

Supplementary Materials: Heterogeneity-Aware Federated Deep Multi-View Clustering towards Diverse Feature Representations

Anonymous Authors

A OVERVIEW

This supplementary material includes the following contents:

- Privacy analysis, including current privacy levels and methods for enhancing privacy. (Section B)
- Communication overhead experiments, including the communication overhead results on two datasets. (Section C)
- Experimental details, including datasets, models, comparison methods, etc. Particularly important is our explanation for why we did not compare with more other federated multi-view clustering methods. (Section D)
- Additional experimental results, including parameter analysis, convergence analysis, etc. Particularly important is our explanation of why the comparison method performs worse in IID than in Non-IID, and we clarify the motivation behind HFMVC’s heterogeneity-aware module. (Section E)
- The generalizability of HFMVC and its potential application scenarios. (Section F)

B PRIVACY

In fact, in HFMVC, the data exchanged between the client and server mainly includes the following:

- (1) In each global round, the client needs to send the local model and high-level features to the server.
- (2) The server aggregates the received local models and then sends the aggregated model back to the client; additionally, it exchanges the received high-level features among clients.
- (3) After the pre-training phase is completed, the server needs to send the results obtained from the heterogeneous evaluation (i.e., a weight coefficient) back to the client.

Since the data in (3) is only a constant, it essentially does not leak client privacy. Nevertheless, to fully protect the clients’ privacy (such as the heterogeneity of their data), it can be encrypted separately [21]. Therefore, the key focus of the privacy analysis lies in (1) and (2), which involve the transmission of model parameters and high-level features during communication between the clients and server.

B.1 Privacy Analysis

We consider the semi-honest adversaries model, where the server is honest yet curious; it reliably collects updates from participants and returns the updated model, but it may be curious about participants’ information and attempt to uncover this information through the received updates. In this scenario, even though the server knows the clients’ local models and high-level features, it cannot infer the clients’ original data. This is because inferring the original data from given outputs and model parameters involves solving a system of linear equations, which is not feasible due to the non-invertibility of the fully connected layer. The encoding process leads

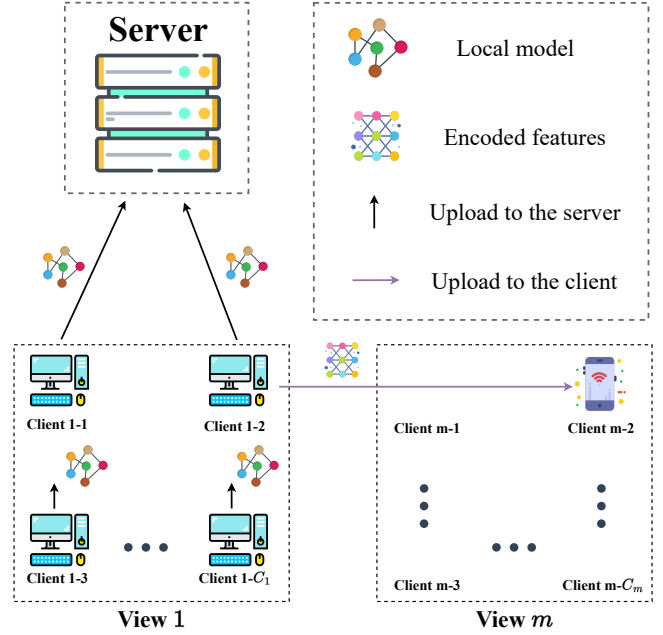


Figure 1: Another feasible approach for applying HFMVC in practice is for the client to send only the local model (encrypted) to the server, while high-level features are sent directly to their corresponding friend clients. This way, high-level features and the local model can be transmitted separately among different entities.

to information loss, making reverse inference of data impossible. Specifically, given the weight matrix W and bias vector b in the fully connected layer, and input X (dimension of 512), the output H (dimension of 128) can be represented as:

$$H = X \cdot W + b. \quad (1)$$

To infer the original data, given the output H , the server needs to infer the original input X . This means solving the equation:

$$X = (H - b) \cdot W^{-1}. \quad (2)$$

Since W is a 512×128 matrix with fewer rows than columns, it is not full rank and is non-invertible. So the server cannot infer the clients’ original data from the high-level features and model parameters.

To further protect client privacy, we consider enhancing the security of the system from the following aspects.

B.2 Enhancing Privacy in Model Aggregation

To ensure security during the model aggregation stage, techniques such as homomorphic encryption [1], differential privacy [5] and secure aggregation protocol [2] can be used to protect the privacy of model parameters. For instance, the following scheme can be used to enhance the level of privacy:

Given n participants, each participant has their own model parameters θ_i , where $\theta_i \in \mathbb{R}^d$, and d is the dimension of the parameters. The goal of the aggregation is to compute the weighted average of all participants' model parameters to generate a global model parameter θ_{global} .

First, each client's local model needs to be encrypted using homomorphic encryption before transmission.

$$E(\theta_i) = \text{Encrypt}(\theta_i, k_i), \quad (3)$$

where $E(\theta_i)$ represents the encrypted model parameters, and k_i is the encryption key of participant i . Then the server performs a weighted average of the encrypted model parameters from each participant directly, without decrypting them.

$$E(\theta_{\text{global}}) = \sum_{i=1}^n w_i E(\theta_i), \quad (4)$$

where w_i is the weight of participant i 's model during the aggregation phase. In HFMVC, we set the weight of each client's model to be equal, i.e., $w_i = 1/n$.

Through the aforementioned homomorphic encryption scheme, the server cannot know the specific parameters of individual models, as all operations are performed on encrypted data. Similarly, secure aggregation protocol [2] can achieve a similar effect.

B.3 Enhancing Privacy in Feature Exchange

Considering the privacy threats and the load on the server when it simultaneously receives local models and high-level features from the client, we propose another feasible approach in practice, as illustrated in Figure 1. The client continues to send its local model to the server as usual, but it sends high-level features selectively to corresponding friend clients. This way, the server cannot access the client's high-level features, while the client can only obtain the high-level features of its friend clients, thus enhancing security through data partitioning. Even if the server colludes with a malicious client to attack another client and obtain its high-level features, the original data cannot be restored due to the encryption of the local model. Furthermore, this approach reduces the server's load since it does not need to receive and relay high-level features, thus avoiding potential server crashes.

C COMMUNICATION COST

In this section, we provide experimental results regarding communication overhead. Table 1 displays the total communication overhead for a single client to reach convergence in various heterogeneous environments and datasets. The results indicate that each client only needs to upload a few hundred megabytes of data, which is entirely acceptable in practice. This holds true for both clients with powerful computing capabilities (such as hospitals and large institutions) and lightweight clients (such as mobile devices).

D EXPERIMENTAL DETAILS

D.1 Datasets

We conduct our experiments on four public datasets, i.e., **MNIST-USPS** [10], **BDGP** [3], **Multi-Fashion** [15], and **Caltech** [6]. Table 2 presents the key features of the relevant datasets, including the number of samples, views, and classes.

D.2 Models

We conduct our experiments using models consistent with many previous works [9, 12, 18–20]. Specifically, for data in view m , with input dimension D_m and output dimension d_m , the encoder has dimensions D_m -500-500-2000-512, while the decoder follows the reverse order, i.e., 512-2000-500-500- D_m . Additionally, we extract high-level features from the encoded features, which have consistent dimensions in the experiments, i.e., 512-128.

D.3 Baselines

We select the following six state-of-the-art (SOTA) methods for comparison: DEMVC [16], SDMVC [17], MFLVC [18], GCFAgg [19], FedDMVC [4] and FCUIF [11]. Please note that among the comparison methods mentioned above, only FedDMVC [4] and FCUIF [11] are specifically designed for distributed environments, while the other methods are centralized solutions. To ensure a fair comparison, we have made slight modifications to them. The following describes the modifications made to these methods:

- DEMVC has proposed a novel collaborative training approach that guides the training of each view in sequence to explore consistent and complementary information from the other views. We have modified the data partitioning strategy of DEMVC to make it compatible with scenarios involving multiple clients.
- SDMVC acquires pseudo-labels through the self-supervision mechanism to establish a unified target distribution, ultimately achieving multi-view discriminative feature learning. Likewise, we have modified its data partitioning strategy to accommodate scenarios involving multiple clients.
- MFLVC aims to learn different levels of features from raw features in a fusion-free manner, including low-level features, high-level features, and semantic labels, with the application of contrastive learning as a crucial strategy. Significantly, in a distributed environment, it is risky for clients to share raw data, making the contrastive learning strategy unfeasible. However, to assess its real performance in a federated setting, we have retained its contrastive learning module, with modifications made to the data partitioning strategy to support multi-client scenarios.
- GCFAggMVC is a multi-view clustering method that employs global and cross-view feature aggregation, with a structurally guided contrastive learning module significantly enhancing prediction accuracy. However, similar to MFLVC, this strategy is challenging to sustain in a distributed environment. Hence, we have made modifications to GCFAggMVC similar to those applied to MFLVC for the purpose of comparison.
- FedDMVC and FCUIF consider multi-view clustering in a federated setting. However, their assumption is too strict

Table 1: The communication overhead required for HFMVC to reach convergence in various heterogeneous environments and across different datasets.

MNIST-USPS				BDGP			
Dirichlet(0.5)	Dirichlet(1.0)	Dirichlet(10)	Dirichlet(∞)	Dirichlet(0.5)	Dirichlet(1.0)	Dirichlet(10)	Dirichlet(∞)
306.3MB	306.3MB	382.9MB	357.3MB	117.9MB	117.9MB	188.6MB	377.1MB

Table 2: Statistics of the related datasets.

Datasets	Samples	Views	Classes	Type
MNIST-USPS	5,000	2	10	Digits
BDGP	2,500	2	5	Images and text
Multi-Fashion	10,000	3	10	Digits
Caltech-5V	1,400	5	7	RGB images

(each client possesses the complete data of one view, i.e., a one-to-one correspondence between views and clients), making it challenging to apply in practice. Therefore, we have modified them to broaden their applicability.

In addition, we conduct a detailed investigation of the following methods related to federated multi-view clustering and provide reasons why comparisons with these methods cannot be made.

- FMSC [13] is a distributed and secure framework that aims to utilize Homomorphic Encryption (HE) and Differential Privacy (DP) to achieve secure and private clustering. However, FMSC focuses on secure clustering in federated settings rather than on high-quality multi-view data mining in heterogeneous environments. Moreover, as a method based on traditional spectral clustering, FMSC’s experimental performance is not satisfactory. More importantly, FMSC is not open-source, and its specific process is difficult to reproduce.
- FedMVL [8] is based on orthogonal non-negative matrix factorization to handle multi-view clustering tasks, primarily addressing issues such as high communication costs, fault tolerance, and stragglers in federated MVC. However, the main issues with FedMVL are: (1) It assumes that M views are distributed among M clients, so the number of clients cannot be freely set. (2) Traditional orthogonal non-negative matrix factorization methods have relatively poorer performance compared to deep learning-based approaches. (3) Its source code is not publicly available, making it challenging to reproduce. Therefore, we did not choose FedMVL as the baseline algorithm.
- FedMVFC [7] is a federated multiview fuzzy C-means clustering method that has demonstrated good performance in practice. However, FedMVFC focuses on a different scenario than HFMVC: it assumes that each client has multi-view data (rather than single-view data) and investigates cross-client information mining based on this assumption. Additionally, FedMVFC is not based on deep learning methods, so we did not choose it for comparison.
- PMCC [14] is built upon the concept decomposition based on local manifold learning, referred to as parallel multiview

concept clustering. However, like most of the methods mentioned above, PMCC is based on traditional distributed learning schemes and is not open-source. Therefore, we did not choose it for comparison.

D.4 Implementation

The batch size in our experiments remains 2500. The learning rate is set to 0.001. All experiments are performed on Windows PC with 11th Gen Intel(R) Core(TM) i7-11700K @ 3.60GHz, one NVIDIA GeForce RTX 2060 with 6GB RAM and one NVIDIA GeForce RTX 3090 GPU with 24GB RAM. The temperature coefficients τ_T and τ_C in contrastive learning are always set to 0.5. All methods undergo a pre-training phase of 500 epochs before training.

E ADDITIONAL EXPERIMENTAL RESULTS

In this section, we present some additional experimental results and analysis as a supplement to the main text.

E.1 Additional Parameter Analysis

In this subsection, we conduct experimental analysis on the two non-critical temperature parameters τ_C (defined in Eq. (4)) and τ_T (defined in Eq. (6)) mentioned in the main text, as depicted in Figure 2. It can be clearly seen that when τ_C and τ_T are between 0.1 and 1, the overall clustering performance is relatively good, with relatively minor fluctuations. Therefore, for such non-critical parameters, we uniformly set them to 0.5 in the experiments.

Additionally, Figure 5 provides experimental results (NMI and ARI) on the parameter analysis of the trade-off coefficients α and β , which are consistent with the conclusions drawn in Figure 4(a) and Figure 4(b) of the main text.

E.2 Additional Convergence Analysis

We plot the convergence curves of individual clients on the BDGP dataset under different heterogeneity settings, as shown in Figure 3. It can be observed that as training progresses, the local loss of each client gradually decreases. This indicates that HFMVC can facilitate the learning process through collaboration among multiple clients, resulting in better global clustering results and also benefiting individual clients.

E.3 Additional Scalability Analysis

Figure 4 illustrates the variation in clustering performance (NMI and ARI) of HFMVC and FedDMVC under different heterogeneous environments as the number of clients changes. It can be observed that, based on these metrics, HFMVC demonstrates superior scalability when faced with multi-client scenarios, consistent with the experimental results (ACC) in the main text.

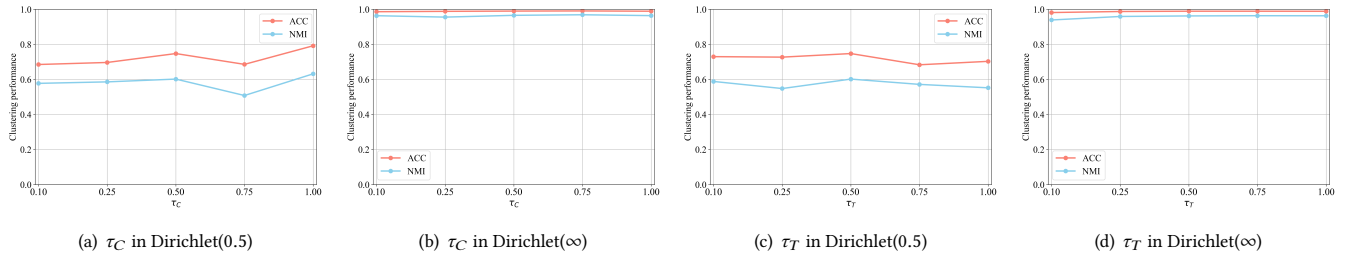


Figure 2: Parameter sensitivity analysis under Dirichlet(∞) (IID) and Dirichlet(0.5) (Non-IID) settings on BDGP.

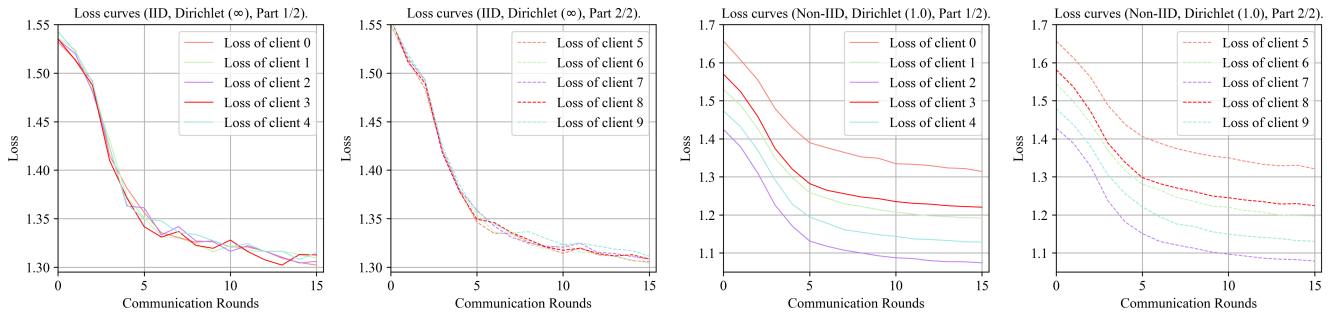


Figure 3: Convergence analysis: the loss curves of individual clients during the training process on the BDGP dataset. And we set up 10 clients in our experiments.

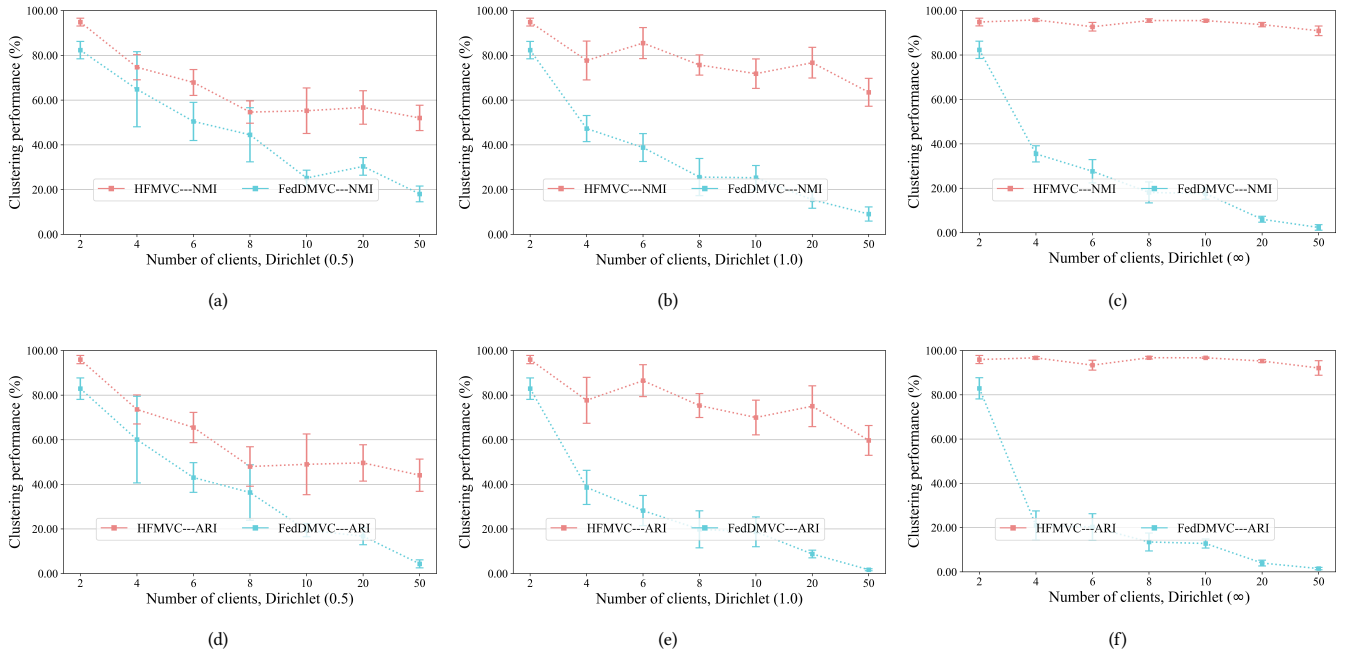


Figure 4: Scalability analysis: clustering performance (NMI and ARI) of HFMVC and FedDMVC under IID and Non-IID scenarios with different numbers of clients.

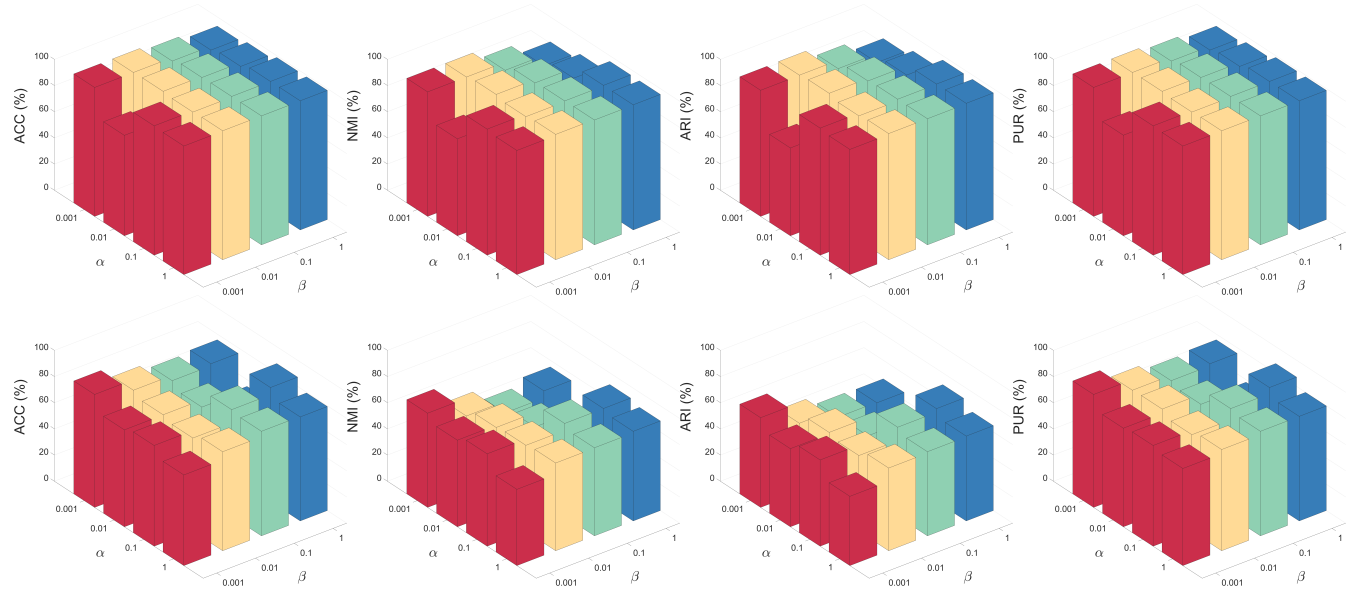


Figure 5: Parameter sensitivity analysis under Dirichlet(∞) (IID) and Dirichlet(0.5) (Non-IID) settings on BDGP.

E.4 Additional Comparison Results

Table 3 supplements the standard deviation of the data for each group and corrects some minor errors in Table 1 of the main text. It is worth noting that unlike HFMVC, the clustering performance of most comparative algorithms tends to decrease as the heterogeneity decreases. We refer to this seemingly anomalous phenomenon as the preference of autoencoders for heterogeneous data. Specifically, if an autoencoder’s data is Non-IID, meaning that the classes and quantities are unequal, the autoencoder only needs to reconstruct these pure (i.e., “heterogeneous” in this context means “pure”) data, resulting in stronger representation capabilities for such heterogeneous data. On the other hand, if the data is IID, each client needs to reconstruct data of the same classes and quantities, making it difficult for individual clients to comprehensively reconstruct each class of data they own, thus leading to poorer overall performance. This characteristic is one of the significant differences between unsupervised and supervised tasks. HFMVC strengthens the communication of information between clients through selective aggregation, thereby improving its clustering performance as the heterogeneity decreases. Therefore, based on this observation, we propose a heterogeneity-aware module, which evaluates the degree of system heterogeneity based on the statistical characteristics of local clustering results from clients after pre-training. It is worth noting that the choice to perform heterogeneity-aware module after pre-training is because at this point, each client has only reconstructed its local data (the model has not been aggregated), thus reflecting the original data characteristics.

F GENERALIZABILITY AND APPLICATIONS

F.1 Generalizability

HFMVC is applicable to various scenarios, even though we primarily assume that each client has data from only one view. For example,

if client 1 possesses data from two views and client 2 has data from one view, we can treat client 1 as two separate clients. In other words, we deploy two autoencoders within client 1, with each autoencoder trained on data from a single view. This approach ensures full compatibility with HFMVC. Therefore, the fundamental assumption we rely on—that each client has data from only one view—can be extended to accommodate more scenarios. This kind of generalizability is lacking in other works [4, 7, 11].

F.2 Applications

HFMVC has wide-ranging applications in practice, including but not limited to the following:

Healthcare. In the healthcare sector, multiple hospitals or research institutions may possess distributed datasets with different views, such as medical images and genomic data, which are often non-IID. HFMVC can facilitate the collaboration and clustering of these data sources while respecting data privacy regulations, thereby promoting joint analysis and knowledge discovery across multiple views. This can improve disease diagnosis, provide personalized treatment recommendations, and enhance understanding of complex medical conditions.

Social Media Analysis. HFMVC can help uncover hidden patterns and structures in multi-modal data sources. By clustering users or content based on multiple views (such as text, images, and social relationships), HFMVC can facilitate targeted advertising, content recommendation, and user analysis.

Finance. Different financial institutions may have diverse customer data and transaction data, including user behavior, transaction patterns, credit scores, and more. HFMVC can assist financial institutions in clustering customers across institutions to discover potential customer groups or anomalous patterns.

Table 3: The clustering performance (mean \pm standard deviation %) across four multi-view benchmark datasets.

Data	Heterogeneity			Dirichlet (0.5)			Dirichlet (1.0)			Dirichlet (10)			IID, Dirichlet(∞)		
	Metrics			ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
MNIST-USPS	DEMVC [16]	38.72 \pm 2.94	32.38 \pm 4.55	20.34 \pm 2.92	36.42 \pm 0.80	13.28 \pm 0.74	7.35 \pm 0.45	25.97 \pm 0.39	16.68 \pm 1.85	8.07 \pm 1.06	29.39 \pm 0.57	6.31 \pm 0.46	4.06 \pm 0.33		
	SDMVC [17]	36.57 \pm 0.79	12.96 \pm 0.24	7.02 \pm 0.20	30.01 \pm 0.75	20.79 \pm 2.27	10.96 \pm 1.54	36.93 \pm 3.56	15.53 \pm 3.99	10.10 \pm 3.20	25.16 \pm 0.95	18.26 \pm 2.07	9.16 \pm 1.71		
	MFLVC [18]	39.74 \pm 3.05	39.55 \pm 3.25	23.11 \pm 2.51	31.89 \pm 1.88	29.65 \pm 1.12	14.88 \pm 1.13	31.88 \pm 2.55	29.65 \pm 1.91	15.25 \pm 1.54	33.36 \pm 4.30	34.71 \pm 4.27	17.44 \pm 3.85		
	GCFAggMVC [19]	49.38 \pm 2.98	43.27 \pm 2.28	28.37 \pm 2.71	42.51 \pm 1.76	36.91 \pm 1.13	21.28 \pm 1.25	30.36 \pm 1.08	24.11 \pm 2.62	11.51 \pm 1.48	26.40 \pm 1.58	22.81 \pm 2.00	10.01 \pm 1.54		
	FedDMVC [4]	46.91 \pm 4.46	41.01 \pm 4.05	28.68 \pm 4.28	43.34 \pm 3.13	31.67 \pm 4.95	25.60 \pm 5.53	29.48 \pm 1.52	21.11 \pm 1.45	11.94 \pm 1.31	26.80 \pm 1.83	20.31 \pm 1.81	10.97 \pm 1.80		
	FCUIF [11]	45.14 \pm 6.59	38.80 \pm 7.09	24.82 \pm 6.32	31.52 \pm 3.35	23.69 \pm 3.48	12.89 \pm 2.84	25.68 \pm 1.32	17.58 \pm 1.94	8.77 \pm 1.23	23.77 \pm 1.61	16.03 \pm 1.96	7.81 \pm 1.23		
	Ours	86.21\pm3.80	84.66\pm5.48	78.96\pm5.22	95.32\pm0.32	91.07\pm0.56	90.29\pm0.63	97.35\pm1.15	94.04\pm1.85	94.31\pm2.38	98.27\pm1.01	96.04\pm1.34	96.31\pm1.95		
	Gain \uparrow	\uparrow 36.83	\uparrow 41.39	\uparrow 50.28	\uparrow 51.98	\uparrow 54.16	\uparrow 64.69	\uparrow 60.42	\uparrow 64.39	\uparrow 79.06	\uparrow 64.91	\uparrow 61.33	\uparrow 78.87		
BDGP	DEMVC [16]	56.78 \pm 8.70	35.01 \pm 8.69	28.66 \pm 9.40	50.94 \pm 1.54	27.17 \pm 4.85	20.24 \pm 2.64	33.44 \pm 3.29	7.28 \pm 0.87	5.10 \pm 1.41	31.21 \pm 0.65	6.98 \pm 1.73	4.95 \pm 1.05		
	SDMVC [17]	57.24 \pm 5.98	36.70 \pm 8.68	27.12 \pm 8.25	51.19 \pm 3.51	31.24 \pm 4.80	21.79 \pm 3.16	35.70 \pm 0.34	15.98 \pm 2.02	10.66 \pm 1.47	43.42 \pm 1.19	18.43 \pm 1.03	13.71 \pm 1.26		
	MFLVC [18]	39.23 \pm 3.88	17.84 \pm 3.74	13.59 \pm 5.70	37.19 \pm 3.92	13.20 \pm 9.13	9.13 \pm 3.16	43.42 \pm 0.45	21.34 \pm 0.87	15.14 \pm 0.53	39.26 \pm 1.41	17.76 \pm 1.41	11.82 \pm 0.73		
	GCFAgg [19]	50.32 \pm 2.46	26.95 \pm 5.81	21.23 \pm 4.54	46.63 \pm 3.20	23.12 \pm 2.69	18.43 \pm 2.40	33.75 \pm 1.51	9.82 \pm 1.70	6.76 \pm 1.17	34.30 \pm 2.36	11.36 \pm 2.27	7.26 \pm 1.37		
	FedDMVC [4]	46.88 \pm 1.08	25.20 \pm 3.47	19.48 \pm 2.94	48.54 \pm 4.50	25.32 \pm 5.47	18.71 \pm 6.70	36.74 \pm 1.74	13.81 \pm 1.70	9.91 \pm 1.04	40.45 \pm 1.69	17.60 \pm 2.53	12.82 \pm 2.06		
	FCUIF [11]	59.76 \pm 7.70	38.89 \pm 9.14	31.36 \pm 11.08	48.01 \pm 4.90	25.18 \pm 5.95	18.31 \pm 5.25	36.59 \pm 4.07	14.44 \pm 2.12	9.38 \pm 3.79	39.62 \pm 5.31	16.58 \pm 5.16	12.46 \pm 5.27		
	Ours	73.32\pm8.44	55.24\pm10.17	48.98\pm13.62	85.69\pm4.66	71.81\pm6.58	69.97\pm7.79	98.42\pm0.47	94.89\pm1.46	96.13\pm1.14	98.67\pm0.13	95.47\pm0.52	96.72\pm0.33		
	Gain \uparrow	\uparrow 13.56	\uparrow 16.35	\uparrow 17.62	\uparrow 34.50	\uparrow 40.57	\uparrow 48.18	\uparrow 55.00	\uparrow 73.55	\uparrow 80.99	\uparrow 55.25	\uparrow 77.04	\uparrow 83.01		
Fashion	DEMVC [16]	39.17 \pm 2.47	34.50 \pm 1.16	21.13 \pm 1.77	35.39 \pm 0.35	31.98 \pm 0.83	17.11 \pm 0.19	26.58 \pm 0.59	18.85 \pm 1.71	8.97 \pm 0.69	30.45 \pm 0.17	26.77 \pm 0.44	15.12 \pm 0.40		
	SDMVC [17]	40.16 \pm 3.00	32.86 \pm 3.61	20.53 \pm 3.91	39.45 \pm 1.25	37.83 \pm 1.46	21.58 \pm 1.49	28.86 \pm 2.10	23.17 \pm 3.10	11.53 \pm 2.04	30.98 \pm 0.71	31.55 \pm 0.52	15.87 \pm 0.37		
	MFLVC [18]	34.00 \pm 2.16	27.68 \pm 2.22	15.65 \pm 2.43	31.72 \pm 2.55	23.49 \pm 2.34	12.56 \pm 1.91	26.51 \pm 1.98	19.87 \pm 2.26	9.43 \pm 1.47	28.32 \pm 1.64	22.46 \pm 2.28	11.34 \pm 1.95		
	GCFAggMVC [19]	54.42 \pm 3.02	54.05 \pm 2.28	36.22 \pm 3.33	50.13 \pm 2.43	51.85 \pm 1.41	33.76 \pm 1.20	29.40 \pm 0.94	31.74 \pm 0.54	13.65 \pm 0.65	34.66 \pm 1.39	45.13 \pm 4.83	24.07 \pm 1.49		
	FedDMVC [4]	33.53 \pm 2.83	28.03 \pm 2.27	15.52 \pm 1.91	34.92 \pm 0.79	28.14 \pm 0.95	16.11 \pm 1.22	29.02 \pm 1.78	23.24 \pm 0.96	11.85 \pm 1.05	38.12 \pm 0.65	38.87 \pm 1.08	23.09 \pm 0.30		
	FCUIF [11]	40.82 \pm 2.93	36.59 \pm 3.87	22.29 \pm 3.61	37.80 \pm 1.91	30.75 \pm 2.33	17.51 \pm 1.33	25.96 \pm 1.81	21.51 \pm 0.65	10.05 \pm 0.62	24.91 \pm 1.50	19.63 \pm 2.18	9.37 \pm 1.44		
	Ours	75.08\pm5.25	80.3\pm4.69	69.08\pm6.02	84.41\pm6.43	85.28\pm1.97	77.18\pm4.72	91.27\pm1.84	88.61\pm1.10	84.44\pm2.19	93.18\pm0.89	89.80\pm0.97	86.85\pm1.36		
	Gain \uparrow	\uparrow 20.66	\uparrow 26.25	\uparrow 32.86	\uparrow 34.28	\uparrow 33.43	\uparrow 43.42	\uparrow 61.87	\uparrow 56.87	\uparrow 70.79	\uparrow 55.06	\uparrow 44.67	\uparrow 62.78		
Caltech-5V	DEMVC [16]	35.89 \pm 1.71	24.91 \pm 2.04	14.88 \pm 1.50	37.87 \pm 2.69	25.13 \pm 3.74	16.02 \pm 2.59	35.01 \pm 3.64	22.27 \pm 5.15	13.80 \pm 3.49	33.13 \pm 3.73	21.17 \pm 6.05	12.08 \pm 4.05		
	SDMVC [17]	38.83 \pm 2.67	25.06 \pm 3.10	15.55 \pm 2.38	40.71 \pm 3.38	28.08 \pm 2.20	19.33 \pm 2.07	34.27 \pm 2.24	18.16 \pm 2.41	10.72 \pm 1.94	29.16 \pm 1.89	14.85 \pm 1.70	8.02 \pm 1.37		
	MFLVC [18]	38.89 \pm 3.13	23.60 \pm 3.54	15.61 \pm 2.41	33.13 \pm 2.87	16.58 \pm 3.27	9.96 \pm 2.60	29.09 \pm 2.25	13.48 \pm 2.57	7.45 \pm 2.04	27.37 \pm 1.93	13.50 \pm 2.30	7.25 \pm 1.51		
	GCFAggMVC [19]	42.65 \pm 3.50	30.56 \pm 1.86	18.62 \pm 1.91	41.28 \pm 3.35	30.39 \pm 4.29	17.94 \pm 3.42	31.61 \pm 2.25	22.00 \pm 3.66	11.84 \pm 2.64	34.00 \pm 4.42	24.95 \pm 4.36	14.09 \pm 3.69		
	FedDMVC [4]	39.93 \pm 3.15	25.56 \pm 4.71	17.66 \pm 4.18	40.39 \pm 4.14	25.39 \pm 5.36	17.50 \pm 4.64	36.27 \pm 1.66	19.13 \pm 2.19	12.25 \pm 2.00	35.30 \pm 2.23	21.67 \pm 1.99	13.29 \pm 1.85		
	FCUIF [11]	50.83 \pm 4.83	40.67 \pm 5.33	28.53 \pm 3.98	38.67 \pm 2.35	25.97 \pm 2.69	16.38 \pm 0.55	32.09 \pm 2.24	19.15 \pm 2.64	10.75 \pm 1.86	32.56 \pm 1.16	19.30 \pm 2.27	11.70 \pm 1.71		
	Ours	60.24\pm8.86	50.02\pm6.52	40.62\pm7.37	63.26\pm2.01	55.04\pm3.17	45.42\pm3.61	69.69\pm5.90	62.02\pm6.17	52.97\pm6.51	66.76\pm4.33	56.97\pm6.20	47.98\pm6.21		
	Gain \uparrow	\uparrow 9.41	\uparrow 9.35	\uparrow 12.09	\uparrow 21.98	\uparrow 24.65	\uparrow 26.09	\uparrow 33.42	\uparrow 39.75	\uparrow 39.17	\uparrow 31.46	\uparrow 32.02	\uparrow 33.89		

REFERENCES

- [1] Abbas Acar, Hidayet Aksu, A Sercuk Uluagac, and Mauro Conti. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)* 51, 4 (2018), 1–35.
- [2] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1175–1191.
- [3] Xiao Cai, Hua Wang, Heng Huang, and Chris Ding. 2012. Joint stage recognition and anatomical annotation of drosophila gene expression patterns. *Bioinformatics* 28, 12 (2012), i16–i24.
- [4] Xinyue Chen, Jie Xu, Yazhou Ren, Xiaorong Pu, Ce Zhu, Xiaofeng Zhu, Zhifeng Hao, and Lifang He. 2023. Federated Deep Multi-View Clustering with Global Self-Supervision. In *Proceedings of the 31st ACM International Conference on Multimedia*. 3498–3506.
- [5] Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*. Springer, 1–12.
- [6] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*. IEEE, 178–178.
- [7] Xingchen Hu, Jindong Qin, Yinghua Shen, Witold Pedrycz, Xinwang Liu, and Jiayuan Liu. 2023. An Efficient Federated Multi-view Fuzzy C-Means Clustering Method. *IEEE Transactions on Fuzzy Systems* (2023).
- [8] Shudong Huang, Wei Shi, Zenglin Xu, Ivor W Tsang, and Jiancheng Lv. 2022. Efficient federated multi-view learning. *Pattern Recognition* 131 (2022), 108817.
- [9] Jiaqi Jin, Siwei Wang, Zhibin Dong, Xinwang Liu, and En Zhu. 2023. Deep Incomplete Multi-view Clustering with Cross-view Partial Sample and Prototype Alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11600–11609.
- [10] Xi Peng, Zhenyu Huang, Jiancheng Lv, Hongyuan Zhu, and Joey Tianyi Zhou. 2019. COMIC: Multi-view clustering without parameter selection. In *International conference on machine learning*. PMLR, 5092–5101.
- [11] Yazhou Ren, Xinyue Chen, Jie Xu, Jingyu Pu, Yonghao Huang, Xiaorong Pu, Ce Zhu, Xiaofeng Zhu, Zhifeng Hao, and Lifang He. 2024. A novel federated multi-view clustering method for unaligned and incomplete data fusion. *Information Fusion* (2024), 102357.
- [12] Huayi Tang and Yong Liu. 2022. Deep safe incomplete multi-view clustering: Theorem and algorithm. In *International Conference on Machine Learning*. PMLR, 21090–21110.
- [13] Hongtao Wang, Ang Li, Bolin Shen, Yuyan Sun, and Hongmei Wang. 2020. Federated multi-view spectral clustering. *IEEE Access* 8 (2020), 202249–202259.
- [14] Hao Wang, Yan Yang, Xiaobo Zhang, and Bo Peng. 2020. Parallel multi-view concept clustering in distributed computing. *Neural Computing and Applications* 32, 10 (2020), 5621–5631.
- [15] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).
- [16] Jie Xu, Yazhou Ren, Guofeng Li, Lili Pan, Ce Zhu, and Zenglin Xu. 2021. Deep embedded multi-view clustering with collaborative training. *Information Sciences* 573 (2021), 279–290.
- [17] Jie Xu, Yazhou Ren, Huayi Tang, Zhimeng Yang, Lili Pan, Yang Yang, Xiaorong Pu, S Yu Philip, and Lifang He. 2022. Self-supervised discriminative feature learning for deep multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering* (2022).
- [18] Jie Xu, Huayi Tang, Yazhou Ren, Liang Peng, Xiaofeng Zhu, and Lifang He. 2022. Multi-level feature learning for contrastive multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16051–16060.
- [19] Weiqing Yan, Yuanyang Zhang, Chenlei Lv, Chang Tang, Guanghui Yue, Liang Liao, and Weisi Lin. 2023. GCFAgg: Global and Cross-view Feature Aggregation for Multi-view Clustering. In *Proceedings of the IEEE/CVF Conference on Computer*

- [20] Xihong Yang, Jin Jiaqi, Siwei Wang, Ke Liang, Yue Liu, Yi Wen, Suyuan Liu, Sihang Zhou, Xinwang Liu, and En Zhu. 2023. Dealmvc: Dual contrastive calibration for multi-view clustering. In *Proceedings of the 31st ACM International Conference on Multimedia*. 337–346.
- [21] Xin Zhou and Xiaofei Tang. 2011. Research and implementation of RSA algorithm for encryption and decryption. In *Proceedings of 2011 6th international forum on strategic technology*, Vol. 2. IEEE, 1118–1121.

697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754

755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812