Communication-Efficient Federated Multi-view Clustering

Jiyuan Liu, Xinwang Liu, Siqi Wang, Xinhang Wan, Dongsheng Li, Kai Lu and Kunlun He

***Abstract*—Federated multi-view clustering is an emerging machine learning paradigm that groups the data with each view distributed on an isolated client while preserving their privacies. Although recent researches have proposed a few feasible solutions, they are severely limited by two drawbacks. In specific, the clients are required to share their data representations at each iteration of model training, leading to heavy communication overhead. On the other hand, existing researches handle large-scale data by employing the matrix factorization and neural network encoding techniques, failing to utilize their similarity information sufficiently. To address these issues, we propose a communication-efficient federated multi-view clustering framework by approximating the data representation with pseudo-label and centroid matrix, where the latter two are shared in model training. Meanwhile, the framework is instanced by incorporating linear kernel function to consider the data pairwise similarities. Note that, corresponding linear kernels are not required to compute explicitly, making the resultant method able to be optimized in linear complexity to the number of samples. Nevertheless, the proposed method is evaluated on benchmark datasets. It not only achieves inspiring results (26.84% accuracy improvement on average, $2.9_\times$-$2153_\times$ computation speedup and 98.4% communication overhead reduction at most) compared with existing federated multi-view clustering methods, but also outperforms centralized multi-view clustering approaches on performance and computation efficiency.**

***Index Terms*—federated multi-view clustering, multi-view clustering, federated learning, linear kernel, matrix approximation**

I. Introduction

MULTI-VIEW clustering is a long-standing unsupervised learning technique to incorporate data information from different sources or modalities. In recent decades, a large amount of multi-view clustering methods have been proposed and achieved satisfactory performances [1]–[6]. Commonly, almost of them assume that the multi-view data are centralized and available unconditionally. However, in numerous real-world applications, the data are distributed across different devices, organizations or owners, and restricted to share due to privacy concerns. For example, when making advertisement delivery, Internet content providers usually match potential target customers by analyzing the user profiles from multiple aspects, such as basic information, financial conditions, shopping records, etc. These profiles are mostly possessed by the providers, banks and shopping platforms which would not exchange their raw data directly. In this background, federated multi-view clustering is emerging to be a promising solution and has attracted a lot of research interest in recent years [7]–[12].

Briefly, federated multi-view clustering is a machine learning paradigm that groups the distributed multi-view data into categories while preserving their privacies [7], [8]. Typically, it is composed of multiple isolated clients and an independent server. In specific, each client corresponds to a data holder which possesses only one view of the data, whereas the server always refers to a trusted third-party. In the clustering procedure, the clients first generate local data representations with their own data and send them to the server. Once received by the server, they are integrated to compute a consensus representation. Thereafter, the consensus is sent back to all clients to update the local data representations. By executing the above processes until convergence, the server will obtain the desired consensus data representation and compute the cluster assignments on its basis. Under the aforementioned learning framework, some researches also customize the generation of local data representations on clients and the integration of information on the server to improve clustering performance, such as [13], [14]. For example, Yan et al. use heterogeneous graph neural networks to extract data features on clients [13]. Feng et al. adopt graph-based regularization to retain the local geometric data structure on the server, obtaining better clustering performance [14].

In order to deal with large-scale data and improve clustering efficiency, existing federated multi-view clustering approaches employ the matrix factorization techniques. For instance, Huang et al. factorize each data view into a low-dimensional non-negative matrix and local clustering assignment matrix (i.e. the local data representation) on all clients, achieving linear complexity [8]. Besides, the neural network encoding is another popular option to reduce the computation complexity. Chen et al. use the fully-connected neural network to encode data on clients [12], while Yan et al. construct a two-layer graph neural network as encoder to generate data representations [13].

Although the existing federated multi-view clustering approaches achieve feasible results, they are severely limited by the following two drawbacks. It is obvious that the aforementioned framework relies on data representations to transmit the

Jiyuan Liu is with the College of Systems Engineering, National University of Defense Technology, Changsha, Hunan, China, 410072. E-mail: liujiyuan13@nudt.edu.cn.

Xinwang Liu, Siqi Wang, Xinhang Wan and Kai Lu is with the College of Computer, National University of Defense Technology, Changsha, Hunan, China, 410072.

Dongsheng Li is with the National Key Laboratory of Parallel and Distributed Computing, National University of Defense Technology, Chang sha, Hunan, China, 410072.

Kunlun He is with the Chinese PLA General Hospital, Beijing, China, 100039.

Corresponding author: Xinwang Liu

Manuscript received January 9, 2025.

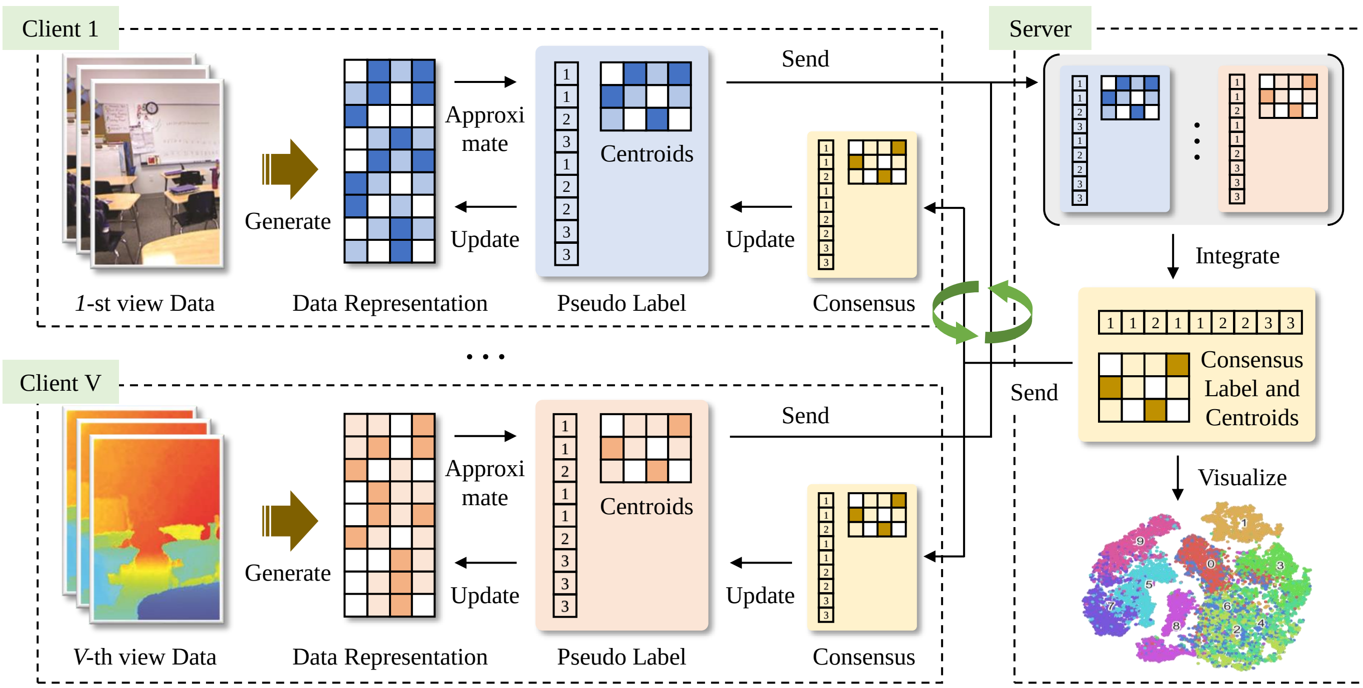

Fig. 1. The proposed communication-efficient federated multi-view clustering framework. It is composed of V clients and a unique server, where each client holds one view of data and the server is responsible for integrating the information from all the clients. For ease of expression, only clients 1 and V are shown while the others are similar and omitted here. In addition, the green circle indicates that the data transmissions between clients and the server are executed cyclically.

information of multiple views between the clients and server in each iteration. Denoting their dimension and the sample number to N and d, respectively, the data representation is of size $N \times d$. Hence, the corresponding communication consumption is at least $\mathcal{O}(tdN)$, in which t refers to the iteration number. Consequently, a larger d, especially when dealing with large-scale data, will induce an obvious increase in computation overhead. On the other hand, although the matrix factorization and neural network encoding techniques successfully decrease the computation complexity in data representation generation, they fail to utilize the similarity information of data samples sufficiently, leaving room for further improvement of the clustering performance.

To address these two issues, we, to the first attempt, propose a **C**ommunication-**e**fficient **F**ederated **M**ulti-view **C**lustering (CeFMC) framework, as shown in Fig. 1. Similar to the common federated learning framework, it is composed of V clients and a unique server, where each client holds one view of data and the server is responsible for integrating the information from all the clients. Specifically, each client first generates the local data representation corresponding to its own data. Instead of transmitting them directly like the existing approaches, the proposed framework approximates them with two parts, including a pseudo-label vector and a centroid matrix, which are sent to the server with a relatively low communication overhead. Once all pseudo-labels and centroids are collected, the server integrates them to compute the consensus. Then, the consensus are sent back to clients and used to update the locally generated pseudo-labels and centroids. By executing the above processes in a number of iterations till convergence, the desired consensus labels can be obtained on the server. Moreover, we also instance the proposed framework by incorporating linear kernel function on the data of each client to consider their pairwise similarities. Note that, corresponding linear kernels are not required to compute explicitly, preventing from introducing the extra computation complexity. To solve the proposed method, we propose an optimization procedure of linear complexity to the number of data samples. Nevertheless, we conduct extensive experiments on seven popular benchmark datasets and the results well support its effectiveness, computation efficiency and communication efficiency over the competing solutions. Overall, the contributions can be summarized as follows:

1) To the first attempt, we propose to approximate the data representation, i.e. the bottleneck of information integration in federated multi-view clustering, with a pseudo-label vector and a centroid matrix and transmit them between the clients and the server, significantly reducing the communication overhead.
2) We develop a communication-efficient federated multi-view clustering framework and instance it by incorporating the linear kernel function on data implicitly, achieving not only a linear computation complexity but also the leading performance.
3) The proposed method is evaluated on benchmarks. It not only achieves inspiring results (26.84% accuracy improvement on average, $2.9_{\times}$-$2153_{\times}$ computation speedup and 98.4% communication overhead reduction at most) compared with existing federated multi-view clustering methods, but also outperforms the centralized

multi-view clustering approaches on performance and computation efficiency.

The rest of the paper is organized as follows. Section II introduces the closely related researches, while Section III gives out the problem definition of federated multi-view clustering formally. Moreover, Section IV presents the proposed communication-efficient federated multi-view clustering framework and corresponding instance method. Additionally, the optimization procedure, computation complexity, communication overhead, convergence and privacy requirement are also introduced and analyzed simultaneously. Nevertheless, the experiment settings and results are provided in Section V. Finally, we make conclusion and present the future work in Section VI.

II. Related work

In this section, we briefly introduce the closely related researches, including multi-view clustering and federated multi-view clustering.

A. Multi-view clustering

With the rapid development of electronic device and information technology, increasing amounts of data are collected from multiple sources or modalities, which is annotated to multi-view data. During the last decades, multi-view clustering has attracted a large volume of interest in research communities, since it can group multi-view data effectively by integrating the complementary information among different views [1]–[6], [15]. Mostly, the existing approaches are derived from classical single-view ones and can be roughly classified into four categories: multiple kernel clustering [6], [16], [17], multi-view subspace clustering [4], [5], [18], multi-view spectral clustering [19]–[22] and multi-view matrix factorization [23], [24]. For example, Liu et al. propose a multiple kernel clustering algorithm on the basis of kernel k-means by assuming that the consensus kernel is a weighted kernel sum of different data views [16]. Zhang et al. develop the latent multi-view subspace clustering algorithm by seeking the underlying consensus latent representation of all data views and simultaneously performing subspace clustering on it [4].

Although the above methods achieve promising results, they are often of high computation complexity and impractical to handle large-scale data, especially with limited computation resources. Therefore, a number of researchers propose to improve their efficiencies by utilizing techniques of linear complexity, such as matrix factorization, anchor graph and neural network encoding. For instance, Wang et al. directly factorize each data view into a basis matrix and a data representation matrix by following the widely-used Non-negative Matrix Factorization (NMF) and impose a diversity regularization on the data representations [25]. Kang et al. first learn an anchor graph with respect to each view and then substitute the complete graphs in multi-view spectral clustering with them, successfully achieving linear complexity to the number of data samples [26]. Yang et al. propose a novel end-to-end deep multi-view clustering approach through encoding the original data into latent representations with deep neural networks and simultaneously incorporating collaborative learning to fuse their complementary information and promote consistent cluster structure for a better clustering solution [27].

Apart from the large-scale data setting, other conditions, such as incomplete data, noisy data and unaligned data, are also sufficiently explored in research communities [28]–[34].

B. Federated multi-view clustering

Federated multi-view clustering is an emerging learning paradigm to group the multi-view data distributed across different clients while preserving their privacies [7]–[12]. In recent researches, the existing methods assume each data holder to be an isolated client and always require an independent server, i.e. a trusted third-party, when building the clustering model.

According to the distribution of multi-view data, they can be roughly categorized into two groups. The first group considers the consequence where each client holds the data observations of all views but partial data samples [7], [35], [36], which is annotated to horizontal federated multi-view clustering for ease of expression. A representative is the federated multi-view fuzzy c-means clustering method in [36]. Briefly, each client independently computes the prototypes on its multi-view data with the local multi-view fuzzy c-means clustering technique and sends them to the server. When the server collects the prototypes from all clients, it aggregates them into the consensus by projecting to a unified embedding space. Also, Bárcena et al. do the similar but design a strategy to alternately execute the object assignment to clusters and simultaneously update the data centers in a collaborative way [7].

The other group of methods takes the consumption that each client holds the data observations of one view but all data samples [8], [13], [14], [37], which is annotated to vertical federated multi-view clustering here. It is worth noting that the proposed CeFMC method belongs to this group whose details are described in the problem definition of Section III. In such setting, Yan et al. build different auto-encoders with heterogeneous graph neural networks on clients to extract data representations and aggregate them into a global one on the server [13]. Feng et al. compute the data representations with matrix tri-factorization on each client and adopt a graph-based regularization to utilize them on the server [14]. To deal with large-scale data, some researchers also adopt the matrix factorization and neural network encoding techniques to reduce the computation complexity. For example, Huang et al. factorize each data view into a low-dimensional non-negative matrix and a local clustering assignment matrix (i.e. the local data representation) on all clients, achieving linear complexity [8]. Chen et al. use the fully-connected neural network to encode data on clients [12], while Yan et al. construct a two-layer graph neural network as the encoder to generate data representations [13].

III. Problem definition

For simplicity and clarity of expression, the table of notations is first provided in Table I. Given the multi-view data

TABLE I
TABLE OF NOTATIONS.

Notation	Meaning
N, V, k	the number of data samples, views and classes
$\mathbf{X}_v$	the v-th view data
$\mathbf{K}_v$	the kernel matrix of the v-th view data
$\mathbf{H}_v$	the generated representation of the v-th view data on the v-th client
$\mathbf{y}_v$, $\mathbf{Y}_v$	the label vector and its one-hot encoding on the v-th client
$\mathbf{C}_v$	the centroid matrix on the v-th client
$\mathbf{H}_c$	the reconstructed representation concatenation of all clients on the server
$\mathbf{y}$, $\mathbf{Y}$	the consensus label vector and its one-hot encoding on the server
λ, β	the trade-off parameters of representation approximation and label integration
$\mathbf{A}^{(0)}$	the initial value of an arbitrary matrix $\mathbf{A}$
$\hat{\mathbf{A}}$, $\bar{\mathbf{A}}$	the temporary matrix in computation of an arbitrary matrix $\mathbf{A}$
obj_v, obj	the objective values of the v-th client and the server

$\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_V\}$ and V isolated clients $\{C_1, C_2, \cdots, C_V\}$, this paper considers the consequence where the v-th view data $\mathbf{X}_v$ is distributed on the v-th client C_v. Also, a trusted third-party is required to group the data into categories by aggregating the data information from all clients and defined to be the server S. Assuming the data ID set of each client to be $\{I_1, I_2, \cdots, I_V\}$, the complete data setting is considered as

$$I_1 = I_2 = \cdots = I_v \neq \emptyset. \tag{1}$$

Since the data of different views are collected and possessed by different organizations in federated multi-view setting, they are not aligned with each other in most cases. Before building the clustering model, we apply the widely explored Private Set Intersection (PSI) technique [38], [39] by default to achieve data alignment and also use $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_V\}$ to denote the aligned data for simplicity of expression.

In the building of clustering model, the clients are forbidden from transmitting their raw multi-view data to each other or the server, while only the encoded data information $\mathbf{D}_v$ and byproducts $\mathbf{B}_v$, such as data representation, hyper-parameter, data centroid in latent space, etc., are shared. Meanwhile, the shared data information and by-products should be protected with encryption by default and we do not discuss this explicitly in the following. Denoting the model of the v-th client to be $f_{\Theta_v}(\cdot)$ and

$$\mathbf{D}_v, \mathbf{B}_v = f_{\Theta_v}(\mathbf{X}_v), \tag{2}$$

one may infer the multi-view data via

$$\mathbf{X}_v = f_{\Theta_v^{-1}}(\mathbf{D}_v, \mathbf{B}_v), \tag{3}$$

in which $f_{\Theta_v^{-1}}(\cdot)$ is the inverse model, leading to the leakage of data privacy. Therefore, the model $f_{\Theta_v}(\cdot)$ and data features of $\mathbf{X}_v$ are also expected to keep private, preventing from the illegal building of $f_{\Theta_v^{-1}}(\cdot)$.

IV. METHODOLOGY

In this section, we first propose the CeFMC framework. Then, its instance is introduced by incorporating linear kernel function to consider the data pairwise similarity. Nevertheless, corresponding optimization procedure is presented. Finally, its computation complexity, communication overhead, convergence and privacy requirements are analyzed.

A. The proposed framework

Similar to the existing federated multi-view clustering approaches, the proposed CeFMC framework is shown in Fig. 1. It is composed of V clients and a unique server, where each client holds one view of data and the server is responsible for utilizing the information of all the clients. Accordingly, four main processes are considered, including representation generation, representation approximation, label integration and alternate update.

Representation Generation. Given the v-th view data $\mathbf{X}_v$ of size $N \times D_v$, client C_v will generate its data representation $\mathbf{H}_v$ of size $N \times d_v$. Note that, N, D_v and d_v refer to the sample number, the dimension of the v-th view data and the dimension of corresponding data representation, respectively. Commonly, there are two possible approaches, where the first projects the data explicitly as

$$\mathbf{H}_v = \mathcal{G}_{\Theta_v}(\mathbf{X}_v), \tag{4}$$

in which $\mathcal{G}_{\Theta_v}(\cdot)$ is a mapping function parameterized by Θ_v, such as neural networks [12]. Also, the data representation can be implicitly generated by imposing proper constraints and being optimized accordingly, i.e.

$$\min_{\mathbf{H}_v} \mathcal{G}(\mathbf{X}_v, \mathbf{H}_v), \quad s.t. \ \mathbf{H}_v \in \Delta, \tag{5}$$

where $\mathcal{G}$ represents the constraints and Δ is the feasible domain of data representation $\mathbf{H}_v$.

Representation Approximation. Once the data representations are generated, the existing researches directly transmit $\mathbf{H}_v$ to the server. Instead, the proposed CeFMC framework approximates it into two parts, including a pseudo-label vector $\mathbf{y}_v$ of size $N \times 1$ and a centroid matrix $\mathbf{C}_v$ of size $c \times d_v$ (c is the number of centroids), via minimizing the following

$$\min_{\mathbf{Y}_v, \mathbf{C}_v} \ell_v(\mathbf{H}_v, \mathbf{Y}_v\mathbf{C}_v), \tag{6}$$

where $\mathbf{Y}_v \in \{0,1\}^{N \times c}$ is the one-hot encoding of $\mathbf{y}_v$. In literature, there are a volume of functions to implement $\ell_v(\cdot)$, such as the L_2-norm of their difference. It is worth noting that each client can select or design its ℓ_v without considering the others, increasing flexibility of the CeFMC framework. Once the immediate variables $\mathbf{y}_v$ and $\mathbf{C}_v$ are obtained, they are transmitted to the server.

Label Integration. With the received pseudo-label $\mathbf{y}_v$ and centroid matrix $\mathbf{C}_v$, server S is supposed to integrate them for further clustering. Denoting the consensus data label and centroids to $\mathbf{y}$ and $\mathbf{C}$, one can minimize the following

$$\min_{\mathbf{y}, \mathbf{C}} \sum_{v=1}^{V} \ell_S(\mathbf{Y}_v\mathbf{C}_v, \mathbf{Y}\mathbf{C}). \tag{7}$$

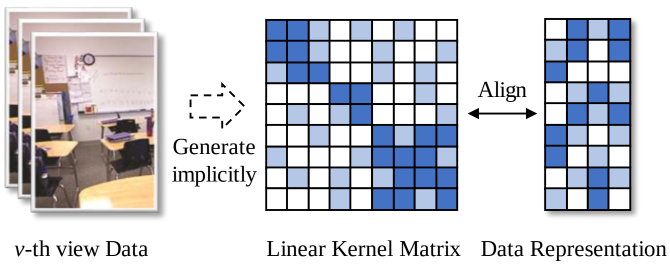

Fig. 2. The generation of data representation on the v-th client. Note that, the linear kernel matrix is generated implicitly to prevent from high computation complexity, hence is represented with dashed arrow.

Similarly, $\mathbf{Y} \in \{0,1\}^{N\times k}$ is the one-hot encoding of label $\mathbf{y}$ and k is the number of clusters. Also, the implementation of $\ell_S(\cdot)$ is the same to that of $\ell_v(\cdot)$ in Eq. (6), where the L_2-norm of their difference is the mostly used in literature. Nevertheless, the server can adopt the feasible $\ell_S(\cdot)$ without considering the clients, increasing flexibility of the CeFMC framework as well.

Alternate Update. By executing the representation generation, representation approximation and label integration only once, the obtained label $\mathbf{y}$ is not optimal, since the received $\mathbf{y}_v$ and $\mathbf{C}_v$ of the v-th client are generated only depending on its data but lacking the guidance from the integrated information of server S. Therefore, the CeFMC framework proposes to send the label $\mathbf{y}$ and centroids $\mathbf{C}$ back to clients for better generation of $\mathbf{y}_v$ and $\mathbf{C}_v$. To execute the above processes alternately, the CeFMC framework equals to minimizing one of the following unified objective functions.

$$\begin{aligned} \min_{\Theta_v,\mathbf{Y}_v,\mathbf{C}_v,\mathbf{Y},\mathbf{C}} \sum_{v=1}^{V} & [\ell_v\left(\mathcal{G}_{\Theta_v}(\mathbf{X}_v), \mathbf{Y}_v\mathbf{C}_v\right) \\ & + \lambda \ell_S(\mathbf{Y}_v\mathbf{C}_v, \mathbf{Y}\mathbf{C})], \end{aligned} \tag{8}$$

or

$$\begin{aligned} \min_{\mathbf{H}_v,\mathbf{Y}_v,\mathbf{C}_v,\mathbf{Y},\mathbf{C}} \sum_{v=1}^{V} & [\mathcal{G}(\mathbf{X}_v, \mathbf{H}_v) + \lambda \ell_v(\mathbf{H}_v, \mathbf{Y}_v\mathbf{C}_v) \\ & + \beta \ell_S(\mathbf{Y}_v\mathbf{C}_v, \mathbf{Y}\mathbf{C})], \end{aligned} \tag{9}$$

where λ and β are trade-off parameters and the former objective is used when generating the data representations explicitly with Eq. (4), while the latter is adopted when generating the data representations implicitly with Eq. (5).

B. The example method

In order to group the distributed multi-view data in practice, the proposed CeFMC framework is implemented by instancing the four components, including the representation generation, representation approximation, label integration and alternate update in the following.

To deal with large-scale data, existing federated multi-view clustering approaches adopt the matrix factorization and neural network encoding techniques to generate the data representations on clients, reducing the computation complexity. Besides, anchor graph is another feasible solution which is widely explored in the multi-view clustering literature. However, these two techniques fail to utilize the pairwise similarity information of data samples sufficiently, preventing from further improvement of clustering performance. To address this issue, we construct a linear kernel matrix implicitly on each client and compute corresponding data representation subsequently, as shown in Fig. 2. In specific, with given the v-th view data $\mathbf{X}_v$, the linear kernel can be represented with

$$\mathbf{K}_v = \mathbf{X}_v\mathbf{X}_v^\top, \tag{10}$$

where $\mathbf{K}_v$ is of size $N \times N$ and its (i,j)-th element is the similarity between the i-th sample and the j-th sample. Here, the linear kernel function is adopted for achieving linear computation complexity in optimization. Also, other kernel functions, such as Gaussian kernel, are compatible but higher computation complexity is required simultaneously. By following the kernel alignment theory [40], we can align kernel matrix $\mathbf{K}_v$ with data representation $\mathbf{H}_v$ by maximizing

$$\max_{\mathbf{H}_v} \frac{< \mathbf{K}_v,\ \mathbf{H}_v\mathbf{H}_v^\top >}{< \mathbf{K}_v,\ \mathbf{K}_v >< \mathbf{H}_v\mathbf{H}_v^\top,\ \mathbf{H}_v\mathbf{H}_v^\top >}, \tag{11}$$

where $< \cdot, \cdot >$ refers to the inner-product of two matrices. As for the feasible domain of data representation $\mathbf{H}_v$, we expect each of its columns to be discriminative to the others, which can be formulated by setting it orthogonal [41], i.e.

$$\mathbf{H}_v^\top\mathbf{H}_v = \mathbf{I}_k. \tag{12}$$

For simplicity, we set the dimensions of $\{\mathbf{H}_v\}_{v=1}^V$ to $N \times k$ universally. With introducing Eq. (12) into Eq. (11), the data representation can be generated by maximizing

$$\max_{\mathbf{H}_v} \operatorname{Tr}(\mathbf{K}_v\mathbf{H}_v\mathbf{H}_v^\top), \quad s.t.\ \mathbf{H}_v^\top\mathbf{H}_v = \mathbf{I}_k. \tag{13}$$

which is the instance of Eq. (5) by incorporating the linear kernel function to consider the pairwise similarities among data samples.

As for the approximation of data representation, we propose to instance Eq. (6) by minimizing the difference between $\mathbf{H}_v$ and $\mathbf{Y}_v\mathbf{C}_v$ with L_2-norm and obtain

$$\min_{\mathbf{Y}_v} \|\mathbf{H}_v - \mathbf{Y}_v\mathbf{C}_v\|_F^2. \tag{14}$$

Note that, centroid matrix $\mathbf{C}_v$ is fixed to the initial value and does not participate in the subsequent optimization for a smaller computation overhead. In addition, the size of label one-hot encoding $\mathbf{Y}_v$ is set to $N \times k$ with respect to all data views and centroid matrix $\mathbf{C}_v$ is of size $k \times k$. Since data representation $\mathbf{H}_v$ is required to be orthogonal in Eq. (12), the above formulation can be transformed to

$$\max_{\mathbf{Y}_v} \operatorname{Tr}(\mathbf{H}_v^\top\mathbf{Y}_v\mathbf{C}_v). \tag{15}$$

Nevertheless, when received all pseudo-labels and centroids from clients, the server first recovers the corresponding data representations by simply multiplying them. Subsequently, the obtained data representations are concatenated into a unified one and the multi-view information is aggregated by minimizing its difference from the consensus data representation. To be concrete, the formulation can be written as

$$\min_{\mathbf{Y},\mathbf{C}} \|\mathbf{H}_c - \mathbf{Y}\mathbf{C}\|_F^2, \quad s.t.\ \mathbf{C}\mathbf{C}^\top = \mathbf{I}_k. \tag{16}$$

in which $\mathbf{H}_c$ is the horizontal concatenation, i.e.

$$\mathbf{H}_c = [\mathbf{Y}_1\mathbf{C}_1, \mathbf{Y}_2\mathbf{C}_2, \cdots, \mathbf{Y}_V\mathbf{C}_V]. \tag{17}$$

At the same time, consensus label one-hot encoding $\mathbf{Y}$ and centroid matrix $\mathbf{C}$ are of size $N \times k$ and $k \times (Vk)$, respectively. Similar to Eq. (14), the above formulation can be transformed to

$$\max_{\mathbf{Y},\mathbf{C}} \ \mathrm{Tr}(\mathbf{H}_c^\top \mathbf{Y}\mathbf{C}), \quad s.t.\ \mathbf{C}\mathbf{C}^\top = \mathbf{I}_k. \tag{18}$$

To achieve a promising consensus label $\mathbf{y}$, we can execute the above processes alternately by unifying Eq. (13), (15) and (18). Consequently, the CeFMC framework of Eq. (9) is instanced into

$$\begin{aligned} \max_{\mathbf{H}_v,\mathbf{Y}_v,\mathbf{Y},\mathbf{C}} \ & \sum_{v=1}^{V} \left[\mathrm{Tr}(\mathbf{K}_v\mathbf{H}_v\mathbf{H}_v^\top) + \lambda \mathrm{Tr}(\mathbf{H}_v^\top \mathbf{Y}_v\mathbf{C}_v)\right] \\ & + \beta \mathrm{Tr}(\mathbf{H}_c^\top \mathbf{Y}\mathbf{C}) \\ s.t.\ & \mathbf{H}_v^\top \mathbf{H}_v = \mathbf{I}_k,\ \mathbf{C}\mathbf{C}^\top = \mathbf{I}_k, \mathbf{Y}_v \in \{0,1\}^{N\times k}, \\ & \mathbf{Y} \in \{0,1\}^{N\times k}, \end{aligned} \tag{19}$$

in which λ and β are the trade-off parameters of representation approximation and label integration. Note that, it is recommended setting them to the same value for better practicality and more details can be found in Section V-G.

C. Optimization

To solve the objective in Eq. (19), we design an alternate optimization procedure of linear complexity. Specifically, it can be separated into three parts: variable initialization, client computation and server computation. Meanwhile, the complete optimization procedure is summarized in Alg. 2.

1) Variable initialization: At the beginning of optimization, the variables are supposed to be initialized. For data representation $\mathbf{H}_v^{(0)}$ on the v-th client, it is set to the left singular vectors of data $\mathbf{X}_v$ corresponding to the k largest singular values,

$$\mathbf{H}_v^{(0)} = \mathbf{U}(:, 1:k), \tag{20}$$

with $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \mathbf{X}_v$ being the Singular Vector Decomposition (SVD), meeting the orthogonal requirement of data representations. Then, the obtained data representations of all clients are sent to the server and concatenated horizontally to a unified one, obtaining

$$\mathbf{H}_c^{(0)} = [\mathbf{H}_1^{(0)}, \mathbf{H}_2^{(0)}, \cdots, \mathbf{H}_V^{(0)}]. \tag{21}$$

Next, the server performs the widely used k-means on $\mathbf{H}_c^{(0)}$ to compute the consensus label $\mathbf{Y}^{(0)}$ and clustering centroid $\hat{\mathbf{C}}^{(0)}$. In this way, the consensus centroid matrix $\mathbf{C}^{(0)}$ is obtained by orthogonalizing $\hat{\mathbf{C}^{(0)}}$ with

$$\mathbf{C}^{(0)} = \mathbf{U}^{(0)}\mathbf{V}^{(0)\top}, \tag{22}$$

where $\mathbf{U}^{(0)}\mathbf{\Sigma}^{(0)}\mathbf{V}^{(0)\top} = \hat{\mathbf{C}}^{(0)}$ is the SVD correspondingly. With segmenting $\mathbf{C}^0$ into V parts, i.e.

$$[\bar{\mathbf{C}}_1^{(0)}, \bar{\mathbf{C}}_2^{(0)}, \cdots, \bar{\mathbf{C}}_V^{(0)}] = \mathbf{C}^{(0)}, \tag{23}$$

Algorithm 1 Optimization of $\mathbf{H}_v$-subproblem

Input: data observation $\mathbf{X}_v$, centroid matrix $\mathbf{C}_v$ and initial data representation $\mathbf{H}_v$
Output: data representation $\mathbf{H}_v$

1: ensure $\mathbf{H}_v\mathbf{H}_v^\top = \mathbf{I}_k$;
2: **repeat**
3: compute $\mathbf{M} = \mathbf{X}_v\mathbf{X}_v^\top\mathbf{H}_v + \lambda\mathbf{Y}_v\mathbf{C}_v$;
4: perform SVD on $\mathbf{M}$, i.e. $\mathbf{U}\Sigma\mathbf{V}^\top = \mathbf{M}$;
5: update $\mathbf{H}_v = \mathbf{U}\mathbf{V}^\top$;
6: **until** convergent

and transmitting them to each client respectively, the pseudo-labels and centroid matrices of the clients can be set to

$$\begin{aligned} &\mathbf{Y}_1^{(0)} = \mathbf{Y}_2^{(0)} = \cdots = \mathbf{Y}_V^{(0)} = \mathbf{Y}^{(0)}, \\ &\mathbf{C}_1^{(0)} = \bar{\mathbf{C}}_1^{(0)}, \mathbf{C}_2^{(0)} = \bar{\mathbf{C}}_2^{(0)}, \cdots, \mathbf{C}_V^{(0)} = \bar{\mathbf{C}}_V^{(0)}. \end{aligned} \tag{24}$$

2) Client computation: In the v-th client, data representation $\mathbf{H}_v$ and pseudo-label $\mathbf{Y}_v$ are optimized alternately, while the local objective value is computed accordingly.

$\mathbf{H}_v$-subproblem. With fixing pseudo-label $\mathbf{Y}_v$, the objective of Eq. (19) can be written to

$$\begin{aligned} \max_{\mathbf{H}_v} \ & \mathrm{Tr}(\mathbf{H}_v^\top \mathbf{X}_v\mathbf{X}_v^\top \mathbf{H}_v) + \lambda \mathrm{Tr}(\mathbf{H}_v^\top \mathbf{Y}_v\mathbf{C}_v) \\ s.t.\ & \mathbf{H}_v^\top \mathbf{H}_v = \mathbf{I}_k \end{aligned} \tag{25}$$

where $\mathbf{K}_v$ is replaced with $\mathbf{X}_v\mathbf{X}_v^\top$, since it is not required to be computed explicitly. It can be observed that Eq. (25) is a quadratic optimization problem on the Stiefel manifold [42] and can be solved by Alg. 1 in linear complexity to the number of data samples.

$\mathbf{Y}_v$-subproblem. With fixing data representation $\mathbf{H}_v$, the objective of Eq. (19) can be written to

$$\max_{\mathbf{Y}_v} \ \mathrm{Tr}(\mathbf{H}_v^\top \mathbf{Y}_v\mathbf{C}_v) + \mathrm{Tr}(\mathbf{H}_c^\top \mathbf{Y}\bar{\mathbf{C}}_v), \tag{26}$$

in which $\bar{\mathbf{C}}_v$ is the v-th horizontal segment of consensus centroid matrix $\mathbf{C}$, comprehensively shown as

$$[\bar{\mathbf{C}}_1, \bar{\mathbf{C}}_2, \cdots, \bar{\mathbf{C}}_V] = \mathbf{C}. \tag{27}$$

Nevertheless, by introducing Eq. (17), Eq. (26) is transformed to

$$\max_{\mathbf{Y}_v} \ \mathrm{Tr}(\mathbf{Y}_v^\top \mathbf{S}), \tag{28}$$

with $\mathbf{S} = \mathbf{H}_v\mathbf{C}_v^\top + \mathbf{Y}\bar{\mathbf{C}}_v\mathbf{C}_v^\top$. Denoting $\mathbf{Y}_v(i,:)$ and $\mathbf{S}(i,:)$ to the i-th row of $\mathbf{Y}_v$ and $\mathbf{S}$, the solution is

$$\mathbf{Y}_v(i,:) = \mathbf{e}_j, \quad s.t.\ j = \arg\max \mathbf{S}(i,:), \tag{29}$$

where $\mathbf{e}_j$ represents the one-hot encoding of constant j.

Local objective. To check if the alternate optimization procedure converges, the v-th client is expected to compute its local objective with the obtained $\mathbf{H}_v$ and $\mathbf{Y}_v$ as

$$obj_v = \mathrm{Tr}(\mathbf{H}_v^\top \mathbf{X}_v\mathbf{X}_v^\top \mathbf{H}_v) + \lambda \mathrm{Tr}(\mathbf{H}_v^\top \mathbf{Y}_v\mathbf{C}_v), \tag{30}$$

which is sent to the server subsequently.

Algorithm 2 Optimization of the CeFMC method

Input: multi-view data $\{\mathbf{X}_v\}_{v=1}^V$, cluster number k, trade-off parameter λ
Output: cluster assignment $\mathbf{y}$

- - - - - - - - - - - - - - - - v**-th Client** C_v - - - - - - - - - - - - - - -

```
# Varible initialization
initialize H_v^(0) with Eq. (20);
send H_v^(0) to Server S;
receive y^(0) and C̄_v^(0) from Server S;
initialize Y_v^(0) and C_v^(0);
# Client computation
t = 1;
while not convergent do
    update data representation H_v with Alg. 1;
    update pseudo-label Y_v with Eq. (29);
    compute the local objective value obj_v^t in Eq. (30);
    send y_v and obj_v^t to Server S;
    receive y and C̄_v from Server S;
    t = t + 1;
end while
```

- - - - - - - - - - - - - - - - **Server** S - - - - - - - - - - - - - - - - -

```
# Varible initialization
receive {H_v^(0)}_{v=1}^V from all clients;
compute Y^(0) and Ĉ^(0) by k-means;
initialize C^(0) with Eq. (22);
send y^(0) and C̄_v^(0) to clients;
# Server computation
t = 1;
while not convergent do
    receive {y_v}_{v=1}^V and {obj_v^t}_{v=1}^V from clients;
    update consensus label Y with Eq. (32);
    update consensus centroids C with Eq. (34);
    send y and {C̄_v}_{v=1}^V to clients;
    compute the overall objective value obj^t with Eq. (35);
    check the convergence condition:
    obj^t − obj^{t−1}/obj^t ≤ δ
    t = t + 1;
end while
output the cluster assignment y;
```

3) Server computation: In the server, consensus label one-hot encoding $\mathbf{Y}$ and centroid matrix $\mathbf{C}$ are optimized alternately, while the overall objective value is computed accordingly.

$\mathbf{Y}$**-subproblem**. With fixing consensus centroid matrix $\mathbf{C}$, the objective of Eq. (19) can be written to

$$\max_{\mathbf{Y}} \ \mathrm{Tr}(\mathbf{Y}^\top \mathbf{Q}), \tag{31}$$

with $\mathbf{Q} = \mathbf{H}_c\mathbf{C}^\top$. Similar to Eq. (28), we can denote $\mathbf{Y}(i,:)$ and $\mathbf{Q}(i,:)$ are the i-th rows of $\mathbf{Y}$ and $\mathbf{Q}$, respectively, and the solution should be

$$\mathbf{Y}(i,:) = \mathbf{e}_j, \quad s.t. \ j = \arg\max \mathbf{Q}(i,:), \tag{32}$$

where $\mathbf{e}_j$ represents the one-hot encoding of constant j.

$\mathbf{C}$**-subproblem**. By fixing consensus label $\mathbf{Y}$, the objective of Eq. (19) can be written to

$$\max_{\mathbf{Y}} \ \mathrm{Tr}(\mathbf{CP}), \quad s.t. \ \mathbf{CC}^\top = \mathbf{I}_k, \tag{33}$$

with $\mathbf{P} = \mathbf{H}_c^\top\mathbf{Y}$. According to Theorem 1 of [43], the solution of Eq. (33) can be obtained by

$$\mathbf{C} = \mathbf{VU}^\top, \tag{34}$$

in which $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors of $\mathbf{P}$ with the SVD being $\mathbf{U\Sigma V}^\top = \mathbf{P}$.

Overall objective. To check if the alternate optimization procedure converges, the server is supposed to compute the overall objective with the fixed $\mathbf{Y}$, $\mathbf{C}$ and the received $\{obj_v\}_{v=1}^V$ via

$$obj = \beta\mathrm{Tr}(\mathbf{H}_c^\top\mathbf{YC}) + \sum_{v=1}^{V} obj_v. \tag{35}$$

D. Computation complexity

The computation complexity of the CeFMC method is analyzed as follows. Primarily, it is well known that SVD on a matrix of size $n \times m$ takes $\mathcal{O}(nm^2)$.

Variable initialization. $\{\mathbf{H}_v^{(0)}\}_{v=1}^V$ is computed with V independent SVD, requiring a $\mathcal{O}((\sum_{v=1}^V D_v)N)$ complexity, while $\mathbf{Y}^{(0)}$ and $\hat{\mathbf{C}}^{(0)}$ are computed with k-means, which is of $\mathcal{O}(Vk^2N)$ complexity. Also, the subsequent computation of $\mathbf{C}^{(0)}$ with SVD requires a $\mathcal{O}(V^2k^2)$ complexity.

Client computation. In the v-th client, $\mathbf{H}_v$ is calculated via Alg. 1. Here, $\mathbf{X}_v\mathbf{X}_v^\top\mathbf{H}_v$ equals to $\mathbf{X}_v(\mathbf{X}_v^\top\mathbf{H}_v)$ of $\mathcal{O}(kD_vN)$ complexity, while $\mathbf{Y}_v\mathbf{C}_v$ can be efficiently computed by *index* operation, i.e. $\mathbf{C}_v(\mathbf{y}_v,:)$ in which $\mathbf{y}_v$ is the corresponding label vector of one-hot encoding $\mathbf{Y}_v$. In this way, the computation of $\mathbf{M}$ only requires $\mathcal{O}(kD_vN)$. Also, the SVD on $\mathbf{M}$ is of $\mathcal{O}(k^2N)$ complexity. Assuming Alg. 1 needs t_1 iterations to converge, the computation complexity of $\mathbf{H}_v$ is $\mathcal{O}(t_1kD_vN + t_1k^2N)$. Nevertheless, the complexity of $\mathbf{Y}_v$ is mainly on the computation of $\mathbf{S}$ which requires a $\mathcal{O}(k^2N + k^3)$ complexity ($\mathbf{Y}\bar{\mathbf{C}}_v\mathbf{C}_v^\top$ equals to $\mathbf{Y}(\bar{\mathbf{C}}_v\mathbf{C}_v^\top)$ with the *index* operation). Moreover, in the calculation of obj_v, $\mathbf{H}_v^\top\mathbf{X}_v\mathbf{X}_v^\top\mathbf{H}_v$ can be computed by $(\mathbf{H}_v^\top\mathbf{X}_v)(\mathbf{X}_v^\top\mathbf{H}_v)$ with a $\mathcal{O}(kD_vN + k^2D_v)$ complexity, while $\mathbf{H}_v^\top\mathbf{Y}_v\mathbf{C}_v$ with a $\mathcal{O}(k^2N + k^2D_v)$ complexity. Taking V clients into account, the overall complexity of client computation is of $\mathcal{O}(t_1k(\sum_{v=1}^V D_v)N + t_1Vk^2N)$.

Server computation. To compute $\mathbf{Y}$, the complexity is mainly on the calculation of $\mathbf{Q}$ in which $\mathbf{H}_c$ is constructed by Eq. (17) with *index* and concatenation operators and $\mathbf{H}_c\mathbf{C}^\top$ requires a $\mathcal{O}(Vk^2N)$. Meanwhile, to compute $\mathbf{C}$, $\mathbf{P}$ can be calculated with the $index$ and sum operators and the SVD on $\mathbf{P}$ requires a $\mathcal{O}(Vk^3)$ complexity. Nevertheless, to compute the overall objective, $\mathbf{H}_c^\top\mathbf{YC}$ is of $\mathcal{O}(Vk^2N)$ complexity.

In summary, assuming that the optimization procedure converges at the t-th iteration, corresponding complexity is $\mathcal{O}(tt_1k(\sum_{v=1}^V D_v)N + tt_1Vk^2N)$, reducing to $\mathcal{O}(N)$, i.e. linear to the number of data samples.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2025.3601533

E. Communication overhead

Different from the centralized learning paradigm, federated learning is often limited by the bottleneck of transmitting data among clients and server when building the target model, since the network availability, stability and bandwidth are hard guaranteed in most cases. Therefore, a low communication overhead always induces to the desired practicality.

To analyze the communication overhead[1] of the CeFMC method (Alg. 2), we can separate it into two stages, i.e. initialization and clustering. In the former, the v-th client needs to send data representation $\mathbf{H}_v^{(0)}$ to the server, requiring transmitting kN *float*s, while receives $\mathbf{y}_v^{(0)}$ and $\bar{\mathbf{C}}_v^{(0)}$, requiring transmitting N *uint*s and k^2 *float*s. In the latter, assuming the optimization procedure converges at the t-th iteration, the v-th client needs to send $\mathbf{y}_v$ and obj_v^t to the server, requiring transmitting tN *uint*s and t *float*s, while receive $\mathbf{y}$ and $\bar{\mathbf{C}}_v$, requiring transmitting tN *uint*s and tk^2 *float*s. To be summarized, the communication overhead of CeFMC method is $(k+1)N+(t+1)k^2+t$ *float*s and $(2t+1)N$ *uint*s between each client and the server. In literature, the existing federated multi-view clustering methods directly transmit the data representations between client and server, requiring at least $2tkN$ *float*s. In comparison, the proposed CeFMC method saves multiple times of communication overheads, making it more practical in real-world applications. By the way, please find corresponding empirical results in Section V-D.

F. Convergence

To solve the maximization problem of Eq. (19), an alternate optimization procedure is proposed in Alg. 2 and is convergent with theoretical guarantee.

For simplicity of expression, the objective can be annotated to

$$\max_{\mathbf{H}_v,\mathbf{Y}_v,\mathbf{Y},\mathbf{C}} L\left(\{\mathbf{H}_v\}_{v=1}^V, \{\mathbf{Y}_v\}_{v=1}^V, \mathbf{Y}, \mathbf{C}\right). \tag{36}$$

Also, we denote $L(\{\mathbf{H}_v^{(t)}\}_{v=1}^V, \{\mathbf{Y}_v^{(t)}\}_{v=1}^V, \mathbf{Y}^{(t)}, \mathbf{C}^{(t)})$ to the objective value at the t-th iteration. In such setting, the optimization procedure proposes to alternately optimize one of the variables while keeping the others fixed at the $t+1$ iteration, inducing to

$$\begin{aligned}
&L\left(\{\mathbf{H}_v^{(t)}\}_{v=1}^V, \{\mathbf{Y}_v^{(t)}\}_{v=1}^V, \mathbf{Y}^{(t)}, \mathbf{C}^{(t)}\right)\\
&\leq L\left(\{\mathbf{H}_v^{(t+1)}\}_{v=1}^V, \{\mathbf{Y}_v^{(t)}\}_{v=1}^V, \mathbf{Y}^{(t)}, \mathbf{C}^{(t)}\right)\\
&\cdots\\
&\leq L\left(\{\mathbf{H}_v^{(t+1)}\}_{v=1}^V, \{\mathbf{Y}_v^{(t+1)}\}_{v=1}^V, \mathbf{Y}^{(t+1)}, \mathbf{C}^{(t+1)}\right),
\end{aligned} \tag{37}$$

which can be simplified to $L^{(t)} \leq L^{(t+1)}$ by denoting $L^{(t)}$ to the objective value at the end of t-th iteration. Meanwhile, the objective in Eq. (19) can be separated into three parts, i.e. $\sum_{v=1}^V \mathrm{Tr}(\mathbf{K}_v\mathbf{H}_v\mathbf{H}_v^\top)$, $\sum_{v=1}^V \mathrm{Tr}(\mathbf{H}_v^\top\mathbf{Y}_v\mathbf{C}_v)$ and $\mathrm{Tr}(\mathbf{H}_c^\top\mathbf{Y}\mathbf{C})$, which are analyzed as follows:

1) Since $\{\mathbf{K}_v\}_{v=1}^V$ are fixed for the given multi-view data, we assume each of their elements no bigger than a constant δ. Meanwhile, due to $\mathbf{H}_v^\top\mathbf{H}_v = \mathbf{I}_k$, it holds that $\mathbf{H}_v(:,j)^\top\mathbf{H}_v(:,j) = 1$, in which $\mathbf{H}_v(:,j)$ represents the j-th column of matrix $\mathbf{H}_v$. This illustrates that each element of $\mathbf{H}_v$ is in range $[-1,1]$. Thus,
$$\sum_{v=1}^V \mathrm{Tr}(\mathbf{K}_v\mathbf{H}_v\mathbf{H}_v^\top) = \sum_{v=1}^V \mathrm{Tr}\left[(\mathbf{H}_v^\top\mathbf{K}_v)\mathbf{H}_v\right] \leq VkN^2\delta. \tag{38}$$
2) It can be seen from the first point that each element of $\mathbf{H}_v$ is in range $[-1,1]$. Similarly, each element of $\mathbf{C}_v$ is also in range $[-1,1]$. Combining the constraint $\mathbf{Y}_v \in \{0,1\}^{N\times k}$, the following holds that
$$\sum_{v=1}^V \mathrm{Tr}(\mathbf{H}_v^\top\mathbf{Y}_v\mathbf{C}_v) \leq Vk^2. \tag{39}$$
3) Since $\mathbf{H}_c = [\mathbf{Y}_1\mathbf{C}_1, \mathbf{Y}_2\mathbf{C}_2, \cdots, \mathbf{Y}_V\mathbf{C}_V]$, each element of $\mathbf{H}_c$ is in range $[-1,1]$. Also, $\mathbf{C}\mathbf{C}^\top = \mathbf{I}_k$, inducing that each element of $\mathbf{C}$ is also in range $[-1,1]$. Therefore,
$$\mathrm{Tr}(\mathbf{H}_c^\top\mathbf{Y}\mathbf{C}) \leq Vk^2. \tag{40}$$

By utilizing the above equations, the objective value of Eq. (19) is no bigger than $VkN^2\delta + (\lambda+\beta)Vk^2$.

To be summarized, the objective value monotonically increases in iteration and is upper bounded, therefore is convergent theoretically.

G. Privacy requirement

Since the proposed method is an instance of the proposed framework, its privacy requirement follows the similar rules in Section III. Concretely, in the building of clustering model, the clients are forbidden from transmitting their raw multi-view data $\mathbf{X}_v$ to each other or the server, while only the pseudo-label $\mathbf{y}_v$ and centroid matrix $\mathbf{C}_v$ are shared. Meanwhile, the shared pseudo-label and centroid matrix should be protected with encryption by default. Also, one may infer data privacies illegally by building the mapping that

$$\mathbf{X}_v = f_{\Theta_v^{-1}}(\mathbf{H}_v), \tag{41}$$

where $f_{\Theta_v^{-1}}(\cdot)$ is the inverse model. Therefore, the data features of $\mathbf{X}_v$ are also expected to keep private. In addition, the generated representation $\mathbf{H}_v$ is recommended to do so but not compulsively.

V. Experiment

In this section, we first introduce the experiment setting, then provide and analyze the results of CeFMC method from four aspects, including effectiveness, computation efficiency, communication efficiency and convergence.

A. Experiment setting

In experiment, the proposed CeFMC method is tested on seven popular benchmark datasets of Table II. Also, they are briefly introduced as follows:

1) ORL[2] [44]. It collects 400 face images of 40 distinct subjects. In experiment, the images are resized to 48×48

[1] In the following, *float* refers to the floating point number, while *uint* to the unsigned integer for brevity.

[2] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

TABLE II
DETAILS OF THE USED BENCHMARK DATASETS.

Dataset	Number of Samples	Number of Views	Number of Clusters
ORL	400	3	40
Flower17	1360	7	17
HW	2000	6	10
BDGP	2500	3	5
RGBD	10335	2	45
DryBean	13611	2	7
AwA	30475	6	50

and three types of features are extracted, including the 4096-D intensity, 3304-D LBP and 6750-D Gabor features.

2) Flower17[3] [45]. It is a flower dataset of 17 categories with 80 images for each class. Seven features are extracted, including 5376-D Color Histogram, 512-D GIST, 5376-D HOG (2×2), 5376-D HOG (3×3), 1239-D LBP, 5376-D SIFT and 5376-D SSIM features.
3) HW[4] [46]. It is composed of 2,000 data points from 0 to 9 ten digit classes with each of 200 data points. In experiment, six features are adopted, i.e. 216-D profile correlations, 76-D Fourier coefficients of the character shapes, 64-D Karhunen-Love coefficients, 6-D morphological features, 240-D pixel averages in 2×3 windows and 47-D Zernike moments.
4) BDGP[5] [47]. It collects 2500 drosophila embryo images of 5 categories. In experiment, SIFT features from three different perspectives, i.e. lateral, dorsal and ventral, are extracted with the dimensions to be 1000-D, 500-D and 250-D, respectively.
5) SUNRGBD[6] [48] (abbr. RGBD). It consists of 10335 RGB and depth image pairs for recognition of both objects and room layouts. In experiment, two data features of both 4096-D are extracted with deep neural networks on the RGB and depth images, respectively.
6) DryBean[7] [49]. It collects 13611 grain images of 7 different registered dry beans taken with a high-resolution camera. In experiment, two features, i.e. 12-D dimensions and 4-D shape forms, are extracted from the images.
7) AwA[8] [50] It contains 30475 images of 50 animal classes with six extracted features, including 2688-D Color Histogram, 2000-D local self-similarity, 252-D PHOG, 2000-D SIFT, 2000-D color SIFT and 2000-D SURF features.

Nevertheless, the proposed CeFMC method is empirically compared with existing federated multi-view clustering methods, including

1) FedMVL [8]. It proposes a novel federated multi-view clustering framework by performing orthogonal matrix factorization on each client and incorporating the coefficient matrices on the server.
2) FedCMv and FedFCMv [7]. They are, respectively, developed by extending the c-means and fuzzy c-means clustering algorithms in the federated learning setting where the data are vertically partitioned on clients.

Note that, only these three methods are considered in experiment, since federated multi-view clustering is an emerging research interest and they are the only available and runnable solutions after detailed search and debugging. Meanwhile, the other methods have the following limitations, preventing them from comparison in exepriments.

1) FMVC-IMK [14] and FMVFCMSP [9]. Their codes are not released publicly. In addition, it is unfair to compare CeFMC with FMVC-IMK, since the former is of linear complexity to sample number while the latter of square complexity. Nevertheless, FMVFCMSP inadvertently uses the data labels when incorporating tensor Schatten-p norm and thus violates the unsupervised setting of clustering, which is discussed in [51] thoroughly.
2) FIM-GNNs [13], FedDMVC [12] and FMCSC [11]. They consider different data settings from the proposed CeFMC method. Specifically, FIM-GNNs and FedDMVC focus on the incomplete setting of distributed multi-view data, while FMCSC concentrates on the consequence where some of clients hold single-view data and the others hold multi-view data. Also, the code of FIM-GNNs is not available publicly.
3) FedMVFPC [36]. Different from the CeFMC setting where each client holds the data observations of one view but all data samples, FedMVFPC considers the setting in which each client holds the data observations of all views but partial data samples.

Moreover, to sufficiently illustrate the superiority of CeFMC, we also compare CeFMC with nine more competitive **centralized** large-scale multi-view clustering approaches of linear complexity, i.e.

1) MNMF [52]. It computes the soft-label matrices on each view with None-negative Matrix Factorization (NMF) and pushes them toward a common consensus in the iterative optimization procedure instead of fixing it directly.
2) RMKC [53]. It proposes to remove the effect of data outliers by incorporating the structured sparsity-inducing $L_{2,1}$-norm and can handle with large-scale data by designing the non-smooth norm based loss function with proved convergence.
3) BMVC [3]. It combines the compact discrete representation learning and binary clustering structure learning in a joint framework by encoding the multi-view image descriptors into a compact common binary code space.
4) LMSuC [26]. By introducing the anchor graph technique, it first learns a smaller graph for each view and

[3]https://www.robots.ox.ac.uk/∼vgg/data/flowers/17/
[4]https://archive.ics.uci.edu/ml/datasets/Multiple+Features/
[5]https://www.fruitfly.org/
[6]http://rgbd.cs.princeton.edu/
[7]https://archive.ics.uci.edu/dataset/602/dry+bean+dataset
[8]https://cvml.ist.ac.at/AwA/

integrates them with spectral clustering, improving the clustering efficiency significantly.

5) OPMC [43]. It proposes a matrix tri-factorization method with the consensus discrete label matrix on each data view to integrate the complementary information.
6) EOMSC [54]. It combines anchor learning and graph construction into a unified framework and imposes a graph connectivity constraint to produce the discrete label directly.
7) TBGL [55]. It proposes a variance based decorrelation anchor selection strategy for bipartite construction and exploits the similarity of inter-view by minimizing the tensor Schatten p-norm.
8) FastMICE [56]. It proposes a fast multi-view clustering approach by capturing the versatile view-wise relationships and designing the hybrid early-late fusion strategy to enable efficient multi-stage data information fusion.
9) UDBGL [57]. It proposes an efficient multi-view clustering approach via learning the unified and discrete bipartite graphs and develops an efficient optimization algorithm of linear complexity accordingly.

To produce the experiment results, we directly run the codes publicly available at the authors' websites without further revision for reproduction. In addition, we grid-search parameters in the range recommended in corresponding papers and report the best. So does the proposed CeFMC method by setting parameter λ and β in $[2^{-10}, 2^{-9}, \cdots 2^{10}]$. Note that, we set $\lambda = \beta$ by default for better practicality and more details are provided in Section V-G. By following the multi-view clustering literature, three widely-used metrics, i.e. Accuracy (ACC), Normalized Mutual Information (NMI) and Purity, are adopted to evaluate the clustering performance. Furthermore, the methods are executed multiple times to remove randomness and their averages with standard variance are presented. Additionally, *OT*, abbr. *Out of Time*, indicates that the result is not obtained before the maximal time limit in the following tables.

TABLE III
PERFORMANCE COMPARISON BETWEEN CEFMC AND EXISTING FEDERATED MULTI-VIEW CLUSTERING APPROACHES.

Dataset	FedCMv	FedFCMv	FedMVL	CeFMC	Gap
			ACC		
ORL	08.55± 0.9	25.75± 0.0	64.45± 4.5	**72.65**± 3.9	08.20↑
Flower17	07.69± 1.7	12.94± 0.0	14.57± 0.5	**34.35**± 0.9	19.78↑
HW	24.32± 8.6	51.53± 2.2	21.05± 1.9	**94.47**± 0.4	42.94↑
BDGP	20.00± 0.0	20.89± 0.2	22.65± 0.3	**54.18**± 6.3	31.53↑
RGBD	**21.29**± 1.4	07.78± 0.1	18.94± 0.4	20.40± 0.9	00.89↓
DryBean	41.75± 0.0	55.73± 1.2	32.07± 3.1	**80.62**± 1.5	24.89↑
AwA	08.29± 0.2	04.16± 0.3	03.39± 0.0	**10.01**± 0.2	01.72↑
Average	18.84	25.54	25.30	**52.38**	26.84↑
			NMI		
ORL	29.71± 1.4	49.34± 0.0	79.51± 2.2	**85.62**± 1.9	06.11↑
Flower17	01.91± 1.7	07.96± 0.0	08.07± 0.6	**33.47**± 1.3	25.40↑
HW	31.79± 16.8	56.76± 0.3	07.54± 1.5	**88.32**± 0.5	31.56↑
BDGP	00.00± 0.0	00.50± 0.1	00.33± 0.0	**29.14**± 5.2	28.64↑
RGBD	**27.71**± 0.8	11.48± 0.0	22.92± 0.5	21.52± 0.3	06.19↓
DryBean	45.72± 0.0	50.46± 1.5	16.43± 2.4	**69.56**± 2.6	19.10↑
AwA	09.76± 0.1	01.39± 1.4	01.04± 0.0	**11.62**± 0.4	01.86↑
Average	20.94	25.41	19.41	**48.47**	23.06↑
			Purity		
ORL	08.55± 0.9	27.50± 0.0	68.85± 3.9	**76.35**± 3.2	07.50↑
Flower17	07.69± 1.7	13.68± 0.0	15.53± 0.4	**35.69**± 1.1	20.16↑
HW	24.32± 8.6	55.89± 2.1	21.86± 1.7	**94.47**± 0.4	38.58↑
BDGP	20.00± 0.0	20.94± 0.2	22.86± 0.3	**54.52**± 6.2	31.66↑
RGBD	**39.10**± 1.1	21.31± 0.2	38.28± 0.7	37.45± 0.5	01.65↓
DryBean	41.75± 0.0	57.15± 1.5	41.85± 3.4	**80.62**± 1.5	23.47↑
AwA	10.10± 0.1	04.78± 1.0	04.49± 0.0	**11.66**± 0.3	01.56↑
Average	21.64	28.75	30.53	**55.82**	25.29↑

* The *Gap* column refers to performance increase or decrease of CeFMC to the best of the others.

B. Effectiveness

To validate effectiveness of the proposed CeFMC, we compare it with the existing federated multi-view clustering methods and corresponding results are presented in Table III. It is obvious that CeFMC outperforms the others by large margins. For instance, it increases the accuracies by 08.20%, 19.78%, 42.94%, 31.53%, 24.89% and 01.72% on ORL, Flower17, HW, BDGP, DryBean ad AwA datasets, respectively. Although it achieves relatively worse performances on RGBD dataset, only slight decreases are observed, i.e. 00.89% ACC, 06.19% NMI and 01.65% purity. At the same time, the FedMVL and FedFCMv methods achieve similar results. In contrast, the FedCMv method obtains the worst performances. Specifically, it only achieves 20.00% ACC, 00.00% NMI and 20.00% purity on BDGP, degrading to random guess, since BDGP is of 5 clusters according to the ground truth. By averaging the performances on all datasets, we can see that CeFMC obtains 26.84% ACC, 23.06% NMI and 25.29% purity increases.

Apart from federated multi-view clustering methods, we also compare the proposed CeFMC with the centralized multi-view clustering approaches in experiment and present the results in Table IV. Specifically, it achieves almost the best or second-best performances on all datasets, except for the NMIs on BDGP, RGBD and AwA and the purity on RGBD. Also, it should be noted that the TBGL method cannot compute the clustering assignments on BDGP, RGBD and AwA datasets, while CeFMC not only can do this, but also achieves satisfactory performances. Meanwhile, we average the results of each approach on all datasets. It can be observed that the proposed CeFMC method achieves the best performance and improves the ACC, NMI and purity by 05.24%, 02.26% and 03.51% over the second-best, respectively. Note that, the centralized multi-view clustering methods are more conductive to achieving better results, since they are of no need to consider the consequence of transmitting information between the clients and server. Therefore, the aforementioned observations validate the effectiveness of CeFMC to a large extent.

In summary, the proposed CeFMC method not only outper-

TABLE IV
PERFORMANCE COMPARISON BETWEEN CEFMC AND CENTRALIZED MULTI-VIEW CLUSTERING APPROACHES.

Dataset	MNMF	RMKC	BMVC	LMSuC	OPMC	EOMSC	TBGL	FastMICE	UDBGL	CeFMC
					ACC					
ORL	66.95± 2.7	57.35± 3.5	57.25± 0.0	59.00± 0.0	58.95± 3.5	62.25± 0.0	57.05± 1.3	70.35± 3.1	56.50± 0.0	**72.65**± 3.9
Flower17	**35.72**± 2.2	23.31± 2.5	27.57± 0.0	31.40± 0.0	29.38± 0.6	28.09± 0.0	11.16± 1.0	28.51± 0.7	32.13± 0.0	34.35± 0.9
HW	65.12± 2.3	70.43± 3.2	86.40± 0.0	92.10± 0.0	81.48± 7.8	76.00± 0.0	75.64± 1.9	88.25± 4.4	80.95± 0.0	**94.47**± 0.4
BDGP	48.59± 0.0	49.63± 5.1	32.00± 0.0	45.00± 0.0	50.70± 0.2	42.08± 0.0	*OT*	51.61± 1.4	39.44± 0.0	**54.18**± 6.3
RGBD	18.62± 0.8	17.89± 0.8	16.39± 0.0	17.71± 0.0	19.40± 0.7	**23.70**± 0.0	*OT*	18.43± 0.9	20.06± 0.0	20.40± 0.9
DryBean	36.09± 4.3	55.24± 4.6	50.45± 0.0	70.60± 0.0	47.63± 0.1	60.23± 0.0	32.84± 4.2	64.03± 1.9	**82.93**± 0.0	80.62± 1.5
AwA	06.77± 0.1	09.01± 0.2	**10.45**± 0.0	08.18± 0.0	09.40± 0.2	08.72± 0.0	*OT*	08.79± 0.2	08.18± 0.0	10.01± 0.2
Average	39.69	40.41	40.07	46.28	42.42	43.01	/	47.14	45.74	**52.38**
					NMI					
ORL	81.11± 1.7	75.53± 1.1	72.19± 0.0	78.91± 0.0	76.32± 2.3	**88.15**± 0.0	70.15± 1.2	84.17± 1.1	73.80± 0.0	85.62± 1.9
Flower17	**34.54**± 1.5	20.87± 2.0	24.88± 0.0	29.34± 0.0	29.05± 0.5	26.53± 0.0	05.78± 1.0	28.14± 0.7	28.83± 0.0	33.47± 1.3
HW	61.48± 2.4	70.71± 3.0	84.03± 0.0	86.49± 0.0	77.94± 3.7	82.08± 0.0	70.39± 1.6	88.04± 2.9	78.83± 0.0	**88.32**± 0.5
BDGP	32.50± 0.0	25.69± 3.1	08.20± 0.0	24.57± 0.0	**35.42**± 0.2	14.59± 0.0	*OT*	33.25± 1.5	15.29± 0.0	29.14± 5.2
RGBD	23.37± 0.3	23.84± 0.4	19.22± 0.0	20.71± 0.0	**25.53**± 0.2	22.49± 0.0	*OT*	23.13± 0.9	19.82± 0.0	21.52± 0.3
DryBean	20.49± 4.8	47.16± 2.0	37.30± 0.0	57.00± 0.0	40.33± 0.0	53.23± 0.0	14.57± 3.8	56.19± 1.4	66.47± 0.0	**69.56**± 2.6
AwA	07.62± 0.1	11.16± 0.3	**12.30**± 0.0	09.03± 0.0	11.95± 0.2	10.10± 0.0	*OT*	10.56± 0.2	08.47± 0.0	11.62± 0.4
Average	37.30	39.28	36.88	43.72	42.36	42.45	/	46.21	41.64	**48.47**
					Purity					
ORL	72.10± 2.4	61.20± 2.6	60.00± 0.0	65.50± 0.0	62.55± 3.6	**92.50**± 0.0	63.45± 1.3	73.95± 2.6	62.00± 0.0	76.35± 3.2
Flower17	**36.87**± 2.3	24.68± 2.5	28.90± 0.0	33.16± 0.0	30.97± 0.3	28.75± 0.0	11.59± 1.0	30.28± 0.5	32.28± 0.0	35.69± 1.1
HW	67.69± 1.0	73.99± 2.9	86.40± 0.0	92.10± 0.0	82.82± 6.5	76.20± 0.0	76.00± 1.7	88.78± 4.2	80.95± 0.0	**94.47**± 0.4
BDGP	52.10± 0.0	50.57± 5.1	33.88± 0.0	45.96± 0.0	53.26± 0.2	42.08± 0.0	*OT*	52.91± 1.8	40.24± 0.0	**54.52**± 6.2
RGBD	38.78± 0.9	38.21± 0.8	33.28± 0.0	35.42± 0.0	**40.29**± 0.5	33.33± 0.0	*OT*	38.22± 1.0	28.12± 0.0	37.45± 0.5
DryBean	42.30± 4.2	59.56± 1.9	57.17± 0.0	72.00± 0.0	56.66± 0.2	61.65± 0.0	33.57± 4.5	71.07± 2.1	**82.93**± 0.0	80.62± 1.5
AwA	08.59± 0.1	11.06± 0.3	**12.19**± 0.0	10.03± 0.0	11.49± 0.2	09.62± 0.0	*OT*	10.93± 0.2	09.41± 0.0	11.66± 0.3
Average	45.49	45.61	44.55	50.60	48.29	49.16	/	52.31	47.99	**55.82**

* *OT*, abbr. Out of Time, indicates that the result is not obtained before the time limit.

TABLE V
TIME CONSUMPTION (IN SECONDS) COMPARISON BETWEEN CEFMC AND EXISTING FEDERATED MULTI-VIEW CLUSTERING APPROACHES.

Dataset	FedCMv	FedFCMv	FedMVL	CeFMC	Speedup
ORL	1273.7	86.4	31.1	**5.2**	6.0×-244.9×
Flower17	2299.4	241.0	81.0	**27.6**	2.9×- 83.3×
HW	75.0	9.9	41.1	**0.2**	49.5×-375.0×
BDGP	24.2	13.3	48.3	**1.4**	9.5×- 34.5×
RGBD	43204.7	1465.1	1837.1	**71.8**	20.4×-601.7×
DryBean	24.9	10.8	3444.8	**1.6**	6.8×- 2153×
AwA	198724.3	6619.2	34435.0	**690.9**	9.6×-287.6×
Avg. order	3.7	2.3	3.0	**1.0**	/

* The *Speedup* column presents the acceleration times of CeFMC to the others, while the *Order* row is the average order of each method on all datasets.

forms existing federated multi-view clustering approaches by large margins, but also obtains better performances over the centralized multi-view clustering approaches, well illustrating its effectiveness.

C. Computation efficiency

As analyzed in Section IV-D, the proposed CeFMC method is of linear complexity to the number of data samples. In the following, we validate its efficiency empirically by comparing with existing federated multi-view clustering approaches. Corresponding time consumptions are collected in Table V. It can be observed that the CeFMC method takes the least time to compute the clustering results, while FedFCMv consumes the second least time on all datasets. In statistics, we sort the time consumptions of all methods on each dataset in ascending order and present their averages at the bottom, where the CeFMC takes the first place, following by FedFCMv, FedMVL and FedCMv subsequently. Meanwhile, the speedup ranges are calculated by dividing the minimal and maximal values of the competing methods with those of CeFMC. It can be seen that the proposed CeFMC accelerates the clustering process by 2.9 times at least and 2153 times at most. These observations sufficiently demonstrate the efficiency of the proposed CeFMC over the competing methods.

Nevertheless, we also compare the time consumptions of CeFMC with those of existing centralized multi-view clustering methods, where the results are recorded in Table VI.

TABLE VI
TIME CONSUMPTION (IN SECONDS) COMPARISON BETWEEN CEFMC AND EXISTING CENTRALIZED MULTI-VIEW CLUSTERING APPROACHES.

Dataset	MNMF	RMKC	BMVC	LMSuC	OPMC	EOMSC	TBGL	FastMICE	UDBGL	CeFMC	Order
ORL	2440.5	5.7	**0.6**	9.2	134.0	5.9	26.5	3.7	18.1	5.2	3
Flower17	1847.0	6.5	**2.1**	37.7	1230.6	6.9	500.2	19.1	69.2	27.6	5
HW	31.5	11.0	3.8	13.4	23.3	1.8	1616.8	1.5	313.6	**0.2**	1
BDGP	128.1	**1.1**	1.6	5.6	116.1	3.6	*OT*	4.0	14.4	1.4	2
RGBD	6339.4	594.2	**6.8**	132.9	9105.0	83.5	*OT*	214.5	5453.5	71.8	2
DryBean	23.3	2.6	3.2	22.2	84.2	59.7	66321.1	26.6	141.1	**1.6**	1
AwA	15939.1	6830.6	**68.9**	2379.8	61798.1	341.1	*OT*	187.9	6034.4	690.9	4
Avg. order	8.3	3.9	**2.0**	5.3	8.4	4.0	9.4	3.7	7.4	2.6	2.6

* The *Order* column is the order of CeFMC on each dataset, while the *Order* row presents the order averages of each methods on all datasets.

TABLE VII
COMMUNICATION OVERHEAD (IN *MB*) COMPARISON BETWEEN CEFMC AND EXISTING FEDERATED MULTI-VIEW CLUSTERING APPROACHES.

Dataset	FedCMv	FedFCMv	FedMVL	CeFMC	Ratio
ORL	5.41	11.12	25.63	**0.5**	2.0% - 9.2%
Flower17	10.93	38.13	86.43	**1.96**	2.3% -17.9%
HW	13.46	28.84	64.09	**1.3**	2.0% - 9.7%
BDGP	3.15	9.44	20.03	**1.19**	5.9% -37.8%
RGBD	215.26	215.26	496.75	**7.77**	1.6% - 3.6%
DryBean	32.71	46.73	101.77	**1.98**	1.9% - 6.1%
AwA	2113.48	2113.48	4882.62	**79.17**	1.6% - 3.7%
Order	2.0	3.0	4.0	**1.0**	/

* The *Ratio* column is the communication overhead ratio of CeFMC to the others on each dataset, while the *Order* row presents the order averages of each method on all datasets.

Specifically, the proposed CeFMC achieves the best results on HW and DryBean datasets, as well, the second-best on BDGP and RGBD datasets. Besides, we sort the methods on all datasets according to their time consumptions in ascending order, and list the orders on the right. It can be seen that CeFMC takes at least the 5-th place. Moreover, the orders are also averaged at the bottom of Table VI. In specific, CeFMC is of 2.6, only behind the BMVC method. Note that, BMVC consists of 6 hyper-parameters, while CeFMC is only of 1 hyper-parameter and therefore more practical, since there is no validation data set to tune parameters in the unsupervised learning setting and they are mostly set by experience. Also, we should notice that the competing methods are all of linear complexity and the proposed CeFMC takes less computation time than them, validating its efficiency to a large extent.

In summary, the proposed CeFMC not only takes the lead in federated multi-view clustering methods, but also is better than most of the existing centralized multi-view clustering methods on time consumption, well illustrating its efficiency.

D. Communication efficiency

Motivated by the heavy communication overhead of existing federated multi-view clustering methods, CeFMC is proposed to promote the communication efficiency between clients and the server. In experiment, we validate this by comparing its communication overhead with those of the existing methods. Corresponding results are collected in Table VII. It is obvious that the FedMVL method consumes the most communication overheads on all datasets, while the proposed CeFMC requires the least. Meanwhile, by dividing the communication overheads of CeFMC with those of competing methods, we can obtain the communication overhead ratios on the right of Table VII. It can be seen that CeFMC only takes 1.6% overheads at most and 37.8% overheads at least of the competing methods. Besides, we sort the methods on all datasets according to their communication overheads in ascending order and compute the averages at the bottom of the table. It can be found that CeFMC takes the lead among all federated multi-view clustering approaches. These benefit from the fact that it approximates the data representation with pseudo label and centroid matrix, and only requires transmitting the latter, significantly reducing the data transmission. To be summarized, the proposed CeFMC saves a large volume of communication overhead when clustering, making it more practical in real-world applications.

E. Ablation study

To promote the communication efficiency of federated multi-view clustering, the proposed CeFMC method approximates the local data representation on clients with a pseudo label vector and a centroid matrix. To validate this, an ablation study is conducted in the following. Specifically, a new approach named CeFMC-H is developed from the CeFMC method by exchanging full data representations $\{\mathbf{H}_v\}_{v=1}^V$ between clients and the server. Note that, its details on the objective and optimization can be found in Appendix. In such case, we test it on the seven benchmark datasets and compare its clustering performances and communication overheads with those of the CeFMC method in Table VIII. It can be observed that the CeFMC method achieves extremely close clustering performances to the CeFMC-H method, i.e. only 0.03% ACC increase, 0.37% NMI decrease and 0.36% purity decrease in average. Nevertheless, the CeFMC method takes much smaller communication overheads, i.e. 50.78% at least while 9.14% at most. Therefore, these two observations well prove that approximating the local data representation with a pseudo-label vector and a centroid matrix can not only well preserve

TABLE VIII
PERFORMANCE COMPARISON OF THE CEFMC-H AND CEFMC METHODS.

Dataset	ACC			NMI			Puirty			Communication Overhead		
	CeFMC-H	CeFMC	Gap	CeFMC-H	CeFMC	Gap	CeFMC-H	CeFMC	Gap	CeFMC-H	CeFMC	Ratio
ORL	72.50± 3.2	72.65± 3.9	0.15↑	85.45± 1.6	85.62± 1.9	0.17↑	76.25± 1.4	76.35± 3.2	0.10↑	2.08	0.50	24.04%
Flower17	35.05± 1.9	34.35± 0.9	0.70↓	33.68± 1.1	33.47± 1.3	0.21↓	36.01± 1.6	35.69± 1.1	0.32↓	3.86	1.96	50.78%
HW	94.57± 0.2	94.47± 0.4	0.10↓	88.44± 0.2	88.32± 0.5	0.12↓	94.57± 0.2	94.47± 0.4	0.1↓	2.90	1.30	44.83%
BDGP	54.56± 7.9	54.18± 6.3	0.38↓	29.79± 6.8	29.14± 5.2	0.65↓	55.35± 7.5	54.52± 6.2	0.83↓	5.45	1.19	21.83%
RGBD	19.61± 0.0	20.40± 0.9	0.79↑	24.27± 0.0	21.52± 0.3	2.75↓	39.22± 0.4	37.45± 0.5	1.77↓	85.03	7.77	9.14%
DryBean	80.11± 0.0	80.62± 1.5	0.51↑	68.98± 0.0	69.56± 2.6	0.58↑	80.11± 0.0	80.62± 1.5	0.51↑	10.91	1.98	18.15%
AwA	10.08± 0.3	10.01± 0.2	0.07↓	11.29± 0.3	11.62± 0.4	0.33↑	11.76± 0.3	11.66± 0.3	0.1↓	211.69	79.17	37.40%
Avg.	52.35	52.38	0.03↑	48.84	48.47	0.37↓	56.18	55.82	0.36↓	/	/	29.45%

* The *Gap* and *Ratio* columns refer to performance increase/decrease and communication overhead ratio of CeFMC to CeFMC-H, while the *Avg.* row represents the average performance or communication overhead on all datasets.

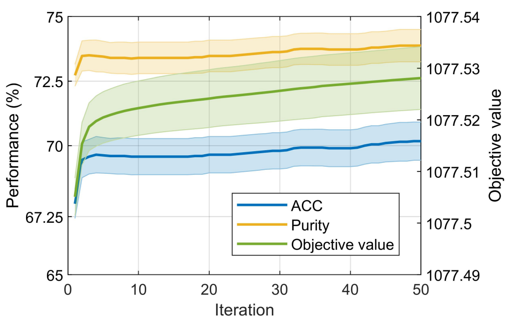

Fig. 3. Objective value and performance of the consensus label at each iteration on ORL dataset. Note that, NMI exceeds the axis range and would twist the figure thus is omitted here.

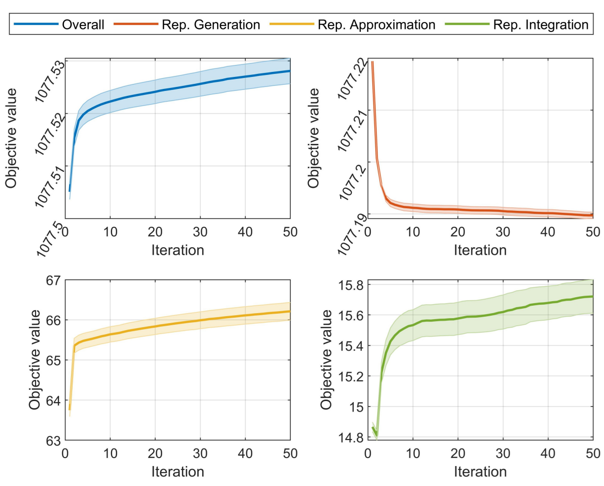

Fig. 4. Objective value variations of the CeFMC objective function and its three processes, i.e. representation generation, representation approximation and label integration, on ORL dataset.

the data information, but also improve the communication efficiency significantly.

F. Convergence

According to the convergence analysis of Section IV-F, the proposed CeFMC method is convergent by adopting the optimization strategy in Section IV-C. To validate this, we run the CeFMC method on ORL dataset and record the objective value of Eq. (19) in every iteration. Meanwhile, the consensus label is also evaluated at each update and corresponding results are accumulated. As a result, they are plotted in Fig. 3, where NMI follows a similar trend but exceeds the axis range and would twist the figure thus is not presented. It can be found that the objective value and clustering performances, i.e. ACC and purity, first increase dramatically and then rise to the maximum monotonically along with iteration. This not only well validates convergence of the alternate optimization procedure, but also confirms the rationality of the proposed CeFMC objective function.

According to Section IV-A, the proposed CeFMC method can be composed of four relatively independent processes, i.e. representation generation, representation approximation, label integration and alternate update, where the first three correspond to Eq.(13), Eq. (15) and Eq. (18), respectively. In such setting, we explore the objective value variations of the three processes and show them in Fig. 4. It can be observed that the objective value of CeFMC increases monotonically, which keeps consistent with that of Fig. 3. Nevertheless, the objective values of the representation generation and label integration also rise monotonically, while that of representation approximation decreases to minimum. This is caused by the fact that data representation $\mathbf{H}_v^{(0)}$ is initialized to the left singular vectors of data $\mathbf{X}_v$ corresponding to the k largest singular values which is the global optimal value of representation generation process in Eq. (13). The phenomenon well illustrates that the data representation computed by each client independently only encodes the data information of one view and is not the most suitable for clustering. With guidance from label integration of the server, the proposed CeFMC method can gradually generate better data representations, improving the clustering performance ultimately. In other words, the optimization strategy of CeFMC is to find the balance among the aforementioned three processes.

G. Parameter Study

As seen from the objective function of Eq. (19), the proposed CeFMC method introduces two hyper-parameters λ and

TABLE IX
PERFORMANCE COMPARISON OF THE PROPOSED CEFMC METHOD IN $\lambda \perp \beta$ AND $\lambda = \beta$ SETTINGS.

Dataset	ACC			NMI			Puirty		
	$\lambda \perp \beta$	$\lambda = \beta$	Gap	$\lambda \perp \beta$	$\lambda = \beta$	Gap	$\lambda \perp \beta$	$\lambda = \beta$	Gap
ORL	73.38± 2.4	72.65± 3.9	0.72↓	86.40± 1.5	85.62± 1.9	0.78↓	76.67± 2.4	76.35± 3.2	0.33↓
Flower17	35.02± 1.2	34.35± 0.9	0.67↓	33.35± 1.1	33.47± 1.3	0.12↑	36.04± 1.5	35.69± 1.1	0.35↓
HW	94.72± 0.3	94.47± 0.4	0.25↓	88.66± 0.3	88.32± 0.5	0.33↓	94.72± 0.3	94.47± 0.4	0.25↓
BDGP	57.27± 8.7	54.18± 6.3	3.10↓	32.84± 7.8	29.14± 5.2	3.71↓	57.51± 8.6	54.52± 6.2	3.00↓
RGBD	20.55± 1.0	20.40± 0.9	0.15↓	20.91± 0.3	21.52± 0.3	0.61↑	36.92± 0.5	37.45± 0.5	0.53↑
DryBean	81.13± 0.0	80.62± 1.5	0.51↓	70.38± 0.0	69.56± 2.6	0.81↓	81.13± 0.0	80.62± 1.5	0.51↓
AwA	10.14± 0.2	10.01± 0.2	0.13↓	11.60± 0.4	11.62± 0.4	-0.02↓	11.71± 0.3	11.66± 0.3	0.06↓
Avg.	53.17	52.38	0.79↓	49.16	48.46	0.70↓	56.39	55.82	0.57↓

* The *Gap* column refers to performance increase or decrease of $\lambda = \beta$ to $\lambda \perp \beta$, while the *Avg.* row represents the average performance on all datasets.

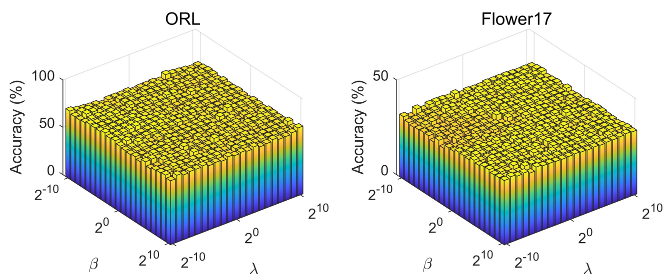

Fig. 5. Accuracies of the proposed CeFMC method with parameter λ and β taking different values in range $\{2^{-10}, 2^{-9}, \cdots, 2^{10}\}$ on ORL and Flower17 datasets.

β to balance the weights of representation generation, representation approximation and label integration processes. In the above experiments, both of them are grid-searched in range $\{2^{-10}, 2^{-9}, \cdots, 2^{10}\}$. To analyze their effects on clustering performance, we collect the corresponding accuracies and present them in Fig. 5. It can be observed that the clustering accuracy keeps relatively stable when λ and β take different values. Meanwhile, we notice that compulsively setting the two parameters with the same value, i.e. $\lambda = \beta$, can obtain similar best performances to setting them independently, i.e. $\lambda \perp \beta$. Specifically, best experiment results in the both settings are compared in Table IX. Hence, we recommend to set $\lambda = \beta$ in real-world applications to improve practicality of the proposed CeFMC method.

VI. CONCLUSION

To keep the data privacy of each client, existing federated multi-view clustering approaches propose to share the encoded data representations among clients and server rather than the original data observations. However, the data representations are transmitted at every iteration of model training, leading to heavy communication overheads. Meanwhile, most of them employ the matrix factorization and neural network encoding techniques to reduce computational complexity, but results in unsatisfactory clustering performance due to the failure of considering the similarity information among data samples. To solve these two issues, this paper first proposes the CeFMC framework by approximating the data representation with a pseudo-label and centroid matrix and sharing them in model training, saving a large volume of communication overhead. Then, the linear kernel function is utilized implicitly to incorporate the pairwise similarities among data in its instantiation. Moreover, we not only introduce its optimization strategy thoroughly, but also analyze its computation complexity, communication overhead and convergence. Nevertheless, we conduct extensive experiments on seven benchmark datasets and compare it with existing federated multi-view clustering and centralized multi-view clustering methods. In both comparisons, the proposed CeFMC method achieves obviously better results, well validating its effectiveness and efficiency. In the near future, we will continue to explore better solutions to promote the efficiency of federated multi-view clustering, since the communication bandwidth is extremely limited in most real-world applications. Meanwhile, the proposed CeFMC method lacks a feasible strategy to evaluate the contribution of each client and we will explore this intensively.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 62306324, 62376279, U24A20333, 62325604, 62276271, 62421002), the Science and Technology Innovation Program of Hunan Province (No. 2024RC3128) and the National University of Defense Technology Research Foundation (No. ZK24-30).

REFERENCES

[1] J. Wen, Z. Zhang, L. Fei, B. Zhang, Y. Xu, Z. Zhang, and J. Li, "A survey on incomplete multiview clustering," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 53, no. 2, pp. 1136–1149, 2023.
[2] M. Chen, J. Lin, X. Li, B. Liu, C. Wang, D. Huang, and J. Lai, "Representation learning in multi-view clustering: A literature review," *Data Sci. Eng.*, vol. 7, no. 3, pp. 225–241, 2022.
[3] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, 2019.
[4] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, 2020.
[5] W. Chang, H. Chen, F. Nie, R. Wang, and X. Li, "Tensorized and compressed multi-view subspace clustering via structured constraint," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10 434–10 451, 2024.

[6] J. Liu, X. Liu, Y. Yang, Q. Liao, and Y. Xia, "Contrastive multi-view kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9552–9566, 2023.

[7] J. L. C. Bárcena, F. Marcelloni, A. Renda, A. Bechini, and P. Ducange, "Federated c-means and fuzzy c-means clustering algorithms for horizontally and vertically partitioned data," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 12, pp. 6426–6441, 2024.

[8] S. Huang, W. Shi, Z. Xu, I. W. Tsang, and J. Lv, "Efficient federated multi-view learning," *Pattern Recognit.*, vol. 131, p. 108817, 2022.

[9] W. Feng, Z. Wu, Q. Wang, B. Dong, and Q. Gao, "Federated fuzzy c-means with schatten-p norm minimization," in *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*. ACM, 2024, pp. 9407–9416.

[10] W. Feng, Z. Wu, Q. Wang, B. Dong, Z. Tao, and Q. Gao, "Federated multi-view clustering via tensor factorization," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*. ijcai.org, 2024, pp. 3962–3970.

[11] X. Chen, Y. Ren, J. Xu, F. Lin, X. Pu, and Y. Yang, "Bridging gaps: Federated multi-view clustering in heterogeneous hybrid views," *CoRR*, vol. abs/2410.09484, 2024.

[12] X. Chen, J. Xu, Y. Ren, X. Pu, C. Zhu, X. Zhu, Z. Hao, and L. He, "Federated deep multi-view clustering with global self-supervision," in *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*. ACM, 2023, pp. 3498–3506.

[13] X. Yan, Z. Wang, and Y. Jin, "Federated incomplete multi-view clustering with heterogeneous graph neural networks," *CoRR*, vol. abs/2406.08524, 2024.

[14] W. Feng, Z. Wu, Q. Wang, B. Dong, Z. Tao, and Q. 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 2024, Jeju, South Korea, August 3-9, 2024*. ijcai.org, 2024, pp. 3971–3979.

[15] Z. Han, C. Zhang, H. Fu, and J. T. Zhou, "Trusted multi-view classification with dynamic evidential fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2551–2566, 2023.

[16] X. Liu, "Simplemkkm: Simple multiple kernel k-means," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5174–5186, 2023.

[17] Z. Ren, S. X. Yang, Q. Sun, and T. Wang, "Consensus affinity graph learning for multiple kernel clustering," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3273–3284, 2021.

[18] J. Liu, X. Liu, Y. Yang, X. Guo, M. Kloft, and L. He, "Multiview subspace clustering via co-training robust data representation," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 10, pp. 5177–5189, 2022.

[19] A. Kumar, P. Rai, and H. D. III, "Co-regularized multi-view spectral clustering," in *25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, 2011, pp. 1413–1421.

[20] Z. Huang, J. T. Zhou, X. Peng, C. Zhang, H. Zhu, and J. Lv, "Multi-view spectral clustering network," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. ijcai.org, 2019, pp. 2563–2569.

[21] Q. Zheng, "Flexible and parameter-free graph learning for multi-view spectral clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 9, pp. 8966–8971, 2024.

[22] Y. Sun, X. Li, Q. Sun, M. Zhang, and Z. Ren, "Improved weighted tensor schatten p-norm for fast multi-view graph clustering," in *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*. ACM, 2024, pp. 1427–1436.

[23] J. Wen, G. Xu, Z. Tang, W. Wang, L. Fei, and Y. Xu, "Graph regularized and feature aware matrix factorization for robust incomplete multi-view clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 5, pp. 3728–3741, 2024.

[24] B. Yang, X. Zhang, F. Nie, F. Wang, W. Yu, and R. Wang, "Fast multi-view clustering via nonnegative and orthogonal factorization," *IEEE Trans. Image Process.*, vol. 30, pp. 2575–2586, 2021.

[25] J. Wang, F. Tian, H. Yu, C. H. Liu, K. Zhan, and X. Wang, "Diverse non-negative matrix factorization for multiview data representation," *IEEE Trans. Cybern.*, vol. 48, no. 9, pp. 2620–2632, 2018.

[26] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 4412–4419.

[27] X. Yang, C. Deng, Z. Dang, and D. Tao, "Deep multiview collaborative clustering," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 34, no. 1, pp. 516–526, 2023.

[28] J. Wen, C. Liu, G. Xu, Z. Wu, C. Huang, L. Fei, and Y. Xu, "Highly confident local structure based consensus graph learning for incomplete multi-view clustering," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 15 712–15 721.

[29] M. Yang, Y. Li, P. Hu, J. Bai, J. Lv, and X. Peng, "Robust multi-view clustering with incomplete information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 1055–1069, 2023.

[30] C. Zhang, Y. Cui, Z. Han, J. T. Zhou, H. Fu, and Q. Hu, "Deep partial multi-view learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2402–2415, 2022.

[31] X. Li, Y. Pan, Y. Sun, Y. Sun, Q. Sun, Z. Ren, and I. W. Tsang, "Scalable unpaired multi-view clustering with bipartite graph matching," *Information Fusion*, p. 102786, 2024.

[32] P. Hu, Z. Huang, D. Peng, X. Wang, and X. Peng, "Cross-modal retrieval with partially mismatched pairs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9595–9610, 2023.

[33] Y. Sun, Y. Qin, Y. Li, D. Peng, X. Peng, and P. Hu, "Robust multi-view clustering with noisy correspondence," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 12, pp. 9150–9162, 2024.

[34] Z. Long, C. Zhu, P. Comon, Y. Ren, and Y. Liu, "Feature space recovery for efficient incomplete multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 9, pp. 4664–4677, 2024.

[35] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[36] X. Hu, J. Qin, Y. Shen, W. Pedrycz, X. Liu, and J. Liu, "An efficient federated multiview fuzzy c-means clustering method," *IEEE Transactions on Fuzzy Systems*, vol. 32, no. 4, pp. 1886–1899, 2024.

[37] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y. Zhang, and Q. Yang, "Vertical federated learning: Concepts, advances, and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3615–3634, 2024.

[38] B. Pinkas, T. Schneider, and M. Zohner, "Faster private set intersection based on OT extension," in *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014*. USENIX Association, 2014, pp. 797–812.

[39] Y. Huang, D. Evans, and J. Katz, "Private set intersection: Are garbled circuits better than custom protocols?" in *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5-8, 2012*. The Internet Society, 2012.

[40] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola, "On kernel-target alignment," in *Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada*. MIT Press, 2001, pp. 367–373.

[41] J. Liu, X. Liu, J. Xiong, Q. Liao, S. Zhou, S. Wang, and Y. Yang, "Optimal neighborhood multiple kernel clustering with adaptive local kernels," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 2872–2885, 2022.

[42] F. Nie, R. Zhang, and X. Li, "A generalized power iteration method for solving quadratic problem on the stiefel manifold," *Sci. China Inf. Sci.*, vol. 60, no. 11, pp. 112 101:1–112 101:10, 2017.

[43] J. Liu, X. Liu, Y. Yang, L. Liu, S. Wang, W. Liang, and J. Shi, "One-pass multi-view clustering for large-scale data," in *IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 12 324–12 333.

[44] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of Second IEEE Workshop on Applications of Computer Vision, WACV 1994, Sarasota, FL, USA, December 5-7, 1994*. IEEE, 1994, pp. 138–142.

[45] M. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*. IEEE Computer Society, 2006, pp. 1447–1454.

[46] M. van Breukelen, R. P. W. Duin, D. M. J. Tax, and J. E. den Hartog, "Handwritten digit recognition by combined classifiers," *Kybernetika*, vol. 34, no. 4, pp. 381–386, 1998.

[47] A. C. Spradling, D. Stern, A. Beaton, E. J. Rhem, T. Laverty, N. Mozden, S. Misra, and G. M. Rubin, "The berkeley drosophila genome project gene disruption project: Single p-element insertions mutating 25% of vital drosophila genes," *Genetics*, vol. 153, no. 1, pp. 135–177, 09 1999.

[48] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *IEEE Conference on Computer*

Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. IEEE Computer Society, 2015, pp. 567–576.

[49] M. Koklu and I. A. Özkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," *Comput. Electron. Agric.*, vol. 174, p. 105507, 2020.

[50] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 2009, pp. 951–958.

[51] J. Liu, X. Liu, C. Li, X. Wan, H. Tan, Y. Zhang, W. Liang, Q. Qu, Y. Feng, R. Guan, and K. Liang, "Large-scale multi-view tensor clustering with implicit linear kernels," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Music City Center, Nashville TN, June 11-15, 2025*. IEEE, 2024.

[52] J. Gao, J. Han, J. Liu, and C. Wang, "Multi-view clustering via joint nonnegative matrix factorization," in *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA*. SIAM, 2013, pp. 252–260.

[53] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. IJCAI/AAAI, 2013, pp. 2598–2604.

[54] S. Liu, S. Wang, P. Zhang, K. Xu, X. Liu, C. Zhang, and F. Gao, "Efficient one-pass multi-view subspace clustering with consensus anchors," in *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Virtual Event, February 22 - March 1, 2022*. AAAI Press, 2022, pp. 7576–7584.

[55] W. Xia, Q. Gao, Q. Wang, X. Gao, C. Ding, and D. Tao, "Tensorized bipartite graph learning for multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5187–5202, 2023.

[56] D. Huang, C. Wang, and J. Lai, "Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 11, pp. 11 388–11 402, 2023.

[57] S. Fang, D. Huang, X. Cai, C. Wang, C. He, and Y. Tang, "Efficient multi-view clustering via unified and discrete bipartite graph learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 35, no. 8, pp. 11 436–11 447, 2024.

Jiyuan Liu received his Ph.D. from National University of Defense Technology (NUDT), China, in 2022. He is now a lecturer with the College of Systems Engineering, NUDT. His current research interests include multi-view clustering, federated learning and anomaly detection. Dr. Liu has published papers in journals and conferences such as IEEE TPAMI, IEEE TKDE, IEEE TNNLS, ICML, NeurIPS, CVPR, ICCV, etc. He serves as program committee member and reviewer on IEEE TPAMI, IEEE TKDE, IEEE TNNLS, ICML, NeurIPS, CVPR, ICCV, etc.

Xinwang Liu received his Ph.D. degree from National University of Defense Technology (NUDT), China, in 2013. He is now a professor with the College of Computer, NUDT. His current research interests include kernel learning, multi-view clustering and unsupervised feature learning. Dr. Liu has published 100+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, ICML, NeurIPS, CVPR, ICCV, AAAI, IJCAI, etc. He is an Associate Editor of IEEE T-NNLS and Information Fusion Journal.

Siqi Wang is currently an assistant research professor in College of Computer, NUDT. His main research include anomaly/outlier detection, pattern recognition and unsupervised learning. His works have been published on leading conferences and journals, such as NeurIPS, AAAI, IJCAI, ACM MM, TPAMI and TIP. He serves as a PC member and reviewer for top-tier conferences like NeurIPS and AAAI and several prestigious journals.

Xinhang Wan received the B.E degree in Computer Science and Technology from Northeastern University, Shenyang, China, in 2021. He is currently pursuing the Ph.D. degree with the National University of Defense Technology (NUDT), Changsha, China. He has published papers in journals and conferences such as IEEE TIP, IEEE TKDE, IEEE TNNLS, ICML, ACMMM, AAAI, etc. His current research interests include multi-view learning, incomplete multi-view clustering, continual clustering, and affordance learning.

Dongsheng Li received the PhD degree from the National University of Defense Technology, Changsha, China, in 2005. He is currently a full professor with the National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China. His research interests include distributed computing, cloud computing, computer networks, and large-scale data management. He was awarded the prize of the National Excellent Doctoral Dissertation of China by Ministry of Education of China, in 2008.

Kai Lu received the B.S. and Ph.D. degrees in computer science and technology from the College of Computer, National University of Defense Technology, Changsha, in 1995 and 1999, respectively. He is currently a Professor with the College of Computer, National University of Defense Technology. His research interests include operating systems, parallel computing, and security.

Kunlun He received his M.D. degree from The 3rd Military Medical University, Chongqing, China in 1988, and PhD degree in Cardiology from Chinese PLA Medical school, Beijing, China in 1999. He worked as a postdoctoral research fellow at Division of circulatory physiology of Columbia University from 1999 to 2003. He is director and professor of Medical Big Data Research Center, Chinese PLA General Hospital, Beijing, China. His research interests include big data and artificial intelligence of cardiovascular disease.