Incomplete Multi-View Unsupervised Federated Feature Selection via Cooperative Particle Swarm Optimization and Tensor-Aligned Learning

Zhiwei Ye[1,2], Songsong Zhang[1], Wen Zhou[1,2*], Libing Wu[3], Jun Shen[4], Ting Cai[1,2*], Mingwei Wang[1,2], Jixin Zhang[1,2]

[1]School of Computer Science, Hubei University of Technology, Wuhan 430000, Hubei, China
[2]Hubei Provincial Key Laboratory of Green Intelligent Computing Power Network, Hubei University of Technology, Wuhan 430000, Hubei, China
[3]School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, Hubei, China
[4]School of Computing and Information Technology, University of Wollongong, Wollongong, 2500, New South Wales, Australia
hgcsyzw@hbut.edu.cn, 102301187@hbut.edu.cn, zw_mmwh@whu.edu.cn, wu@whu.edu.cn, jun_shen@uow.edu.au, caiting@hbut.edu.cn, wmwscola@hbut.edu.cn, zhangjx@hbut.edu.cn

Abstract

With the widespread adoption of multi-view data in numerous fields, multi-view unsupervised feature selection (MUFS) has made notable strides in both feature pruning and missing-view completion. Nonetheless, existing MUFS methods typically rely on centralized servers, which cannot meet real-world demands for privacy preservation and distributed learning, and they often suffer from the suboptimal solutions and weak convergence guarantees. To address these challenges, IMUFFS, an incomplete multi-view unsupervised federated feature selection via cooperative particle swarm optimization (CPSO) and tensor-aligned learning (TAL) is proposed. Specifically, each client executes CPSO-TAL at two stages: (i) an external optimization phase that involves a CPSO, inspired by the co-evolutionary mechanism of hybrid breeding optimization algorithm, performing a global search in the feature space, and (ii) an internal optimization phase that leverages TAL with imputation and CP decomposition, where CP decomposition reduces dimensionality by decomposing the original tensor into a sum of core components, to learn low-dimensional embeddings, while simultaneously updating anchor graphs and view preference weights, thereby harmonizing imputation and representation learning. On the server side, a federated aggregation strategy using adaptive normalized mutual information (NMI) weighting combines the locally optimized feature selection (FS) weights and NMI scores from clients, ensuring privacy while improving the quality of FS and convergence. Extensive experiments on multiple datasets demonstrate that IMUFFS consistently outperforms state-of-the-art methods, yielding more effective and robust FS and enhancing better missing-view completion.

Introduction

Multi-view data is ubiquitous in real-world scenarios, where each sample is described by diverse features captured from multiple perspectives. These features often exhibit correlations and complementarities across various views (Hao, Liu, and Gao 2025; Ran, Yuan, and Zheng 2025). Due to the heterogeneous nature of feature sources and the labor-intensive process of data labeling, the resulting datasets are typically high-dimensional and unlabeled (Hu et al. 2025; Duan et al. 2025). With the rise of big data, selecting the most informative features from these unlabeled multi-view data has become a critical challenge in diverse fields, including recommendation systems, image recognition, and social network analysis. As for a well-established unsupervised learning method, multi-view unsupervised feature selection (MUFS) has garnered significant attention, with the aim of enhancing the performance of downstream tasks by reducing dimensionality and eliminating redundant features (Liang et al. 2023; Zhang et al. 2024).

Existing MUFS methods might generally be categorized into two categories. The first involves combining the features from multiple views into a single unified view, followed by the application of single-view feature selection (FS) techniques. Representative methods, such as GBUFS (Xu et al. 2023), SPCAFS (Li et al. 2021), and HSL (Mi et al. 2024), are easy to implement but are not able to fully capture the underlying correlations present in the multi-view data. The second directly conducts FS on multi-view data, seeking to uncover the latent relationships between different views. Notable methods in this category include SCMvFS (Cao and Xie 2024), JMVFG (Fang et al. 2023), and SDFS (Zhou et al. 2023). These approaches leverage similarity graphs or graph learning frameworks to enhance the consistency across views and improve the performance of FS. However, these methods typically assume that each sample has a complete set of features in every view, which does not hold in the real world.

Such a challenge is particularly prevalent in some practical applications, for instance, medical diagnostic systems, where patient records and test results are typically accessible. However, privacy concerns often lead to missing personal health information (Han et al. 2022). Missing data is a common hurdle, especially in multi-view data. Although methods like MIMB (Wang et al. 2024) and IMC-MCL (Yin et al. 2025) are capable of handling incomplete multi-view

*Equal contribution as co-corresponding authors.

data, they usually treat missing value imputation and FS as separate processes, failing to fully exploit the synergy between the two. To address this, TIME-FS (Huang et al. 2025) introduces a joint optimization framework that integrates missing value recovery, FS, and low-dimensional representation learning. While effective in handling missing-view data, this centralized MUFS method does not satisfy the privacy requirements of distributed settings and leaves significant room for improvement in FS efficiency, solution quality, and convergence speed.

Another challenge is that existing MUFS methods largely depend on centralized frameworks that assume all data is stored on a central server, which raises serious concerns about privacy and data transmission. Although some studies have explored integrating federated learning (FL) with multi-view domains (Chen et al. 2024; Gao et al. 2025), they primarily focus on downstream tasks such as clustering or representation learning and do not address the upstream FS problem in federated settings, leaving federated MUFS completely unexplored. In the real world, data are often distributed, especially when dealing with sensitive information such as personal or medical records (Hao, Gao, and Hu 2025; Li et al. 2025). FL offers an effective solution by enabling collaborative learning across clients without centralized data sharing. Introducing MUFS into the FL framework for the first time not only bridges a long-standing research gap between MUFS and FL but also establishes a methodological foundation for privacy-preserving and distributed multi-view data analysis, with significant theoretical and practical implications.

In recent years, researchers have explored meta-heuristic algorithms such as the Particle Swarm Optimization algorithm (PSO) (Song et al. 2022), Genetic Algorithms (GA) (Fu et al. 2022), and Ant Colony Optimization algorithm (ACO) (Ye et al. 2023), all of which enhance global search capabilities and improve the stability and convergence of FS. Ye *et al.* (Mei et al. 2025) introduced a hybrid breeding optimization algorithm (HBO), inspired by the heterosis theory in nature, which has demonstrated impressive performance in FS. PSO, known for its robustness and rapid convergence, integrates the co-evolutionary mechanism of HBO with PSO's global search capability, might offer a powerful approach to improving both the performance and stability of MUFS models.

To address these challenges, we propose IMUFFS, an incomplete multi-view unsupervised federated FS framework that integrates cooperative PSO (CPSO) with tensor-aligned learning (TAL). They are incorporated together in a FL framework to jointly optimize missing value recovery, FS, and low-dimensional representation learning. Clients perform CPSO and TAL in two stages: (i) External optimization with CPSO, inspired by the three-line co-evolutionary mechanism of HBO, performs global updates using PSO within each line, conducting a global search in the feature space, and (ii) internal optimization using TAL-based imputation and CP decomposition to learn embeddings while updating anchor graphs and view-preference matrices, harmonizing imputation and representation. On the server side, a federated aggregation strategy with adaptive normalized mutual information (NMI) weighting combines locally optimized FS weight matrices and NMI scores from clients, ensuring privacy while enhancing global performance. Extensive experiments demonstrate that IMUFFS not only preserves privacy but also substantially outperforms existing centralized MUFS methods in FS quality, convergence, and missing-view imputation. The main contributions of this paper are as follows:

- To the best of our knowledge, IMUFFS is the first framework to combine FL with MUFS. This method performs joint optimization of missing value recovery, FS, and low-dimensional representation learning within the FL framework by executing both external and internal optimizations of CPSO and TAL at the client side. On the server side, a federated aggregation strategy via adaptive NMI weighting, is employed to ensure privacy while improving the quality and convergence of MUFS solutions.
- We introduce a CPSO that takes advantage of the co-evolutionary mechanism of HBO with the global search capability of PSO. The population is split into three groups, which are updated via PSO. This algorithm effectively avoids the local optima as external optimization at the client side, achieving efficient and robust FS, while filling a gap in current MUFS research and evolutionary computation algorithms.
- We propose a federated aggregation strategy via adaptive NMI weighting, which allows the server to efficiently aggregate FS results and evaluation metrics from clients. By leveraging FS consistency and NMI weighting across clients, this strategy optimizes the aggregation process, eliminates the need for raw data transmission, hence not only enhances the global performance of the model but also strengthens privacy protection, minimizing the risk of data leakage.

The Proposed Method

This section provides detail of the proposed IMUFFS, focusing on client-side operations, server-side processes, and analyses of privacy protection, communication overhead, and time complexity, respectively.

Client-Side Operation

Each client executes CPSO-TAL (As shown in Fig. 1) in a two-stage manner. In the external optimization phase, CPSO performs a global search over the local feature-subset space. In the subsequent internal optimization phase, TAL, together with imputation and CP decomposition, is used to learn low-dimensional embeddings, while updating anchor graphs and view-preference weights, thereby harmonizing imputation and representation learning. The detailed procedure is as follows.

Client-side MUFS Matrix Initialization. Each client holds a local multi-view dataset $\mathbf{X} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(V)}\}$, with each view v represented by $\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}$, comprising n samples and d_v features. The goal is to learn a feature selection matrix $\mathbf{W}^{(v)} \in \mathbb{R}^{d_v \times c}$ for each view that maps the original features into a common c-dimensional space,

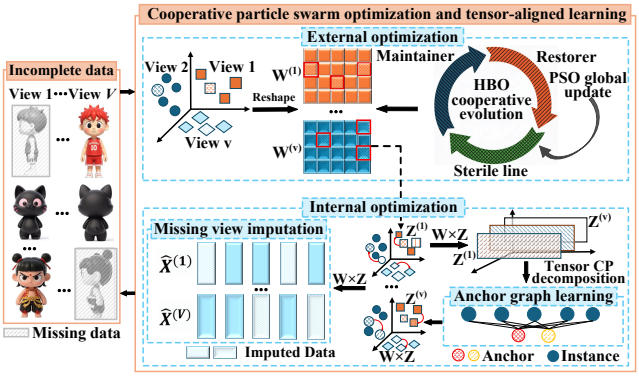

Figure 1: The flowchart of CPSO-TAL.

where c usually denotes the number of clusters or categories. The total number of parameters to be optimized for each client is the sum of the parameters across all views, given by $\dim(\mathbf{W}) = \sum_{v=1}^{V} d_v \times c$.

Each matrix $\mathbf{W}^{(v)}$ is initialized by reshaping a randomly sampled vector $\mathbf{w}_{\text{rec}}^{(v)} \in \mathbb{R}^{d_v c}$ into a matrix:

$$\mathbf{W}^{(v)} = \text{reshape}(\mathbf{w}_{\text{rec}}^{(v)}, d_v, c), \tag{1}$$

where $\text{reshape}(\cdot, d_v, c)$ converts a vector of length $d_v c$ into a $d_v \times c$ matrix in row-major order. The full local FS representation is $\mathbf{W} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(\mathbf{V})}\}$.

External Optimization: Collaborative Particle Swarm Optimization. Inspired by the co-evolutionary mechanism of HBO, the particles are first sorted according to the fitness value and divided into three complementary subgroups: the maintainer line (top 1/3 of particles with the highest fitness), the restorer line (middle 1/3 with moderate fitness), and the sterile line (bottom 1/3 with the lowest fitness). Each subgroup independently executes the procedure of primary PSO (Song et al. 2022), updating both velocity and position.

Subsequently, the child individual is generated by combining two parent particles, p_1 (maintainer group) and p_2 (restorer group), using a weight α sampled from $\text{Uniform}(0, 1)$. A Gaussian mutation is then applied by adding zero-mean, unit-variance noise scaled by the mutation factor σ, i.e., *child* $= \alpha \times \text{parent}_1 + (1 - \alpha) \times \text{parent}_2$ + (mutation factor $\times$ Gaussian noise).

The offspring will replace the worst-performing individuals in the sterile line, effectively eliminating low-quality solutions and replacing them with better ones, thereby enhancing the global convergence of the swarm. The global best particle $\mathbf{x}_{\text{gbest}} \in \mathbb{R}^{W_{\text{dim}}}$ is reshaped into FS matrices $\mathbf{W}^{(v)}$ for each view v. Specifically, each $\mathbf{x}_{\text{gbest}}^{(v)}$ of size $d_v \times k$ is reshaped to form $\mathbf{W}^{(v)}$, where d_v is the number of features for view v, and k represents the number of selected features.

Internal Optimization: Joint Completion and Representation Learning. After each external optimization, the globally searched $\mathbf{W}^{(v)}$ is used to guide the optimization of other variables in the internal optimization. The internal refinement module, inspired by TIME-FS (Huang et al. 2025), performs structured tensor decomposition for missing-view completion, representation learning, and view alignment, following its original derivation with necessary adaptations. The outputs are then fed back to guide the external evolutionary FS process.

(1) Missing-View Completion: Let $\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}$ denote the observed data matrix for view v, with d_v features and n samples. To indicate missing entries, we define a binary mask $\mathbf{G}^{(v)} \in \{0, 1\}^{d_v \times n}$, where $\mathbf{G}_{ij}^{(v)} = 1$ if $X_{ij}^{(v)}$ is missing, and 0 otherwise. The completed matrix is given by

$$\hat{\mathbf{X}}^{(v)} = \mathbf{X}^{(v)} + \mathbf{E}^{(v)} \odot \mathbf{G}^{(v)}, \tag{2}$$

where $\mathbf{E}^{(v)}$ is a learnable imputation matrix and $\odot$ denotes the Khatri–Rao product.

(2) Joint Embedding and View Weighting: Each view v is mapped to a low-dimensional space via $\mathbf{Z}^{(v)} \in \mathbb{R}^{k \times n}$ and a feature selector $\mathbf{W}^{(v)} \in \mathbb{R}^{d_v \times k}$. The joint objective is defined as

$$\begin{aligned} \min_{\{\mathbf{W}^{(v)}, \mathbf{Z}^{(v)}, \alpha_v\}} & \sum_{v=1}^{V} \alpha_v \left(\|\hat{\mathbf{X}}^{(v)} - \mathbf{W}^{(v)} \mathbf{Z}^{(v)}\|_F^2 \right) \\ & + \lambda \operatorname{tr}\left((\mathbf{W}^{(v)} \odot \mathbf{Q}\mathbf{W}^{(v)})^\top \mathbf{W}^{(v)} \right) \\ & + \beta \sum_{v=1}^{V} \|\mathbf{W}^{(v)}\|_{2,1}. \end{aligned} \tag{3}$$

Here, $\|\cdot\|_F$ is the Frobenius norm, $\|\cdot\|_{2,1}$ induces row sparsity, $\mathbf{Q}$ is a structure-aware matrix, and $\lambda, \beta > 0$ are regularization parameters. The weight α_v for each view is computed based on the reconstruction loss d_v and is normalized as follows: α_v is proportional to the normalized reconstruction cost d_v, which includes the reconstruction error and a regularization term involving $\mathbf{Q}$, with the normalization controlled by the factor $\gamma \in (0, 1)$ to adjust the sharpness of the weight distribution.

(3) Tensor-CP Fusion for Multi-View Alignment: To capture cross-view correlations, we stack all embeddings into a third-order tensor:

$$\mathcal{Z} = [\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(V)}] \in \mathbb{R}^{k \times n \times V}, \tag{4}$$

where $\mathbf{Z} \in \mathbb{R}^{k \times n \times \mathbf{V}}$ is the third-order tensor obtained by stacking the embeddings $\mathbf{Z}^{(v)} \in \mathbb{R}^{k \times n}$ of $\mathbf{V}$ views, with k being the embedding dimensionality, n the number of samples, and $\mathbf{V}$ the number of views.

We then apply rank-r CP decomposition to express this tensor as

$$\mathcal{Z} \approx [\![\mathbf{A}, \mathbf{P}, \mathbf{M}]\!] = \sum_{j=1}^{r} \mathbf{a}_j \circ \mathbf{p}_j \circ \mathbf{m}_j, \tag{5}$$

where $\mathbf{A} \in \mathbb{R}^{k \times r}$ encodes anchor bases in the embedding space, $\mathbf{P} \in \mathbb{R}^{n \times r}$ captures sample-to-anchor affinities, $\mathbf{M} \in \mathbb{R}^{V \times r}$ represents view-level coefficients, and $\circ$ denotes the outer product.

(4) Anchor Graph Refinement and Update: We update the CP factors via alternating least-squares on each unfolding of $\mathcal{Z}$.

(i) Anchor matrix update:

$$\mathbf{A} = \mathbf{Z}_{(1)}(\mathbf{P} \odot \mathbf{M}) \left[(\mathbf{P} \odot \mathbf{M})^\top (\mathbf{P} \odot \mathbf{M})\right]^{-1}, \quad (6)$$

where $\mathbf{Z}_{(1)} \in \mathbb{R}^{k \times nV}$ is the mode-1 unfolding.

(ii) Preference matrix update:

$$\mathbf{P} = \mathbf{Z}_{(3)}(\mathbf{MA}) \left[(\mathbf{MA})^\top (\mathbf{MA})\right]^{-1}, \quad (7)$$

where $\mathbf{Z}_{(3)} \in \mathbb{R}^{V \times kn}$ is the mode-3 unfolding.

(iii) Consensus graph update:

$$\begin{aligned} \mathbf{Y} &= -\mathbf{Z}_{(2)}(\mathbf{PA}) \left[\eta \mathbf{I} + (\mathbf{PA})^\top (\mathbf{PA})\right]^{-1}, \\ \mathbf{M} &= \max\left(\frac{1 + \mathbf{Y1}}{r}\mathbf{1}^\top - \mathbf{Y},\, 0\right), \end{aligned} \quad (8)$$

where $\mathbf{Z}_{(2)} \in \mathbb{R}^{n \times kV}$ is the mode-2 unfolding, $\eta > 0$ controls the sparsity of $\mathbf{M}$, and $\mathbf{1}$ is a vector of ones used to enforce non-negativity and normalization.

(iv) Missing value refinement: After representation learning, the missing entries are refined using:

$$\mathbf{E}^{(v)} = \left(\mathbf{W}^{(v)}\mathbf{Z}^{(v)}\right) \odot \mathbf{G}^{(v)}, \quad (9)$$

where $\mathbf{G}^{(v)}$ indicates missing entries and the operation selectively updates only those positions.

(v) Feature selector synchronization: To ensure consistency with the external optimizer, the internal feature selectors $\mathbf{W}^{(v)}$ are reset after each cycle to the best values $\mathbf{W}^{(v)}_{\text{best}}$ found during the global CPSO process. This update ensures that each view's feature selector is synchronized with the global optimization process.

(5) Overall Objective Function: The internal module jointly minimizes reconstruction loss, structural and sparsity regularization, imputation cost, and CP-based alignment:

$$\begin{aligned} \min_{\substack{\{\mathbf{W}^{(v)},\mathbf{Z}^{(v)},\mathbf{E}^{(v)}\} \\ \mathbf{A},\mathbf{P},\mathbf{M},\boldsymbol{\alpha}}} & \sum_{v=1}^{V} \alpha_v \left\|\hat{\mathbf{X}}^{(v)} - \mathbf{W}^{(v)}\mathbf{Z}^{(v)}\right\|_F^2 \\ & + \lambda \sum_v \operatorname{tr}\big((\mathbf{W}^{(v)} \odot \mathbf{Q}\mathbf{W}^{(v)})^\top \mathbf{W}^{(v)}\big) \\ & + \beta \sum_v \|\mathbf{W}^{(v)}\|_{2,1} + \mu \sum_v \|\mathbf{E}^{(v)}\|_F^2 \\ & + \tau \left\|\mathcal{Z} - [\![\mathbf{A}, \mathbf{P}, \mathbf{M}]\!]\right\|_F^2 \\ \text{s.t. } & \alpha_v \geq 0,\ \sum_v \alpha_v = 1,\ \mathbf{M_1} = \mathbf{1},\ \mathbf{M} \geq 0. \end{aligned} \quad (10)$$

Here, $\hat{\mathbf{X}}^{(v)}$ is the imputed data of view v; $\mathbf{W}^{(v)}$ selects and re-weights features, while $\mathbf{Z}^{(v)}$ is its low-dimensional representation. The coefficient α_v is an adaptive non-negative view weight with $\sum_v \alpha_v = 1$. $\mathbf{Q}$ encodes prior structure (e.g., a graph Laplacian), and λ, β, μ, $\tau > 0$ are regularization hyperparameters for structural alignment, sparsity, imputation penalty, and CP decomposition consistency, respectively. The tensor $\mathcal{Z}$ is approximated by its CP factors $(\mathbf{A}, \mathbf{P}, \mathbf{M})$, where $\mathbf{M} \in \mathbb{R}^{V \times r}$ is constrained to be non-negative and row-normalized: $\mathbf{M_1} = \mathbf{1}$.

(6) Update Strategy: All variables (i.e., $\mathbf{Z}^{(v)}$, $\mathbf{A}$, $\mathbf{P}$, $\mathbf{M}$, $\mathbf{E}^{(v)}$, and $\mathbf{W}^{(v)}$) are updated in an alternating fashion until convergence. This coordinated internal loop enables robust missing-view completion, discriminative embedding, and view-aligned representation, thereby enhancing the performance of MUFS.

Algorithm 1: Implementation of IMUFFS

Input: Incomplete multi-view dataset $\mathbf{X} = \{\mathbf{X}^{(v)}\}_{v=1}^{\mathbf{V}}$.
Parameters: Number of clients $\mathbf{K}$, clusters $\mathbf{c}$, hyperparameters $\lambda, \eta, \gamma, r, k$, communication rounds $\mathbf{T}$.
Output: Global $\{\bar{W}^{(v)}\}_{v=1}^{V}$.

```
Initialize W^(v) using Eq. (1), and let t = 0.
for t = 1 to T do
    for client k = 1 to K do
        External Optimization: Evaluate fitness (Eq. (10)), update W^(v).
        Internal Loop: Perform imputation (Eq. (2)), joint embedding (Eq. (3)), Tensor-CP fusion (Eqs. (4–5)), and update anchor graph variables A, P, Y, E^(v) (Eqs. (6–9)).
        Update objective (Eq. (10)).
        Compute NMI and send {W_k^(v), NMI_k} to server.
    end for
    Aggregate global FS matrix {W̄^(v)}_{v=1}^V on server (Eq. (11)), then broadcast to clients.
    Check stopping condition.
end for
return {W̄^(v)}_{v=1}^V
```

Server-Side Operations

Federated Aggregation Strategy via Adaptive NMI Weighting. In FL, non-IID client data often results in uneven local FS quality, and simply averaging client models would degrade the global representation. To address this, we propose a federated aggregation strategy via adaptive NMI weighting, which assigns each client a weight proportional to the clustering quality of its selected features.

After every local training round, client v uploads its FS matrix $\mathbf{W}^{(v)} \in \mathbb{R}^{d \times c}$. The server evaluates clustering quality via normalized mutual information NMI_v and updates the global model as

$$\bar{\mathbf{W}} = \sum_{v=1}^{K} \alpha_v \, \mathbf{W}^{(v)}, \qquad \sum_{v=1}^{K} \alpha_v = 1, \quad (11)$$

where $\bar{\mathbf{W}}$ denotes the aggregated FS matrix; K is the number of participating clients; α_v represents the adaptive weight of client v; and $\mathbf{W}^{(v)}$ is the local FS matrix learned on client v. The weight α_v is calculated as $\alpha_v = \frac{\mathbf{NMI}_v}{\sum_{j=1}^{K} \mathbf{NMI}_j}$, normalizing the NMI of client v to ensure $\sum_{v=1}^{K} \alpha_v = 1$. This strategy preserves privacy, addresses client heterogeneity, and improves the stability of the global model.

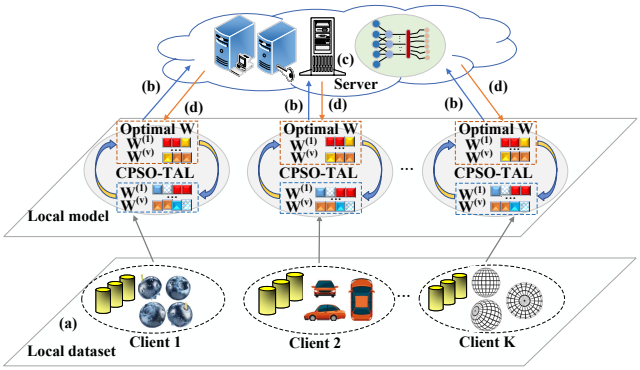

Figure 2: The framework of IMUFFS: (a) Load the local dataset; (b) Upload locally optimized $\mathbf{W}^v$ and NMI; (c) Apply federated aggregation; (d) Broadcast the aggregated results to all clients for further optimization.

After adaptive aggregation of the FS matrices from all clients, the server distributes the updated global FS matrix to each client for initialization in the next training round. The early stopping mechanism is triggered when the global evaluation metric, such as average NMI, changes by less than a predefined threshold (e.g., 1×10^{-3} in this work), prompting the server to broadcast a termination signal, indicating convergence and halting further training.

Privacy and Communication Overhead Analysis

As shown in Fig. 2, the proposed IMUFFS framework allows clients to perform FS locally through CPSO-TAL and transmit model parameters using the federated aggregation strategy with adaptive NMI weighting, ensuring that raw data remains local and minimizing leakage risks. As only abstract parameters are shared, privacy is enhanced. In each round, clients upload local parameters, which the server aggregates into a global model and broadcasts back, with a communication cost of $\mathcal{O}(d \cdot c)$, much lower than transmitting raw data, thus reducing bandwidth usage and further protecting privacy.

Time Complexity Analysis

The entire procedure of IMUFFS is shown in Algorithm 1, where its time complexity involves both client-side computation and server-side aggregation, with client-side computation being the dominant factor. Each client performs CPSO-TAL optimization over $\mathbf{P}$ candidate feature subsets for $\mathbf{E}$ external iterations, incurring a cost of $O(\mathbf{EPVdc})$. The internal optimization phase has a complexity of $O(\mathbf{EP}[Vdnk \cdot \mathbf{c} + rV(\mathbf{c}+nk)])$. On the server side, aggregating and broadcasting feature weights from $\mathbf{K}$ clients incurs a per-round cost of $O(\mathbf{KVdc})$. Over $\mathbf{R}$ rounds, the total complexity is:

$$O(\mathbf{RKEP}[\mathbf{V}dnk \cdot \mathbf{c} + r\mathbf{V}(c+nk)] + \mathbf{V}dc), \qquad (12)$$

where $\mathbf{V}$, r, nk, and $\mathbf{V}dc$ represent data dimensions, computational load, client data size, and constants, respectively. n and k represent the size of the data set and the number of clients, respectively.

Datasets	Views	Samples	Features	Classes
Caltech101	6	1474	48 / 40 / 254 / 1984 / 512 / 928	7
COIL20	3	1440	30 / 19 / 30	20
Digit4k	4	2000	240 / 216 / 47 / 64	10
HandWritten	2	544	4657 / 1125	5
ORL_mtv	3	400	4096 / 3304 / 6750	40
WebKB	3	2100	540 / 640 / 256	21

Table 1: Dataset description.

Experiments

Experimental Settings

Datasets. Our method is evaluated on six real-world datasets across different fields, including object recognition, text classification, handwritten digit recognition, and face recognition. Table 1 provides a detailed summary of the six datasets (Xie, Li, and Sun 2023; Ma et al. 2024; Huang et al. 2025). To simulate the incomplete multi-view scenario, we adopt the strategy from (Lin et al. 2022), where a proportion of samples is randomly omitted from each view to induce missing data. To assess robustness under varying degrees of incompleteness, the missing ratios (MR) are set to $\{10\%, 20\%, 30\%, 40\%, 50\%\}$.

Comparison Methods and Parameter Settings. To evaluate the effectiveness of the proposed IMUFFS, which is the first federated method for MUFS, we compare it with several representative centralized MUFS approaches, including TIME-FS, which represents the State of the Art in MUFS methods (Huang et al. 2025), TRCA-CGL (Liang et al. 2023), SDFS (Zhou et al. 2023), CDMvFS (Cao, Xie, and Li 2024), JMVFG (Fang et al. 2023), SCMvFS (Cao and Xie 2024), and CDNMF (Duan et al. 2024). For completeness, we include a baseline that uses the full set of original features (AllFea). The main parameters of IMUFFS are 10 federated rounds, $\gamma = 6$, $\lambda = 0.01$, $\eta = 1$, with both subspace and anchor count set to c, a maximum of 100 iterations, and an external optimization population size $P = 30$, while the parameters of other comparison methods are set according to their respective literature.

Comparison Schemes. To guarantee a fair comparison, we apply a unified preprocessing strategy for competing methods that cannot handle incomplete multi-view data natively. All baselines, excluding TIME-FS, use mean values to fill missing data before FS. We perform a sensitivity study on four crucial hyperparameters—the sparsity coefficient λ ($\{0.001, 0.005, 0.01, 0.05\}$), the multi-view weighting exponent γ ($\{1.5, 2, 3, 4, 5, 6\}$), the anchor-graph regularisation weight η ($\{0.01, 0.1, 1, 10, 100\}$), and the population size P for the external optimisation stage ($\{10, 20, 30, 40, 50\}$). Because the optimal number of selected features is difficult to predetermine in unsupervised settings (Li et al. 2017), we vary the feature-selection ratio (FR) from 10 % to 90 % in steps of 10 %, thereby assessing the robustness and adaptability of the model across different dimensionalities. All methods are evaluated with k-means clustering, using clustering accuracy (ACC) and NMI as standard MUFS

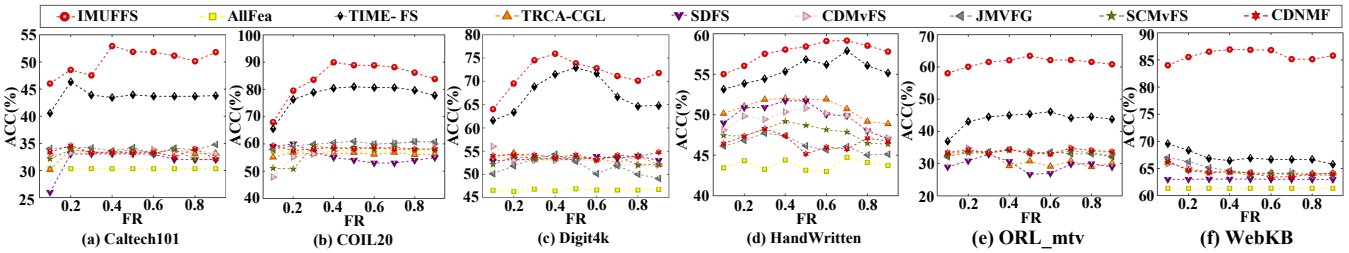

Figure 3: ACC of various methods on six datasets under varying feature selection ratios (5 Clients).

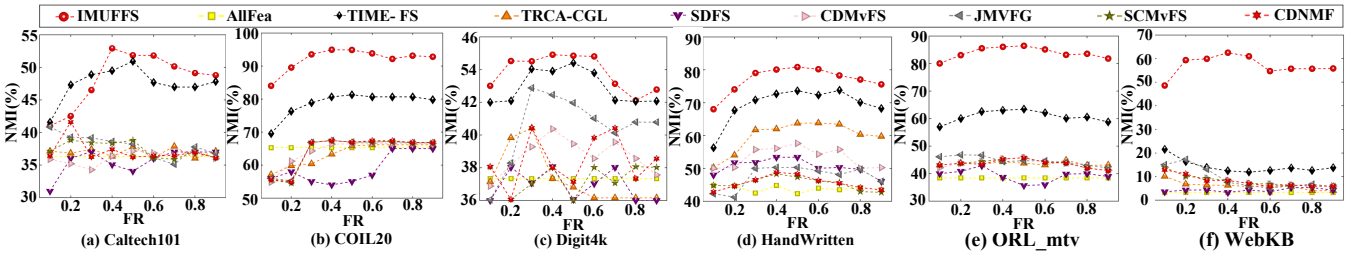

Figure 4: NMI of various methods on six datasets under varying feature selection ratios (5 Clients).

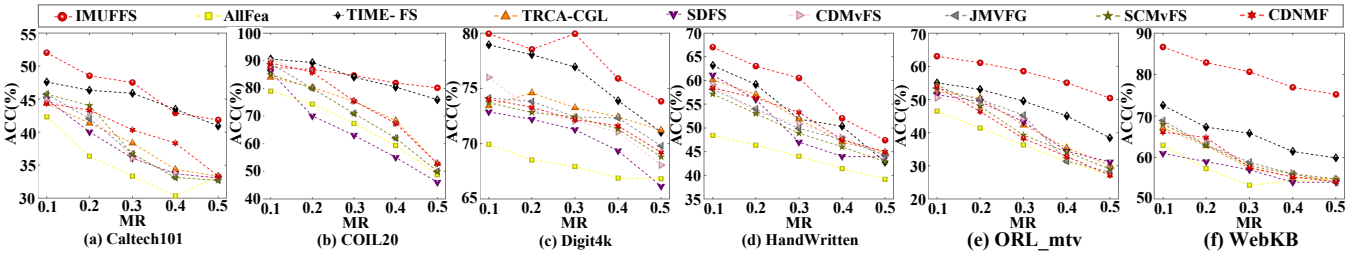

Figure 5: ACC of various methods on six datasets with varying missing ratios (5 Clients).

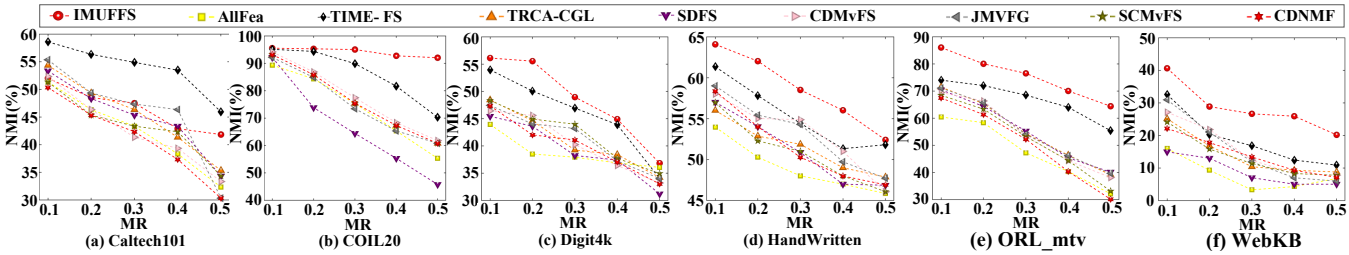

Figure 6: NMI of various methods on six datasets with varying missing ratios (5 Clients).

metrics (Huang et al. 2025); each experiment is repeated 30 times, and the average result is reported. Experiments are performed in MATLAB R2022a on a desktop equipped with an Intel i7-13700KF (3.40GHz), 32GB RAM, RTX 3060 (12GB), and 2.29TB storage. The source code is available at: https://github.com/Zsong1010/IMUFFS.

Experimental Results and Analysis

Performance Comparisons. To evaluate the effectiveness of IMUFFS, we first set up 5 clients and ran each method 30 times, reporting the average clustering performance under different FR and MR. Figs. 3–4 present ACC and NMI results of different methods under varying FR (10%–90%) with MR fixed at 40%. As shown in these figures, IMUFFS consistently outperforms the other methods across the entire range of FR from 10% to 90%. Specifically, on the Caltech101, ORL_mtv, and WebKB datasets, IMUFFS achieves an average improvement of over 10% in both ACC and NMI compared to the best results of the state-of-the-art centralized method (TIME-FS). On the COIL20, Digit4k, and HandWritten datasets, IMUFFS achieves an average improvement of over 6% in ACC and NMI compared to the second-best methods. Furthermore, Figs. 5–6 show ACC and NMI performance of different methods under varying

Datasets	IMUFFS		IMUFFS-I		IMUFFS-II	
	ACC	NMI	ACC	NMI	ACC	NMI
Caltech101	52.85*	53.42*	48.41	47.46	47.21	46.82
COIL20	90.55*	96.12*	85.81	91.97	84.98	90.14
Digit4k	76.65*	45.64*	71.39	40.14	70.80	39.07
HandWritten	65.18*	65.43*	57.95	65.23	56.62	59.77
ORL_mtv	86.73*	87.27*	78.81	82.97	77.08	81.64
WebKB	87.10*	62.87*	80.44	50.59	79.82	47.15

Table 2: ACC(%) and NMI(%) comparison of IMUFFS and its variants, where * indicates a significant improvement (Wilcoxon test, $p < 0.05$).

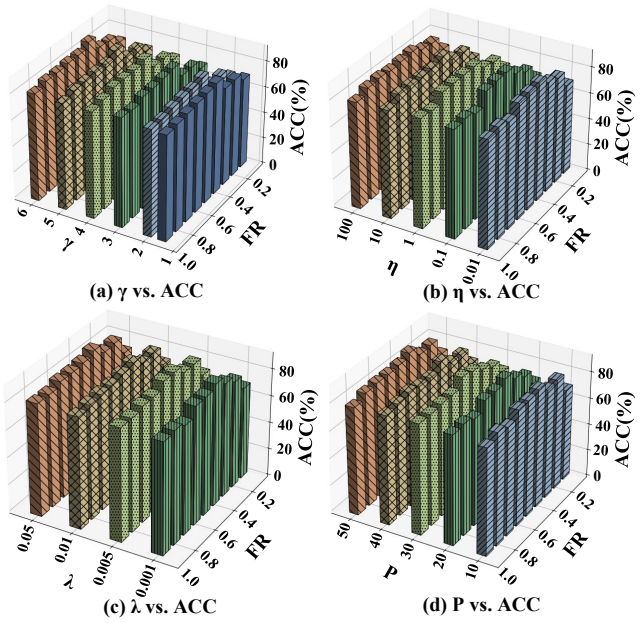

Figure 7: ACC of IMUFFS with varying parameters (γ, η, λ, P) and FR on the COIL20 dataset.

MR, with FR fixed at 40%. In most cases, IMUFFS consistently achieves superior performance over competing methods. The exceptional performance of IMUFFS is attributed to its integration of CPSO, adaptive missing view imputation, anchor graph learning, and adaptive NMI-weighted aggregation strategy into an FL framework. This synergistic integration ensures that each component enhances the overall performance, significantly improving the clustering results.

Ablation Study. To evaluate the contribution of the proposed module in IMUFFS, we perform ablation studies comparing it with two variants: (i) IMUFFS-I: It removes the external optimization, relying solely on internal optimization; (ii) IMUFFS-II: It removes the internal optimization, relying solely on external optimization, with missing values filled using mean imputation. Table 2 presents the average results over 30 runs of the ablation experiments conducted on six datasets, with an MR of 40% and an FR of 40%, and a client setting of 5. The results show that the performance of IMUFFS-I significantly deteriorates compared

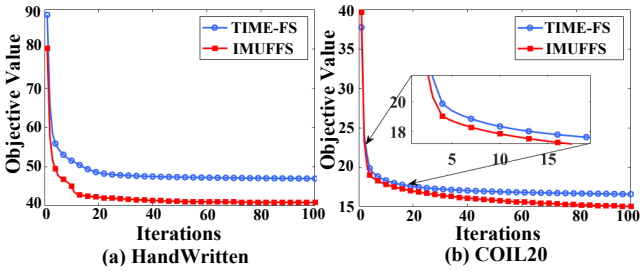

Figure 8: The convergence comparison of IMUFFS and TIME-FS on the HandWritten and COIL20 datasets.

to IMUFFS, emphasizing the crucial role of jointly learning adaptive imputation and feature selection in improving performance. Additionally, IMUFFS outperforms IMUFFS-II, highlighting that incorporating consistent anchor graphs and view-specific weighting improves the accuracy of local structure modeling.

Parameter Sensitivity and Convergence Analysis. IMUFFS has four key tuning parameters: γ, η, λ, and population size P. We demonstrate how performance varies with these parameters and FR, using 5 clients and a 40% MR dataset, with the average of 30 runs, as shown in Fig. 7. Due to space constraints, only ACC results on the COIL20 dataset are reported. For results on other datasets, please refer to the appendix. Fig. 7 shows that IMUFFS performs stably when γ, η, λ, and P are within certain ranges, while FR significantly impacts performance, remaining a challenge in MUFS. Fig. 8 compares IMUFFS's convergence with the latest method (TIME-FS) on the HandWritten and COIL20 datasets, demonstrating that IMUFFS outperforms TIME-FS in both convergence speed and solution quality. Faster convergence improves performance by reducing iterations and computational time, allowing the model to reach the optimal solution more quickly.

Conclusion

In this paper, we introduce IMUFFS, an FL-based approach for incomplete MUFS, integrating CPSO with TAL and a federated aggregation strategy via adaptive NMI weighting. The method effectively addresses the challenges of missing views and privacy protection within an FL framework, enabling efficient and secure FS. Extensive experiments demonstrate that IMUFFS consistently surpasses state-of-the-art MUFS methods on multiple real-world datasets, significantly enhancing the quality of FS, convergence, and computational efficiency while preserving privacy. The combination of CPSO, adaptive missing-view imputation, anchor graph learning, and federated aggregation strategy via adaptive NMI weighting ensures robust solutions while preserving privacy. Future research would focus on developing more advanced imputation techniques and extending the method to larger datasets to further verify its performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62376089, U23A20318, 62302154), the Natural Science Foundation of Hubei Province (2024AFB882), and the Program for Scientific and Technological Innovation Teams of Young and Middle-Aged Researchers in Higher Education Institutions of Hubei Province (T2023007).

References

Cao, Z.; and Xie, X. 2024. Structure learning with consensus label information for multi-view unsupervised feature selection. *Expert Systems with Applications*, 238: 121893.

Cao, Z.; Xie, X.; and Li, Y. 2024. Multi-view unsupervised feature selection with consensus partition and diverse graph. *Information Sciences*, 661: 120178.

Chen, X.; Ren, Y.; Xu, J.; Lin, F.; Pu, X.; and Yang, Y. 2024. Bridging gaps: Federated multi-view clustering in heterogeneous hybrid views. *Advances in Neural Information Processing Systems*, 37: 37020–37049.

Duan, M.; Song, P.; Zhou, S.; Mu, J.; and Liu, Z. 2024. Consensus and discriminative non-negative matrix factorization for multi-view unsupervised feature selection. *Digital Signal Processing*, 154: 104668.

Duan, S.; Sun, Y.; Peng, D.; Duan, G.; Peng, X.; and Hu, P. 2025. Deep Fuzzy Multi-view Learning for Reliable Classification. In *Forty-second International Conference on Machine Learning*.

Fang, S.-G.; Huang, D.; Wang, C.-D.; and Tang, Y. 2023. Joint multi-view unsupervised feature selection and graph learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(1): 16–31.

Fu, T.; Gao, W.; Coley, C.; and Sun, J. 2022. Reinforced genetic algorithm for structure-based drug design. *Advances in Neural Information Processing Systems*, 35: 12325–12338.

Gao, M.; Zheng, H.; Feng, X.; and Tao, R. 2025. Multimodal fusion using multi-view domains for data heterogeneity in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 16736–16744.

Han, T.; Gong, X.; Feng, F.; Zhang, J.; Sun, Z.; and Zhang, Y. 2022. Privacy-preserving multi-source domain adaptation for medical data. *IEEE journal of biomedical and health informatics*, 27(2): 842–853.

Hao, P.; Gao, W.; and Hu, L. 2025. Embedded feature fusion for multi-view multi-label feature selection. *Pattern Recognition*, 157: 110888.

Hao, P.; Liu, K.; and Gao, W. 2025. Uncertainty-Aware Global-View Reconstruction for Multi-View Multi-Label Feature Selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 17068–17076.

Hu, R.; Gan, J.; Zhan, M.; Li, L.; and Wei, M. 2025. Unsupervised Kernel-based Multi-view Feature Selection with Robust Self-representation and Binary Hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 17287–17294.

Huang, Y.; Lu, M.; Huang, W.; Yi, X.; and Li, T. 2025. Timefs: joint learning of tensorial incomplete multi-view unsupervised feature selection and missing-view imputation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 17503–17510.

Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; and Liu, H. 2017. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6): 1–45.

Li, J.; Li, B.; Zhang, X.; Ma, X.; and Li, Z. 2025. MDMNI-DGD: A novel graph neural network approach for druggable gene discovery based on the integration of multi-omics data and the multi-view network. *Computers in Biology and Medicine*, 185: 109511.

Li, Z.; Nie, F.; Bian, J.; Wu, D.; and Li, X. 2021. Sparse PCA via $\ell_{2,p}$, 2, p-Norm Regularization for Unsupervised Feature Selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 5322–5328.

Liang, C.; Wang, L.; Liu, L.; Zhang, H.; and Guo, F. 2023. Multi-view unsupervised feature selection with tensor robust principal component analysis and consensus graph learning. *Pattern recognition*, 141: 109632.

Lin, Y.; Gou, Y.; Liu, X.; Bai, J.; Lv, J.; and Peng, X. 2022. Dual contrastive prediction for incomplete multi-view representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4447–4461.

Ma, Y.; Shen, X.; Wu, D.; Cao, J.; and Nie, F. 2024. Cross-view approximation on grassmann manifold for multiview clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4): 7772–7777.

Mei, M.; Zhang, S.; Ye, Z.; Wang, M.; Zhou, W.; Yang, J.; Zhang, J.; Yan, L.; and Shen, J. 2025. A cooperative hybrid breeding swarm intelligence algorithm for feature selection. *Pattern Recognition*, 111901.

Mi, Y.; Chen, H.; Luo, C.; Horng, S.-J.; and Li, T. 2024. Unsupervised feature selection with high-order similarity learning. *Knowledge-Based Systems*, 285: 111317.

Ran, W.; Yuan, W.; and Zheng, Y. 2025. You Always Recognize Me (YARM): Robust Texture Synthesis Against Multi-View Corruption. In *Forty-second International Conference on Machine Learning*.

Song, X.; Zhang, Y.; Gong, D.; Liu, H.; and Zhang, W. 2022. Surrogate sample-assisted particle swarm optimization for feature selection on high-dimensional data. *IEEE transactions on evolutionary computation*, 27(3): 595–609.

Wang, H.; Yao, M.; Chen, Y.; Xu, Y.; Liu, H.; Jia, W.; Fu, X.; and Wang, Y. 2024. Manifold-based incomplete multi-view clustering via bi-consistency guidance. *IEEE Transactions on Multimedia*.

Xie, X.; Li, Y.; and Sun, S. 2023. Deep multi-view multi-class twin support vector machines. *Information Fusion*, 91: 80–92.

Xu, W.; Huang, M.; Jiang, Z.; and Qian, Y. 2023. Graph-based unsupervised feature selection for interval-valued information system. *IEEE Transactions on Neural Networks and Learning Systems*.

Ye, H.; Wang, J.; Cao, Z.; Liang, H.; and Li, Y. 2023. DeepACO: Neural-enhanced ant systems for combinatorial optimization. *Advances in neural information processing systems*, 36: 43706–43728.

Yin, J.; Wang, P.; Sun, S.; and Zheng, Z. 2025. Incomplete Multi-View Clustering via Multi-Level Contrastive Learning. *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, C.; Fang, Y.; Liang, X.; Wu, X.; Jiang, B.; et al. 2024. Efficient multi-view unsupervised feature selection with adaptive structure learning and inference. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, 5443–5452.

Zhou, S.; Song, P.; Yu, Y.; and Zheng, W. 2023. Structural regularization based discriminative multi-view unsupervised feature selection. *Knowledge-Based Systems*, 272: 110601.