

# Federated Incomplete Multi-View Clustering with Tensorized Low-Rank Constraint

Wei Feng<sup>1</sup>, Danting Liu<sup>2</sup>, Qianqian Wang<sup>3</sup>, Mengping Jiang<sup>3</sup>, Bin Liu<sup>1\*</sup>

<sup>1</sup>College of Information Engineering, Northwest A&F University

<sup>2</sup>School of Computer Science and Technology, Xi'an Jiaotong University

<sup>3</sup>School of Telecommunication Engineerings, Xidian University

wei.feng@nwfufu.edu.cn, 2196123604@stu.xjtu.edu.cn, qqwang@xidian.edu.cn, mpjiang@foxmail.com, liubin0929@nwsuaf.edu.cn

## Abstract

Federated Multi-View Clustering has gained increasing attention for its ability to discover complementary clustering structures of distributed multi-view data while preserving data privacy. However, real-world clients often only have access to partial views, and the view incompleteness poses great challenges to federated multi-view feature fusion to exploit consistent and complementary information. Moreover, efficiency is highly expected in federated scenarios due to the limited resources of each client. To alleviate these issues, we propose **Federated Incomplete Multi-View Clustering with Tensorized Low-Rank Constraint (FIMVC-TLRC)**, which incorporates anchors to improve efficiency and is able to address prevalent view incompleteness issue in federated scenarios. FIMVC-TLRC aligns the local anchor graphs and employs a tensorized low-rank constraint based on the tensor Schatten  $p$ -norm to enforce the consistency of the data representations learned by each client. Besides, a federated optimization framework is developed to jointly optimize the construction and alignment of anchor graphs, thus enabling collaborative and privacy-preserving training. Experimental results on multiple datasets demonstrate its effectiveness.

## Introduction

Multi-view clustering (MVC) has emerged as an effective unsupervised method for handling multi-view data since it fully exploits the complementary and consistent information between views (Fang et al. 2023; Liu et al. 2024). Existing MVC methods can be broadly classified into deep MVC (Cui et al. 2024; Wen et al. 2023) and heuristic MVC (Wan et al. 2023), and both of them achieve impressive performance in clustering. Compared with deep MVC methods, heuristic methods achieve advantages in interoperability and hence still receive increasing attention. Despite the numerous heuristic MVC approaches developed, most of them assume that the multi-view data is located in a single party, ignoring the real-world multi-view data is likely to be collected and maintained by different entities. In this case, the data holders may not be willing to share data with others for privacy concerns, and thus the data isolation leads to a great challenge in the deployment of MVC algorithms.

Federated learning (Li et al. 2020; Zhang et al. 2021; Meng et al. 2024), which allows different entities to collaboratively train a global model without privacy leakage, has emerged as a possible solution to the above challenge. In particular, vertical federated learning (VFL) (Huang et al. 2022a) focuses on the scenario where different clients share the same sample space but different feature spaces, which is well-suited for distributed multi-view data processing by considering each view to be a subset of features. However, MVC requires the exploitation of consistent and complementary information embedded in different views, which is rarely explored by existing VFL works. As a result, VFL generally cannot obtain satisfactory performance on multi-view data.

To address this issue, several federated MVC (FedMVC) algorithms have been proposed. For example, Chen *et al.* (2023) proposed a self-supervised deep FedMVC method to learn consistent clustering assignments; Huang *et al.* (2022b) developed a non-negative matrix factorization (NMF) based FedMVC method. Despite the outstanding performance of these methods, their effectiveness hinges on the completeness of data across all views, while incomplete multi-view data is ubiquitous in practical scenarios due to issues like sensor failure and transmission noises. Although incomplete multi-view clustering (IMVC) has been widely studied in centralized settings, only a few works explore it in federated scenarios. Ren *et al.* (2024) proposed a deep Federated IMVC (FedIMVC) method with collaborative missing view imputation and cluster alignment. Nevertheless, heuristic FedIMVC methods still remain underexplored, despite their inherent advantages of simplicity and interpretability.

Currently, heuristic FedIMVC confronts several challenges. First, heuristic methods might produce high computational complexity when handling high-dimensional data. Moreover, in centralized settings, the missing views can be imputed with the data from other views, while in federated scenarios, the data isolation and privacy requirements hinder the information exchange between different views. As a result, the features for clustering results learned by different clients suffer from an inconsistency issue, resulting in unsatisfactory clustering performance. How to fuse local clustering outcomes from different clients into a unified global result and meanwhile keep data security is an important chal-

\*Corresponding Author

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

lenge to be solved.

To overcome these challenges, we propose a novel heuristic FedIMVC method named **Federated Incomplete Multi-View Clustering with Tensorized Low-Rank Constraint (FIMVC-TLRC)**, which incorporates anchor graph for clustering to improve its efficiency and scalability when handling high-dimensional data. To tackle the inconsistency problem due to local training, FIMVC-TLRC first aligns locally learned graphs with the global guidance matrix produced by the server-side model, which incorporates tensor Schatten p-norm to refine locally learned data representations from different clients, thereby efficiently exploring the consistent information between views. Finally, we design a federated optimization algorithm to optimize the model without exchanging any raw data. The main contributions are as follows:

- We propose a novel and efficient FedIMVC method based on anchor graph, *i.e.*, FIMVC-TLRC, which aligns the local graphs and introduces a tensor Schatten p-norm for local multi-view representation refinement to explore the consistent information in each view.
- We develop a federated optimization framework to support joint training of the construction and alignment of anchor graphs as well as clustering model.
- We conduct extensive experiments to demonstrate the effectiveness and superiority of our method.

## Related Work

### Federated Multi-view Clustering

Federated learning is a distributed machine learning framework designed to train a unified model by coordinating the training process on local data across multiple clients while maintaining data privacy for each client (Qi et al. 2025a,b). It can be broadly classified into Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL). Federated learning has quickly expanded its applications into the multi-view learning domain. In the context of FedMVC, Chen *et al.* (2023) developed a federated deep multi-view clustering method with global self-supervision. Huang *et al.* (2022b) constructed a federated multi-view clustering method using Non-negative Matrix Factorization (NMF) and K-means, which effectively reduces computational costs. Despite their effectiveness, these FedMVC methods all assume complete views and are not suitable for FedIMVC. Recently, Ren *et al.* (2024) proposed a FedIMVC method based on deep autoencoders to learn local features and perform adaptive imputation globally, which utilizes deep models and incurs high computational costs, making it unsuitable for scenarios with stringent efficiency requirements.

### Incomplete Multi-View Clustering

Existing incomplete multi-view clustering (IMVC) methods can be categorized into four types (Wen et al. 2022): deep learning-based IMVC (Xu et al. 2022), matrix factorization-based IMVC (Jia et al. 2021), kernel learning-based

IMVC (Wu, Feng, and Yuan 2024), and graph learning-based IMVC (Fu et al. 2024). Deep learning-based methods leverage neural networks to capture complex patterns and relationships within the data. Among the latter three methods, graph methods excel at exploring intrinsic geometric structures and exhibit impressive performance on linearly non-separable data. However, these methods are all designed for centralized scenarios, and how to deploy graph-based IMVC under federated settings efficiently remains underexplored.

## Proposed Method

### Problem Statement

**Problem Definition:** In the federated scenario, we consider a system consisting of a central server  $\mathcal{S}$  and  $M$  clients  $\{\mathcal{C}_1, \dots, \mathcal{C}_M\}$ , where incomplete multi-view data  $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^M\}$  is distributed across these clients. The incomplete local data held by  $\mathcal{C}_m$  is denoted as  $\mathbf{X}^m \in \mathbb{R}^{d_m \times n}$ , where  $d_m$  denotes the feature dimension of the  $m$ -th client and  $n$  denotes the sample number, but only  $n_m$  samples are available. Our proposed method leverages local computations at each client and coordinates these with the server to obtain a unified global clustering outcome. This approach ensures that sensitive data remains on the clients while achieving effective clustering across distributed datasets.

**Security Model and Design Goals:** FedMVC aims to empower all clients to collaboratively train a multi-view clustering model without leaking any personal information. In this work, we adopt the semi-honest adversary model, where all clients and the server are curious about others' private information but strictly follow the predefined protocol to behave honestly. However, they may attempt to infer others' private data from the information they have obtained during the protocol execution. Our underlying model adopts an anchor-based method and further performs feature extraction on this basis, thereby avoiding the exchange of raw data. Therefore, we have the following goals:

- **Security Goal:** Neither of the server and clients could obtain the raw data, which is sensitive to the data privacy.
- **Availability Goal:** The proposed FedIMVC method should be able to effectively partition the distributed incomplete multi-view data into disjoint clusters in a collaborative manner. The clustering performance should be comparable to the centralized MVC models.

### Objectives

FIMVC-TLRC is composed of four procedures: Initialization, Local Training, Global Training, and Aggregation. Specifically, the Initialization procedure initializes the necessary parameters on each client and server; the Local Training and Global Training respectively train the local models  $\{f^m(\cdot)\} (1 \leq m \leq M)$  on the client  $\mathcal{C}_m$  and a global model  $f_g(\cdot)$  for consistency assurance on the server  $\mathcal{S}$ . Aggregation is used to produce the final clustering result. In what follows, we introduce the local and global objectives in detail.

**Objective for Local Training:** For each client  $\mathcal{C}_m$ , it trains a local clustering model  $\mathbf{F}^m \leftarrow f^{(m)}(\hat{\mathbf{F}}^m, \mathbf{X}^m)$  on

their local data  $\mathbf{X}^m \in \mathbb{R}^{d_m \times n}$ , where  $\mathbf{F}^m$  represents the learned local high-level representations, and  $\hat{\mathbf{F}}^m$  represents guidance matrix output by the global model  $f_g(\cdot)$  from  $\mathcal{S}$  for  $\mathcal{C}_m$ ;  $f_g^{(m)}(\cdot)$  refers to taking the  $m$ -th component of  $f_g(\cdot)$ 's output. To efficiently exploit the complementary information of multi-view data, each client first constructs a local anchor graph and then utilizes a structural alignment module to align the anchors.

**Local Anchor Graph Construction:** Traditional graph-based IMVC methods generally require constructing a graph from all samples, resulting in computational complexity of  $\mathcal{O}(n^3)$  and space complexity of  $\mathcal{O}(n^2)$ , which limits their scalability. For better efficiency, our clustering method is built on anchor graph, which establishes relationships between a small number of representative anchors and the samples. For each client  $\mathcal{C}_m$ , it can build its anchor graph with local data by optimizing the following objective:

$$\begin{aligned} \min_{\mathbf{G}^m, \mathbf{A}^m, \alpha_m} \quad & \alpha_m^2 \|\mathbf{X}^m \mathbf{W}^m - \mathbf{A}^m \mathbf{G}^m \mathbf{W}^m\|_F^2 + \beta \|\mathbf{G}^m\|_F^2 \\ \text{s.t.} \quad & \mathbf{A}^{m\top} \mathbf{A}^m = \mathbf{I}, \quad \mathbf{G}^m \geq 0, \mathbf{G}^{m\top} \mathbf{1} = 1 \end{aligned} \quad (1)$$

where  $\mathbf{A}^m \in \mathbb{R}^{d_m \times \sigma}$  denotes the anchor matrix from the  $m$ -th view by  $\mathcal{C}_m$ ,  $\sigma$  is the anchor number;  $\mathbf{W}^m \in \mathbb{R}^{n \times n}$  is the diagonal indicator matrix, and  $\mathbf{W}_{ii}^m = 0$  means the  $i$ -th sample of the  $m$ -th view is missing;  $\mathbf{G}^m \in \mathbb{R}^{\sigma \times n}$  is the view-specific anchor graph;  $\mathbf{G}^m \mathbf{W}^m$  can be considered as the similarities between  $\sigma$  anchors and  $n_m$  available instances;  $\alpha_m$  and  $\beta$  are parameters to balance the influence of each term, and  $\sum_{m=1}^M \alpha_m = 1$ . We enforce the orthogonality of  $\mathbf{A}^m$  to enhance discriminability, *i.e.*,  $\mathbf{A}^{m\top} \mathbf{A}^m = \mathbf{I}_m$ .

**Local Anchor Alignment:** The local anchors and anchor graphs are selected independently by each client, which leads to two problems, *i.e.*, (1) anchor alignment: the anchors selected by each client does not correspond to those selected by other clients; and (2) inconsistency issue: the learned anchor graphs do not characterize the consistent features of each sample, degrading the final clustering performance. To address these issues, we employ the server to refine the anchor graphs to be consistent based on tensor learning, rather than enforce a single and consistent anchor graph to be learned. The refined anchor graphs are distributed to each client as guidance to instruct local anchor graph learning. Then, we leverage a permutation matrix  $\mathbf{H}^m$  to structurally align the local anchor graph  $\mathbf{G}^m$  the guidance matrix from the server, resulting in view-specific representations  $\mathbf{F}^m$ :

$$\begin{aligned} \min_{\alpha_m, \mathbf{A}^m, \mathbf{G}^m, \mathbf{H}^m, \mathbf{F}^m} \quad & \alpha_m^2 \|\mathbf{X}^m \mathbf{W}^m - \mathbf{A}^m \mathbf{G}^m \mathbf{W}^m\|_F^2 \\ & + \beta \|\mathbf{G}^m\|_F^2 + \gamma \|\mathbf{H}^m \mathbf{G}^m - \mathbf{F}^m\|_F^2 \\ \text{s.t.} \quad & \mathbf{A}^{m\top} \mathbf{A}^m = \mathbf{I}, \mathbf{H}^{m\top} \mathbf{H}^m = \mathbf{I}, \mathbf{G}^m \geq 0, \\ & \mathbf{G}^{m\top} \mathbf{1} = 1, \mathbf{F}^m \mathbf{F}^{m\top} = \mathbf{I}, \mathbf{F}^m = f_g^m(\{\mathbf{F}^m\}) \end{aligned} \quad (2)$$

where  $\gamma$  is the trade-off parameter to balance the influence between anchor graph construction and the alignment module; The core idea is to enforce the local learned high-level representation  $\mathbf{F}^m$  to be similar to the refined representation learned by  $f_g(\cdot)$ . It should be noted that the guidance matrix is dynamic rather than constant during training rounds.

**Objective for Global Training:** The goal of the global training is to refine local high-level representation  $\{\mathbf{F}^m\} (1 \leq m \leq M)$  via tensor learning to exploit the consistent and complementary information. Denote the refined representation for each view as  $\hat{\mathbf{F}}^m$ , and  $\hat{\mathcal{F}}$  is a tensor constructed by stacking all the  $\hat{\mathbf{F}}^m$ , we refine the local representations by minimizing the tensor low-rank objective of based on tensor Schatten-p norm and simultaneously enforcing the refined representation to be similar to the outputs of local models  $\{f^{(m)}(\cdot)\}$  via a constraint. Following this line, the global objective to be optimized is as follows

$$\min_{\hat{\mathcal{F}}} \lambda \|\hat{\mathcal{F}}\|_{S_p}^p \text{ s.t. } \hat{\mathbf{F}}^m = f^{(m)}(\hat{\mathbf{F}}^m, \mathbf{X}^m) (1 \leq m \leq M) \quad (3)$$

where  $\|\cdot\|_{S_p}^p$  represents the tensor Schatten-p norm. Then,  $f_g^m(\cdot)$  returns  $\hat{\mathbf{F}}^m$  the guidance for  $\mathcal{C}_m$  by taking the  $m$ -th frontal slice of  $\hat{\mathcal{F}}$ ;  $\lambda$  is a hyper-parameter to balance the influence of the global training.

## Federated Optimization

The optimization of local objectives and global objective requires the interaction between  $\mathcal{S}$  and  $\mathcal{C}_m$  by sharing the model output with each other. We adopt the Alternative Iteration Method for optimization, and in this case, the collaborative optimization of the local and global objectives is equivalent to optimizing the following objective:

$$\begin{aligned} \min \quad & \sum_{m=1}^M \underbrace{\left\{ \alpha_m^2 \|\mathbf{X}^m \mathbf{W}^m - \mathbf{A}^m \mathbf{G}^m \mathbf{W}^m\|_F^2 + \beta \|\mathbf{G}^m\|_F^2 \right\}}_{\text{Local Anchor Graph Construction}} \\ & + \gamma \underbrace{\sum_{m=1}^M \|\mathbf{H}^m \mathbf{G}^m - \mathbf{F}^m\|_F^2}_{\text{Local Anchor Alignment}} + \underbrace{\lambda \|\mathcal{F}\|_{S_p}^p}_{\text{Global Training Objective}} \\ \text{s.t.} \quad & \alpha^\top \mathbf{1} = 1, \mathbf{A}^{m\top} \mathbf{A}^m = \mathbf{I}, \mathbf{H}^{m\top} \mathbf{H}^m = \mathbf{I}, \\ & \mathbf{G}^m \geq 0, \mathbf{G}^{m\top} \mathbf{1} = 1, \mathbf{F}^m \mathbf{F}^{m\top} = \mathbf{I} \end{aligned} \quad (4)$$

where  $\mathcal{F}$  is a tensor by stacking all  $\mathbf{F}^m$ s.

In our federated framework, the influence of  $\mathcal{S}$  and  $\mathcal{C}_m$  on each other is achieved with the constraint  $\mathbf{F}^m = f^m(\mathbf{X}^m)$  and  $\hat{\mathbf{F}}^m = f_g^m(\{\mathbf{F}^m\})$ , which enforces the local representations to be similar to be the globally refined representation. Since multi-view data is distributed across multiple clients, and for the sake of privacy, sensitive data should not be shared, which brings a great challenge for optimization. However, the high-level representation  $\mathbf{F}^m$  learned based on local anchor graph projection does not lead to data leakage and can be transmitted between the server and clients. Based on these analyses, we illustrate our federated optimization framework as follows.

**System Initialization:** The server and all the clients negotiate the consistent public parameters. Each client initializes  $\mathbf{H}^m$  and  $\mathbf{F}^m$  as identity matrices and sets  $\alpha_m$  to  $1/M$ , while the server initializes  $\hat{\mathcal{F}}$  as empty tensors.

**Local Training:** Suppose the guidance matrix output by the global model  $f_g^m(\{\mathbf{F}_m\})$  is  $\hat{\mathbf{F}}^m$ , and by introducing Augmented Lagrange Multiplier (ALM) method, the objective optimized by  $\mathcal{C}_m$  is transformed to:

$$\begin{aligned} & \min_{\alpha_m, \mathbf{A}^m, \mathbf{G}^m, \mathbf{H}^m, \mathbf{F}^m} \alpha_m^2 \|\mathbf{X}^m \mathbf{W}^m - \mathbf{A}^m \mathbf{G}^m \mathbf{W}^m\|_F^2 \\ & + \beta \sum_{m=1}^M \|\mathbf{G}^m\|_F^2 + \gamma \|\mathbf{H}^m \mathbf{G}^m - \mathbf{F}^m\|_F^2 \quad (5) \\ & + \frac{\rho}{2} \|\mathbf{F}^m - \hat{\mathbf{F}}^m + \frac{\mathbf{B}^m}{\rho}\|_F^2 \end{aligned}$$

where  $\mathbf{B}^m$  is the Lagrange multiplier,  $\rho$  is penalty parameter. By performing Alternative Iteration Method on it, the local training can be divided into the following subproblems.

(1). **Update  $\mathbf{A}^m$ :** Removing the irrelevant terms, the local objective becomes:

$$\min_{\mathbf{A}^{m\top} \mathbf{A}^m = \mathbf{I}} \alpha_m^2 \|\mathbf{X}^m \mathbf{W}^m - \mathbf{A}^m \mathbf{G}^m \mathbf{W}^m\|_F^2 \quad (6)$$

Rewrite the objective with traces and remove the irrelevant terms, Eq.(6) becomes:

$$\max_{\mathbf{A}^{m\top} \mathbf{A}^m = \mathbf{I}} \text{Tr}(\mathbf{A}^{m\top} \mathbf{Q}^m) \quad (7)$$

where  $\mathbf{Q}^m = \mathbf{X}^m \mathbf{W}^m \mathbf{W}^{m\top} \mathbf{G}^{m\top}$ . The optimal solution of  $\mathbf{A}^m$  can be obtained by computing  $\mathbf{F}^{m*} = \mathbf{U}^m \mathbf{V}^{m\top}$ , where  $\mathbf{U}^m$  and  $\mathbf{V}^m$  are the left and right singular matrices obtained by performing SVD decomposition on  $\mathbf{Q}^m$ .

(2). **Update  $\mathbf{G}^m$ :**

$$\begin{aligned} & \min_{\mathbf{G}^m \geq 0, \mathbf{G}^{m\top} \mathbf{1} = 1} \alpha_m^2 \|\mathbf{X}^m \mathbf{W}^m - \mathbf{A}^m \mathbf{G}^m \mathbf{W}^m\|_F^2 \\ & + \beta \|\mathbf{G}^m\|_F^2 + \gamma \|\mathbf{H}^m \mathbf{G}^m - \mathbf{F}^m\|_F^2 \quad (8) \end{aligned}$$

To fully leverage global information to guide the update of local anchor graphs, we use the consistency high-level representation  $\mathbf{F} = \frac{1}{M} \sum_{m=1}^M \mathbf{F}^m$  aggregated by the server to replace the local representation  $\mathbf{F}^m$  in this step. Then, after a simple algebraic derivation, the optimization of Eq.(8) can be formulated as:

$$\begin{aligned} & \min_{\mathbf{G}^m \geq 0, \mathbf{G}^{m\top} \mathbf{1} = 1} \text{Tr}(\mathbf{G}^{m\top} \mathbf{G}^m (\alpha_m^2 \mathbf{W}^m \mathbf{W}^{m\top} + (\beta + \gamma) \mathbf{I})) \\ & - 2\text{Tr}(\mathbf{G}^{m\top} (\alpha_m^2 \mathbf{A}^{m\top} \mathbf{X}^m \mathbf{W}^m \mathbf{W}^{m\top} + \gamma \mathbf{H}^{m\top} \mathbf{F})) \quad (9) \end{aligned}$$

Since  $\mathbf{W}$  is a diagonal matrix, it is easy to obtain that  $\alpha_m^2 \mathbf{W}^m \mathbf{W}^{m\top} + (\beta + \gamma) \mathbf{I}$  is a diagonal matrix and all the diagonal elements are non-zero. Denote  $\alpha_m^2 \mathbf{W}^m \mathbf{W}^{m\top} + (\beta + \gamma) \mathbf{I}$  and  $\alpha_m^2 \mathbf{A}^{m\top} \mathbf{X}^m \mathbf{W}^m \mathbf{W}^{m\top} + \gamma \mathbf{H}^{m\top} \mathbf{F}$  as  $\mathbf{Q}_1^m$  and  $\mathbf{Q}_2^m$ , Eq. (9) is transformed to:

$$\min_{\mathbf{G}^m \geq 0, \mathbf{G}^{m\top} \mathbf{1} = 1} \text{Tr}(\frac{1}{2} \mathbf{G}^{m\top} \mathbf{G}^m - \mathbf{G}^{m\top} \mathbf{Q}_2^m (\mathbf{Q}_1^m)^{-1}) \quad (10)$$

Introducing the Lagrange multiplier to Eq. (10), we get:

$$\begin{aligned} \mathcal{L}(\mathbf{G}^{(m)}, \nu, \Lambda) & = \text{Tr}(\frac{1}{2} \mathbf{G}^{m\top} \mathbf{G}^m - \mathbf{G}^{m\top} \mathbf{Q}_2^m (\mathbf{Q}_1^m)^{-1}) \\ & - \nu^T \mathbf{G}^{m\top} \mathbf{1} - \text{Tr}(\Lambda^T \mathbf{G}^m) \quad (11) \end{aligned}$$

where  $\nu$  and  $\Lambda$  are Lagrange multipliers. Given the derivation of  $\mathcal{L}(\mathbf{G}^m, \nu, \Lambda)$  with regard to  $\mathbf{G}^m$ :  $\mathbf{G}^m - \mathbf{Q}_2^m (\mathbf{Q}_1^m)^{-1} - \nu \mathbf{1} - \Lambda$ , the solution is given according to the Karush-Kuhn-Tucker (KKT) condition as follows:

$$\begin{cases} \mathbf{G}^m - \mathbf{Q}_2^m (\mathbf{Q}_1^m)^{-1} - \nu \mathbf{1} - \Lambda = 0 \\ \mathbf{G}^{m\top} \mathbf{1} = 1 \\ \Lambda^T \mathbf{G}^m = 0 \end{cases} \quad (12)$$

Using Newton method to optimize  $\nu$ , the solution is:

$$\mathbf{G}^m = \max(\mathbf{Q}_2^m (\mathbf{Q}_1^m)^{-1} - \nu \mathbf{1}, 0) \quad (13)$$

(3). **Update  $\mathbf{H}_m$ :** The objective can be written as:

$$\begin{aligned} & \min_{\mathbf{H}^{m\top} \mathbf{H}^m = \mathbf{I}} \gamma \|\mathbf{H}^m \mathbf{G}^m - \mathbf{F}^m\|_F^2 \\ & \Leftrightarrow \max_{\mathbf{H}^{m\top} \mathbf{H}^m = \mathbf{I}} \text{Tr}(\mathbf{H}^{m\top} \mathbf{F}^m \mathbf{G}^{m\top}) \quad (14) \end{aligned}$$

Similar to optimizing  $\mathbf{G}^m$ , we replace  $\mathbf{F}^m$  with  $\mathbf{F}$ , and  $\mathbf{H}^m$  can be optimized by SVD decomposition on  $\mathbf{F} \mathbf{G}^{m\top}$ .

(4). **Update  $\alpha_m$ :** We only focus on the term related to  $\alpha_m$ :

$$\min_{\alpha^T \mathbf{1} = 1} \alpha_m^2 \|\mathbf{X}^m \mathbf{W}^m - \mathbf{A}^m \mathbf{G}^m \mathbf{W}^m\|_F^2 \quad (15)$$

Let  $\tau_m = \|\mathbf{X}^m \mathbf{W}^m - \mathbf{A}^m \mathbf{G}^m \mathbf{W}^m\|_F^2$ , we apply the Cauchy-Schwarz inequality as:

$$\left( \sum_{m=1}^M \alpha_m^2 \tau_m \right) \left( \sum_{m=1}^M \frac{1}{\tau_m} \right) \geq \left( \sum_{m=1}^M \alpha_m \right)^2 \quad (16)$$

The equality holds when  $\alpha_m = k/\tau_m$ , where  $k$  is a constant. Based on the constraint  $\alpha^T \mathbf{1} = 1$ , we can solve for  $k$  as  $k = \sum_{m=1}^M 1/\tau_m$ . Finally, the optimal  $\alpha_m$  is given by:

$$\alpha_m = \frac{1/\tau_m}{\sum_{m=1}^M 1/\tau_m} \quad (17)$$

**Step 5. Update  $\mathbf{F}^m$ :**

$$\min_{\mathbf{F}^m \mathbf{F}^{m\top} = \mathbf{I}} \gamma \|\mathbf{H}^m \mathbf{G}^m - \mathbf{F}^m\|_F^2 + \frac{\rho}{2} \|\mathbf{F}^m - \hat{\mathbf{F}}^m + \frac{\mathbf{B}^m}{\rho}\|_F^2 \quad (18)$$

where  $\mathbf{B}^m$  is Lagrange multiplier and is updated by  $\mathbf{B}^m = \mathbf{B}^m + \rho(\mathbf{F}^m - \hat{\mathbf{F}}^m)$ . Eq. (18) can be transformed into:

$$\begin{aligned} & \max_{\mathbf{F}^m} (\gamma \mathbf{H}^m \mathbf{G}^m + \frac{\rho}{2} (\hat{\mathbf{F}}^m - \frac{\mathbf{B}^m}{\rho})) \mathbf{F}^{m\top} \\ & \text{s.t., } \mathbf{F}^m \mathbf{F}^{m\top} = \mathbf{I} \quad (19) \end{aligned}$$

We solve the optimal  $\mathbf{F}^m$  by performing SVD on  $\gamma \mathbf{H}^m \mathbf{G}^m + \frac{\rho}{2} (\hat{\mathbf{F}}^m - \frac{\mathbf{B}^m}{\rho})$ . The optimal  $\mathbf{F}^m$  is sent to  $\mathcal{S}$  for refinement.

**Global Training:** The only variable to be updated by  $\mathcal{S}$  is  $\hat{\mathcal{F}}$ . By performing ALM, the global objective function w.r.t  $\hat{\mathcal{F}}$  becomes:

$$\min_{\hat{\mathcal{F}}} \gamma \|\hat{\mathcal{F}}\|_{S_p}^2 + \frac{\rho}{2} \|\mathcal{F} - \hat{\mathcal{F}} + \frac{\mathcal{B}}{\rho}\|_F^2 \quad (20)$$

where  $\mathcal{B}$  is Lagrange multiplier and is updated by  $\mathcal{B} = \mathcal{B} + \rho(\mathcal{F} - \hat{\mathcal{F}})$ ,  $\rho$  is the penalty parameter. The solution is

$$\hat{\mathcal{F}}^* = \text{ifft}(\mathbf{U} * \mathcal{D}_{\frac{\gamma}{\rho}, p}(\mathcal{P}) * \mathbf{V}^T) \quad (21)$$

where  $\mathcal{P} = \mathcal{F} + \frac{\hat{\mathcal{F}}}{\rho}$ , and  $\mathbf{U}$  and  $\mathbf{V}$  are acquired via t-SVD of  $\mathcal{P}$ , i.e.,  $\mathcal{P} = \mathbf{U} * \mathcal{S} * \mathbf{V}^T$ .

---

**Algorithm 1: FIMVC-TLRC**

---

**Input:** Incomplete data  $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_M\}$ ; Missing indicator matrix  $\mathbf{W}_m$ ; cluster number  $K$ .

```
1: Each client  $\mathcal{C}_m$  initializes  $\mathbf{W}^m, \mathbf{H}^m$  and  $\alpha_m$ ;  
2: while not converged do  
3:   for  $m = 1$  to  $M$  do  
4:      $\triangleright$  On  $m$ -th client  $\mathcal{C}_m$   
5:     Update weight  $\alpha_m$  according to Eq. (17)  
6:     Update  $\mathbf{A}^m, \mathbf{G}^m, \mathbf{F}^m, \mathbf{H}^m$   
7:     Compute  $\tau_m$   
8:     Send  $\mathbf{F}^m$  and  $1/\tau_m$  to the global server  
9:   end for  
10:   $\triangleright$  On the Server  $\mathcal{S}$   
11:  Update  $\hat{\mathcal{F}}$   
12:  Aggregate  $\mathbf{F}_m$  into  $\mathbf{F}$   
13:  Send  $\hat{\mathbf{F}}_m$  and  $\mathbf{F}$  to  $\mathcal{C}_m$   
14:  Compute  $\sum_{m=1}^M 1/\tau_m$  and send it to all clients  
15: end while  
16: Obtain  $\mathbf{U}$  by performing SVD on  $\mathbf{F}$   
Output: Perform k-means on  $\mathbf{U}$  to obtain labels
```

---

**Aggregation:** After the model converges,  $\mathcal{S}$  computes  $\mathbf{F} = \frac{1}{M} \sum_{m=1}^M \mathbf{F}^m$  and then performs SVD on  $\mathbf{F}$  to learn  $\mathbf{U}$  and applies k-means on  $\mathbf{U}$  to obtain the clustering result.

### Workflow

FIMVC-TLRC requires the transmission of certain parameters to support the training of collaborative and privacy-preserving models. To better illustrate the working procedures of FIMVC-TLRC, we briefly introduce its workflow here. Firstly, each client  $\mathcal{C}_m$  conducts initialization by setting  $\mathbf{G}^m$  as zero matrix and  $\mathbf{H}^m$  and  $\mathbf{F}^m$  as identity matrix, and setting  $\alpha_m$  with  $1/M$ . After initialization,  $\mathcal{C}_m$  updates  $\mathbf{A}^m, \mathbf{G}^m, \mathbf{H}^m, \mathbf{F}^m$ , and calculates  $\tau_m$ . Then, each client sends  $\tau_m$  and  $\mathbf{F}^m$  to the server  $\mathcal{S}$ . The server  $\mathcal{S}$ , after receiving all the parameters, stacks all the  $\mathbf{F}^m$  into a tensor  $\mathcal{F}$ , updates  $\hat{\mathcal{F}}$  according to Eq.(21), and updates  $\mathcal{W}$ . Subsequently,  $\mathcal{S}$  will also calculate  $\sum_{m=1}^M \frac{1}{\tau_m}$  and global consistent representation  $\mathbf{F} = \left[ \sum_{m=1}^M \frac{\mathbf{F}_m}{M} \right]$ , send them to all clients, and send the updated  $\hat{\mathbf{F}}^m$  to  $\mathcal{C}_m$ , where client  $\mathcal{C}_m$  can update  $\alpha_m$  using Eq.(17). This process is repeated until the model converges. Finally, the server aggregates the representations into a unified one and obtain the final clustering results. We summarize the algorithm in Algorithm 1.

### Complexity Analysis

**Computational Complexity:** (1) **Local client-side:** Each client's computational load includes updating  $\mathbf{A}^m, \mathbf{G}^m, \mathbf{F}^m, \mathbf{H}^m$ , and  $\alpha_m$ . In each local iteration, the computational complexity of updating  $\mathbf{A}^m, \mathbf{F}^m$ , and  $\mathbf{H}^m$  primarily stems from SVD decomposition, with approximate complexities of  $\mathcal{O}(n\sigma d_m + \sigma^2 d_m), \mathcal{O}(n\sigma^2 M)$ , and  $\mathcal{O}(n\sigma^2 M + M^3 \sigma)$ . The time complexity for updating  $\mathbf{G}^m$  and  $\alpha_m$  is dominated by matrix multiplication, with a computational cost of  $\mathcal{O}(n\sigma d_m)$ . Therefore, the total computational complex-

ity of the client is  $\mathcal{O}(n(\sigma d_m + \sigma^2 M) + \sigma^3 M + \sigma^2 d_m)$ . (2) **Global server-side:** The server needs to update  $\hat{\mathcal{F}}$  and  $\mathcal{W}$ , where the primary computational cost arises from the DFT and SVD operations, and the computational complexity is  $\mathcal{O}(Mn\sigma \log(Mn) + M^2 n\sigma)$ .

**Communication Complexity:** Furthermore, we analyze the transmission load between each client and the global server. **Local Client  $\rightarrow$  Global Server:** Each client transmits its local  $\mathbf{F}^m$  to the global server before aggregation. For all clients, the amount of data transmitted in each communication round is  $\mathcal{O}(n\sigma M)$ . **Global Server  $\rightarrow$  Local Client:** The global server sends the  $m$ -th slice of  $\hat{\mathcal{F}}$  to the  $m$ -th client along with the computed  $\sum_{m=1}^M 1/\tau_m$  after aggregation. Thus, the total amount of data transmitted in each communication round is  $\mathcal{O}(n\sigma M)$ .

### Experiment

**Datasets:** We evaluate our method on eight public multi-view datasets: (1)**Yale**; (2)**WebKB** (Blum and Mitchell 1998); (3) **BDGP** (Cai et al. 2012); (4)**Caltech101-7** (Fei-Fei, Fergus, and Perona 2004); (5)**NGs** (Hussain, Bisson, and Grimal 2010); (6)**Sonar** (Sejnowski and Gorman 1988); (7)**SentencesNYU v2(RGB-D)** (Silberman et al. 2012); (8)**CiteSeer** (Apté, Damerau, and Weiss 1994). Following (Li et al. 2022), we randomly remove samples from each view for incomplete data. For the federated setting, our experiments include a server and multiple clients, each holding a view of the data.

**Compared Methods:** We compared our methods with nine clustering methods: (1) **BSV** (Zhao, Liu, and Fu 2016); (2)**Concat** (Zhao, Liu, and Fu 2016); (3)**DAIMC** (Hu and Chen 2019); (4)**PIC** (Wang et al. 2019); (5)**IMVC\_AGL** (Wen, Xu, and Liu 2018); (6)**IMVC-CBG** (Wang et al. 2022); (7)**VKMC** (Huang et al. 2022a); (8)**FedMVL** (Huang et al. 2022b); (9)**TensorFMVC** (Feng et al. 2024). where (1)-(6) are centralized multi-view methods, (7)-(9) are federated multi-view methods.

### Experimental Results

Figure 1, Table 1, and Table 2 present the clustering results on eight multi-view datasets. The following points are easy to observe: (1) Compared with existing centralized IMVC methods, FIMVC-TLRC demonstrates superior performance across most datasets. For example, on the Yale dataset, our method outperforms the second-best method in terms of ACC, NMI, and PUR by 11.54%, 11.87%, and 11.87%, respectively. Additionally, as shown in Figure 1, the clustering performance of FIMVC-TLRC consistently surpasses that of other methods, demonstrating the effectiveness of Local Anchor Construction and Tensorial Representation Fusion; (2) Compared to the existing vertical federated method VKMC, both approaches meet the requirements for data privacy protection. However, our method significantly outperforms VKMC in overall performance, as it fully explores the consistency structure between views through a tensor regularization term and can be applied to incomplete data; (3) In contrast to FedMVL, our approach is

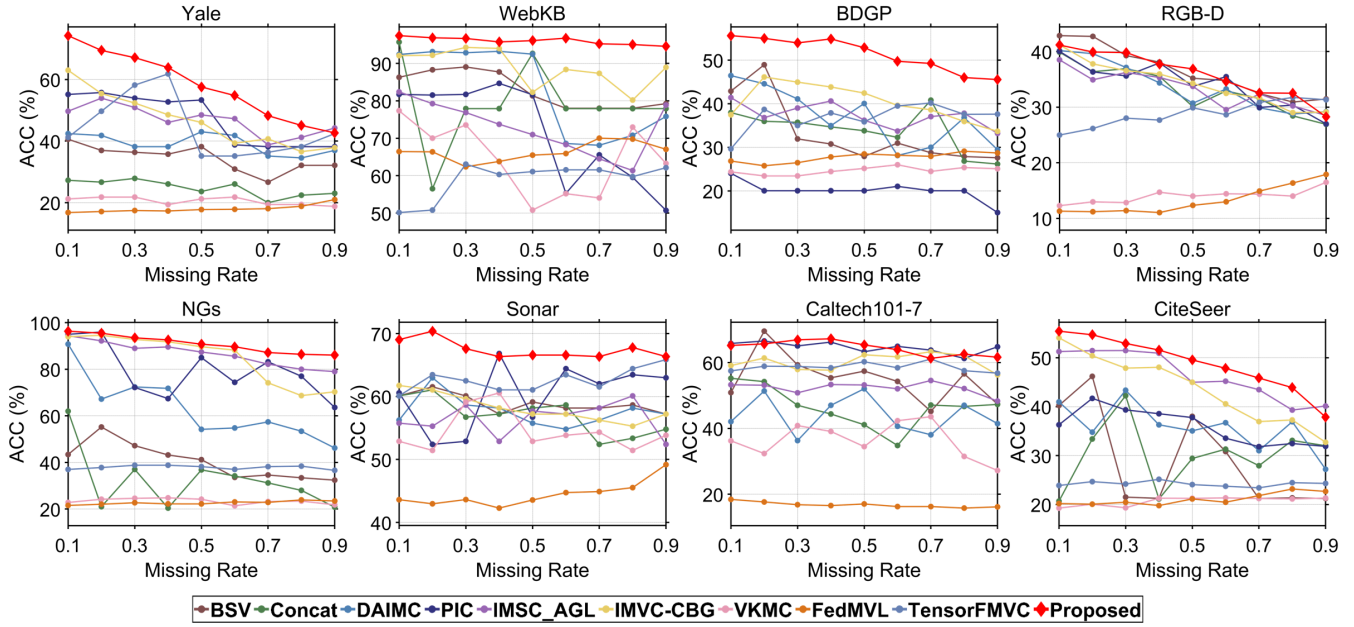


Figure 1: Clustering performance of FIMVC-TLRC on eight multi-view datasets with different missing rates.

Datasets	Yale			WebKB			BDGP			Caltech101-7		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
BSV	34.41	35.62	36.30	82.89	16.43	82.93	33.07	9.23	33.41	55.10	36.98	75.90
Concat	24.78	29.56	28.22	79.16	13.80	81.69	33.80	9.12	34.03	46.42	35.31	77.94
DAIMC	39.12	44.73	41.62	83.05	31.30	<u>86.29</u>	36.83	11.83	37.16	43.99	39.74	83.14
PIC	<u>47.13</u>	50.99	48.30	71.38	7.37	80.42	20.04	0.16	20.16	<b>64.63</b>	<b>51.57</b>	<b>85.89</b>
IMSC_AGL	46.73	50.29	48.01	72.93	6.66	78.80	37.31	12.89	38.65	52.30	42.59	83.78
IMVC-CBG	46.65	<u>52.84</u>	48.94	<u>88.89</u>	<u>45.85</u>	83.43	40.29	17.84	40.86	60.20	43.75	81.06
VKMC	20.54	24.99	21.42	64.56	2.69	78.12	24.64	1.82	25.67	36.39	3.58	57.65
FedMVL	18.02	16.31	19.47	66.36	17.45	68.55	27.70	5.05	33.65	16.78	7.98	22.54
TensorFMVC	44.24	51.75	46.06	58.94	3.53	78.11	<u>44.24</u>	<u>19.38</u>	<u>46.06</u>	58.62	5.29	58.94
FIMVC-TLRC	<b>58.67</b>	<b>64.71</b>	<b>60.81</b>	<b>96.05</b>	<b>70.04</b>	<b>96.05</b>	<b>51.41</b>	<b>22.12</b>	<b>51.43</b>	<u>64.41</u>	<u>49.65</u>	<u>84.14</u>

Table 1: Clustering performance of ACC(%), NMI(%), and PUR(%) on Yale, WebKB, BDGP, and Caltech101-7 datasets.

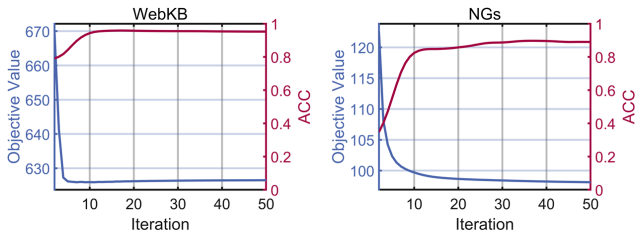


Figure 2: The convergence curves of FIMVC-TLRC.

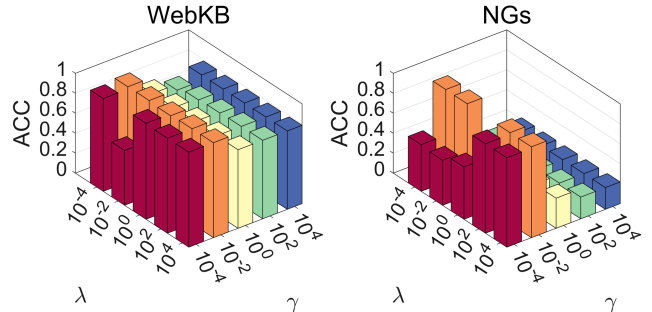


Figure 3: ACC w.r.t.  $\gamma$  and  $\lambda$ .

more efficient due to it is based on anchor graph, and it leverages tensor low-rank constraints to uncover complementary information between views in incomplete data, resulting in superior clustering performance.

**Ablation Experiments:** We conducted ablation studies and summarize the results in Table 3. We evaluate the per-

Datasets	NGs			Sonar			RGB-D			CiteSeer		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
BSV	40.47	18.39	41.31	58.92	1.79	58.92	<u>35.58</u>	<b>26.80</b>	48.04	29.09	6.81	29.87
Concat	32.38	14.20	32.96	56.94	1.81	57.05	33.18	26.09	45.80	30.13	7.65	30.85
DAIMC	63.13	45.62	64.27	57.59	1.81	57.59	33.68	23.27	46.51	35.81	13.39	38.09
PIC	79.31	62.46	79.58	60.26	4.14	60.42	35.50	25.74	<u>48.16</u>	35.93	13.05	37.92
IMSC_AGL	<u>86.60</u>	66.74	<u>86.60</u>	56.46	1.49	56.62	33.26	19.85	44.11	<u>46.47</u>	<u>21.00</u>	<u>50.58</u>
IMVC-CBG	84.92	<u>71.48</u>	84.92	58.19	2.29	58.19	34.23	25.27	45.54	43.65	18.17	46.11
VKMC	23.38	2.09	23.64	54.49	<u>6.59</u>	55.02	14.01	3.22	40.89	20.69	3.89	21.78
FedMVL	22.66	2.15	26.89	44.49	1.75	46.63	13.27	4.65	17.18	21.08	1.89	25.51
TensorFMVC	37.86	14.56	38.30	<u>62.61</u>	5.08	<u>62.61</u>	28.81	24.75	47.29	24.20	4.95	24.50
FIMVC-TLRC	<b>90.90</b>	<b>77.39</b>	<b>90.90</b>	<b>67.45</b>	<b>9.22</b>	<b>67.45</b>	<b>35.94</b>	<u>26.64</u>	<b>48.63</b>	<b>48.85</b>	<b>23.94</b>	<b>51.87</b>

Table 2: Clustering performance of ACC(%), NMI(%), and PUR(%) on NGs, Sonar, RGB-D, and CiteSeer

Dataset	Yale			RGB-D			BDGP			NGs		
	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity
Fixed-Anchor	40.41	44.79	41.80	17.32	6.71	30.67	23.36	4.93	24.65	27.13	2.41	27.61
w.o. A-A & T-R	43.22	49.91	45.00	32.11	21.67	39.67	43.14	17.15	44.80	61.01	36.89	61.14
w.o. T-R	44.44	50.34	46.34	34.82	26.61	46.89	47.89	20.22	49.12	86.44	73.56	83.33
FIMVC-TLRC	<b>58.61</b>	<b>64.71</b>	<b>60.81</b>	<b>35.94</b>	<b>26.64</b>	<b>48.63</b>	<b>51.41</b>	<b>22.12</b>	<b>51.43</b>	<b>90.90</b>	<b>77.39</b>	<b>90.90</b>

Table 3: Clustering results of the ablation experiments.

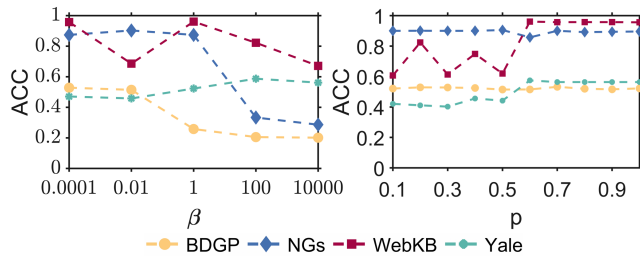


Figure 4: ACC w.r.t.  $\beta$  and  $p$ .

formance of FIMVC-TLRC in four cases: (1) Fixed anchor graph FIMVC-TLRC (**Fixed-Anchor**): The anchors are initialized by k-means cluster centers without updating during the optimization process. (2) FIMVC-TLRC without anchors alignment and tensor regularization (**w.o. A-A & T-R**); (3) FIMVC-TLRC without tensor regularization (**w.o. T-R**); (4) the complete FIMVC-TLRC (**FIMVC-TLRC**). Compared to **Fixed-Anchor**, our method achieves better clustering performance across all datasets, demonstrating the effectiveness of locally constructing learnable anchor graphs. In the NGs dataset, **w.o. T-R** improves ACC, NMI, and PUR compared to **w.o. A-A & T-R**, further validating the effectiveness of the structure alignment module. Additionally, in the Yale dataset, **FIMVC-TLRC** achieves superior performance compared to **w.o. T-R**, indicating that tensor regularization effectively captures consistent information across different views, thereby enhancing clustering performance. The ablation experiment demonstrates the

contribution and the necessity of each component.

**Convergence Analysis:** Figure 2 presents the convergence curves of ACC and the objective function on the We-bKB, and NGs datasets. The results indicate that both ACC and the objective loss converge within limited iterations.

**Parameter Analysis:** Our objective function involves four hyperparameters:  $\gamma$ ,  $\lambda$ ,  $\beta$ , and  $p$ . Figure 3 illustrates the ACC variations across four datasets when  $\gamma$  and  $\lambda$  take values in the set of  $\{0.0001, 0.01, 1, 100, 10000\}$ . From these results, we can observe that  $\gamma$  set to 0.01 yields better performance, while  $\lambda$  has minimal impact. Figure 4 shows the ACC variations for  $\beta$  and  $p$ . The clustering performance is optimal when  $\beta$  takes values between 0.0001 and 1, and the performance remains high when  $p$  is between 0.1 and 0.9.

## Conclusion

This work presents a novel tensorial federated incomplete multi-view clustering with local anchor alignment (FIMVC-TLRC). The proposed method constructs view-specific anchor graphs to capture complementary information and improve anchor correspondence across views through a structural alignment module. Additionally, it leverages a tensor Schatten p-norm regularizer to fuse representations and effectively exploits consistent information between various views. To support collaborative training under the distributed data settings, we developed a federated unified framework to jointly optimize the construction and alignment of anchor graphs. Experimental results on multiple datasets demonstrate the effectiveness of FIMVC-TLRC.

## Acknowledgments

This research is supported by the National Natural Science Foundation of China (No.62376226), Shaanxi's Key Research and Development Program (No.2024NC-ZDCYL-05-05), Xi'an Key Technology Research Projects for Key Agricultural Industry Chains (No. 2024JH-NYZD-0027) and the Yangling Demonstration Zone Science and Technology Plan Project under Grant (No.2024NY-14).

## References

- Apté, C.; Damerau, F.; and Weiss, S. M. 1994. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems (TOIS)*, 12(3): 233–251.
- Blum, A.; and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 92–100.
- Cai, X.; Wang, H.; Huang, H.; and Ding, C. 2012. Joint stage recognition and anatomical annotation of drosophila gene expression patterns. *Bioinformatics*, 28(12): i16–i24.
- Chen, X.; Xu, J.; Ren, Y.; Pu, X.; Zhu, C.; Zhu, X.; Hao, Z.; and He, L. 2023. Federated deep multi-view clustering with global self-supervision. In *Proceedings of the 31st ACM International Conference on Multimedia*, 3498–3506.
- Cui, C.; Ren, Y.; Pu, J.; Li, J.; Pu, X.; Wu, T.; Shi, Y.; and He, L. 2024. A novel approach for effective multi-view clustering with information-theoretic perspective. *Advances in Neural Information Processing Systems*, 36.
- Fang, U.; Li, M.; Li, J.; Gao, L.; Jia, T.; and Zhang, Y. 2023. A comprehensive survey on multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(12): 12350–12368.
- Fei-Fei, L.; Fergus, R.; and Perona, P. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, 178–178. IEEE.
- Feng, W.; Wu, Z.; Wang, Q.; Dong, B.; Tao, Z.; and Gao, Q. 2024. Federated multi-view clustering via tensor factorization. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 3962–3970.
- Fu, Y.; Li, Y.; Huang, Q.; Cui, J.; and Wen, J. 2024. Anchor graph network for incomplete multiview clustering. *IEEE Transactions on Neural Networks and Learning Systems*.
- Hu, M.; and Chen, S. 2019. Doubly aligned incomplete multi-view clustering. *arXiv preprint arXiv:1903.02785*.
- Huang, L.; Li, Z.; Sun, J.; and Zhao, H. 2022a. Coresets for Vertical Federated Learning: Regularized Linear Regression and  $K$ -Means Clustering. *Advances in Neural Information Processing Systems*, 35: 29566–29581.
- Huang, S.; Shi, W.; Xu, Z.; Tsang, I. W.; and Lv, J. 2022b. Efficient federated multi-view learning. *Pattern Recognition*, 131: 108817.
- Hussain, S. F.; Bisson, G.; and Grimal, C. 2010. An improved co-similarity measure for document clustering. In *2010 ninth international conference on machine learning and applications*, 190–197. IEEE.
- Jia, X.; Dong, X.; Chen, M.; and Yu, X. 2021. Missing data imputation for traffic congestion data based on joint matrix factorization. *Knowledge-Based Systems*, 225: 107114.
- Li, L.; Fan, Y.; Tse, M.; and Lin, K.-Y. 2020. A review of applications in federated learning. *Computers & Industrial Engineering*, 149: 106854.
- Li, Z.; Tang, C.; Zheng, X.; Liu, X.; Zhang, W.; and Zhu, E. 2022. High-order correlation preserved incomplete multi-view subspace clustering. *IEEE Transactions on Image Processing*, 31: 2067–2080.
- Liu, H.; Lou, T.; Zhang, Y.; Wu, Y.; Xiao, Y.; Jensen, C. S.; and Zhang, D. 2024. EEG-based multimodal emotion recognition: A machine learning perspective. *IEEE Transactions on Instrumentation and Measurement*, 73: 1–29.
- Meng, L.; Qi, Z.; Wu, L.; Du, X.; Li, Z.; Cui, L.; and Meng, X. 2024. Improving global generalization and local personalization for federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36.
- Qi, X.; Li, M.; Zhou, S.; Feng, W.; and Qi, Z. 2025a. Federated Learning for Science: A Survey on the Path to a Trustworthy Collaboration Ecosystem.
- Qi, Z.; Meng, L.; Li, Z.; Hu, H.; and Meng, X. 2025b. Cross-Silo Feature Space Alignment for Federated Learning on Clients with Imbalanced Data. In *The 39th Annual AAAI Conference on Artificial Intelligence (AAAI-25)*, 19986–19994.
- Ren, Y.; Chen, X.; Xu, J.; Pu, J.; Huang, Y.; Pu, X.; Zhu, C.; Zhu, X.; Hao, Z.; and He, L. 2024. A novel federated multi-view clustering method for unaligned and incomplete data fusion. *Information Fusion*, 108: 102357.
- Sejnowski, T.; and Gorman, R. 1988. Connectionist bench (sonar, mines vs. rocks). *UCI Machine Learning Repository*, 10: C5T01Q.
- Silberman, N.; Hoiem, D.; Kohli, P.; and Fergus, R. 2012. Indoor segmentation and support inference from rgb-d images. In *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, 746–760. Springer.
- Wan, X.; Liu, X.; Liu, J.; Wang, S.; Wen, Y.; Liang, W.; Zhu, E.; Liu, Z.; and Zhou, L. 2023. Auto-weighted multi-view clustering for large-scale data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 10078–10086.
- Wang, H.; Zong, L.; Liu, B.; Yang, Y.; and Zhou, W. 2019. Spectral perturbation meets incomplete multi-view data. *arXiv preprint arXiv:1906.00098*.
- Wang, S.; Liu, X.; Liu, L.; Tu, W.; Zhu, X.; Liu, J.; Zhou, S.; and Zhu, E. 2022. Highly-efficient incomplete large-scale multi-view clustering with consensus bipartite graph. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9776–9785.

- Wen, J.; Xu, Y.; and Liu, H. 2018. Incomplete multiview spectral clustering with adaptive graph learning. *IEEE transactions on cybernetics*, 50(4): 1418–1429.
- Wen, J.; Zhang, Z.; Fei, L.; Zhang, B.; Xu, Y.; Zhang, Z.; and Li, J. 2022. A survey on incomplete multiview clustering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(2): 1136–1149.
- Wen, Y.; Wang, S.; Liang, K.; Liang, W.; Wan, X.; Liu, X.; Liu, S.; Liu, J.; and Zhu, E. 2023. Scalable incomplete multi-view clustering with structure alignment. In *Proceedings of the 31st ACM international conference on multimedia*, 3031–3040.
- Wu, T.; Feng, S.; and Yuan, J. 2024. Low-Rank Kernel Tensor Learning for Incomplete Multi-View Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15952–15960.
- Xu, J.; Li, C.; Ren, Y.; Peng, L.; Mo, Y.; Shi, X.; and Zhu, X. 2022. Deep incomplete multi-view clustering via mining cluster complementarity. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 8761–8769.
- Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; and Gao, Y. 2021. A survey on federated learning. *Knowledge-Based Systems*, 216: 106775.
- Zhao, H.; Liu, H.; and Fu, Y. 2016. Incomplete multi-modal visual data grouping. In *IJCAI*, 2392–2398.