ROBUST FEDERATED MULTI-VIEW CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Multi-View Clustering (FMVC) has attracted widespread attention in the field of privacy-preserving data analysis. However, existing methods usually suffer from noise interference in practical distributed scenarios: the raw data collected by different clients generally contain noise, outliers, and local distortions, which lead to unstable local representations and clustering structures, resulting in degraded clustering performance. To address the problem, we propose a robust federated multi-view clustering method (RFMVC). For each client, instead of learning clustering structure directly on the original noisy data, our method learns a denoised feature space and learns clustering structure in it to preserve the key structural information of the original data while effectively suppressing the influence of noise and local perturbations on clustering results. For the server, it employs the tensor Schatten p-norm to impose a unified constraint on the cluster indicator matrices of all clients, and combines it with an adaptive client weighting mechanism to fuse multi-client information, thereby enhancing cross-client consistency while achieving denoising at the global structural level. Experimental results show that the proposed method outperforms methods on multiple real-world multi-view datasets, which verifies its robustness.

1 INTRODUCTION

Multi-view data can describe the same object from different feature spaces. Since different views usually characterize data from different perspectives, they can provide information with both consistency and complementarity. Compared with single-view learning methods Yang et al. (2018; 2019; 2020), multi-view learning can obtain more comprehensive and reliable data representations by jointly exploiting the shared and complementary information among different views, thereby achieving better learning performance Huang et al. (2019). Due to these advantages, multi-view clustering has attracted extensive attention in recent years as an important unsupervised tool in multi-view learning Chao et al. (2021). Its core objective is to jointly utilize the shared and complementary information among different views in an unsupervised manner to group samples into different clusters. Therefore, multi-view clustering Kumar et al. (2011); Gao et al. (2023); Li et al. (2015); Pei et al. (2022) has become an important research topic in pattern recognition and data mining.

Multi-view clustering methods have been widely studied and can be roughly categorized into three types. Among them, multi-view matrix factorization-based methods realize cross-view information fusion by learning shared latent representations Liu et al. (2013); multi-view k-means-based methods improve clustering performance by jointly optimizing clustering structures across different views Cai et al. (2013); Wan et al. (2023); multi-view spectral clustering-based methods characterize inter-sample relationships through graph structures and spectral embeddings, thereby enhancing the adaptability to complex data distributions Houthuys et al. (2018). Although these methods perform well in centralized scenarios, traditional multi-view clustering methods are difficult to directly apply to distributed scenarios where raw data cannot be centralized due to privacy, security, or data-sharing constraints. This limitation has also promoted the development of federated multi-view clustering.

Federated learning Yang & Sinaga (2025); Lei & Chen (2025) supports collaborative training among multiple parties without uploading raw data, and is therefore well suited to privacy-sensitive scenarios McMahan et al. (2017). In this context, federated multi-view clustering has gradually emerged as a new research direction. Huang et al. (2022) proposed FedMVL, which constructs a federated multi-view learning framework based on orthogonal matrix factorization. Subsequently, Chen et al.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(2023) proposed FedDMVC, which introduces global self-supervision to help clients learn view-specific clustering representations and handle incomplete multi-view data in distributed environments. On this basis, subsequent studies have further extended this research direction to integrated optimization and higher-order structural modeling. Feng et al. (2024a) proposed TensorFMVC, which is built upon k-means and introduces a third-order tensor to model the complementary information and spatial structure among different views. Although existing federated multi-view clustering methods have achieved some progress, most of them assume that the local data or intermediate representations provided by each client are of high quality. However, the raw data collected by different clients usually contain noise, outliers, and local distortions, which directly affect feature extraction, thereby reducing the reliability and stability of the clustering results Zhao et al. (2021). Therefore, how to learn more robust data representations within the federated multi-view clustering framework has become a key problem that remains to be solved urgently.

To address the above issue, we propose a robust federated multi-view clustering method (RFMVC), which aims to learn a cleaner data feature space for clustering, such that it effectively suppresses noise interference while preserving the intrinsic clustering structure of the original data. The overall framework is shown in Fig. 1. Specifically, RFMVC first introduces a robust representation learning mechanism to construct a clean feature space on each client, which effectively suppresses noise interference while preserving the key structural information of the original data. Then, we integrate denoised feature learning and local clustering structure learning into a unified objective function, so that the two can mutually reinforce each other during the optimization process. For the server, we use the tensor Schatten p-norm to jointly model the local cluster indicator matrices uploaded by different clients, so as to capture the higher-order correlations and complementarity among multi-client clustering results, and further introduce an adaptive client weighting mechanism to jointly constrain and fuse the information from different clients. The main contributions of this paper are summarized as follows:

- We propose RFMVC, a novel FMVC method, which learns a denoised feature space on each client and further learns clustering structures in this space, thereby effectively improving the robustness of the clustering process against noise and local perturbations.
- We develop a global consistency modeling strategy that integrates the tensor Schatten p-norm with adaptive client weighting, thereby effectively enhancing information fusion and collaborative modeling among different clients in federated scenarios.
- We conduct experiments on multiple datasets to verify the effectiveness and robustness of the proposed method in terms of clustering performance and stability.

2 Proposed Method

2.1 Problem Statement

In the federated multi-view clustering scenario, the data on different clients describe the same batch of objects from different perspectives, and thus the data on each client can be regarded as an independent view. Assume that there are M clients in the system. The m-th client has a local data matrix $\mathbf{X}^{(m)} \in \mathbb{R}^{N \times d_m}$, where N denotes the number of samples, d_m denotes the feature dimension of the m-th view, and the weight of each client is denoted by α_m.

1) Objective for Local Training: In practical distributed environments, the data collected by each client are usually affected by factors such as noise, outliers, and local distortions. To alleviate this issue, this paper introduces a denoised representation matrix $\mathbf{Z}^{(m)} \in \mathbb{R}^{N \times d_m}$ to characterize a data representation that is more suitable for clustering on the m-th client. Compared with the original data matrix $\mathbf{X}^{(m)}$, the denoised representation $\mathbf{Z}^{(m)}$ allows moderate adjustment of sample representations, thereby enhancing the robustness of the data to noise and abnormal perturbations while preserving the main structural information of the original data.

In this paper, the distance between any two denoised representations $\mathbf{z}_i^{(m)}$ and $\mathbf{z}_j^{(m)}$ on the m-th client is defined as $\mathbf{D}_{ij}^{(m)} = \left\| \mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)} \right\|_2^2$. Accordingly, we construct the distance matrix $\mathbf{D}^{(m)}$ and learn a discrete cluster indicator matrix $\mathbf{Y}^{(m)} \in \mathrm{Ind}$. To prevent the denoised representation

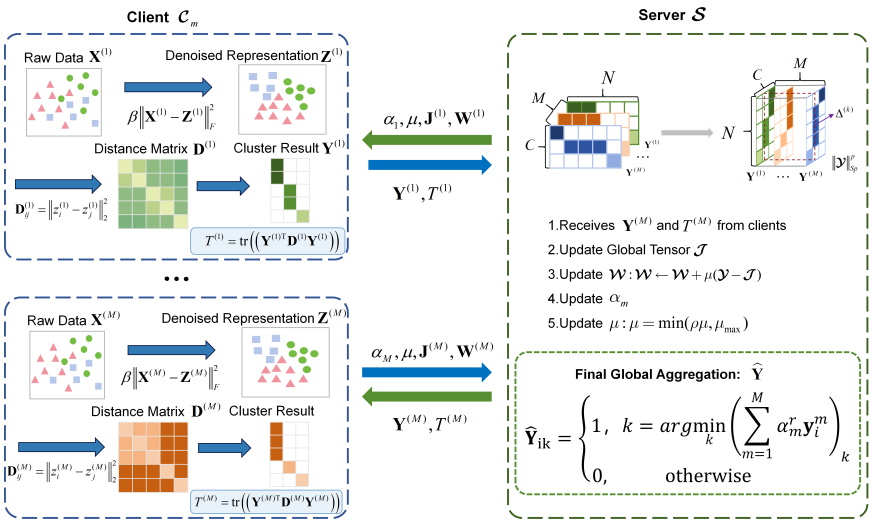


Figure 1: The overall framework of RFMVC. (1) Clients learn denoised representations and clustering structures, then transmit $\mathbf{Y}^{(m)}$ and $T^{(m)}$ to the server. (2) The server enforces global consistency across clients and aggregates local results to obtain the final clustering assignments.

from deviating excessively from the original data, we introduce the fidelity term $\|\mathbf{X}^{(m)}-\mathbf{Z}^{(m)}\|_F^2$. This term is used to measure the difference between the original data and the denoised representation. Therefore, the local objective function of the m-th client can be written as

$$\begin{aligned}\min_{\mathbf{Z}^{(m)},\mathbf{Y}^{(m)},\alpha}\quad & \alpha_m^r\,\mathrm{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right)+\beta\left\|\mathbf{X}^{(m)}-\mathbf{Z}^{(m)}\right\|_F^2\\ \text{s.t.}\quad & \mathbf{D}_{ij}^{(m)}=\left\|\mathbf{z}_i^{(m)}-\mathbf{z}_j^{(m)}\right\|_2^2,\ \mathbf{Y}^{(m)}\in\mathrm{Ind},\ \sum_{m=1}^{M}\alpha_m=1,\ \alpha_m\geq 0.\end{aligned}\tag{1}$$

where β is used to control the degree of denoising of the representation $\mathbf{Z}^{(m)}$. The first term is the local clustering loss term, which is used to measure the consistency between the cluster indicator matrix $\mathbf{Y}^{(m)}$ and the local distance structure $\mathbf{D}^{(m)}$ on the m-th client; the second term is the denoised representationa pproximation term, which is used to preserve the structural relationship between the denoised representation and the original data and improve the robustness of the local representation. Their joint optimization not only mitigates the adverse effects of noise on clustering but also uncovers a representation structure well suited to the clustering task, thereby enabling local clients to obtain more accurate cluster-indicator matrices.

2) Objective for Global Training: To exploit the complementary information and higher-order correlations among different clients, we introduce the tensor Schatten p-norm as a global consistency regularization term. The server stacks the cluster indicator matrices $\{\mathbf{Y}^{(m)}\}_{m=1}^{M}$ learned by the clients along the third dimension to construct $\boldsymbol{\mathcal{Y}}$, and introduces an auxiliary variable $\boldsymbol{\mathcal{J}}$ to represent the global consensus clustering structure. The server-side model is formulated as follows:

$$\min_{\boldsymbol{\mathcal{J}}}\quad \|\boldsymbol{\mathcal{J}}\|_{S_p}^p,\quad \text{s.t.}\quad \mathbf{J}^{(m)}=f_m\left(\mathbf{J}^{(m)},\mathbf{X}^{(m)}\right),\quad 1\leq m\leq M.\tag{2}$$

where $\boldsymbol{\mathcal{Y}}\in\mathbb{R}^{N\times C\times M}$ denotes the third-order tensor constructed from the cluster indicator matrices $\mathbf{Y}^{(m)}$ learned by M clients, $\boldsymbol{\mathcal{J}}$ denotes the global consensus clustering representation learned on the server, and $f_m(\cdot)$ denotes the local clustering model of the m-th client. Moreover, $\|\boldsymbol{\mathcal{J}}\|_{\mathbb{S}_p}^p$ denotes the tensor Schatten p-norm regularization term. By minimizing this regularization term, the proposed model captures the shared low-rank structure among the clustering results of different clients, thereby enhancing cross-client clustering consistency and suppressing noise interference.

2.2 FEDERATED OPTIMIZATION

The complete objective of RFMVC is formulated as

$$\begin{aligned}\min_{\mathbf{Z}^{(m)},\mathbf{Y}^{(m)},\boldsymbol{\alpha}} \quad & \sum_{m=1}^{M} \alpha_m^r \operatorname{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right) + \beta\left\|\mathbf{X}^{(m)} - \mathbf{Z}^{(m)}\right\|_F^2 + \lambda\left\|\boldsymbol{\mathcal{Y}}\right\|_{S_p}^p \\ \text{s.t.} \quad & \mathbf{D}_{ij}^{(m)} = \left\|\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}\right\|_2^2,\ \mathbf{Y}^{(m)} \in \text{Ind},\ \sum_{m=1}^{M}\alpha_m = 1,\ \alpha_m \geq 0.\end{aligned} \tag{3}$$

To optimize the objective in Eq. (3), we first employ the augmented Lagrange multiplier method to convert the original problem into an equivalent constrained optimization problem. By introducing the Lagrange multiplier and an auxiliary variable $\boldsymbol{\mathcal{J}}$ to substitute for $\boldsymbol{\mathcal{Y}}$, we obtain the following optimization problem:

$$\begin{aligned}\min_{\boldsymbol{\mathcal{J}},\mathbf{Y}^{(m)},\mathbf{Z}^{(m)},\boldsymbol{\alpha}} \quad & \sum_{m=1}^{M} \alpha_m^r \operatorname{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right) + \beta\left\|\mathbf{X}^{(m)} - \mathbf{Z}^{(m)}\right\|_F^2 + \lambda\left\|\boldsymbol{\mathcal{J}}\right\|_{S_p}^p \\ & + \frac{\mu}{2}\left\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{J}} + \frac{\boldsymbol{\mathcal{W}}}{\mu}\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{D}_{ij}^{(m)} = \left\|\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}\right\|_2^2,\ \mathbf{Y}^{(m)} \in \text{Ind},\ \sum_{m=1}^{M}\alpha_m = 1,\ \alpha_m \geq 0.\end{aligned} \tag{4}$$

where μ denotes the penalty parameter. Since data from different views are distributed across different clients, the optimization procedure is divided into client-side optimization and server-side optimization. On the client side, the local clustering structure is updated by alternately optimizing $\mathbf{Z}^{(m)}$ and $\mathbf{Y}^{(m)}$, while on the server side, the global clustering structure and client weights are updated by optimizing $\boldsymbol{\mathcal{J}}$ and α_m.

1) Local Training: By removing the terms in Eq. (4) that are irrelevant to $\mathbf{Z}^{(m)}$ and $\mathbf{Y}^{(m)}$, the optimization objective function for client $\boldsymbol{\mathcal{C}}_m$ can be obtained as

$$\begin{aligned}\min_{\mathbf{Z}^{(m)},\mathbf{Y}^{(m)}} \quad & \alpha_m^r \operatorname{tr}\Big((\mathbf{Y}^{(m)})^\top\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\Big) + \beta\left\|\mathbf{X}^{(m)} - \mathbf{Z}^{(m)}\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{Y}^{(m)} - \mathbf{J}^{(m)} + \frac{\mathbf{W}^{(m)}}{\mu}\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{D}_{ij}^{(m)} = \left\|\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}\right\|_2^2,\ \mathbf{Y}^{(m)} \in \text{Ind}.\end{aligned} \tag{5}$$

Step 1. **Update $\mathbf{Y}^{(m)}$:** The optimization problem of $\mathbf{Y}^{(m)}$ in Eq. (5) can be written as

$$\begin{aligned}\min_{\mathbf{Y}^{(m)}} \quad & \alpha_m^r \operatorname{tr}\Big((\mathbf{Y}^{(m)})^\top\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\Big) + \frac{\mu}{2}\left\|\mathbf{Y}^{(m)} - \mathbf{J}^{(m)} + \frac{\mathbf{W}^{(m)}}{\mu}\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{D}_{ij}^{(m)} = \left\|\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}\right\|_2^2,\ \mathbf{Y}^{(m)} \in \text{Ind}.\end{aligned} \tag{6}$$

Since $\mathbf{Y}^{(m)}$ is a discrete cluster indicator matrix, Eq. (6) is a discrete optimization problem. To solve this problem, we adopt a row-wise optimization strategy. Let $\mathbf{P}^{(m)} = \mathbf{J}^{(m)} - \frac{\mathbf{W}^{(m)}}{\mu}$. By optimizing $\mathbf{Y}^{(m)}$ row by row, the elements of its optimal solution can be computed by

$$\hat{\mathbf{Y}}_{ik}^{(m)} = \begin{cases} 1, & k = \arg\min\limits_k \left(2\alpha_m^r\left(\mathbf{Y}^{(m)}\right)^\top\left(\mathbf{d}_i^{(m)}\right)^\top - \mu\left(\mathbf{p}_i^{(m)}\right)^\top\right)_k, \\ 0, & \text{otherwise.}\end{cases} \tag{7}$$

If $\mathbf{Y}^{(m)}$ is updated directly according to Eq. (7), explicitly computing and storing the distance matrix $\mathbf{D}^{(m)}$ require $\mathcal{O}(N^2 d_m)$ time and $\mathcal{O}(N^2)$ space, respectively. To avoid explicitly constructing

$\mathbf{D}^{(m)}$, let $\mathbf{A}_k$ denote the set of samples in the k-th cluster. Then,

$$\begin{aligned}\left((\mathbf{Y}^{(m)})^\top(\mathbf{d}_i^{(m)})^\top\right)_k &= \sum_{j=1}^{N}\mathbf{D}_{ij}^{(m)}\mathbf{Y}_{jk}^{(m)} = \sum_{\mathbf{z}_j^{(m)}\in\mathbf{A}_k}\left\|\mathbf{z}_i^{(m)}-\mathbf{z}_j^{(m)}\right\|_2^2 \\ &= N_k\left\|\mathbf{z}_i^{(m)}\right\|_2^2 - 2(\mathbf{z}_i^{(m)})^\top\mathbf{s}_k^{(m)} + q_k^{(m)}.\end{aligned} \tag{8}$$

where $N_k = |\mathbf{A}_k|$, $\mathbf{s}_k^{(m)} = \sum_{\mathbf{z}_j^{(m)}\in\mathbf{A}_k}\mathbf{z}_j^{(m)}$, and $q_k^{(m)} = \sum_{\mathbf{z}_j^{(m)}\in\mathbf{A}_k}\|\mathbf{z}_j^{(m)}\|_2^2$. Therefore, only these three statistics need to be maintained for each cluster. When updating the label of the i-th sample, the total distances to different clusters are compared together with the guidance provided by $(\mathbf{p}_i^{(m)})^\top$, and the cluster minimizing the objective function is selected. Using these cluster statistics reduces the per-iteration time complexity to $\mathcal{O}(NCd_m)$ and the space complexity to $\mathcal{O}(NC + Cd_m)$.

Step 2. **Update $\mathbf{Z}^{(m)}$:** By retaining only the terms related to $\mathbf{Z}^{(m)}$ in the optimization objective Eq. (5) we can obtain:

$$\begin{aligned}\min_{\mathbf{Z}^{(m)}} \quad & \alpha_m^r \operatorname{tr}\left((\mathbf{Y}^{(m)})^\top\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right) + \beta\left\|\mathbf{X}^{(m)}-\mathbf{Z}^{(m)}\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{D}_{ij}^{(m)} = \left\|\mathbf{z}_i^{(m)}-\mathbf{z}_j^{(m)}\right\|_2^2\end{aligned} \tag{9}$$

Optimizing the rows of $\mathbf{Z}^{(m)}$ alternately and setting the derivative with respect to $\mathbf{z}_i^{(m)}$ to zero yields

$$\mathbf{z}_i^{(m)} = \frac{2\alpha_m^r\sum_{\mathbf{z}_j^{(m)}\in\mathbf{A}_{k_i},\, j\neq i}\mathbf{z}_j^{(m)} + \beta\mathbf{x}_i^{(m)}}{2\alpha_m^r\left(|\mathbf{A}_{k_i}|-1\right)+\beta} \tag{10}$$

*2) Global Training:****Step 1.*** **Update $\mathcal{J}$:** To update $\mathcal{J}$, we fix all other variables and obtain the following subproblem:

$$\min_{\mathcal{J}}\lambda\|\mathcal{J}\|_{S_p}^p + \frac{\mu}{2}\left\|\mathcal{Y}-\mathcal{J}+\frac{\mathcal{W}}{\mu}\right\|_F^2. \tag{11}$$

Accordingly, the solution is

$$\mathcal{J}^* = \mathcal{U}*\operatorname{ifft}\left(\Gamma_{\frac{\lambda}{\mu}}(\widetilde{\mathcal{T}})\right)*\mathcal{V}^\top, \tag{12}$$

where $\mathcal{T} = \mathcal{Y} + \mathcal{W}/\mu$, $\widetilde{\mathcal{T}} = \operatorname{fft}(\mathcal{T}, [], 3)$, and $\mathcal{U}$ and $\mathcal{V}$ are obtained via t-SVD of $\mathcal{T}$, i.e., $\mathcal{T} = \mathcal{U}*\mathcal{M}*\mathcal{V}^\top$ Gao et al. (2020).

It can be seen that updating $\mathcal{J}$ requires $\mathcal{Y}$, while $\mathcal{Y}$ is constructed from all local indicator matrices. Therefore, before updating $\mathcal{J}$, all clients need to transmit their local indicator matrices $\mathbf{Y}^{(m)}$ to the server.

Step 2. **Update $\boldsymbol{\alpha}$:** When $\mathbf{Y}^{(m)}$ and $\mathcal{J}$ are fixed, Eq. (4) can be written as

$$\min_{\boldsymbol{\alpha}}\mathcal{L}(\alpha_1,\ldots,\alpha_M) = \min_{\boldsymbol{\alpha}}\sum_{m=1}^{M}\alpha_m^r\operatorname{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right). \tag{13}$$

To handle the constraints $\sum_{m=1}^M\alpha_m = 1$ and $\alpha_m \geq 0$, the augmented Lagrange multiplier method (ALM) is adopted. Based on the KKT conditions, the solution is given by $\frac{\partial\mathcal{L}}{\partial\alpha_m} = 0, \eta_m\alpha_m = 0, \eta_m \geq 0, 1-\sum_{m=1}^M\alpha_m = 0$. The solution to this equation is given by:

$$\alpha_m = \frac{\left(\operatorname{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right)\right)^{\frac{1}{1-r}}}{\sum_{m=1}^M\left(\operatorname{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right)\right)^{\frac{1}{1-r}}} \tag{14}$$

Algorithm 1 Robust Federated Multi-View Clustering

Input: $\{\mathbf{X}^{(m)}\}_{m=1}^{M}$, C; $\rho = 1.1$, $\mu = 10^{-4}$, $\mu_{\max} = 12 \times 10^{12}$, p, r, λ, β.
Output: Global indicator matrix $\hat{\mathbf{Y}}$.

```
Initialize Y^(m), Z^(m) = X^(m), α_m = 1/M, 𝒥 = 𝒴, 𝒲 = 0.
while not converged do
    for m = 1 to M do
        Receive J^(m), W^(m), μ, and α_m.
        Compute P^(m), N_k, s_k^(m), and q_k^(m).
        Update Y^(m) and Z^(m); compute T^(m).
        Send {Y^(m), T^(m)} to server 𝒮.
    end for
    Construct 𝒴; update 𝒥, 𝒲, α, and μ.
    Broadcast J^(m), W^(m), μ, and α_m.
end while
Aggregate {Y^(m)}_{m=1}^M to obtain Ŷ via Eq. (16).
return Ŷ.
```

Considering the sensitivity of the original local client data and the representation $\mathbf{Z}$, all clients transmit their computed

$$\begin{aligned} T^{(m)} &= \operatorname{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right) \\ &= \sum_{k=1}^{C}\sum_{\mathbf{z}_i^{(m)}\in\mathbf{A}_k}\left(|\mathbf{A}_k|\|\mathbf{z}_i^{(m)}\|_2^2 - 2\mathbf{z}_i^{(m)\top}\sum_{\mathbf{z}_j^{(m)}\in\mathbf{A}_k}\mathbf{z}_j^{(m)} + \sum_{\mathbf{z}_j^{(m)}\in\mathbf{A}_k}\|\mathbf{z}_j^{(m)}\|_2^2\right) \end{aligned} \tag{15}$$

To the server $\boldsymbol{\mathcal{S}}$. The server then updates the weights α_m corresponding to all clients according to Eq. (14). Subsequently, the server sends these weights back to the corresponding clients.

2.3 Federated Workflow

The federated workflow consists of the following three stages:

Stage 1: Initialization stage: All clients initialize their representations as $\mathbf{Z}^{(m)} = \mathbf{X}^{(m)}$ and initialize the indicator matrix $\mathbf{Y}^{(m)}$ such that every consecutive C rows form an identity matrix. Then, each client sends $\mathbf{Y}^{(m)}$ to the server $\boldsymbol{\mathcal{S}}$ to construct the third-order tensor $\boldsymbol{\mathcal{Y}}$. The server $\boldsymbol{\mathcal{S}}$ initializes $\alpha_m = \frac{1}{M}, \quad \boldsymbol{\mathcal{J}} = \boldsymbol{\mathcal{Y}}, \quad \boldsymbol{\mathcal{W}} = 0$.

Stage 2: Iterative optimization: At each communication round, each client $\boldsymbol{\mathcal{C}}_m$ updates $\mathbf{Y}^{(m)}$ according to Eq. (7) and $\mathbf{Z}^{(m)}$ according to Eq. (10), and then sends the non-sensitive indicator matrix $\mathbf{Y}^{(m)}$ and $T^{(m)}$ to the server $\boldsymbol{\mathcal{S}}$. After receiving the information from all clients, $\boldsymbol{\mathcal{S}}$ stacks all $\mathbf{Y}^{(m)}$ to construct $\boldsymbol{\mathcal{Y}}$, updates $\boldsymbol{\mathcal{J}}$ according to Eq. (12), updates $\boldsymbol{\mathcal{W}}$ by $\boldsymbol{\mathcal{W}} \leftarrow \boldsymbol{\mathcal{W}} + \mu(\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{J}})$, and updates α_m. Finally, $\boldsymbol{\mathcal{S}}$ sends $\mathbf{J}^{(m)}$, $\mathbf{W}^{(m)}$, and α_m back to $\boldsymbol{\mathcal{C}}_m$.

Stage 3: Final aggregation: Repeat Stage 2 until convergence. Then, $\boldsymbol{\mathcal{C}}_m$ sends the final local indicator matrix $\mathbf{Y}^{(m)}$ to $\boldsymbol{\mathcal{S}}$. The final clustering assignment matrix is computed on the server as

$$\hat{\mathbf{Y}}_{ik} = \begin{cases} 1, & k = \arg\max_k \left(\sum_{m=1}^{M} \alpha_m^r \mathbf{y}_i^{(m)}\right)_k, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Algorithm 1 summarizes the complete procedure.

2.4 Complexity Analysis

1) Computational Complexity Analysis: Assuming $C \ll N$ and $M \ll N$, we analyze their computational complexities separately. For $\boldsymbol{\mathcal{C}}_m$, updating $\mathbf{Y}^{(m)}$ using cluster statistics requires $\mathcal{O}(NCd_m)$

Table 1: Clustering performance (%) of different methods on six multi-view datasets.

Dataset	Metric	CMKOA	AWMVC	OMVCDR	FastMICE	CRLSNMF	Orth-NMF	FMVC-IMK	SFOMVC-TR	RFMVC
BDGP	ACC	48.92	50.16	48.32	49.86	53.84	50.72	62.72	40.60	**92.80**
	NMI	22.76	33.55	33.19	42.14	30.44	19.13	32.85	11.20	**81.65**
	PUR	48.92	50.16	51.68	49.86	54.97	50.72	62.72	40.60	**92.80**
20NGs	ACC	56.81	50.40	33.20	51.44	56.00	54.60	56.20	97.20	**100.00**
	NMI	30.42	36.69	11.75	33.23	35.56	35.60	30.98	92.35	**100.00**
	PUR	56.81	50.60	34.20	63.20	57.48	56.00	56.20	97.20	**100.00**
MFEAT	ACC	78.75	86.95	55.75	89.42	80.84	89.35	31.75	92.95	**97.05**
	NMI	75.30	78.00	56.81	87.19	69.49	86.30	28.29	91.17	**94.51**
	PUR	78.75	86.95	59.75	91.44	80.84	89.35	32.90	92.95	**97.05**
ORL	ACC	100.00	80.00	65.00	75.85	70.85	73.00	94.00	94.75	**100.00**
	NMI	100.00	89.80	80.05	87.77	83.51	86.16	94.85	97.49	**100.00**
	PUR	100.00	81.50	66.75	80.65	75.30	76.00	94.00	95.75	**100.00**
MSRC	ACC	100.00	84.29	66.19	86.48	70.00	97.14	85.72	100.00	**100.00**
	NMI	100.00	72.70	57.93	77.53	59.43	93.87	53.52	100.00	**100.00**
	PUR	100.00	84.29	67.14	86.48	71.90	97.14	85.72	100.00	**100.00**
NUSWIDE	ACC	56.94	13.88	41.50	51.02	32.81	54.18	39.22	58.90	**84.66**
	NMI	42.43	15.48	14.27	21.14	7.78	34.35	41.94	43.36	**65.60**
	PUR	56.94	51.42	41.66	51.10	34.56	54.18	13.80	58.90	**84.66**

time per iteration. By maintaining the feature sum $\mathbf{s}_{k_i}^{(m)}$, we have $\sum_{\mathbf{z}_j^{(m)} \in \mathbf{A}_{k_i}, j \neq i} \mathbf{z}_j^{(m)} = \mathbf{s}_{k_i}^{(m)} - \mathbf{z}_i^{(m)}$. Thus, updating one sample and all $\mathbf{Z}^{(m)}$ requires $\mathcal{O}(d_m)$ and $\mathcal{O}(Nd_m)$, respectively. The total computational complexity on $\boldsymbol{\mathcal{C}}_m$ over t_1 iterations is $\mathcal{O}(t_1 NCd_m)$. For $\boldsymbol{\mathcal{S}}$, updating the third-order tensor $\boldsymbol{\mathcal{J}}$ requires $\mathcal{O}(2MNC\log(MC) + M^2CN)$. Therefore, the total computational complexity over t_1 communication rounds is $\mathcal{O}(t_1(2MNC\log(MC) + M^2CN))$.

2) Communication Complexity Analysis: After local updating, $\boldsymbol{\mathcal{C}}_m$ sends $T^{(m)}$ and $\mathbf{Y}^{(m)}$ to $\boldsymbol{\mathcal{S}}$, resulting in $\mathcal{O}(NC)$ communication complexity per round. After global updating, $\boldsymbol{\mathcal{S}}$ sends $\mathbf{J}^{(m)}$, $\mathbf{W}^{(m)}$, μ, and α_m to $\boldsymbol{\mathcal{C}}_m$, which also requires $\mathcal{O}(NC)$ communication per round.

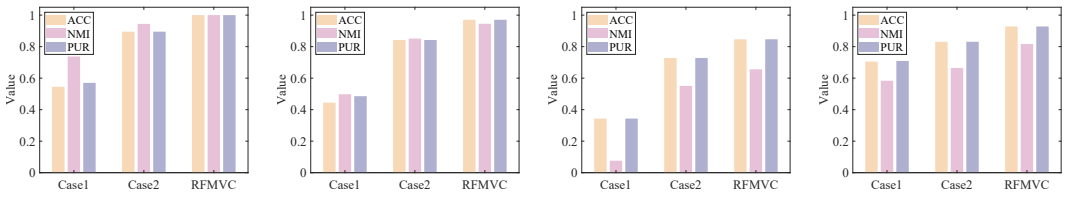
3 Experiment

3.1 Experimental Settings

We implement all the methods with MATLAB 2023b on a Desktop running Windows 11 equipped with an Intel Core i513400 CPU (2.50GHz) and 32GB DDR4 RAM. We compare our method with eight methods on six datasets. We adopt the federated setting described in the problem formulation, where each client holds one view of the same set of samples.

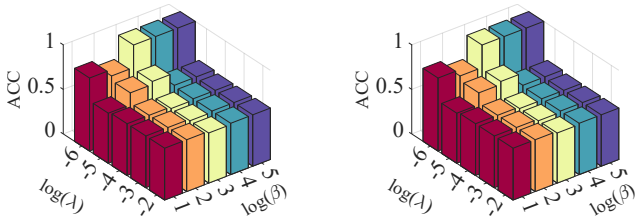
1) Datasets: We evaluate all methods on six real-world multi-view datasets. Except for 20NGs, all the other datasets are class-balanced, while 20NGs is a class-imbalanced dataset: **BDGP** Cai et al. (2012), **20NGs** Lang (1995), **MFEAT** van Breukelen et al. (1998), **ORL** Samaria & Harter (1994), **MSRC** Liu et al. (2010), **NUSWIDE** Chua et al. (2009).

2) Comparison Methods: We compare RFMVC with eight representative clustering methods. The first six methods are centralized multi-view clustering approaches, whereas FMVC-IMK and SFOMVC-TR are federated multi-view clustering methods: **CMKOA** Lu et al. (2023), **AWMVC** Wan et al. (2023), **OMVCDR** Wan et al. (2024), **FastMICE** Huang et al. (2023), **CRL-SNMF** Dong et al. (2024), **Orth-NMF** Li et al. (2023), **FMVC-IMK** Feng et al. (2024b), and **SFOMVC-TR** Feng et al. (2025).



(a) ORL (b) MFEAT (c) NUSWIDE (d) BDGP

Figure 2: Ablation study on four multi-view datasets.



(a) MSRC-ACC (b) MFEAT-ACC

Figure 3: ACC of RFMVC on MSRC and MFEAT under different values of β and λ.

3.2 Clustering Performance

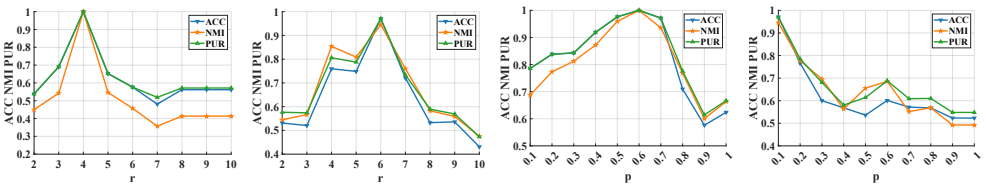
Table 1 reports the clustering results of RFMVC and the comparison methods on six datasets. Overall, RFMVC outperforms the comparison methods in terms of ACC, NMI, and PUR on all six datasets. This demonstrates that denoised representation learning can preserve the original structural information while reducing the adverse effects of noise, local perturbations, and abnormal structures, thereby improving the stability and reliability of clustering. The improvements also indicate that RFMVC can more effectively fuse information from different clients and learn a consistent clustering structure from multi-client representations. RFMVC shows good adaptability to different types of datasets, with particularly significant improvements on datasets such as NUSWIDE and BDGP. For example, on BDGP, RFMVC achieves 92.80%, 81.65%, and 92.80% in ACC, NMI, and PUR, respectively, substantially outperforming FMVC-IMK, which obtains 62.72%, 32.85%, and 62.72%. These results indicate that RFMVC can effectively calibrate and constrain client-side representations without transmitting the original data, enabling the clustering process to focus on stable and effective structural information and thus producing more accurate and robust clustering results.

3.3 Ablation Study

To verify the effectiveness of the denoised representation term and the global consistency regularization term in the model, we designed ablation experiments on four datasets, and the results are shown in Fig. 2. Specifically, Case 1 removes both terms, while Case 2 removes only the denoised representation term. The performance of Case 1 drops significantly on all datasets, indicating that both components play an important role in improving the clustering performance of the model. Case 2 improves performance on some datasets by exploiting cross-client correlations and complementary information, but remains inferior to the full RFMVC model due to noise and outliers.

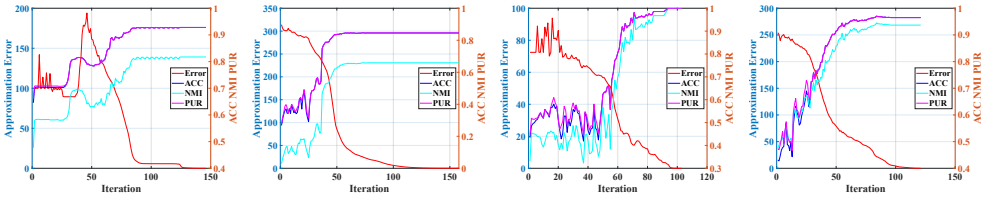
3.4 Parameter Analysis

1) Parameters β and λ: We investigate the effects of β and λ on MSRC and MFEAT. As shown in Fig. 3: (1) As β increases, the clustering performance of the model gradually improves. However, an excessively small β may cause representation distortion and the loss of useful information, whereas an excessively large β limits the corrective capability of the denoised representation; (2) A larger λ enhances the sparsity and robustness of the model and suppresses noise interference, but an excessively large value may cause the loss of discriminative information. A smaller λ weakens the



(a) MSRC-r (b) MFEAT-r (c) MSRC-p (d) MFEAT-p

Figure 4: Parameter sensitivity of RFMVC with respect to r and p on the MSRC and MFEAT datasets.



(a) BDGP (b) NUSWIDE (c) MSRC (d) MFEAT

Figure 5: Convergence curves of RFMVC on the BDGP, NUSWIDE, MSRC, and MFEAT datasets.

regularization effect and makes the model more prone to overfitting noise. Therefore, appropriate values of β and λ should be selected to achieve the best clustering performance.

2) Parameters r and p:

As shown in Fig. 4, both r and p significantly affect clustering performance. (1) Parameter r controls the sensitivity of client weights α_m^r. A small r yields insufficient weight differentiation, whereas a large r causes extreme weight allocation, weakening client complementarity and reducing performance. (2) Parameter p controls the Schatten p-norm regularization of the latent structure. Its optimal value varies across datasets; thus, an appropriate p is essential for stable clustering performance.

3.5 Convergence Analysis

To verify the convergence of the proposed model, we record the error between $\mathbf{J}^{(m)}$ and $\mathbf{Y}^{(m)}$ in each iteration on the BDGP and NUSWIDE datasets, together with the variation of clustering performance metrics, where $\text{Error} = \left\|\mathbf{J}^{(m)} - \mathbf{Y}^{(m)}\right\|_F^2$. As shown in Fig. 5, the error exhibits a clear downward trend and gradually approaches 0, while ACC, NMI, and PUR gradually improve and become stable. This indicates that the model can progressively reduce the discrepancy between $\mathbf{J}^{(m)}$ and $\mathbf{Y}^{(m)}$ and obtain stable clustering results within a limited number of iterations, verifying the effectiveness and stability of the proposed model.

4 Conclusion

To address noise interference and anomalous perturbations in federated multi-view clustering, this paper proposes a robust federated multi-view clustering method, termed RFMVC. By learning denoised representations, RFMVC suppresses noise while preserving the underlying clustering structure. In addition, a global consistency regularization term is introduced to exploit higher-order correlations and complementary information among clients. Experiments on six public datasets demonstrate that RFMVC achieves the best or tied-best performance on the evaluated datasets. Ablation studies, parameter-sensitivity analyses, and convergence experiments further support the effectiveness and stability of the proposed components. Overall, RFMVC effectively improves the robustness and reliability of federated multi-view clustering in complex data environments.

LLM USAGE STATEMENT

During the preparation of this manuscript, we employed Large Language Models (LLMs) for language assistance, primarily to refine academic expression, improve sentence structure, and check grammar, thereby enhancing the accuracy, clarity, and overall readability of the manuscript.

All content processed with LLM assistance was carefully reviewed and revised by the authors. The authors take full responsibility for the research content, written presentation, and final manuscript, ensuring compliance with academic ethics and research integrity standards.

REFERENCES

Xiao Cai, Hua Wang, Heng Huang, and Chris Ding. Joint stage recognition and anatomical annotation of drosophila gene expression patterns. *Bioinformatics*, 28(12):i16–i24, 2012.

Xiao Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on big data. In *Ijcai*, volume 13, pp. 2598–2604, 2013.

Guoqing Chao, Shiliang Sun, and Jinbo Bi. A survey on multiview clustering. *IEEE Transactions on Artificial Intelligence*, 2(2):146–168, 2021.

Xinyue Chen, Jie Xu, Yazhou Ren, Xiaorong Pu, Ce Zhu, Xiaofeng Zhu, Zhifeng Hao, and Lifang He. Federated deep multi-view clustering with global self-supervision. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 3498–3506, 2023.

Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, pp. 1–9, 2009.

Yuzhu Dong, Hangjun Che, Man-Fai Leung, Cheng Liu, and Zheng Yan. Centric graph regularized log-norm sparse non-negative matrix factorization for multi-view clustering. *Signal Processing*, 217:109341, 2024.

Wei Feng, Zhenwei Wu, Qianqian Wang, Bo Dong, Zhiqiang Tao, and Quanxue Gao. Federated multi-view clustering via tensor factorization. In *IJCAI*, pp. 3962–3970, 2024a.

Wei Feng, Zhenwei Wu, Qianqian Wang, Bo Dong, Zhiqiang Tao, and Quanxue Gao. Efficient Federated Multi-View Clustering with Integrated Matrix Factorization and K-Means. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 3971–3979. International Joint Conferences on Artificial Intelligence Organization, 2024b.

Wei Feng, Danting Liu, Qianqian Wang, Wenqi Liang, and Zheng Yan. Scalable federated one-step multi-view clustering with tensorized regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16586–16594, 2025.

Quanxue Gao, Pu Zhang, Wei Xia, Deyan Xie, Xinbo Gao, and Dacheng Tao. Enhanced tensor rpca and its application. *IEEE transactions on pattern analysis and machine intelligence*, 43(6): 2133–2140, 2020.

Quanxue Gao, Qianqian Wang, Han Lu, Wei Xia, and Xinbo Gao. Rethinking k-means from manifold learning perspective. *arXiv preprint arXiv:2305.07213*, 2023.

Lynn Houthuys, Rocco Langone, and Johan A. K. Suykens. Multi-view kernel spectral clustering. *Information Fusion*, 44:46–56, 2018.

Dong Huang, Chang-Dong Wang, and Jian-Huang Lai. Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11388–11402, 2023.

Shudong Huang, Zhao Kang, Ivor W Tsang, and Zenglin Xu. Auto-weighted multi-view clustering via kernelized graph learning. *Pattern Recognition*, 88:174–184, 2019.

Shudong Huang, Wei Shi, Zenglin Xu, Ivor W Tsang, and Jiancheng Lv. Efficient federated multi-view learning. *Pattern Recognition*, 131:108817, 2022.

Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. *Advances in neural information processing systems*, 24, 2011.

Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 331–339, 1995.

Yifan Lei and Xiaohong Chen. Decentralized federated multi-view sparse subspace clustering: Y. lei and x. chen. *Annals of Mathematics and Artificial Intelligence*, pp. 1–25, 2025.

Jing Li, Quanxue Gao, QIANQIAN WANG, Ming Yang, and Wei Xia. Orthogonal non-negative tensor factorization based multi-view clustering. In *Advances in Neural Information Processing Systems*, volume 36, pp. 18186–18202, 2023.

Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. Large-scale multi-view spectral clustering via bipartite graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM international conference on data mining*, pp. 252–260. SIAM, 2013.

Wei Liu, Junfeng He, and Shih-Fu Chang. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th international conference on machine learning*, pp. 679–686. Citeseer, 2010.

Han Lu, Quanxue Gao, Qianqian Wang, Ming Yang, and Wei Xia. Centerless multi-view k-means based on the adjacency matrix. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8949–8956, 2023.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. Pmlr, 2017.

Feiping Nie, Heng Huang, and Chris Ding. Low-rank matrix recovery via efficient schatten p-norm minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pp. 655–661, 2021.

Shenfei Pei, Huimin Chen, Feiping Nie, Rong Wang, and Xuelong Li. Centerless clustering. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):167–181, 2022.

Ferdinando S Samaria and Andy C Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE workshop on applications of computer vision*, pp. 138–142. IEEE, 1994.

M. van Breukelen, R. P. W. Duin, D. M. J. Tax, and J. E. den Hartog. Handwritten digit recognition by combined classifiers. *Kybernetika*, 34(4):381–386, 1998.

Xinhang Wan, Xinwang Liu, Jiyuan Liu, Siwei Wang, Yi Wen, Weixuan Liang, En Zhu, Zhe Liu, and Lu Zhou. Auto-weighted multi-view clustering for large-scale data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10078–10086, 2023.

Xinhang Wan, Jiyuan Liu, Xinbiao Gan, Xinwang Liu, Siwei Wang, Yi Wen, Tianjiao Wan, and En Zhu. One-step multi-view clustering with diverse representation. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

Miin-Shen Yang and Kristina P. Sinaga. Federated multi-view k-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2446–2459, 2025.

X. Yang, C. Deng, X. Liu, and F. Nie. New $l_{2,1}$-norm relaxation of multi-way graph cut for clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Xu Yang, Cheng Deng, Kun Wei, Junchi Yan, and Wei Liu. Adversarial learning for robust deep clustering. *Advances in Neural Information Processing Systems*, 33:9098–9108, 2020.

Xiaowei Zhao, Feiping Nie, Rong Wang, and Xuelong Li. Robust fuzzy k-means clustering with shrunk patterns learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(3):3001–3013, 2021.

A Description of Datasets

Real-world multi-view datasets are adopted in the experiments, as shown in Table 2. Except for 20NGs, all the other datasets are class-balanced, while 20NGs is a class-imbalanced dataset.

Table 2: Details of the datasets.

Dataset	Views	Samples	Clusters	Dimensions
BDGP	2	2500	5	1750/79
20NGs	3	500	5	2000/2000/2000
MFEAT	6	2000	10	216/76/64/6/240/47
ORL	3	400	40	4096/3304/6750
MSRC	5	210	7	24/576/512/256/254
NUSWIDE	5	5000	5	65/226/145/74/129

B Tensor Schatten p-Norm

The tensor Schatten p-norm is defined as follows.

Definition 1 *The tensor Schatten p-norm for $\mathcal{M}$ is defined as*

$$\|\mathcal{M}\|_{S_p} = \left(\sum_{i=1}^{\min(n_1,n_2)} \sum_{k=1}^{n_3} \left(\overline{\mathcal{M}}^k(i) \right)^p \right)^{\frac{1}{p}} \tag{17}$$

$\overline{\mathcal{M}}^k(i)$ is the i-th singular value of matrix $\overline{\mathcal{M}}^k$ and $\overline{\mathcal{M}} = \mathrm{fft}(\mathcal{M}, [], 3)$. With $0 < p < 1$, the tensor Schatten p-norm is better rank approximation than other norms Lu et al. (2023); Nie et al. (2021).

The k-th frontal slice $\Delta^{(k)}$ is composed of $\mathbf{Y}^1_{:,k}, \mathbf{Y}^2_{:,k}, \ldots, \mathbf{Y}^M_{:,k}$, which describe the relationship between all samples and the k-th sample in all views. Since the relationship between all N samples and the k-th cluster across views should be consistent, minimizing the Schatten p-norm achieves a spatial low-rank structure for $\Delta^{(k)}$, thereby helping to fully explore the complementary information.

C Detailed Optimization of RFMVC

C.1 Update of $\mathbf{Y}^{(m)}$

The optimization problem of $\mathbf{Y}^{(m)}$ corresponding to the local subproblem in the main paper can be written as

$$\begin{aligned} \min_{\mathbf{Y}^{(m)}} \quad & \alpha_m^r \operatorname{tr}\Big((\mathbf{Y}^{(m)})^\top \mathbf{D}^{(m)} \mathbf{Y}^{(m)} \Big) + \frac{\mu}{2} \left\| \mathbf{Y}^{(m)} - \mathbf{J}^{(m)} + \frac{\mathbf{W}^{(m)}}{\mu} \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{D}^{(m)}_{ij} = \left\| \mathbf{z}^{(m)}_i - \mathbf{z}^{(m)}_j \right\|_2^2, \quad \mathbf{Y}^{(m)} \in \mathrm{Ind}. \end{aligned} \tag{18}$$

Since $\mathbf{Y}^{(m)}$ is a discrete cluster indicator matrix, Eq. (18) is a discrete optimization problem. To solve this problem, a row-wise optimization strategy can be adopted. Let $\mathbf{P}^{(m)} = \mathbf{J}^{(m)} - \frac{\mathbf{W}^{(m)}}{\mu}$.

Then Eq. (18) can be rewritten as

$$\min_{\mathbf{y}_i^{(m)}} \quad \mathbf{y}_i^{(m)} \left(2\alpha_m^r (\mathbf{Y}^{(m)})^\top (\mathbf{d}_i^{(m)})^\top - \mu (\mathbf{p}_i^{(m)})^\top\right). \tag{19}$$

The row-wise form shows that the update of $\mathbf{Y}^{(m)}$ depends on $(\mathbf{Y}^{(m)})^\top (\mathbf{d}_i^{(m)})^\top$. We next derive this term without explicitly constructing the full distance matrix $\mathbf{D}^{(m)}$. According to Eq. (19), we have

$$(\mathbf{Y}^{(m)})^\top (\mathbf{d}_i^{(m)})^\top = \sum_{j=1}^{N} \mathbf{D}_{ij}^{(m)} (\mathbf{y}_j^{(m)})^\top.$$

Therefore, the k-th element can be written as

$$\left((\mathbf{Y}^{(m)})^\top (\mathbf{d}_i^{(m)})^\top\right)_k = \sum_{j=1}^{N} \mathbf{D}_{ij}^{(m)} \mathbf{Y}_{jk}^{(m)} = \sum_{\mathbf{z}_j^{(m)} \in \mathbf{A}_k} \mathbf{D}_{ij}^{(m)} = \sum_{\mathbf{z}_j^{(m)} \in \mathbf{A}_k} \|\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}\|_2^2. \tag{20}$$

where $\mathbf{A}_k$ denotes the set of samples corresponding to the k-th cluster.

To avoid explicitly computing the full distance matrix $\mathbf{D}^{(m)}$, Eq. (20) is further expanded as

$$\begin{aligned} \sum_{\mathbf{z}_j^{(m)} \in \mathbf{A}_k} \|\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}\|_2^2 &= \sum_{\mathbf{z}_j^{(m)} \in \mathbf{A}_k} \left(\|\mathbf{z}_i^{(m)}\|_2^2 - 2(\mathbf{z}_i^{(m)})^\top \mathbf{z}_j^{(m)} + \|\mathbf{z}_j^{(m)}\|_2^2\right) \\ &= |\mathbf{A}_k| \|\mathbf{z}_i^{(m)}\|_2^2 - 2(\mathbf{z}_i^{(m)})^\top \sum_{\mathbf{z}_j^{(m)} \in \mathbf{A}_k} \mathbf{z}_j^{(m)} + \sum_{\mathbf{z}_j^{(m)} \in \mathbf{A}_k} \|\mathbf{z}_j^{(m)}\|_2^2. \end{aligned} \tag{21}$$

Therefore, the sum of distances can be computed using three cluster statistics: $N_k = |\mathbf{A}_k|, \quad \mathbf{s}_k^{(m)} = \sum_{\mathbf{z}_j^{(m)} \in \mathbf{A}_k} \mathbf{z}_j^{(m)}, \quad q_k^{(m)} = \sum_{\mathbf{z}_j^{(m)} \in \mathbf{A}_k} \left\|\mathbf{z}_j^{(m)}\right\|_2^2$. Substituting this result into the row-wise objective gives the update rule of $\mathbf{Y}^{(m)}$ in the main paper.

C.2 Update of $\mathbf{Z}^{(m)}$

By retaining only the terms related to $\mathbf{Z}^{(m)}$ in the local optimization objective, we obtain:

$$\begin{aligned} \min_{\mathbf{Z}^{(m)}} \quad & \alpha_m^r \operatorname{tr}\left((\mathbf{Y}^{(m)})^\top \mathbf{D}^{(m)} \mathbf{Y}^{(m)}\right) + \beta \left\|\mathbf{X}^{(m)} - \mathbf{Z}^{(m)}\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{D}_{ij}^{(m)} = \left\|\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}\right\|_2^2 \end{aligned} \tag{22}$$

Considering that the rows of $\mathbf{Z}^{(m)}$ are mutually coupled, direct optimization is difficult. Therefore, we optimize it row by row. The subproblem in Eq. (22) is transformed into the following row-wise form:

$$\min_{\mathbf{z}_i^{(m)}} \quad 2\alpha_m^r \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\|\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}\right\|_2^2 \left\langle \mathbf{y}_i^{(m)}, \mathbf{y}_j^{(m)\top} \right\rangle + \beta \left\|\mathbf{x}_i^{(m)} - \mathbf{z}_i^{(m)}\right\|_2^2. \tag{23}$$

Taking the derivative of Eq. (23) with respect to $\mathbf{z}_i^{(m)}$ and setting it to zero, we obtain

$$4\alpha_m^r \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\langle \mathbf{y}_i^{(m)}, \mathbf{y}_j^{(m)\top} \right\rangle \left(\mathbf{z}_i^{(m)} - \mathbf{z}_j^{(m)}\right) + 2\beta \left(\mathbf{z}_i^{(m)} - \mathbf{x}_i^{(m)}\right) = 0. \tag{24}$$

Therefore, the final update formula of $\mathbf{z}_i^{(m)}$ can be written as

$$\mathbf{z}_i^{(m)} = \frac{2\alpha_m^r \sum_{\mathbf{z}_j^{(m)} \in \mathbf{A}_{k_i}, j \neq i} \mathbf{z}_j^{(m)} + \beta \mathbf{x}_i^{(m)}}{2\alpha_m^r \left(|\mathbf{A}_{k_i}| - 1\right) + \beta} \tag{25}$$

C.3 UPDATE OF $\mathcal{J}$

We first introduce the following theorem:

Theorem 1 *For two third-order tensors $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{C} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, let $\mathcal{U}$ and $\mathcal{V}$ be obtained from the t-SVD of $\mathcal{C}$, i.e., $\mathcal{C} = \mathcal{U} * \mathcal{M} * \mathcal{V}^\top$. Then, the optimal solution to the function*

$$\min_{\mathcal{B}} \ \tau \|\mathcal{B}\|_{S_p}^p + \frac{1}{2}\|\mathcal{B} - \mathcal{C}\|_F^2 \tag{26}$$

is given by

$$\mathcal{B}^* = \mathcal{U} * \text{ifft}\Big(\Gamma_\tau(\tilde{\mathcal{C}})\Big) * \mathcal{V}^\top \tag{27}$$

where $\tilde{\mathcal{C}} = \text{fft}(\mathcal{C}, [], 3)$, and $\Gamma_\tau(\tilde{\mathcal{C}}) \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a third-order diagonal tensor whose diagonal elements can be computed by the generalized soft-thresholding (GST) algorithm proposed in Gao et al. (2020).

By Theorem 1, the solution is given by

$$\mathcal{J}^* = \mathcal{U} * \text{ifft}\Big(\Gamma_{\frac{\lambda}{\mu}}(\tilde{\mathcal{T}})\Big) * \mathcal{V}^\top \tag{28}$$

where

$$\mathcal{T} = \mathcal{Y} + \frac{\mathcal{W}}{\mu} = \mathcal{U} * \mathcal{M} * \mathcal{V}^\top \tag{29}$$

C.4 UPDATE OF $\boldsymbol{\alpha}$

When $\mathbf{Y}^{(m)}$ and $\mathcal{J}$ are fixed, the subproblem w.r.t. $\boldsymbol{\alpha}$ can be written as

$$\min_{\boldsymbol{\alpha}} \mathcal{L}(\alpha_1, \alpha_2, \cdots, \alpha_M) = \min_{\boldsymbol{\alpha}} \sum_{m=1}^{M} \alpha_m^r \ \text{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right) \tag{30}$$

To handle the constraints $\sum_{m=1}^M \alpha_m = 1$ and $\alpha_m \geq 0$, the augmented Lagrange multiplier method (ALM) is adopted and incorporated into the original optimization problem Eq. (30), yielding

$$\begin{aligned} &\min \mathcal{L}(\alpha_1, \ldots, \alpha_M, \eta, \eta_1, \ldots, \eta_M) \\ &= \min \sum_{m=1}^{M} \alpha_m^r \ \text{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right) + \eta\left(1 - \sum_{m=1}^{M} \alpha_m\right) - \sum_{m=1}^{M} \eta_m \alpha_m \end{aligned} \tag{31}$$

where η is the Lagrange multiplier corresponding to the equality constraint $\sum_{m=1}^M \alpha_m = 1$, and η_m is the Lagrange multiplier corresponding to the inequality constraint $\alpha_m \geq 0$ (equivalently, $-\alpha_m \leq 0$), which is used to ensure that the weights satisfy the normalization and nonnegativity constraints. Based on the KKT conditions, Eq. (31) can be solved as follows: $\frac{\partial \mathcal{L}}{\partial \alpha_m} = 0, \eta_m \alpha_m = 0, \eta_m \geq 0, 1 - \sum_{m=1}^M \alpha_m = 0$. The solution to this equation is given by:

$$\alpha_m = \frac{\left(\text{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right)\right)^{\frac{1}{1-r}}}{\sum_{m=1}^{M}\left(\text{tr}\left(\mathbf{Y}^{(m)\top}\mathbf{D}^{(m)}\mathbf{Y}^{(m)}\right)\right)^{\frac{1}{1-r}}} \tag{32}$$