



# Federated Deep Multi-View Clustering with Global Self-Supervision

Xinyue Chen

School of Computer Science and Engineering, University of Electronic Science and Technology of China  
martinachen2580@gmail.com

Xiaorong Pu

School of Computer Science and Engineering, University of Electronic Science and Technology of China  
puxiaor@uestc.edu.cn

Zhifeng Hao

College of Science, Shantou University  
haozhifeng@stu.edu.cn

Jie Xu

School of Computer Science and Engineering, University of Electronic Science and Technology of China  
jieuwork@outlook.com

Ce Zhu

School of Information and Communication Engineering, University of Electronic Science and Technology of China  
eczhu@uestc.edu.cn

Yazhou Ren\*

School of Computer Science and Engineering, University of Electronic Science and Technology of China  
yazhou.ren@uestc.edu.cn

Xiaofeng Zhu

School of Computer Science and Engineering, University of Electronic Science and Technology of China  
seanzhuxf@gmail.com

Lifang He

Department of Computer Science and Engineering, Lehigh University  
lih319@lehigh.edu

## ABSTRACT

Federated multi-view clustering has the potential to learn a global clustering model from data distributed across multiple devices. In this setting, label information is unknown and data privacy must be preserved, leading to two major challenges. First, views on different clients often have feature heterogeneity, and mining their complementary cluster information is not trivial. Second, the storage and usage of data from multiple clients in a distributed environment can lead to incompleteness of multi-view data. To address these challenges, we propose a novel federated deep multi-view clustering method that can mine complementary cluster structures from multiple clients, while dealing with data incompleteness and privacy concerns. Specifically, in the server environment, we propose sample alignment and data extension techniques to explore the complementary cluster structures of multiple views. The server then distributes global prototypes and global pseudo-labels to each client as global self-supervised information. In the client environment, multiple clients use the global self-supervised information and deep autoencoders to learn view-specific cluster assignments and embedded features, which are then uploaded to the server for refining the global self-supervised information. Finally, the results of our extensive experiments demonstrate that our proposed

method exhibits superior performance in addressing the challenges of incomplete multi-view data in distributed environments.

## CCS CONCEPTS

- Computing methodologies → Cluster analysis; • Theory of computation → Unsupervised learning and clustering.

## KEYWORDS

Multi-view clustering, federated learning, global self-supervision

### ACM Reference Format:

Xinyue Chen, Jie Xu, Yazhou Ren, Xiaorong Pu, Ce Zhu, Xiaofeng Zhu, Zhifeng Hao, and Lifang He. 2023. Federated Deep Multi-View Clustering with Global Self-Supervision. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3581783.3612027>

## 1 INTRODUCTION

Multimedia technologies have led to the emergence of a large amount of multi-view or multi-modal data, which often lack label information [4, 44–46]. To explore useful consistent and complementary information among multiple views in an unsupervised manner, researchers have proposed various multi-view clustering methods [21, 29, 40]. However, these methods typically operate in a centralized environment and cannot handle isolated data stored in various distributed devices/silos due to privacy concerns in industry competition. Fortunately, federated learning offers a potential solution for such scenarios by enabling the training of a unified model without exposing sensitive data stored on individual devices.

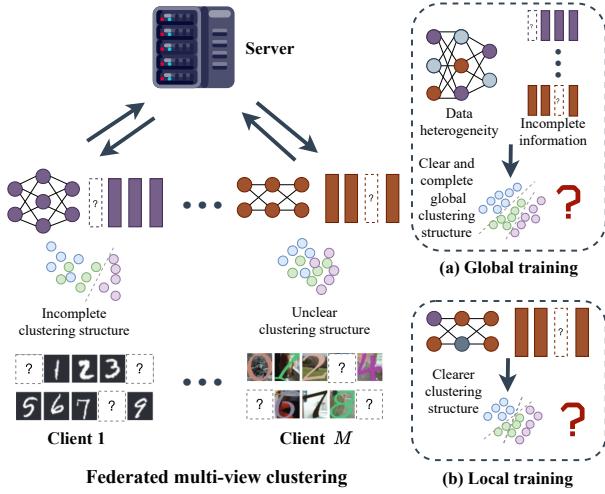
Federated multi-view learning [26, 50] is a relatively new machine learning paradigm that has gained significant attention in recent years. It is designed to learn a global model from multi-view

\*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 979-8-4007-0108-5/23/10...\$15.00  
<https://doi.org/10.1145/3581783.3612027>



**Figure 1: Problem illustration of federated multi-view clustering.** (a) During global training, how can the server address feature heterogeneity and incomplete information in multi-view data to obtain a clear global clustering structure? (b) During local training, how can we alleviate the unclear clustering structure of the client with low-quality views?

data that are distributed across different devices, and it often incorporates existing machine learning methods, such as multi-view matrix factorization [15, 19], ensemble learning [8, 14], and deep models [17, 42]. By combining these methods with the federated learning approach, it becomes possible to address the challenges posed by distributed multi-view data, such as data privacy and feature heterogeneity.

Clustering analysis with federated multi-view learning, known as federated multi-view clustering (FedMVC), has recently been shown to be an effective method for handling multi-view/modal data without label information in distributed environments. However, despite its potential, FedMVC is still a relatively underexplored area of research. Addressing the challenges of FedMVC is crucial, and there are two main obstacles to overcome. Firstly, due to the feature heterogeneity of multi-view data and the complexity of clustering, traditional federated learning solutions struggle to identify complementary cluster structures. Even if each client can cluster local data separately, the feature heterogeneity of the datasets may obscure certain clusters that only become apparent when the data are combined. Some federated clustering methods [10, 30] extend traditional clustering algorithms to federated learning settings, but they have limited capability to learn feature representations and struggle to handle complex heterogeneous multi-view data. Secondly, the data storage and usage of multiple clients in distributed environments can lead to incomplete multi-view data [3, 43]. For example, medical tests distributed across different healthcare institutions can be considered as different views, but patients do not undergo all the corresponding tests at each institution, which makes most methods based on information completeness assumptions unavailable in such scenarios.

To overcome the challenges outlined above, we introduce a novel federated deep multi-view clustering method, FedDMVC. Our method is designed for scenarios where a dataset with  $M$  views is distributed across  $M$  clients, and the samples of each client do not have exact overlaps. The primary aim of our approach is to leverage global self-supervised information within a federated learning setting to extract complementary cluster structures from the data distributed across multiple clients. The general framework of FedDMVC is illustrated in Figure 2. Initially, the server distributes global self-supervised information, such as global prototypes and global pseudo-labels, to each client. Then, each client utilizes deep autoencoders and the global self-supervised information to learn view-specific cluster assignments and embedded features, which are then uploaded to the server for the next iteration.

In general, existing vertical federated learning methods involve parties sharing embeddings in a private manner [5, 7], followed by a server model that captures the complex interactions of the embeddings. For the proposed FedDMVC method, we additionally upload the cluster assignments of each client to the server. To address challenge 1, we construct global self-supervised information on the server to mitigate the heterogeneity of local datasets, and explore complementary cluster structures from multiple views across multiple clients. To tackle challenge 2, we propose sample alignment and data extension techniques that leverage global prototypes and view-specific patterns to impute incomplete data based on sample commonality and view versatility.

We summarize our contributions in this paper as follows:

- We propose a novel federated deep multi-view clustering method that can effectively mine complementary cluster structures from multi-view data across multiple clients.
- We propose a method to expand data from global prototypes and view-specific patterns based on sample commonality and view versatility, thereby addressing the incompleteness of multi-view data in distributed environments.
- Our proposed method can facilitate the flow and sharing of information among clients while ensuring privacy. Extensive experiments conducted on public datasets demonstrate that our method outperforms state-of-the-art techniques.

## 2 RELATED WORK

### 2.1 Multi-View Clustering

Multi-view clustering (MVC) methods aim to improve clustering performance by leveraging consistent and complementary information among multiple views. Traditional MVC methods utilize classical machine learning techniques such as non-negative matrix factorization (NMF), subspace and graph learning. Liu et al. [25] proposed an NMF-based method to handle multi-view data. Zhao et al. [48] utilized a deep semi-NMF structure to extract more consistent information. Similarly, subspace-based MVC methods achieve data clustering by exploring shared representations among multiple views. For example, Li et al. [22] constructed a mutual multilayer subspace representation associated with latent representation to better recognize clustering structures. Zheng et al. [49] introduced an effective feature cascaded multi-view subspace clustering to explore the consistency information of multi-view data. Graph-based MVC methods can exploit graph structure information to improve

the recognition of clustering patterns. For instance, Wang et al. [33] used multi-graph Laplacian regularized low-rank representation for multi-view graph clustering. Fan et al. [11] integrated self-supervised training with graph autoencoder reconstruction in a unified framework for attribute multi-view graph clustering.

In recent years, deep learning based MVC methods have been attracting increasing attention, among which MVC methods based on deep autoencoders have achieved remarkable achievements [9]. Deep autoencoders learn embedded features by optimizing the reconstruction loss between input and output [28, 37]. They are usually combined with existing clustering methods to explore the unified clustering structure among multiple views. For example, Abavisani et al. [1] first used autoencoder architecture for multi-view subspace clustering. Although Xu et al. [38, 39] proposed deep imputation-free frameworks for addressing the incompleteness of multi-view data, data privacy issues in federated environments and the utilization of complementary information in incomplete parts of data have not been well studied.

Most traditional and deep MVC methods usually operate in centralized environments [41], which is difficult to handle data privacy leakage and data isolation issues. Although some distributed MVC methods [6, 18] have been proposed and can be applied in distributed environments, they are not suitable for addressing the unique challenges introduced by federated learning, such as feature heterogeneity and incompleteness of multi-view data across multiple clients. In this paper, we propose a novel federated deep multi-view clustering method that can solve the above issues.

## 2.2 Federated Multi-View Learning

Federated multi-view learning presents effective solutions to the challenge of multi-view learning in federated environments. It can be roughly classified into three categories: (1) Some FedMVL methods [15, 19] extend the federated learning framework to include multi-view matrix factorization, which involves aggregating or selecting the low-rank matrix for each view on the server. For example, Flanagan et al. [15] combined multi-view matrix factorization with a federated learning framework used for personalized recommendations. Huang et al. [19] first considered the issues of high communication costs and proposed an NMF-based federated learning framework for the multi-view clustering task. (2) Ensemble-based FedMVL methods, which usually train single learner locally with data from each client and exploit the differences among multiple learners on the server to improve the learning performance. For instance, Feng et al. [14] proposed a multi-participant multi-class vertical federated learning framework that trains separate models for each participant. Che et al. [8] proposed a generic federated multi-view learning framework that can be applied to both vertical and horizontal multi-view data distributions. (3) Deep learning techniques are applied to FedMVL, including but not limited to (a) Deep structured semantic models are used to map users and items to a shared semantic space within a federated multi-view setting [17]; (b) DeepMood architecture is used for late fusion in a federated learning setting at the session level [42].

Although existing FedMVL methods have designed appropriate frameworks according to the different distributions or characteristics of multi-view data, most studies have focused on labeled

data and are not directly applicable to unsupervised multi-view environments. In addition, all FedMVL methods assume complete information, but in the real world, not all samples have complete views due to the data storage and usage of multiple clients. Unlike previous methods, our method can adapt to unsupervised multi-view environments and address the issue of incomplete views for samples. Moreover, we design a mechanism to discover and leverage global self-supervision information, which enhances the quality of the local model for each client and yields a high-quality global clustering structure on the server.

## 3 METHODOLOGY

### 3.1 Problem Setting

In this paper, we propose a novel federated deep multi-view clustering method, which can collaborate multi-view data distributed across different clients to mine complementary cluster structures, while addressing data incompleteness and privacy. We focus on the cross-silo federation learning scenario, where all clients participate in each round of communication. In a federated multi-view setting, multi-view data with  $M$  views, denoted by  $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^M\}$ , are distributed among  $M$  silos. For client  $m$ , its data are represented as  $\mathbf{X}^m \in \mathbb{R}^{N_m \times D_m}$ , where  $D_m$  is the dimensionality of samples in the  $m$ -th view and  $N_m$  is the number of samples in the  $m$ -th client,  $m = 1, \dots, M$ . It should be noted that there are differences in the number of samples, sample features and clustering distribution of each client, but there are also some overlapping samples among clients. In this scenario, we clarify two goals:

**Global goal.** It is expected to obtain a high-quality global clustering structure on the server that is comparable in performance to a model trained on centralized data collected from clients.

**Local goal.** It is expected to improve the clustering performance of each client by considering global information, resulting in better performance than the model trained with each client's data independently.

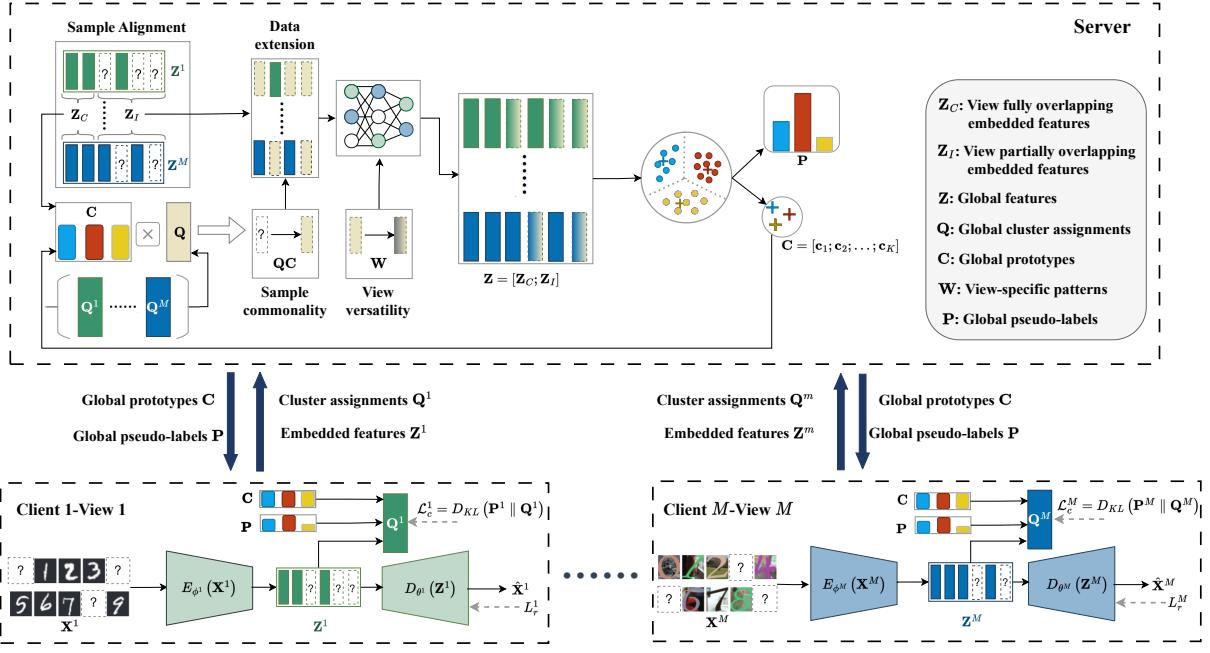
### 3.2 Local Training

We construct a local model for each client using the same approach and enhance it by considering the global prototypes  $\mathbf{C}$  and global pseudo-labels  $\mathbf{P}$  obtained from the server. We analyze the local training process for client  $m$  as follows.

Deep autoencoder has been widely employed in various unsupervised environments owing to their ability to effectively capture the essential features of the data [13, 24, 47]. Therefore, we utilize an autoencoder to project the client's data into a low-dimensional space, preserving the privacy of the original data while capturing informative latent features for clustering. The proposed method can be expressed by minimizing the following reconstruction loss:

$$\mathcal{L}_r^m = \|\mathbf{X}^m - D_{\theta^m}(\mathbf{Z}^m)\|_F^2 = \sum_{i=1}^{N_m} \left\| \mathbf{x}_i^m - D_{\theta^m} \left( E_{\phi^m}(\mathbf{x}_i^m) \right) \right\|_2^2, \quad (1)$$

where  $\mathbf{Z}^m$  denotes the low-dimensional feature embedding of client  $m$ ,  $E_{\phi^m}$  and  $D_{\theta^m}$  denote its encoder and decoder networks, respectively. The encoder is  $E_{\phi^m}(\mathbf{X}^m; \phi^m) : \mathbf{X}^m \in \mathbb{R}^{N_m \times D_m} \mapsto \mathbf{Z}^m \in \mathbb{R}^{N_m \times d_m}$  and the decoder is  $D_{\theta^m}(\mathbf{Z}^m; \theta^m) : \mathbf{Z}^m \in \mathbb{R}^{N_m \times d_m} \mapsto$



**Figure 2: The framework of FedDMVC. It contains a server and  $M$  clients. (1) Server: the server aggregates information uploaded by the clients and proposes sample alignment and data extension. After that, the server proceeds to construct global features  $Z$ , obtain global pseudo-labels  $P$ , and explore the complementary cluster structures among multiple views. (2) Clients: For client  $m$ , we utilize the global self-supervised information and deep autoencoders to learn view-specific cluster assignments  $Q^m$  and embedded features  $Z^m$ , which are then uploaded to the server for refining the global self-supervised information.**

$\hat{X}^m \in \mathbb{R}^{N_m \times D_m}$ , where  $d_m$  is the dimensionality of embedded features,  $\phi^m$  and  $\theta^m$  are learnable parameters of autoencoder network.

Inspired by popular single-view deep clustering methods [16, 37], we use a parameterized mapping  $\mathcal{M}_m(Z^m; \mathbf{U}^m) : Z^m \in \mathbb{R}^{N_m \times d_m} \mapsto Q^m \in \mathbb{R}^{N_m \times K}$  to obtain soft cluster assignments  $Q^m$ , where  $K$  is the number of categories to be clustered. Concretely,

$$q_{ij}^m = \frac{\left(1 + \left\|z_i^m - \mathbf{u}_j^m\right\|_2^2\right)^{-1}}{\sum_{j=1}^K \left(1 + \left\|z_i^m - \mathbf{u}_j^m\right\|_2^2\right)^{-1}} \in Q^m, \quad (2)$$

where  $q_{ij}^m$  is the probability that the embedded feature  $z_i^m$  is assigned to the  $j$ -th cluster,  $\mathbf{U}^m = [\mathbf{u}_1^m; \mathbf{u}_2^m; \dots; \mathbf{u}_K^m] \in \mathbb{R}^{K \times d_m}$  represent the learnable parameters, can be initialized with the global prototypes  $C$ .

For client  $m$ , we can convert the global pseudo-labels  $P$  to supervised information  $P^m$  on that client by a mapping  $\mathcal{F}_m(P) : P \in \mathbb{R}^{N \times K} \mapsto P^m \in \mathbb{R}^{N_m \times K}$ , where  $N$  represents the total number of samples on all clients. Furthermore, the clustering loss between the pseudo-labels  $P^m$  and its own cluster assignment distribution  $Q^m$  is optimized:

$$\mathcal{L}_c^m = D_{KL}(P^m \parallel Q^m) = \sum_{i=1}^{N_m} \sum_{j=1}^K p_{ij}^m \log \frac{p_{ij}^m}{q_{ij}^m}. \quad (3)$$

So, the total loss of client  $m$  consists of two parts:

$$\mathcal{L}^m = \mathcal{L}_r^m + \gamma \mathcal{L}_c^m, \quad (4)$$

where  $\gamma$  is a trade-off coefficient between the clustering and reconstruction losses. The reconstruction loss  $\mathcal{L}_r^m$  ensures the representation capability of the embedded features to the client's original data. Optimizing the clustering loss  $\mathcal{L}_c^m$  will make the distribution of  $Q^m$  sharper and mine complementary information from other clients by minimizing the  $KL$  divergence between  $Q^m$  and  $P^m$ .

### 3.3 Global Training

In our framework, to facilitate information flow, each client uploads its embedded features and cluster assignments to the server. The server plays a critical role in discovering global self-supervised information, achieving high-quality global clustering, and addressing the challenges of feature heterogeneity and incomplete information in multi-view data by utilizing sample alignment and data extension techniques.

After receiving the cluster assignments from each client, the server averages them to obtain the global cluster assignments:

$$Q = \sum_{m=1}^M Q^m A^m, \quad (5)$$

where  $Q \in \mathbb{R}^{N \times K}$ . It is worth noting that the clusters represented by  $Q^m$  in each client do not necessarily correspond to each other. Therefore, we denote  $l_i^m = \text{argmax}_j q_{ij}^m, q_{ij}^m \in Q^m$  and then treat  $l^1$  as an anchor to modify  $l^m$  on the remaining clients by minimizing

the following matching formula:

$$\begin{aligned} & \min_{\mathbf{A}^m} \mathbf{M}^m \mathbf{A}^m \\ & \text{s.t. } \mathbf{A}^m (\mathbf{A}^m)^T = \mathbf{I}_K, \end{aligned} \quad (6)$$

where  $\mathbf{A}^m$  is a boolean matrix used to adjust the arrangement of  $\mathbf{Q}^m$  and  $\mathbf{M}^m \in \mathbb{R}^{K \times K}$  denotes the cost matrix.  $\mathbf{M}^m = \max_{i,j} \tilde{m}_{ij}^m - \bar{\mathbf{M}}^m$  and  $\tilde{m}_{ij}^m = \sum_{n=1}^N \mathbb{1}[l_n^m = i] \mathbb{1}[l_n^m = j]$ , where  $\mathbb{1}[\cdot]$  represents the indicator function. The optimization of Eq. (6) is performed using the Hungarian algorithm [20].

After receiving the embedded features uploaded by each client, the server concatenates them to generate the global features:

$$\mathbf{Z} = [\mathbf{Z}^1, \mathbf{Z}^2, \dots, \mathbf{Z}^M] \in \mathbb{R}^{N \times \sum_{m=1}^M d_m}. \quad (7)$$

On the server, we employ an indicator matrix  $\mathbf{H} \in \{0, 1\}^{N \times M}$ , where  $h_{im} \in \mathbf{H}$ ,  $h_{im} = 1$  denotes client  $m$  has data for the  $i$ -th sample, and otherwise  $h_{im} = 0$ . Moreover, we denote  $\mathbf{Z} = [\mathbf{Z}_C; \mathbf{Z}_I]$ . For each  $\mathbf{z}_i \in \mathbf{Z}$ , if there exists  $\sum_{m=1}^M h_{im} = M$ , then  $\mathbf{z}_i \in \mathbf{Z}_C$ ; otherwise  $\mathbf{z}_i \in \mathbf{Z}_I$ .

By leveraging the overlapping samples across clients, we can obtain the global prototypes  $\mathbf{C}$  using the following objective:

$$\min_{\mathbf{C}} \|\mathbf{Z}_C - \mathbf{C}\|_F^2 = \min_{\{\mathbf{c}_j\}_{j=1}^K} \sum_{\mathbf{z}_i \in \mathbf{Z}_C} \sum_{j=1}^K \|\mathbf{z}_i - \mathbf{c}_j\|_2^2, \quad (8)$$

where  $\mathbf{C} \in \mathbb{R}^{K \times \sum_{m=1}^M d_m}$  and  $\mathbf{c}_j = [\mathbf{c}_j^1, \mathbf{c}_j^2, \dots, \mathbf{c}_j^M]$ . The global prototypes represent the shared common pattern among samples belonging to the same cluster, obtained by aligning the overlapping client samples.

To extract view-specific patterns  $\mathbf{W}$  from each client's data, we utilize the following optimization:

$$\begin{aligned} & \min_{\mathbf{W}} \|\mathbf{Z}_C - \mathbf{W} \mathbf{Q} \mathbf{C}\|_F^2 \\ & = \min_{\{\mathbf{W}^m\}_{m=1}^M} \sum_{\mathbf{z}_i \in \mathbf{Z}_C} \sum_{m=1}^M \|\mathbf{z}_i^m - \mathbf{W}^m \mathbf{q}_i \mathbf{C}^m\|_2^2, \end{aligned} \quad (9)$$

where  $\mathbf{q}_i \in \mathbf{Q}$ . We can leverage information from the global prototypes  $\mathbf{C}$ , global cluster assignments  $\mathbf{Q}$ , and view-specific patterns  $\mathbf{W}$  to impute the unavailable embedded features  $\mathbf{z}_i^m$ . Specifically,  $\mathbf{Q} \mathbf{C}$  and  $\mathbf{W}$  impute the unavailable embedded features from the perspective of sample commonality and view versatility, respectively. In this case, when  $h_{im} = 0$  for  $\mathbf{z}_i^m$ , the calculation is as follows:

$$\mathbf{z}_i^m = \mathbf{W}^m \mathbf{q}_i \mathbf{C}^m \in \mathbf{Z}_I. \quad (10)$$

By starting with the global common structure,  $\mathbf{z}_i^m$  combines the common characteristics of samples with the versatile features of views, resulting in effective data extension. This imputation method enables the utilization of shared information and partially overlapping parts of samples among various clients, facilitating the mining of more accurate global pseudo-labels  $\mathbf{P}$  later.

We concatenate the embedded features uploaded by each client with the features obtained by expanding the data in Eq. (10) to update global features  $\mathbf{Z} = [\mathbf{Z}_C; \mathbf{Z}_I]$ . Then we adopt  $K$ -means [27] on the global features to obtain the global clustering structure and

---

**Algorithm 1** Federated Deep Multi-View Clustering (FedDMVC)

---

**Input:** Data with  $M$  views  $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^M\}$ , which are distributed on  $M$  local silos, number of clusters  $K$ , Epoch  $E$ .

**Output:** Global clustering predictions.

```

1: while not reaching  $E$  epochs do
2:   for  $m = 1$  to  $M$  do in parallel
3:     if  $E == 1$  then
4:       Get  $\theta^m, \phi^m$ , and  $\mathbf{U}^m$  by pretraining autoencoder.
5:     else
6:       Update  $\mathbf{U}^m$  by global prototypes  $\mathbf{C}$ .
7:     while not reach the maximum iterations  $T_1$  do
8:       Optimize the total loss function by Eq. (4).
9:     end while
10:    end if
11:    Upload  $\mathbf{Z}^m$  and  $\mathbf{Q}^m$  to the server.
12:  end for
13:  Update global cluster assignments  $\mathbf{Q}$  by Eqs. (5)-(6).
14:  Obtain global prototypes  $\mathbf{C}$  by Eq. (8).
15:  while not reach the maximum iterations  $T_2$  do
16:    Impute the unavailable embedded features by Eq. (9).
17:  end while
18:  Update global features  $\mathbf{Z}$  by Eq. (10).
19:  Obtain global pseudo-labels  $\mathbf{P}$  by Eqs. (11)-(13).
20:  Distribute  $\mathbf{C}$  and  $\mathbf{P}$  to each client.
21: end while
22: Calculate the clustering predictions by Eq. (14).

```

---

calculate the cluster centroids:

$$\min_{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K} \sum_{i=1}^N \sum_{j=1}^K \|\mathbf{z}_i - \mathbf{c}_j\|^2. \quad (11)$$

After that, we can use the Student's  $t$ -distribution to measure the similarity between global features and cluster centroids as follows:

$$s_{ij} = \frac{\left(1 + \|\mathbf{z}_i - \mathbf{c}_j\|^2\right)^{-1}}{\sum_j \left(1 + \|\mathbf{z}_i - \mathbf{c}_j\|^2\right)^{-1}} \in \mathbf{S}. \quad (12)$$

In this way, the confidence  $s_{ij}$  is high when  $\mathbf{z}_i$  is closer to  $\mathbf{c}_j$ . We use the function  $\mathcal{E}(\mathbf{S})$  to enhance the confidence and obtain the global pseudo-labels  $\mathbf{P}$ :

$$p_{ij} = \mathcal{E}(s_{ij}) = \frac{\left(s_{ij} / \sum_j s_{ij}\right)^2}{\sum_j \left(s_{ij} / \sum_j s_{ij}\right)^2} \in \mathbf{P}. \quad (13)$$

Furthermore, the global clustering predictions are calculated by

$$y_i = \arg \max_j (p_{ij}). \quad (14)$$

In summary, the server effectively utilizes the information uploaded by clients to mine global prototypes and global pseudo-labels based on sample commonality and view versatility, and discovers a clear global clustering structure.

### 3.4 Optimization

Algorithm 1 provides a detailed description of the optimization procedure, which comprises two main parts: the clients and the server. The clients are responsible for parallel training of the local model. In the first round, they perform pretraining of the autoencoder. In the following rounds, they use the global self-supervision information discovered by the server to enhance the quality of their local models. The server aligns and imputes the unavailable embedded features, utilizing the information uploaded by clients to address the issue of incomplete sample overlap. In addition, the server discovers the global prototypes and global pseudo-labels from the global features, and obtains global clustering predictions. Clients and the server alternately iterate through  $E$  epochs.

**Complexity Analysis.** Suppose  $K$ ,  $M$ , and  $N$  represent the number of clusters, clients and total samples, respectively. Let  $H$  denote the maximum number of neurons in autoencoders' hidden layers,  $W$  denote the maximum number of hidden neurons in the network on the server, and  $Z$  denote the maximum dimensionality of embedded features. Generally  $N \gg V, K, M$  holds. In Algorithm 1, for client  $m$ , the complexities to optimize Eq. (1) and Eq. (2) are  $O(NH^2)$  and  $O(NZK)$ , respectively. For server, the complexities to optimize Eq. (6) and Eq. (9) are  $O(MK^3 + NMK)$  and  $O(NW)$ , respectively, while the complexity to optimize Eq. (11) is  $O(NMZK)$ . In conclusion, the total complexity of our algorithm is  $O(NH^2 + NMZK + MK^3 + NW)$  in each iteration, which is linear to the data size  $N$ .

## 4 EXPERIMENT

### 4.1 Experimental Settings

**Datasets.** Our experiments are carried out on four widely used datasets. Specifically, **Reuters** [2] contains 1200 articles in 6 categories, with each article written in five different languages and treated as five separate text views. **Scene** [12] includes 4,485 scene images in 15 classes with three views. **Handwritten Numerals (HW)**<sup>1</sup> contains 2000 samples in 10 categories corresponding to numerals 0-9, each constituted by the six visual views. **Fashion-MV** [36] contains images from 10 categories, where we treat six different styles of one object as six views, to better simulate the federated learning environment with six clients.

Note that in our federated setting, multiple views of these datasets are distributed among different clients and are isolated from each other. In addition, to evaluate the effectiveness of our method in handling incomplete multi-view data, we randomly remove some samples from arbitrary views, resulting in the incomplete dataset, following [43]. Also, we define the sample overlapping rate  $\delta = m/n$  among clients, where  $n$  is the size of the dataset and  $m$  is the number of samples with fully overlapping views for all clients.

**Comparing Methods.** We select several pertinent algorithms to serve as comparison methods. Since our method is essentially distributed, we include two distributed multi-view clustering methods as comparison methods, i.e., RMKMC [6] and CaMVC [18]. Likewise, our method can be applied to IMVC for handling incomplete multi-view data. We compare our method with five state-of-the-art

IMVC methods, i.e., CDIMC-net [34], GIMC-FLSD [35], HCP-IMSC [23], IMVC-CBG [32] and DSIMVC [31].

For fair comparisons, we conduct FedDMVC and baselines under two settings, i.e.,  $\delta = 0.5$  (denoted by Partially) and  $\delta = 1$  (denoted by Fully). As the first two baselines are unable to handle partially overlapping data directly, we preprocess them by filling the incomplete parts with the mean value of the entire view.

**Evaluation Metrics.** We evaluate the clustering effectiveness using three metrics: clustering accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI). A higher value for each metric indicates better clustering performance.

### 4.2 Clustering Results

Table 1 shows the quantitative comparison of FedDMVC and baseline models in the Partially and Fully scenarios. Due to the high algorithmic complexity, HCP-IMSC was unable to be executed on the Fashion-MV dataset. From Table 1, we can observe that the proposed method outperforms all baseline models for different scenarios on all datasets. Compared with the second-best methods CDIMC-net, GIMC-FLSD and DSIMVC, FedDMVC has considerable improvements especially on Reuters, Scene and HW. The results demonstrate that our method is effective in handling both complete and incomplete information, while ensuring data privacy in a federated setting. Particularly in handling incomplete information, the superior performance of FedDMVC validates the effectiveness of our proposed strategy of utilizing sample commonality and view versatility for data extension.

To further investigate the robustness of our proposed method, we conduct experiments on Reuters with overlapping rates varying from 0.1 to 1 with an interval of 0.1. As shown in Figure 3, our FedDMVC significantly outperforms the baseline methods across all overlapping rates. Moreover, the performance of FedDMVC shows substantial improvement with increasing overlapping rates. The results indicate that FedDMVC is robust to varying degrees of sample overlapping across clients. Additionally, FedDMVC can effectively estimate the data distribution by leveraging available information, even when the overlapping rate is low.

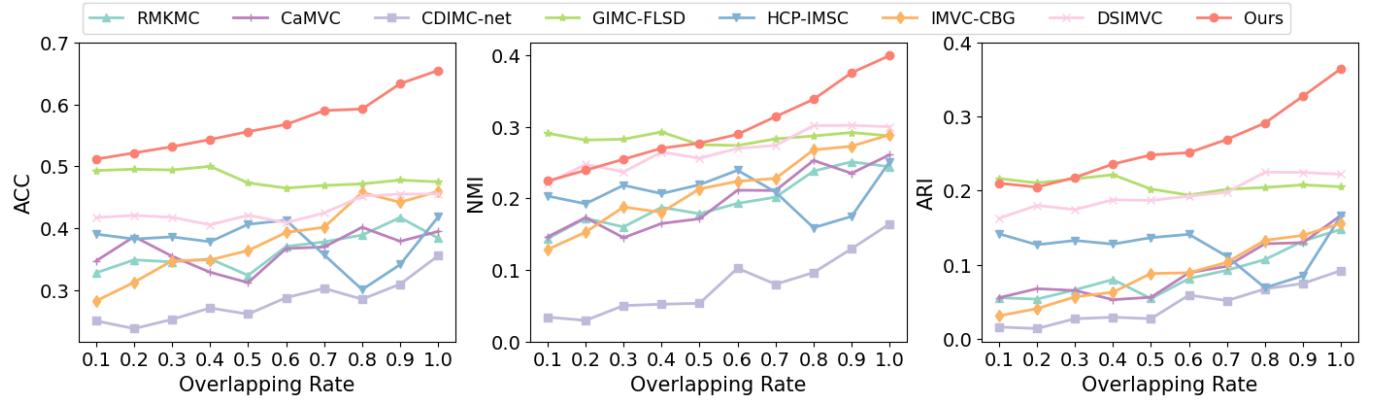
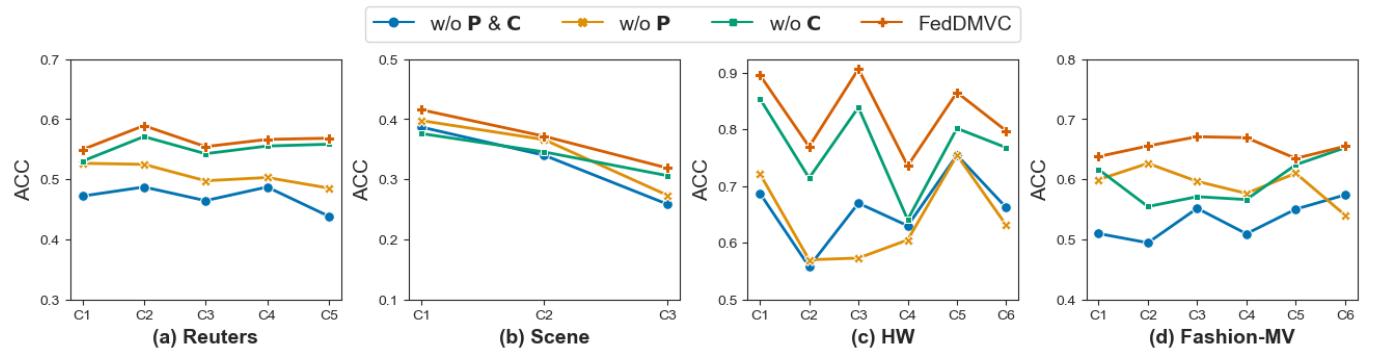
### 4.3 Model Analysis

**Ablation Study in Each Client's Local Model.** To further validate the effectiveness of the global information included in our proposed method on the local models of each client, we conduct an ablation study, as shown in Figure 4. The figure depicts different scenarios, with different colored lines representing each scenario. The label "w/o" denotes the absence of global information in the method. If the client does not consider any global information, it is equivalent to using autoencoder to extract the embedded features of its own raw data and then performing local clustering. If the client does not consider the global pseudo-labels  $P$ , it corresponds to only using Eq. (1) for optimization. If the client does not consider the global prototype  $C$ , it means that the client uses Eq. (4) for optimization but still updates the clustering mapping using local centroids. The results indicate that incorporating both global pseudo-labels  $P$  and global prototypes  $C$  is advantageous for improving local clustering performance. Furthermore, it is observed that  $P$  has a greater influence on optimizing the local clustering structure than  $C$ .

<sup>1</sup><https://archive.ics.uci.edu/ml/datasets.php>

**Table 1: Experiments on four datasets. The best result in each column is shown in bold and the second-best is underlined.**

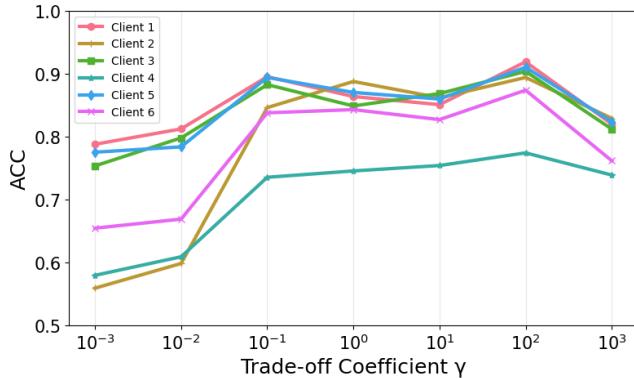
Overlapping	Methods	Reuters			Scene			HW			Fashion-MV		
		ACC	NMI	ARI									
Partially	RMKMC [6]	0.324	0.178	0.054	0.276	0.252	0.118	0.648	0.628	0.508	0.550	0.654	0.467
	CaMVC [18]	0.313	0.171	0.056	0.296	0.293	0.147	0.730	0.673	0.585	0.501	0.582	0.391
	CDIMC-net [34]	0.179	0.040	0.001	0.306	0.319	0.153	0.798	0.820	0.736	0.604	0.701	0.522
	GIMC-FLSD [35]	0.473	0.274	0.202	0.300	0.264	0.135	0.242	0.163	0.033	0.709	0.738	0.603
	HCP-IMSC [23]	0.438	0.261	0.178	0.325	0.273	0.143	0.809	0.778	0.719	–	–	–
	IMVC-CBG [32]	0.364	0.213	0.088	0.268	0.270	0.144	0.471	0.473	0.237	0.468	0.439	0.202
	DSIMVC [31]	0.421	0.256	0.187	0.278	0.304	0.145	0.762	0.736	0.650	0.800	<b>0.801</b>	0.665
Fully	FedDMVC (ours)	<b>0.566</b>	<b>0.299</b>	<b>0.249</b>	<b>0.393</b>	<b>0.343</b>	<b>0.225</b>	<b>0.893</b>	<b>0.824</b>	<b>0.790</b>	<b>0.820</b>	<b>0.785</b>	<b>0.690</b>
	RMKMC [6]	0.384	0.244	0.148	0.407	0.406	0.230	0.741	0.739	0.636	0.532	0.737	0.556
	CaMVC [18]	0.395	0.261	0.166	0.370	0.368	0.203	0.769	0.766	0.684	0.500	0.687	0.510
	CDIMC-net [34]	0.356	0.164	0.092	0.387	0.407	0.193	0.845	0.901	0.826	0.696	0.801	0.642
	GIMC-FLSD [35]	0.475	0.287	0.205	0.347	0.370	0.186	0.422	0.474	0.298	0.787	0.827	0.729
	HCP-IMSC [23]	0.418	0.251	0.166	0.380	0.330	0.183	0.826	0.793	0.743	–	–	–
	IMVC-CBG [32]	0.460	0.289	0.156	0.300	0.316	0.164	0.604	0.618	0.480	0.585	0.594	0.426
DSIMVC [31]	DSIMVC [31]	0.434	0.272	0.204	0.284	0.322	0.152	0.817	0.792	0.735	0.905	<b>0.915</b>	0.853
	FedDMVC (ours)	<b>0.655</b>	<b>0.419</b>	<b>0.364</b>	<b>0.451</b>	<b>0.429</b>	<b>0.280</b>	<b>0.965</b>	<b>0.925</b>	<b>0.924</b>	<b>0.925</b>	<b>0.904</b>	<b>0.856</b>

**Figure 3: Performance analysis on Reuters with different overlapping rates.****Figure 4: Global information ablation experiments for each client on four datasets with the overlapping rate of 0.5.**

**Table 2: Ablation experiments of the data extension process on the server on four datasets with the overlapping rate of 0.5.**

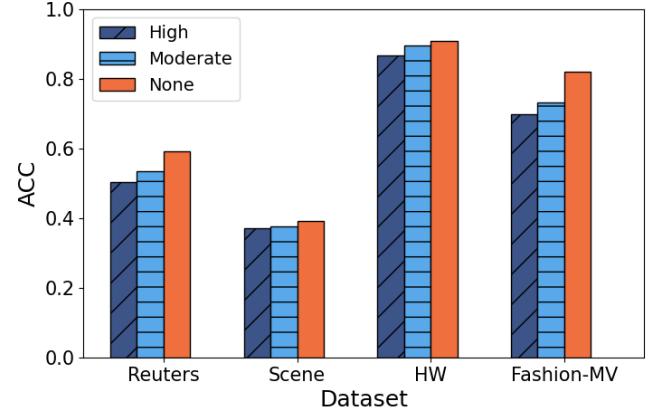
	Variants	Reuters			Scene			HW			Fashion-MV		
		ACC	NMI	ARI									
(A)	w/o QC & W	0.3375	0.1138	0.0831	0.2415	0.1968	0.0949	0.4910	0.3741	0.2752	0.3322	0.3312	0.1974
(B)	w/o W	0.5550	0.2945	0.2474	0.3426	0.3046	0.1701	0.8305	0.7979	0.7510	0.7058	0.7623	0.6214
(C)	FedDMVC	<b>0.5658</b>	<b>0.2993</b>	<b>0.2494</b>	<b>0.3927</b>	<b>0.3426</b>	<b>0.2248</b>	<b>0.8928</b>	<b>0.8071</b>	<b>0.7899</b>	<b>0.8203</b>	<b>0.7848</b>	<b>0.6900</b>

**Variants of Data Extension Process on the Server.** To further verify the effectiveness of our proposed method’s data extension process on the server, we conduct ablation studies on Eq. (8) and Eq. (9). Table 2 shows the global clustering results with different variants included. Similarly, w/o represents that the variants are not included in the method. (A) represents not using any strategy to impute unavailable embedded features. In this case, we directly use the global clustering distribution  $Q$  obtained from Eq. (5) to obtain global information and global clustering structure. (B) considers the commonality of all samples, but lacks estimates of different views from different clients. The results analysis shows that (B) outperforms (A), indicating that QC estimates sample commonality and is representative of some data features. (C) consists of the complete components of our method and outperforms (B). By considering sample commonality and view versatility, we achieve high-quality data extension and obtain a clear global clustering structure.

**Figure 5: ACC with different  $\gamma$  on HW when  $\delta = 0.5$ .**

**Parameter Analysis.** Throughout the training process of each client, the loss function defined in Eq. (4) incorporates a trade-off coefficient parameter,  $\gamma$ , which serves to balance the clustering and reconstruction losses. Here we test the sensitivity of this parameter by varying  $\gamma$  from  $[10^{-3}, 10^{-2}, \dots, 10^3]$ . As shown in Figure 5, the  $\gamma$  range between  $[10^{-1}, 10^2]$  is found to be robust for each client in FedDMVC. This indicates that each client needs to consider both losses to achieve a better clustering structure, and highlights the importance of considering global information. Without loss of generality, we set  $\gamma = 0.1$  for all datasets in our experiments.

**Attributes of Federated Learning.** To explore the heterogeneity of sample sizes among clients in federated learning, we introduce Dirichlet distribution when constructing incomplete datasets. A smaller Dirichlet parameter  $\alpha$  leads to more heterogeneous splits,

**Figure 6: Sensitivity to imbalanced sample sizes among clients on four datasets with the overlapping rate of 0.5.**

resulting in highly imbalanced sample sizes among clients. Figure 6 illustrates three levels of heterogeneity by setting  $\alpha$  to  $10^{-2}$  (high),  $10^0$  (moderate), and  $10^2$  (none) on four datasets. The results show that FedDMVC performs well even in highly heterogeneous scenarios, with only a slight decrease in performance.

## 5 CONCLUSION

In this paper, we propose a novel federated deep multi-view clustering method, which can collaborate multi-view data stored in different clients to mine complementary cluster structures. Firstly, we construct global self-supervised information on the server and explore complementary cluster structures across multiple views from multiple clients. Furthermore, we propose sample alignment and data extension to impute incomplete data based on sample commonality and view versatility. More importantly, the process of discovering and utilizing global self-supervised information enables the flow and sharing of information across clients in a privacy-preserving manner. Numerous experiments demonstrate that our method outperforms centralized methods that cannot protect data privacy, demonstrating the effectiveness of our proposed method.

## ACKNOWLEDGMENTS

This work was supported in part by National Key Research and Development Program of China (2020YFC2004300 and 2020YFC2004302), National Natural Science Foundation of China (61971052), Lehigh’s grants (S00010293 and 001250), and National Science Foundation (MRI 2215789 and IIS 1909879).

## REFERENCES

- [1] Mahdi Abavisani and Vishal M Patel. 2018. Deep multimodal subspace clustering networks. *JSTSP* (2018), 1601–1614.
- [2] Massih R Amini, Nicolas Usunier, and Cyril Goutte. 2009. Learning from multiple partially observed views—an application to multilingual text categorization. In *NeurIPS*. 1–9.
- [3] Irene Balelli, Santiago Silva, and Marco Lorenzi. 2022. A Differentially Private Probabilistic Framework for Modeling the Variability Across Federated Datasets of Heterogeneous Multi-View Observations. *arXiv preprint arXiv:2204.07352* (2022).
- [4] Steffen Bickel and Tobias Scheffer. 2004. Multi-view clustering. In *ICDM*. 19–26.
- [5] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2016. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482* (2016).
- [6] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Multi-view k-means clustering on big data. In *IJCAI*.
- [7] Timothy J Castiglia, Anirban Das, Shiqiang Wang, and Stacy Patterson. 2022. Compressed-vfl: Communication-efficient learning with vertically partitioned data. In *ICML*. 2738–2766.
- [8] Sicong Che, Zhaoming Kong, Hao Peng, Lichao Sun, Alex Leow, Yong Chen, and Lifang He. 2022. Federated multi-view learning for private medical data integration and analysis. *TIST* (2022), 1–23.
- [9] Man-Sheng Chen, Ling Huang, Chang-Dong Wang, and Dong Huang. 2020. Multi-view clustering in latent embedding space. In *AAAI*. 3513–3520.
- [10] Don Kurian Dennis, Tian Li, and Virginia Smith. 2021. Heterogeneity for the win: One-shot federated clustering. In *ICML*. 2611–2620.
- [11] Shaohua Fan, Xiao Wang, Chuan Shi, Emiao Lu, Ken Lin, and Bai Wang. 2020. One2multi graph autoencoder for multi-view graph clustering. In *WWW*. 3070–3076.
- [12] Li Fei-Fei and Pietro Perona. 2005. A bayesian hierarchical model for learning natural scene categories. In *CVPR*. 524–531.
- [13] Fangxiang Feng, Xiaojie Wang, and Ruifan Li. 2014. Cross-modal retrieval with correspondence autoencoder. In *ACM MM*. 7–16.
- [14] Siwei Feng and Han Yu. 2020. Multi-participant multi-class vertical federated learning. *arXiv preprint arXiv:2001.11154* (2020).
- [15] Adrian Flanagan, Were Oyomno, Alexander Grigorievskiy, Kuan E Tan, Suleiman A Khan, and Muhammad Ammad-Ud-Din. 2021. Federated multi-view matrix factorization for personalized recommendations. In *ECML-PKDD*. 324–347.
- [16] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. 2017. Improved deep embedded clustering with local structure preservation. In *IJCAI*. 1753–1759.
- [17] Mingkai Huang, Hao Li, Bing Bai, Chang Wang, Kun Bai, and Fei Wang. 2020. A federated multi-view deep learning framework for privacy-preserving recommendations. *arXiv preprint arXiv:2008.10808* (2020).
- [18] Shudong Huang, Yazhou Ren, and Zenglin Xu. 2018. Robust multi-view data clustering with multi-view capped-norm k-means. *Neurocomputing* (2018), 197–208.
- [19] Shudong Huang, Wei Shi, Zenglin Xu, Ivor W Tsang, and Jiancheng Lv. 2022. Efficient federated multi-view learning. *PR* (2022), 108817.
- [20] Roy Jonker and Ton Volgenant. 1986. Improving the Hungarian assignment algorithm. *Operations Research Letters* (1986), 171–175.
- [21] Lusi Li, Zhiqiang Wan, and Haibo He. 2021. Incomplete multi-view clustering with joint partition and graph learning. *TKDE* (2021), 589–602.
- [22] Ruihuang Li, Changqing Zhang, Huazhu Fu, Xi Peng, Tianyi Zhou, and Qinghua Hu. 2019. Reciprocal multi-layer subspace learning for multi-view clustering. In *ICCV*. 8172–8180.
- [23] Zhenglai Li, Chang Tang, Xiao Zheng, Xinwang Liu, Wei Zhang, and En Zhu. 2022. High-order correlation preserved incomplete multi-view subspace clustering. *TIP* (2022), 2067–2080.
- [24] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. 2021. Completer: Incomplete multi-view clustering via contrastive prediction. In *CVPR*. 11174–11183.
- [25] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *SDM*. 252–260.
- [26] Yi Liu, Li Zhang, Ning Ge, and Guanghao Li. 2020. A systematic literature review on federated learning: From a model quality perspective. *arXiv preprint arXiv:2012.01973* (2020).
- [27] J MacQueen. 1967. Classification and analysis of multivariate observations. In *BSMSP*. 281–297.
- [28] Xi Peng, Jiashi Feng, Shijie Xiao, Wei-Yun Yau, Joey Tianyi Zhou, and Songfan Yang. 2018. Structured autoencoders for subspace clustering. *TIP* (2018), 5076–5086.
- [29] Liangrui Ren, Guoxian Yu, Jun Wang, Lei Liu, Carlotta Domeniconi, and Xiangliang Zhang. 2022. A Diversified Attention Model for Interpretable Multiple Clusterings. *TKDE* (2022), 1–12.
- [30] Morris Stallmann and Anna Wilbik. 2022. Towards Federated Clustering: A Federated Fuzzy  $c$ -Means Algorithm (FFCM). *arXiv preprint arXiv:2201.07316* (2022).
- [31] Huayi Tang and Yong Liu. 2022. Deep safe incomplete multi-view clustering: Theorem and algorithm. In *ICML*. 21090–21110.
- [32] Siwei Wang, Xinwang Liu, Li Liu, Wenxuan Tu, Xinzhang Zhu, Jiyuan Liu, Sihang Zhou, and En Zhu. 2022. Highly-efficient incomplete large-scale multi-view clustering with consensus bipartite graph. In *CVPR*. 9776–9785.
- [33] Yang Wang, Wenjie Zhang, Lin Wu, Xuemin Lin, Meng Fang, and Shirui Pan. 2016. Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering. *arXiv preprint arXiv:1608.05560* (2016).
- [34] Jie Wen, Zhihao Wu, Zheng Zhang, Lunke Fei, Bob Zhang, and Yong Xu. 2020. CDIMC-net: Cognitive Deep Incomplete Multi-view Clustering Network. In *IJCAI*. 3230–3236.
- [35] Jie Wen, Zheng Zhang, Zhao Zhang, Lunke Fei, and Meng Wang. 2020. Generalized incomplete multiview clustering with flexible locality structure diffusion. *IEEE transactions on cybernetics* (2020), 101–114.
- [36] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).
- [37] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*. 478–487.
- [38] Jie Xu, Chao Li, Liang Peng, Yazhou Ren, Xiaoshuang Shi, Heng Tao Shen, and Xiaofeng Zhu. 2023. Adaptive feature projection with distribution alignment for deep incomplete multi-view clustering. *TIP* 32 (2023), 1354–1366.
- [39] Jie Xu, Chao Li, Yazhou Ren, Liang Peng, Yujie Mo, Xiaoshuang Shi, and Xiaofeng Zhu. 2022. Deep Incomplete Multi-View Clustering via Mining Cluster Complementarity. In *AAAI*. 8761–8769.
- [40] Jie Xu, Yazhou Ren, Xiaoshuang Shi, Heng Tao Shen, and Xiaofeng Zhu. 2023. UNTIE: Clustering analysis with disentanglement in multi-view information fusion. *Information Fusion* 100 (2023), 101937.
- [41] Jie Xu, Yazhou Ren, Huayi Tang, Zhiheng Yang, Lili Pan, Yang Yang, Xiaorong Pu, Philip S. Yu, and Lifang He. 2023. Self-Supervised Discriminative Feature Learning for Deep Multi-View Clustering. *TKDE* 35, 07 (2023), 7470–7482.
- [42] Xiaohang Xu, Hao Peng, Lichao Sun, Md Zakirul Alam Bhuiyan, Lianzhong Liu, and Lifang He. 2021. Fedmod: Federated learning on mobile health data for mood detection. *arXiv preprint arXiv:2102.09342* (2021).
- [43] Mouxing Yang, Yunfan Li, Peng Hu, Jinfeng Bai, Jiancheng Lv, and Xi Peng. 2022. Robust multi-view clustering with incomplete information. *TPAMI* (2022), 1055–1069.
- [44] Xihong Yang, Xiaochang Hu, Sihang Zhou, Xinwang Liu, and En Zhu. 2022. Interpolation-based contrastive learning for few-label semi-supervised learning. *TNNLS* (2022).
- [45] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Xinwang Liu, and En Zhu. 2022. Contrastive Deep Graph Clustering with Learnable Augmentation. *arXiv preprint arXiv:2212.03559* (2022).
- [46] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. 2023. Cluster-guided Contrastive Graph Clustering Network. In *AAAI*. 10834–10842.
- [47] Changqing Zhang, Yajie Cui, Zongbo Han, Joey Tianyi Zhou, Huazhu Fu, and Qinghua Hu. 2020. Deep partial multi-view learning. *TPAMI* (2020), 2402–2415.
- [48] Handong Zhao, Zhengming Ding, and Yun Fu. 2017. Multi-view clustering via deep matrix factorization. In *AAAI*.
- [49] Qinghai Zheng, Jihua Zhu, Zhongyu Li, Shamin Pang, Jun Wang, and Yaochen Li. 2020. Feature concatenation multi-view subspace clustering. *Neurocomputing* (2020), 89–102.
- [50] Hangyu Zhu, Haoyu Zhang, and Yaochu Jin. 2021. From federated learning to federated neural architecture search: a survey. *Complex & Intelligent Systems* (2021), 639–657.