Metrics: We use three widely used clustering metrics for evaluation, *i.e.*, Accuracy (ACC), Normalized Mutual Information(NMI), and Adjusted Rand Index(ARI).

Comparison Methods

To validate the effectiveness and superiority of our method, we conducted comparisons with seven state-of-the-art IMVC methods, which include five centralized incomplete multiview clustering methods: **PIC** [Wang *et al.*, 2019], **DAIMC** [Hu and Chen, 2018], **COMPLETER** [Lin *et al.*, 2021], **DSIMVC** [Tang and Liu, 2022], and **CPSPAN** [Jin *et al.*, 2023]; and two recent FedMVC methods: **MGCD** [Sun *et al.*, 2025] and **FCUIF** [Ren *et al.*, 2024a], under missing rate settings of η at [0.1, 0.3, 0.5, 0.7, 0.9].

4.2 Experimental Results

Tables **??** to **??** provide detailed clustering performance comparisons of our method and the benchmark methods across different missing rates and datasets. To eliminate random variations, we conducted five repeated experiments for all methods and reported the average values. The results are presented with the mean value first, followed by the MAD (Mean Absolute Deviation). Additionally, the best results are highlighted in bold, and the second-best results are underlined.

The experimental results demonstrate that, compared to the two most recent federated IMVC methods, our approach achieves the best performance in most cases. In contrast to MGCD, which does not account for missing data, our method outperforms it in all missing rate scenarios. This indicates that missing data can severely impact the clustering performance of federated IMVC.

Compared to FCUIF, our method outperforms it on most datasets and missing rates. Particularly on the HW dataset, while our method shows no significant advantage at low missing rates, at high missing rates, FCUIF experiences a considerable performance drop, whereas the performance decline of our method is smaller than that of FCUIF. This demonstrates that our feature completion approach is effective. Moreover, our method avoids uploading feature embeddings, reducing the risk of privacy leakage compared to existing deep IMVC methods.

To visually observe the impact of missing rate changes on clustering performance, we present the error band diagram of all clustering metrics and comparison methods on the HW dataset in Figure 2. As can be seen from the figure, under different missing rate settings, our method consistently maintains high clustering performance. Particularly in high missing rate scenarios ($\eta = 0.9$), our method even outperforms all benchmark methods, including centralized methods, proving the robustness of our feature completion approach.

4.3 Ablation Study

We validate the effectiveness of each module in the proposed method through ablation experiments. The experimental results are shown in Table **??**. The four modules considered

Dataset	η	Centralized					Federated		
		PIC	DAIMC	COMPLETER	DSIMVC	CPSPAN	MGCD	FCUIF	OURS
HW	0.1	82.85±0.0	86.90±0.0	74.88±12.81	84.06±0.23	90.98±1.90	78.35±0.0	96.17±0.24	**97.64±0.03**
	0.3	71.20±0.0	80.95±0.0	71.19±4.34	79.23±3.88	90.19±1.57	56.08±0.0	94.81±0.21	**97.87±0.06**
	0.5	85.15±0.0	80.80±0.0	64.38±12.52	76.80±3.70	89.39±1.63	39.29±0.0	93.87±0.06	**97.21±0.07**
	0.7	80.80±0.0	71.00±0.0	74.44±5.26	76.08±2.55	89.92±1.82	26.06±0.0	91.29±0.31	**96.19±0.19**
	0.9	87.40±0.0	70.25±0.0	40.66±3.83	46.09±1.97	90.61±1.11	14.06±0.0	87.72±0.48	**96.01±0.21**
OutdoorScene	0.1	63.88±0.0	51.45±0.0	42.02±4.04	58.97±2.19	60.54±2.83	43.73±0.0	**71.61±1.31**	70.80±0.65
	0.3	61.61±0.0	44.64±0.0	48.84±3.20	59.46±0.93	58.91±2.84	31.11±0.0	**72.14±0.92**	71.14±0.85
	0.5	59.67±0.0	50.29±0.0	53.22±2.40	58.07±1.55	62.59±3.44	21.35±0.0	**71.23±0.79**	71.10±0.43
	0.7	61.42±0.0	51.64±0.0	56.06±5.97	53.65±2.29	60.24±2.45	16.43±0.0	68.74±0.44	**69.31±0.44**
	0.9	53.35±0.0	50.63±0.0	44.14±5.37	39.90±1.49	60.83±3.18	17.90±0.0	63.62±2.24	**66.05±0.52**
NoisyMNIST	0.1	**97.29±0.0**	39.78±0.0	77.31±7.10	80.20±7.55	47.01±0.89	28.17±0.0	68.42±1.23	80.23±1.55
	0.3	**86.33±0.0**	37.31±0.0	78.41±2.53	78.59±2.56	46.22±0.63	20.38±0.0	68.75±2.20	83.21±1.59
	0.5	**80.51±0.0**	33.43±0.0	61.59±1.59	75.19±2.40	46.11±0.91	16.52±0.0	63.43±0.94	77.35±1.97
	0.7	**77.05±0.0**	43.72±0.0	54.89±2.42	59.07±3.91	45.62±0.57	14.21±0.0	57.03±1.77	70.39±0.70
	0.9	63.09±0.0	38.46±0.0	33.33±3.94	39.35±4.32	44.26±0.87	20.86±0.0	53.86±1.51	**70.96±2.40**

Table 1: Comparison of clustering accuracy (ACC) under different missing rates.

Dataset	η	Centralized					Federated		
		PIC	DAIMC	COMPLETER	DSIMVC	CPSPAN	MGCD	FCUIF	OURS
HW	0.1	82.93±0.0	77.18±0.0	74.07±9.86	81.20±0.56	83.92±2.13	75.75±0.0	91.44±0.39	**94.44±0.08**
	0.3	75.90±0.0	70.64±0.0	71.94±3.69	77.15±3.16	83.02±1.98	53.22±0.0	88.88±0.41	**95.98±0.13**
	0.5	84.53±0.0	70.21±0.0	69.23±8.47	73.11±2.27	82.28±1.89	35.41±0.0	87.65±0.18	**93.65±0.18**
	0.7	83.03±0.0	68.64±0.0	74.24±1.51	70.71±2.26	83.01±1.82	16.08±0.0	83.34±0.47	**91.50±0.38**
	0.9	85.92±0.0	68.89±0.0	44.70±1.83	50.83±4.70	83.36±1.46	1.31±0.0	77.72±0.72	**91.48±0.37**
OutdoorScene	0.1	53.19±0.0	43.30±0.0	43.04±2.53	51.45±0.47	50.49±1.99	33.34±0.0	**56.78±0.78**	54.14±0.67
	0.3	53.78±0.0	36.80±0.0	42.00±1.50	50.70±0.22	50.43±1.76	20.86±0.0	**56.78±0.64**	55.28±1.20
	0.5	50.18±0.0	38.16±0.0	43.46±1.78	49.84±0.72	52.73±0.83	8.80±0.0	55.39±0.68	**55.30±0.65**
	0.7	43.35±0.0	39.20±0.0	48.02±2.92	46.07±1.51	51.13±1.24	0.85±0.0	53.25±0.41	**53.50±0.43**
	0.9	35.93±0.0	32.88±0.0	35.96±4.41	30.31±1.95	49.36±2.08	4.46±0.0	47.16±1.15	**48.24±0.66**
NoisyMNIST	0.1	**93.24±0.0**	30.78±0.0	76.29±2.76	74.52±4.43	46.27±1.17	19.16±0.0	60.58±1.23	75.59±1.79
	0.3	**84.42±0.0**	31.82±0.0	72.28±2.22	70.53±1.16	45.64±0.74	14.72±0.0	58.60±1.51	74.73±0.84
	0.5	71.71±0.0	26.50±0.0	63.39±2.29	67.08±2.05	45.32±1.47	7.56±0.0	51.28±0.64	**73.15±0.59**
	0.7	**71.79±0.0**	32.79±0.0	54.25±2.14	52.43±3.39	44.32±1.20	4.02±0.0	44.32±0.73	62.86±0.65
	0.9	60.10±0.0	27.96±0.0	31.42±3.43	32.93±5.34	41.25±1.44	17.02±0.0	40.68±1.57	**61.17±2.07**

Table 2: Comparison of clustering normalized mutual information (NMI) under different missing rates.

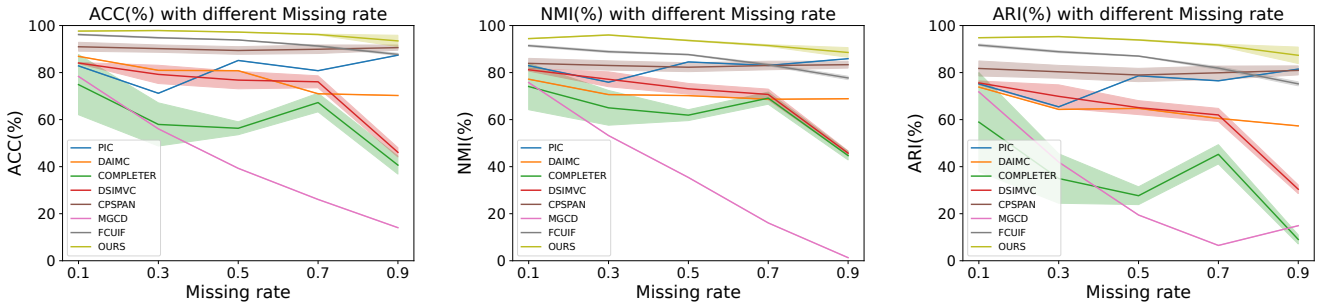

Figure 2: Error band plot of performance as missing rate increases

are: M1 represents the GCN-based feature completion module, M2 represents the weighted fusion module, and M3 and M4 represent the feature reconstruction loss and the structural reconstruction loss in GAE, respectively. The ablation experiment results show that, regardless of whether the missing rate is high or low, M4 has a significant impact on the clustering results. At high missing rates, all modules are indispensable for clustering performance. This demonstrates the effective-

Dataset	η	Centralized					Federated		
		PIC	DAIMC	COMPLETER	DSIMVC	CPSPAN	MGCD	FCUIF	OURS
HW	0.1	75.20±0.0	73.84±0.0	58.95±21.04	75.61±0.83	81.75±3.20	71.78±0.0	91.68±0.49	**94.82±0.07**
	0.3	65.44±0.0	64.34±0.0	51.129±9.3	69.89±4.90	80.34±2.72	41.96±0.0	88.86±0.42	**95.30±0.12**
	0.5	78.64±0.0	64.78±0.0	42.63±14.27	65.10±2.97	78.97±2.83	19.44±0.0	87.01±0.13	**93.87±0.17**
	0.7	76.50±0.0	60.53±0.0	58.70±4.43	61.99±2.78	79.91±2.86	6.51±0.0	81.90±0.62	**91.73±0.40**
	0.9	81.48±0.0	57.33±0.0	9.05±1.93	30.40±1.84	80.93±1.93	14.90±0.0	75.21±0.85	**91.41±0.43**
OutdoorScene	0.1	46.39±0.0	34.84±0.0	21.19±3.60	37.24±1.29	38.67±2.44	24.87±0.0	**49.23±0.84**	46.56±0.92
	0.3	44.76±0.0	28.94±0.0	21.77±3.50	36.91±0.53	38.08±1.93	16.22±0.0	**49.37±0.89**	46.56±0.89
	0.5	41.22±0.0	30.74±0.0	29.09±3.70	36.00±1.07	41.64±1.30	5.74±0.0	48.00±0.91	**48.79±0.40**
	0.7	35.64±0.0	32.71±0.0	35.01±4.43	32.22±1.56	39.58±1.89	0.15±0.0	44.96±0.51	**45.19±0.48**
	0.9	31.36±0.0	26.47±0.0	24.50±4.92	19.11±1.24	38.68±2.49	1.45±0.0	38.81±1.46	**40.13±0.63**
NoisyMNIST	0.1	**94.10±0.0**	20.73±0.0	67.10±5.26	69.01±6.82	31.94±1.25	11.63±0.0	52.82±1.50	69.08±1.57
	0.3	**82.04±0.0**	19.94±0.0	66.05±2.73	64.73±2.07	31.32±0.88	7.22±0.0	51.43±2.26	69.82±1.48
	0.5	70.88±0.0	15.22±0.0	49.47±2.10	60.41±2.81	31.04±1.58	2.82±0.0	45.03±0.75	**66.53±1.59**
	0.7	**66.18±0.0**	22.12±0.0	41.35±3.21	42.58±3.61	31.21±1.10	1.86±0.0	37.60±1.34	56.69±0.70
	0.9	50.11±0.0	18.41±0.0	12.00±4.39	17.63±1.54	27.62±1.20	7.50±0.0	33.67±1.53	**54.69±2.26**

Table 3: Comparison of clustering adjusted rand index (ARI) under different missing rates.

η	M1	M2	M3	M4	ACC	NMI	ARI
0.1		✓	✓	✓	96.40	90.48	90.90
	✓		✓	✓	97.30	93.94	94.09
	✓	✓		✓	97.35	93.95	94.19
	✓	✓	✓		87.85	86.81	82.57
	✓	✓	✓	✓	**97.64**	**94.44**	**94.82**
0.9		✓	✓	✓	93.47	88.55	87.35
	✓		✓	✓	87.75	87.29	83.11
	✓	✓		✓	89.90	85.42	82.02
	✓	✓	✓		74.10	76.79	64.67
	✓	✓	✓	✓	**96.01**	**91.48**	**91.41**

Table 4: Ablation study on HW.

ness of each module in our proposed method.

4.4 Parameter Analysis

We conducted experiments to perform parameter analysis on several key parameters involved in the proposed method. The experimental results are shown in Figure 3 and Figure 4. Specifically, we tested the clustering performance on the HW dataset under a missing rate setting of $\eta = 0.9$, varying the following parameters: The number of neighbors k used for KNN graph construction, ranging from 5 to 45 with an interval of 5. The maximum number of communication rounds R, ranging from 1 to 15 with an interval of 1. The clustering loss balance coefficient $\alpha \in [10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100]$ The number of selected anchors r, ranging from 10 to 1000.

From Figure 3, it can be observed that the clustering performance is optimal when k is around 20. As R increases, the clustering performance initially improves and then stabilizes at a maximum value. From Figure 4, it is evident that α maintains stable clustering performance within the range of 0.1 to 10, and a small number of anchor points (more than 120 on HW) in each cluster is sufficient to achieve stable performance for our method.

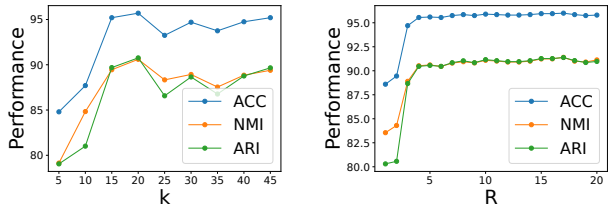

Figure 3: Clustering performance with different k and R

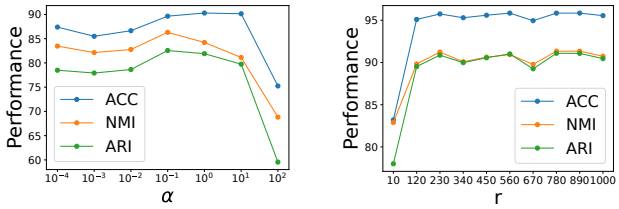

Figure 4: Clustering performance with different α and r

5 Conclusion

In this paper, we propose a federated multi-view clustering method with incomplete attribute imputation. First, a global consensus graph is constructed from the existing data, and clients use the global graph structure information to perform local feature completion. Besides, to reduce privacy leakage risks and communication overhead, we extract an anchor graph from the autoencoder embeddings to replace the direct uploading of embedding features or graphs. Finally, weighted fusion of the anchor graphs is performed on the server to obtain the global clustering assignment, which guides the client model training for self-supervised learning. Experimental results demonstrate the superiority of our method, particularly the robustness in high missing rate scenarios.

Acknowledgments

This work is supported by the National Science Foundation of China under Grant No. 62037001, the Key Research and Development Project in Shaanxi Province (2024PT-ZCK-89), the Fundamental Research Funds for the Central Universities (ZYTS25267, QTZX25004), and the Science and Technology Project of Xi'an (Grant 2022JH-JSYF-0009), Open Project of Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, Anhui University (No. MMC202416), Selected Support Project for Scientific and Technological Activities of Returned Overseas Chinese Scholars in Shaanxi Province 2023-02, and the Xidian Innovation Fund (Project NoYJSJ25007).

References

[Chao *et al.*, 2024] Guoqing Chao, Yi Jiang, and Dianhui Chu. Incomplete contrastive multi-view clustering with high-confidence guiding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11221–11229, 2024.

[Chen *et al.*, 2023a] Jie Chen, Hua Mao, Wai Lok Woo, and Xi Peng. Deep multiview clustering by contrasting cluster assignments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16752–16761, 2023.

[Chen *et al.*, 2023b] Xinyue Chen, Jie Xu, Yazhou Ren, Xiaorong Pu, Ce Zhu, Xiaofeng Zhu, Zhifeng Hao, and Lifang He. Federated deep multi-view clustering with global self-supervision. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3498–3506, 2023.

[Chen *et al.*, 2025a] Jianpeng Chen, Yawen Ling, Jie Xu, Yazhou Ren, Shudong Huang, Xiaorong Pu, Zhifeng Hao, Philip S Yu, and Lifang He. Variational graph generator for multiview graph clustering. *IEEE Trans. on Neural Networks and Learning Systems*, 2025.

[Chen *et al.*, 2025b] Man-Sheng Chen, Xi-Ran Zhu, Jia-Qi Lin, and Chang-Dong Wang. Contrastive multiview attribute graph clustering with adaptive encoders. *IEEE Trans. Neural Networks Learn. Syst.*, 36(4):7184–7195, 2025.

[Feng *et al.*, 2024] Cong Feng, Ao Li, Haoyue Xu, Hailu Yang, and Xinwang Liu. Deep incomplete multiview clustering via local and global pseudo-label propagation. *IEEE Trans. on Neural Networks and Learning Systems*, 2024.

[Hu and Chen, 2018] Menglei Hu and Songcan Chen. Doubly aligned incomplete multi-view clustering. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2262–2268, 2018.

[Huang *et al.*, 2022] Shudong Huang, Wei Shi, Zenglin Xu, Ivor W Tsang, and Jiancheng Lv. Efficient federated multiview learning. *Pattern Recognition*, 131:108817, 2022.

[Huang *et al.*, 2023] Zongmo Huang, Yazhou Ren, Xiaorong Pu, Shudong Huang, Zenglin Xu, and Lifang He. Self-supervised graph attention networks for deep weighted multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7936–7943, 2023.

[Jiang *et al.*, 2024] Zhangqi Jiang, Tingjin Luo, and Xinyan Liang. Deep incomplete multi-view learning network with insufficient label information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12919–12927, 2024.

[Jin *et al.*, 2023] Jiaqi Jin, Siwei Wang, Zhibin Dong, Xinwang Liu, and En Zhu. Deep incomplete multi-view clustering with cross-view partial sample and prototype alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11600–11609, 2023.

[Li *et al.*, 2020] Xuelong Li, Han Zhang, Rong Wang, and Feiping Nie. Multiview clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 44(1):330–344, 2020.

[Li *et al.*, 2023a] Guopeng Li, Dan Song, Wei Bai, Kun Han, and Ratnasingham Tharmarasa. Consensus and complementary regularized non-negative matrix factorization for multi-view image clustering. *Information Sciences*, 623:524–538, 2023.

[Li *et al.*, 2023b] Songze Li, Duanyi Yao, and Jin Liu. Fedvs: Straggler-resilient and privacy-preserving vertical federated learning for split models. In *International Conference on Machine Learning*, pages 20296–20311. PMLR, 2023.

[Liang *et al.*, 2022] Weixuan Liang, Xinwang Liu, Sihang Zhou, Jiyuan Liu, Siwei Wang, and En Zhu. Robust graph-based multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7462–7469, 2022.

[Liang *et al.*, 2025] Xinyan Liang, Pinhan Fu, Yuhua Qian, Qian Guo, and Guoqing Liu. Trusted multi-view classification via evolutionary multi-view fusion. In *Proceedings of the 13th International Conference on Learning Representations*, pages 1–14, 2025.

[Lin *et al.*, 2021] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. Completer: Incomplete multi-view clustering via contrastive prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[Lin *et al.*, 2024] Jia-Qi Lin, Man-Sheng Chen, Xi-Ran Zhu, Chang-Dong Wang, and Haizhang Zhang. Dual information enhanced multiview attributed graph clustering. *IEEE Trans. on Neural Networks and Learning Systems*, 2024.

[Liu *et al.*, 2023] Chengliang Liu, Jie Wen, Zhihao Wu, Xiaoling Luo, Chao Huang, and Yong Xu. Information recovery-driven deep incomplete multiview clustering network. *IEEE Trans. on Neural Networks and Learning Systems*, 2023.

[Liu *et al.*, 2024] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin

Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Trans. on Knowledge and Data Engineering*, 2024.

[Monadjemi *et al.*, 2002] Amir Monadjemi, BT Thomas, and Majid Mirmehdi. Experiments on high resolution images towards outdoor scene classification. 2002.

[Pu *et al.*, 2024] Jingyu Pu, Chenhang Cui, Xinyue Chen, Yazhou Ren, Xiaorong Pu, Zhifeng Hao, S Yu Philip, and Lifang He. Adaptive feature imputation with latent graph for deep incomplete multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14633–14641, 2024.

[Qiao *et al.*, 2024] Dong Qiao, Chris Ding, and Jicong Fan. Federated spectral clustering via secure similarity reconstruction. *Advances in Neural Information Processing Systems*, 36, 2024.

[Ren *et al.*, 2024a] Yazhou Ren, Xinyue Chen, Jie Xu, Jingyu Pu, Yonghao Huang, Xiaorong Pu, Ce Zhu, Xiaofeng Zhu, Zhifeng Hao, and Lifang He. A novel federated multi-view clustering method for unaligned and incomplete data fusion. *Information Fusion*, 108:102357, 2024.

[Ren *et al.*, 2024b] Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, S Yu Philip, and Lifang He. Deep clustering: A comprehensive survey. *IEEE Trans. on Neural Networks and Learning Systems*, 2024.

[Rossi *et al.*, 2022] Emanuele Rossi, Henry Kenlay, Maria I Gorinova, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features. In *Learning on graphs conference*, pages 11–1. PMLR, 2022.

[Sun *et al.*, 2021] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9311–9319, 2021.

[Sun *et al.*, 2025] Bohang Sun, Yongjian Deng, Yuena Lin, Qiuru Hai, Zhen Yang, and Gengyu Lyu. Graph consistency and diversity measurement for federated multi-view clustering. *The 39th Annual AAAI Conference on Artificial Intelligence*, 2025.

[Tang and Liu, 2022] Huayi Tang and Yong Liu. Deep safe incomplete multi-view clustering: Theorem and algorithm. In *International Conference on Machine Learning*, pages 21090–21110. PMLR, 2022.

[Teng *et al.*, 2024] Ge Teng, Ting Mao, Chen Shen, Xiang Tian, Xuesong Liu, Yaowu Chen, and Jieping Ye. Urrl-imvc: Unified and robust representation learning for incomplete multi-view clustering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2888–2899, 2024.

[Wang *et al.*, 2015] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International conference on machine learning*, pages 1083–1092. PMLR, 2015.

[Wang *et al.*, 2019] Hao Wang, Linlin Zong, Bing Liu, Yan Yang, and Wei Zhou. Spectral perturbation meets incomplete multi-view data. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3677–3683, 2019.

[Wang *et al.*, 2023] Senhong Wang, Jiangzhong Cao, Fangyuan Lei, Jianjian Jiang, Qingyun Dai, and Bingo Wing-Kuen Ling. Multiple kernel-based anchor graph coupled low-rank tensor learning for incomplete multi-view clustering. *Applied Intelligence*, 53(4):3687–3712, 2023.

[Winn and Jojic, 2005] John Winn and Nebojsa Jojic. Locus: Learning object classes with unsupervised segmentation. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 756–763. IEEE, 2005.

[Wu *et al.*, 2024] Tingting Wu, Songhe Feng, and Jiazheng Yuan. Low-rank kernel tensor learning for incomplete multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15952–15960, 2024.

[Xiao *et al.*, 2023] Shunxin Xiao, Shide Du, Zhaoliang Chen, Yunhe Zhang, and Shiping Wang. Dual fusion-propagation graph neural network for multi-view clustering. *IEEE Trans. on Multimedia*, 25:9203–9215, 2023.

[Xu *et al.*, 2021] Jie Xu, Yazhou Ren, Guofeng Li, Lili Pan, Ce Zhu, and Zenglin Xu. Deep embedded multi-view clustering with collaborative training. *Information Sciences*, 573:279–290, 2021.

[Yang *et al.*, 2022] Xihong Yang, Xiaochang Hu, Sihang Zhou, Xinwang Liu, and En Zhu. Interpolation-based contrastive learning for few-label semi-supervised learning. *IEEE Trans. on Neural Networks and Learning Systems*, 35(2):2054–2065, 2022.

[Zhang *et al.*, 2022] Hongyuan Zhang, Jiankun Shi, Rui Zhang, and Xuelong Li. Non-graph data clustering via $\mathcal{O}(n)$ bipartite graph convolution. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 45(7):8729–8742, 2022.

[Zhao *et al.*, 2017] Handong Zhao, Zhengming Ding, and Yun Fu. Multi-view clustering via deep matrix factorization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[Zhao *et al.*, 2024] Wenhui Zhao, Guangfei Li, Haizhou Yang, Quanxue Gao, and Qianqian Wang. Embedded feature selection on graph-based multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17016–17023, 2024.

[Zhao *et al.*, 2025] Zihua Zhao, Ting Wang, Haonan Xin, Rong Wang, and Feiping Nie. Multi-view clustering via high-order bipartite graph fusion. *Information Fusion*, 113:102630, 2025.