Deep Multiview Collaborative Clustering

Xu Yang, Member, IEEE, Cheng Deng, Senior Member, IEEE, Zhiyuan Dang, and Dacheng Tao, Fellow, IEEE

Abstract—The clustering methods have absorbed even-increasing attention in machine learning and computer vision communities in recent years. In this article, we focus on the real-world applications where a sample can be represented by multiple views. Traditional methods learn a common latent space for multiview samples without considering the diversity of multiview representations and use K-means to obtain the final results, which are time and space consuming. On the contrary, we propose a novel end-to-end deep multiview clustering model with collaborative learning to predict the clustering results directly. Specifically, multiple autoencoder networks are utilized to embed multi-view data into various latent spaces and a heterogeneous graph learning module is employed to fuse the latent representations adaptively, which can learn specific weights for different views of each sample. In addition, intraview collaborative learning is framed to optimize each single-view clustering task and provide more discriminative latent representations. Simultaneously, interview collaborative learning is employed to obtain complementary information and promote consistent cluster structure for a better clustering solution. Experimental results on several datasets show that our method significantly outperforms several state-of-the-art clustering approaches.

Index Terms—Collaborative learning, heterogeneous graph learning, multiview adaptive fusion, multiview clustering.

I. INTRODUCTION

In many real-world applications, abundant data collected from different views, including multiple modalities or multiple types of features, are available where each view captures different information, but they have the same clustering structure. Thus, it is essential to integrate and cluster multiview data. Numerous traditional clustering methods have been proposed in the past decades, such as K-means [1], nonnegative matrix factorization clustering [2], and spectral clustering [3]–[5]. These methods might be feasible for single-view data to accomplish the clustering task but are invalid to effectively combine multiview information to improve clustering performance.

Recently, various of multiview clustering algorithms have emerged [6]. Chaudhuri et al. [7] projected the multiview data into a lower dimensional subspace and clustered the data via canonical correlation analysis (CCA), which computes two sets of variables and maximizes the correlation between samples in the embedding space. Xia et al. [8] obtained a shared low-rank transition probability matrix as an input to the Markov chain for spectral clustering, and subsequently, Kumar and Daumé [9] imposed a co-regularized strategy on spectral clustering. Tao et al. [10] constructed a consensus partition of data across different views based on view-specific basic partitions (BPs). Motivated by kernel learning [11], Wang et al. [12] integrated heterogeneous features represented in terms of kernel matrices for clustering, and the structured low-rank representation is proposed by factorizing into the latent low-dimensional data-cluster representations for multiview learning.

Motivated by the recent advances in deep generative methods [13], [14], which have a powerful capability to capture nonlinear structures of data without supervised information, deep multiview clustering methods absorb more and more attention. SplitAE [15] has been introduced to be effective for multiview learning in speech and vision tasks based on deep autoencoder networks. Based on CCA, a deep neural network (DNN) extension of CCA (DCCA) [16] is employed to extract nonlinear features for each view, and the correlations between the extracted features were maximized by CCA on the top layer. Then, the deep canonically correlated autoencoder (DCCAE) network [17] was developed to optimize the canonical correlation between the learned features and the reconstruction errors of the autoencoder networks.

Deep subspace clustering, which assumes that a sample can be represented as a linear combination of other samples with a self-expressive layer, can construct an optimal subspace affinity matrix for spectral clustering. Abavisani and Patel [18] presented an autoencoder-based approach, where a fully connected layer is introduced between the encoder and the decoder to project different view data into a common latent space. This method needs to calculate the eigenvectors of the Laplacian matrix corresponding to the affinity matrix, and cluster the eigenvectors using K-means to obtain the final results. The computational cost and memory footprint can become overwhelming even for medium-scale datasets, not to mention large-scale datasets. Multiview spectral clustering network (MvSCN) [19], aiming to learn a common space from

of each sample to achieve adaptive fusion. Then, the fused module, which can learn specific weights for different views, representations are fused with a heterogeneous graph learning multiview data into various latent spaces, and the latent representations are fused with a heterogeneous graph learning module, which can learn specific weights for different views of each sample to achieve adaptive fusion. Then, the fused latent representations are fed to a clustering network to predict multiview clustering results directly with collaborative learning. Specifically, the intraview collaborative learning, which can provide discriminative information of the latent representations, is framed to optimize each single-view clustering task. Simultaneously, interview collaborative learning is employed to harvest the complementary information and promote a consistent cluster structure for multiview samples. In this way, multiple clustering tasks can learn from each other and improve overall performance.

The highlights of this article are three folds.

1) **End-to-End Learning Pipeline:** Unlike traditional methods [19], [20] that only learn deep representations or the subspace affinity matrices followed by learning-free $K$-means clustering, we propose a more effective network pipeline for deep multiview clustering. Specifically, the fusion of multiview latent representations is fulfilled by heterogeneous graph learning and fed to a clustering network for clustering prediction. Our end-to-end learning paradigm has functional advantages: 1) It is deterministic, and thus, more stable than existing methods using $K$-means whose results are sensitive to initial seeds; 2) our method is more flexible as it allows a single sample as input for clustering prediction, while $K$-means has to be performed in an offline mode; and 3) our method allows for incremental learning for clustering, in the sense of SGD training, thus being more space-saving and time-efficient in contrast to $K$-means that has to run all the samples in one shot.

2) **Collaborative Learning for Deep Multiview Clustering:** To overcome the discriminative information loss caused by mapping multiview data into a common space as commonly encountered by existing methods [18], we adopt collaborative learning to explore multiview samples. This is in contrast to the peer method [21] that collaboratively learns the subspace affinity matrix and clustering results, and to the best of our knowledge, this is the first work for collaborative learning for deep multiview clustering. Specifically, each single-view clustering task is optimized in an intraview collaborative manner to extract diverse and discriminative latent representations.

Multiple single-view clustering results and multiview clustering are jointly learned to obtain complementary information for improving the overall performance.

3) **State-of-the-Art Performance:** Empirical experiments demonstrate that our method significantly outperforms several state-of-the-art approaches on three datasets, including both traditional and deep network-based ones.

The rest of this article is organized as follows. We introduce some related work that may support and illuminate the reasons behind our work in Section II. Section III presents our proposed method with accompanying theoretical analysis. Section IV presents our experimental results and analysis. Finally, Section V concludes our work.

## II. Related Work

### A. Multiview Clustering

Nowadays, we have easier access to data that contains heterogeneous features representing samples from different views in many scientific fields, and numerous multiview clustering methods have been proposed in order to exploit the diverse and complementary information contained in different views [22]. Some methods use the multikernel learning strategy to solve this problem, where different predefined kernels are utilized to deal with different views. These kernels are combined either linearly or nonlinearly in order to arrive at a unified kernel. Multiview clustering method via CCA [7], which uses CCA to project the multiview high-dimensional data into a low-dimensional subspace, is proposed to compute two sets of variables and maximize the correlation between them. Then, the work [23] presented a binary multiview clustering (BMVC) framework, which partially reduces the computation cost and memory footprint, meanwhile obtaining better performance.

In addition, graph-based methods have been proposed to explore the relationships between samples [24]. Some methods adopt nonnegative matrix factorization techniques for multiview clustering, seeking a common latent factor among multiview data. The multimodal spectral clustering algorithm [25] is proposed to integrate multiple image features, which learns a commonly shared Laplacian matrix by unifying different models and adds a nonnegative relaxation to improve the robustness of image clustering. Moreover, a co-regularized approach for MVSC is introduced to co-regularize the clustering hypotheses making different graphs agree with each other [9]. Multiview subspace clustering methods [26], [27] perform clustering on the subspace representation of each view simultaneously [28]. Relying on the importance of both low-rank and sparsity constraints in the construction of the affinity matrix [29], multiview low-rank sparse subspace clustering [30] introduces the objective that balances between the agreement across different views, while at the same time encourages sparsity and low-rankness of the solution. Moreover, Zhu et al. [31] proposed a structured general and specific multiview subspace clustering method, where the structural general representation matrix keeps the similarity relationship of data and the specific representation matrices exploit the diversity between different matrices.
MVSC [32] is proposed with bipartite graph. Latent multiview subspace clustering (LMSC) [33] is introduced to seek underlying latent representations and simultaneously performs data reconstruction based on the learned latent representations. A self-weighted multiview clustering method [34] is proposed to cluster feature points by incorporating their motion and context similarities based on $\ell_1$ and $\ell_2$ norm. In general, graph-based methods are pretty conspicuous for efficiency and excellent clustering performance. Under the manifold assumption, graph-based methods trade labeled and unlabeled examples as vertices of a graph and utilize edges to propagate information from labeled ones to unlabeled ones.

### B. Deep Multiview Clustering

DNNs composed of multiple nonlinear transformations can learn a better feature representation than traditional shallow models. A deep autoencoder network, where the goal is to extract a common representation, is utilized to reconstruct the inputs of multiple views. In this scenario, a common encoder is utilized to extract common representations for all views, and different decoders are used to reconstruct view-specific input features from the common representation [35], [36]. SplitAE [15] has been introduced to be effective for multiview learning in speech and vision tasks based on deep autoencoder networks. Based on CCA, a DCCA [16] is employed to extract nonlinear features for each view and the correlations between the extracted features were maximized by CCA on the top layer. Then, the DCCA networks [17] were developed to optimize the canonical correlation between the learned features and the reconstruction errors of the autoencoder networks for two views together.

Recently, deep MVSC and multiview subspace clustering have earned increasing interest. Multiview clustering via deep matrix decomposition [37] is adopted to learn the hierarchical semantics of multiview data in a layerwise fashion. MvdSCN [20] exploited two subnetworks, i.e., diversity network and universality network, to capture different information of the samples and then obtain the results with spectral clustering. The affinity matrix of samples obtained by the self-expressive layer is based on the assumption that the reconstruction loss can make the samples with the same class having similar representations in the latent space. DMSC-UDL [38] combines global and local structures with a self-expression layer. The global and local structures help each other forward and achieve a small distance between samples of the same cluster. However, in fact, the reconstruction loss does not coincide with this expectation effectively. Despite the success of existing MvC works, most of them are nonparametric shallow models which have been proven ineffective and inefficient to handle real-world data, especially considering the large-scale setting and the complex data distribution.

To overcome these disadvantages, however, to the best of our knowledge, only a few deep multiview clustering approaches have been devoted to predicting clustering results with multiview features directly, which calls for a new mechanism to solve it.

### C. Deep Graph Learning

Graph neural networks (GNNs) are introduced and the convolution operation is defined in the Fourier domain by computing the eigendecomposition of the graph Laplacian, resulting in potentially intense computations and nonspatially localized filters. Kipf and Welling [39] simplified the previous method by restricting the filters to operate in a one-step neighborhood around each node. Hamilton introduced GraphSAGE [40], a method for computing node representations in an inductive manner. This technique operates by sampling a fixed-size neighborhood of each node and then performing a specific aggregator over it. More recently, attention mechanisms have become almost a standard module in many sequence-based tasks. When an attention mechanism is used to compute a representation of a single sequence, it is commonly referred to as self-attention or intra-attention. The graph attention network is proposed to compute the hidden representations of each node by attending over its neighbors, following a self-attention strategy [41]. This article presents a heterogeneous graph learning module to fuse the latent representations, which can learn specific weights for different views of each sample to achieve adaptive fusion, and the fused latent representations are fed to a clustering network to predict multiview clustering results directly with collaborative learning.

### III. METHODOLOGY

As aforementioned, our proposed method consists of multiple autoencoder networks, dedicating to build view-specific latent representations, and two fully connected layers followed each encoder network is applied to optimize each single-view clustering task and explore more discriminative information for a better clustering solution. A heterogeneous graph learning module is adopted to fuse the latent representations adaptively with a more reasonable weight to different views of each sample, and then the fused latent representations are fed into a clustering network to predict multiview clustering results. The outputs of each single-view clustering task and the multiview clustering task are optimized together, which can be alternatively supervised and updated from each other until obtaining ideal clustering results. The framework is shown in Fig. 1.

Let $\tilde{x}^v = \{\tilde{x}_1^v, \ldots, \tilde{x}_n^v\}$ be the input samples, and $\tilde{z}^v = \{\tilde{z}_1^v, \ldots, \tilde{z}_n^v\}$ be their corresponding latent representations, where $\tilde{z}_i^v \in \mathbb{R}^d$ is learned by the autoencoder network, where $v \in V$ represents the number of different views. The decoder network is utilized to reconstruct the samples with $\tilde{z}^v$.

We adopt multiple clustering layers: $\tilde{z}_i^v \rightarrow \tilde{y}_i^v \in \mathbb{R}^K$, to predict the pseudolabels of each single view. The deep neural network is utilized to output the final multiview clustering results $\tilde{y}_i^v \in \mathbb{R}^K$, where $K$ is the number of cluster.

#### A. Deep Multiview Clustering

We first train multiple autoencoder networks to embed the multiview data into various latent spaces and learn the latent representations $\tilde{z}^v$ with the reconstruction loss as follows:

$$
\mathcal{L}_r = \sum_{v=1}^{V} \| \tilde{x}^v - \breve{x}^v \|_F^2
$$

(1)
Fig. 1. Our framework consists of multiple autoencoder networks, which can focus on the latent representations of each view respectively. The fully connected layers after each encoder network are applied to optimize the latent representations for optimal clustering. The multiview representations are fused by a heterogeneous graph learning module, and then the fused representations are utilized to predict final clustering results directly. The multiple single-view clustering tasks and the multiview clustering task are optimized in a multiview collaborative scheme.

where \( \hat{X}^v \) is the samples reconstructed by the decoder networks, and then the latent representations are utilized to predict the clustering results directly without using the ground-truth labels. The first priority of predicting the clustering results is to integrate the multiple latent representations and obtain the most consistent representations with the real data distributions. However, cascading or averaging multiview features directly cannot obtain optimal results due to the views of data are inherently strong or weak, which means the final results will be degraded if we fail to distinguish different views. It is really elusive to pursue good performance while rely on adaptive parameter searching to optimize the multiview representations. Thus, we design a multiview latent representations adaptive fusion model based on a heterogeneous graph learning module, and the first step of the module is to calculate the mean of the multiview latent representations

where \( \bar{z}_i \) is the concatenation operation. The attention mechanism, parametrized by a weight vector \( \mathbf{a} \), is a single-layer feedforward neural network, and the nonlinear LeakyReLU is adopted as the activation. \( \bar{z}_i^v \) represents the specific weight for different views of each sample. The normalized attention coefficients are utilized to compute a linear combination of the multiview latent representations, and the final representations for each sample are

In addition, we adopt multihead attention to stabilize the learning process. Specifically, we execute the attention transformation of (4) independently \( U \) times with different weight vector \( \mathbf{a} \), and then their results are averaged, as shown in Fig. 2, resulting in the following output
latent representations:
\[
\tilde{z}^{\text{fusion}}_{i} = \frac{1}{U} \sum_{u=1}^{U} \sum_{v=1}^{V} \tilde{e}_{i}^{u}(u) \tilde{z}^{u}_{i}. \tag{5}
\]

The fused multiview representations are adopted to predict the final clustering results by a clustering network, and the multiple graph affinity matrices are utilized to supervise the clustering results. The clustering can effectively explore the relationships between samples to reduce intraclass differences and produce better clustering results than K-means. We use the latent representations \(\tilde{Z}^{v}\) of each view to compute the nonnegative graph affinity matrix \(\tilde{W}^{v}\)
\[
\tilde{W}^{v}_{ij} = e^{-\frac{||\tilde{z}_{i}^{v} - \tilde{z}_{j}^{v}||^{2}}{2\sigma^{2}}}. \tag{6}
\]

The multiple graph affinity matrices are utilized to supervise the clustering results. Simply combining multiple views directly through affinity matrix addition or feature concatenation may lead to limited performance. We combine the multiple graph affinity matrices with adaptive weights of different views to supervise the learning of the clustering network, and the network is defined as
\[
\tilde{A} = \sum_{v=1}^{V} \alpha_{v} \tilde{W}^{v} \tag{7}
\]
where \(\sum_{v=1}^{V} \alpha_{v} = 1\) denotes the important weights of different view which are learned adaptively. The graph affinity matrix \(\tilde{A}\) is calculated with multiple latent representations and large only if \(\tilde{x}_{i}\) and \(\tilde{x}_{j}\) are probably from the same subspace in multiple views. The loss function of clustering network can be defined as
\[
\mathcal{L}_{c} = \sum_{i,j=1}^{V} \tilde{A}_{ij} \| \tilde{y}_{i} - \tilde{y}_{j} \|^2. \tag{8}
\]

The objective function is utilized to optimize the clustering network as a loss function, where \(\tilde{Y}\) is adopted by a softmax output layer. \(\alpha_{v}\) is dependent on the solution \(\tilde{Y}\) and we obtain the results by alternatively updating. The \(\alpha_{v}\) can be represented as
\[
\alpha_{v} = \frac{1/(\sum_{u=1}^{V} \tilde{W}_{ij}^{u} \| \tilde{y}_{i} - \tilde{y}_{j}^{u} \|^2 )}{\sum_{v=1}^{V} 1/(\sum_{u=1}^{V} \tilde{W}_{ij}^{u} \| \tilde{y}_{i} - \tilde{y}_{j}^{u} \|^2 )}. \tag{9}
\]

In order to prevent that all points are grouped into the same cluster, the output \(\tilde{Y}\) is required to be orthonormal when we calculate the loss function [43]. Before calculating the loss function, we first linearly project \(\tilde{Y}\) to enforce the orthogonality constraint. The weights of the linear projection are computed through its QR decomposition, which is obtained by the Cholesky decomposition, producing the orthogonalized output \(\tilde{Y}\). Then, we calculate the loss function to ensure that \(\tilde{Y}\) does not collapse into a class, and this process does not require additional training parameters. In this way, the objective function of multiview clustering can be expressed as
\[
\min_{\tilde{Y}} \mathcal{L}_{c} + \lambda_{1} \mathcal{L}_{c}. \tag{10}
\]

B. Multiview Collaborative Learning

Considering that the latent representations will directly affect the subsequent fusion and clustering performance, we also need to harvest more discriminative information to improve the final clustering performance. It is not reasonable to directly enforce multiview samples to a common space due to multiview samples may describe samples from different perspectives. The intraview collaborative learning, where the clustering layer and the encoder network of each view are optimized together, is framed to explore more diverse and discriminative information for better clustering solution. Specifically, we optimize each single-view clustering task simultaneously for discriminative latent representations. The clustering model with a softmax output layer after each encoder network to learn each single-view clustering. Denote \(\tilde{F}^{v}\) as the output of the softmax layer, the graph affinity matrices shown in (6) are utilized to supervise the output by the same logic. In the clustering phase, we unify the latent representation learning and the clustering results using KL divergence and the loss is
\[
\mathcal{L}_{\text{icol}} = \sum_{v=1}^{V} \left( \sum_{i} \mathcal{KL}(p((\tilde{f}_{i}^{v} , \tilde{z}_{i}^{v} )) | | q(\tilde{f}_{i}^{v} , \tilde{z}_{i}^{v})) + \frac{\beta}{2} \sum_{i,j} \tilde{W}_{ij}^{v} \| \tilde{f}_{i}^{v} - \tilde{f}_{j}^{v} \|^2 \right). \tag{11}
\]

where \(p\) is the learned distribution of latent representations, and \(q\) represents the ideal distribution.

In each single-view clustering task, the latent representations \(\tilde{Z}^{v}\) is utilized to learn the clustering results \(\tilde{F}^{v}\), while \(\tilde{F}^{v}\) is employed to optimize \(\tilde{Z}^{v}\) according to the KL divergence. Once \(\tilde{Z}^{v}\) is optimized, the performance also improves for the graph affinity matrix \(\tilde{W}^{v}\). Thus, the intro-view collaborative learning between the \(\tilde{Z}^{v}\) and \(\tilde{F}^{v}\) is utilized to explore the discriminative capability of the latent representations, which facilitates subsequent fusion and clustering.

For multiview clustering, the most important consideration is that one should preserve the clustering quality within a single view and the clustering consistency across different views. In other words, the multiview latent representations of each sample are diverse, but the cluster structure of each view should be consistent. To explore more information from multiview samples, interview collaborative learning, which promotes consistent cluster structure for multiview samples, is employed to harvest the complementary information. We optimize the similarity between \(\tilde{y}_{i}\) and \(\tilde{f}_{i}^{v}\) to achieve collaborative learning between different clustering tasks, where different clustering results can learn together and supervise each other to obtain a consistent solution and improve overall performance. As such, we optimize the modules collaboratively to supervise each other and the cross-entropy loss of multiview collaborative learning is
\[
\mathcal{L}_{\text{mcol}} = \sum_{v=1}^{V} \sum_{i} \ln \left( 1 + e^{-\tilde{z}_{i}^{v} \tilde{y}_{i}} \right). \tag{12}
\]
Combining multiview clustering network, the overall loss of our model is

$$\min_{Y} \sum_{F} L_{r} + \lambda_1 L_{c} + \lambda_2 L_{mcol} + \lambda_3 L_{icol}. \quad (13)$$

The first two terms are utilized to optimize multiview clustering results with the shared graph affinity matrix, the next one is adopted to learn the latent representations more accurately and exploit various supervised information, and the last one is utilized to enforce the consistency of multiple clustering tasks.

Fig. 3 shows the optimization process of our method. The graph affinity matrices are learned with multiple latent representations, and the multiple single-view clustering results are optimized by the graph affinity matrices. During the iterative process, the distribution of multiview representations will be optimized accordingly, forming an ideal intro-view collaborative learning. And then, the multiple single-view clustering results are utilized to supervise multiview clustering and enforce the consistent cluster structure of multiview samples. In this way, the intro-view collaborative learning improves the discriminative capability of the latent representations while exploiting more diverse information from multiview data, and the interview collaborative learning improves the overall performance of multiple clustering tasks.

Once the classification result $\tilde{Y}$ of multiview samples is obtained, we can directly infer the cluster labels through the GNN output

$$\tilde{s}_i = \arg\max \tilde{y}_i \quad (14)$$

where $\tilde{s}_i$ is the cluster label of the image $\tilde{x}_i$.

Algorithm 1 Algorithm of Deep Multiview Collaborative Clustering

Input:
Unlabeled multi-view data $\{\tilde{x}^1, \ldots, \tilde{x}^V\}$, number of clusters $K$, $\beta$, $\lambda_1$, $\lambda_2$ and $\lambda_3$.

Initialization:
Pre-train the autoencoder by minimizing the reconstruction error of each view.

while not converge do
1: For each mini-batch multi-view data
2: Integrate the multiple representations with Eq. 5.
3: Calculate the graph affinity matrices $W^v$ with Eq. 6.
4: Train the autoencoder networks and the multi-view clustering network to minimize loss function in Eq. 10
5: Update $\tilde{F}^v$ with Eq. 11.
6: Update $\tilde{Y}$ with Eq. 12.
7: Jointly update all the parameters by minimizing Eq. 13.
end while

Output: Classification result $\tilde{Y}$ of multi-view samples.

IV. EXPERIMENTS

In this section, we evaluate the effectiveness of the proposed clustering method in three datasets and then compare the performance with several state-of-the-arts.

A. Datasets

In order to show that our method works well with various kinds of datasets, we choose the following datasets.

The Extended YaleB [18]: The Extended Yale-B dataset based on the facial regions is utilized to evaluate the performance on the face dataset. We extract facial components

Fig. 3. Multiview collaborative learning. The graph affinity matrices are learned with multiple latent representations, and the multiple single-view clustering results are optimized by the graph affinity matrices. During the iterative process, the distribution of multiview representations will be optimized accordingly, forming an ideal intro-view collaborative learning. And then, the multiple single-view clustering results are utilized to supervise multiview clustering and enforce the consistent cluster structure of multiview samples. In this way, the intro-view collaborative learning improves the discriminative capability of the latent representations while exploiting more diverse information from multiview data, and the interview collaborative learning improves the overall performance of multiple clustering tasks.
from the images and use them along with the entire face for clustering. Here, the views do not share any direct spatial correspondence.

**Digits-MNISTM** [44], [45]: We adopt MNIST and MNIST-M to test the performance, where MNIST consists of 70,000 gray digits images of size 28 × 28 and MNIST-M dataset consists of MNIST digits blended with random color patches from the BSDS500 dataset.

**Fashion-MNIST** [46]: This dataset has the same number of images and the same image size as MNIST, but it is fairly more complicated. Instead of digits, it consists of various types of fashion products. Specifically, we use the original dataset as view 1, and randomly select within-class images to add additive noisy as view 2.

**STL-10:** This dataset consists of color images of 96 × 96 pixel size, in which there are ten classes with 1300 examples each. We extract the deep features of STL-10 to pursue better performance. Specifically, the deep features are extracted from the fully connected layers of four powerful networks, i.e., the AlexNet, VGG16, and ResNet50, all of which are pretrained on ILSVRC2012.

### B. Clustering Metrics

To evaluate the clustering results, we use three standard evaluation metrics: Accuracy (ACC), normalized mutual information (NMI) [2], and adjusted rand index (ARI). The best mapping between cluster assignments and true labels is computed using the Hungarian algorithm to measure accuracy [47]. For completeness, we define ACC with

\[
\text{ACC} = \max_m \frac{\sum_{i=1}^n 1(l_i = m(c_i))}{n}
\]  

(15)

where \(l_i\) and \(c_i\) are the true label and predicted cluster of data point \(x_i\).

NMI calculates the normalized measure of similarity between two labels of the same data, which is defined as

\[
\text{NMI} = \frac{I(l; c)}{\max[H(l), H(c)]}
\]

(16)

where \(I(l; c)\) denotes the mutual information between true label \(l\) and predicted cluster \(c\), and \(H\) represents their entropy. Results of NMI do not change by permutations of clusters (classes), and they are normalized to the range of [0, 1], with 0 meaning no correlation and 1 exhibiting perfect correlation.

Another quantitative metric is the ARI, which is scaled between -1 and 1. It computes the similarity between two clusters by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the ground truth and predicted clusters. The larger the ARI, the better the clustering performance.

### C. Implementation Details

In our experiments, we set \(\lambda_1 = 1, \lambda_2 = 1,\) and \(\lambda_3 = 0.5\), respectively. \(\beta = 0.1\) is the weight to balance the KL divergence and the graph loss. For all the experiments, we pretrain the convolutional autoencoder for 100 epochs with a learning rate \(1 \times 10^{-3}\), then decrease the learning rate to \(1 \times 10^{-5}\) multiview collaborative learning. The channel numbers and kernel sizes of the autoencoder network are shown in Table I, and the dimension of latent space is set to 120. The deep spectral clustering network consists of four fully connected layers. We adopt ReLU as the nonlinear activations and use RMSProp to minimize the loss for all our experiments. The batch size is set to 1024. The weight \(W_i^v\) is calculated as the nearest neighbor graph, and the number of neighbors is set to 5.

### D. Comparison Methods

Co-Reg [9] co-regularizes clustering hypotheses of different views to be consistent with each other. RMSC [48] constructs a joint low-rank transition probability matrix for spectral clustering. DMF [37] conducts multiview clustering with the deep matrix factorization framework with maximizing mutual information of different views, which enforce the final-layer nonnegative representation of each view to be the same. SwMC [49] computes the weights to represent the importance of different views and directly assign the cluster label to each data point. DCCA [16] is the nonlinear extension of CCA, which extracts both nonlinear features for each view and the canonical correlations between different views using kernel technique and DNNs, respectively. DCCAE [50] adds the autoencoder regularization term to DCCA for better-reconstructing inputs. MVEC [10] extends ensemble clustering to multiview cases, which generates a group of view-specific BPs to extract a consensus one shared by multiple views. BMVC [23] realizes large-scale clustering by integrating compact collaborative discrete representation learning and binary clustering structure learning into a unified framework. MvSCN [19] aims to learn a common space with the view-specific representations from multiview data. DMSC [18] adopts the self-expressive layers and multiple decoders for different spatial fusion-based approaches.

### E. Evaluation of Clustering Algorithms

In this section, we evaluate our methods on two mid-size datasets and one small face dataset, where the Digits-MNISTM...
TABLE II
CLUSTERING PERFORMANCE OF DIFFERENT ALGORITHMS ON THREE DATASETS-BASED USING THREE METRICS: ACC, NMI, AND ARI. BEST RESULTS IN BOLD

<table>
<thead>
<tr>
<th>Method</th>
<th>Digits-MNIST</th>
<th>Fashion-MNIST</th>
<th>The Extended YaleB</th>
<th>STL-10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>NMI</td>
<td>ARI</td>
<td>ACC</td>
</tr>
<tr>
<td>Best View</td>
<td>83.1</td>
<td>85.6</td>
<td>82.7</td>
<td>63.5</td>
</tr>
<tr>
<td>Co-Reg [9]</td>
<td>84.6</td>
<td>86.2</td>
<td>85.3</td>
<td>62.1</td>
</tr>
<tr>
<td>RMSC [48]</td>
<td>86.7</td>
<td>88.1</td>
<td>86.3</td>
<td>63.0</td>
</tr>
<tr>
<td>DMP [37]</td>
<td>87.2</td>
<td>89.3</td>
<td>88.2</td>
<td>62.1</td>
</tr>
<tr>
<td>SwMC [49]</td>
<td>89.4</td>
<td>90.1</td>
<td>89.8</td>
<td>65.1</td>
</tr>
<tr>
<td>DCCA [16]</td>
<td>86.8</td>
<td>87.5</td>
<td>83.9</td>
<td>66.3</td>
</tr>
<tr>
<td>DCCA [50]</td>
<td>85.2</td>
<td>87.3</td>
<td>88.2</td>
<td>68.3</td>
</tr>
<tr>
<td>MVEC [10]</td>
<td>88.6</td>
<td>90.2</td>
<td>87.5</td>
<td>66.2</td>
</tr>
<tr>
<td>BMVC [23]</td>
<td>87.3</td>
<td>89.1</td>
<td>88.3</td>
<td>67.6</td>
</tr>
<tr>
<td>DMSC [18]</td>
<td>87.5</td>
<td>88.3</td>
<td>89.2</td>
<td>69.0</td>
</tr>
<tr>
<td>MsSCN [19]</td>
<td>91.3</td>
<td>93.1</td>
<td>90.3</td>
<td>72.3</td>
</tr>
</tbody>
</table>

| Ours     | 93.2 | 95.2 | 92.1 | 74.3 | 77.1 | 65.3 | 98.9 | 98.6 | 96.5 | 94.2 | 88.6 | 87.9 |

Fig. 4. Visualization of latent representation distributions on Digit-MNISTM dataset. Top: before our optimization; Bottom: after our optimization. (a) View1. (b) View2. (c) View3. (d) View1. (e) View2. (f) View3.

The Digits-MNIST and Fashion-MNIST datasets consist of 10,000 test samples. We adopt our autoencoder to extract deep features and then test the final results of comparison methods. DMSC cannot apply to the whole dataset due to the memory and computation issue, and we only select 5000 samples with fixed intervals to generate a subset to show the clustering performance. On Fashion-MNIST, we employ residual blocks with convolutional layers to capture multiview features. The Extended YaleB only contains 2424 samples. We adopt the learned representations to construct the affinity matrix and obtain the final results with spectral clustering. We run our method with ten random trials and report the average performance. The clustering results are shown in Table II, where we can notice that our proposed method outperforms the competing methods on these datasets.
It demonstrates that our method can effectively optimize the distribution of the latent representations and obtain a better clustering result.

We visualized the subspace representations distribution of different views in the Digits-MNISTM dataset, and the results are shown in Fig. 4, where Fig. 4(a)–(c) is the distributions of latent representations from original deep autoencoder networks, and Fig. 4(d)–(f) is the results of our method with collaborative learning. The results show that the latent representations obtained by our method have a more clear distribution structure.

### F. Ablation Studies

We evaluate our method on two large-scale datasets, where the Digits-MNISTM and Fashion-MNIST contain a total of 70,000 samples with 60,000 training and 10,000 testing samples, and we compare different strategies for training our model. For training the model, we analyze the following approaches: the results of different views according to (11). In addition, we also report the results of multiview clustering (DMC) according to (10), where the fused latent representations and the clustering results are learned using KL divergence, DMC with intro-view collaborative learning, and DMC with collaborative learning. From the results shown in Table III, we can see that multiview collaborative learning can effectively improve the clustering performance distinctly, and multiple single-view clustering tasks can enhance the complementarity of information from various views and provide effective support for the multiview clustering task. Furthermore, we show the important weights $\alpha$ on the Digits-MNISTM dataset, DMC_Avg is the clustering results with the averaged graph affinity matrices $\tilde{A} = (1/V) \sum_{v=1}^{V} \tilde{W}$. It indicates that our weighted adaptive algorithm can capture the importance of the multiple graph affinity matrices. It has higher flexibility of our model due to the algorithm can directly predict the label information of the data without the need for further steps. Specifically, we can input any number of test samples to directly obtain the clustering results once the model training is complete, which is more convenient and applicable than the methods obtaining the final results with $K$-means.

In addition, Fig. 5(a) is the fused latent representations with average, while Fig. 5(b) is the fused latent representations with specific weights for different views of each sample obtained from the heterogeneous graph learning module. We can find that learning a proper set of weights can help us better integrate multiview samples according to the results. Moreover, we investigate the parameter sensitivity on Digit-MNISTM, and the results are shown in Fig. 6, where Fig. 6(a) represents the results of ACC from different parameters and Fig. 6(b) is the results of NMI. They both indicate that the proposed method is insensitive to the parameter combinations.

Finally, we verify the change of the loss function and the clustering results with each iteration in the Digits-MNISTM dataset, and the results are shown in Fig. 7. The results demonstrate that the algorithm can effectively improve the clustering performance of each single-view clustering task while achieving optimal results for the multiview clustering task. It is worth mentioning that our multiview collaborative method can effectively improve the performance of other view samples. View3, which has the worst clustering performance

---

### TABLE III

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Metrics</th>
<th>view1</th>
<th>view2</th>
<th>view3</th>
<th>ADMC_Avg</th>
<th>DMC</th>
<th>DMC_jCL</th>
<th>DMC_CL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digits-MNISTM</td>
<td>ACC</td>
<td>87.2</td>
<td>80.8</td>
<td>77.5</td>
<td>88.5</td>
<td>89.9</td>
<td>91.2</td>
<td>93.0</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>88.6</td>
<td>83.5</td>
<td>81.8</td>
<td>89.3</td>
<td>92.9</td>
<td>94.1</td>
<td>95.1</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>86.3</td>
<td>81.7</td>
<td>79.3</td>
<td>87.5</td>
<td>89.1</td>
<td>91.8</td>
<td>91.7</td>
</tr>
<tr>
<td>Fashion-MNIST</td>
<td>ACC</td>
<td>70.2</td>
<td>66.8</td>
<td></td>
<td>71.5</td>
<td>72.4</td>
<td>73.1</td>
<td>73.9</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>73.1</td>
<td>67.6</td>
<td></td>
<td>72.8</td>
<td>75.4</td>
<td>76.9</td>
<td>76.7</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>59.6</td>
<td>57.2</td>
<td></td>
<td>60.5</td>
<td>62.7</td>
<td>64.8</td>
<td>65.0</td>
</tr>
</tbody>
</table>

---

Fig. 5. Visualization of the fused latent representation distributions on Digit-MNISTM. (a) Data points in the latent subspace of average features. (b) Data points in the latent subspace of our model. (a) Average. (b) Our Method.

Fig. 6. ACC and NMI of Our method with different $\lambda_2$ and $\lambda_3$ on Digit-MNISTM. (a) ACC. (b) NMI.
after initialization, will gradually improve its performance to achieve similar results to other views.

V. CONCLUSION

In this article, we propose a novel end-to-end deep multi-view clustering framework, which has multiple single-view clustering tasks and one multiview clustering task. The single-view clustering task optimized by the intraview collaborative learning is framed to provide more discriminative information of the latent representations. In addition, a heterogeneous graph learning module is employed to fuse the latent representations adaptively, which can learn a specific weight for different views of each sample. Furthermore, interview collaborative learning, which achieves mutual learning of multiple clustering tasks, is employed to harvest the complementary and consistent information of multiview data. Unlike traditional methods, only learn a joint affinity matrix or deep features with multiview samples and use $K$-means to obtain the final results, the proposed method which can predict clustering labels directly is more convenient and applicable for practical applications, and we will explore the applicability of the proposed model to existing clustering methods in future work. Experimental results on several datasets show that our method significantly outperforms several state-of-the-art clustering approaches.

REFERENCES


Cheng Deng (Senior Member, IEEE) received the B.E., M.S., and Ph.D. degrees in signal and information processing from Xidian University, Xi’an, China, in 2001, 2006, and 2009, respectively. He is currently a Full Professor with the School of Electronic Engineering, Xidian University. He has authored and coauthored more than 100 scientific articles at top venues, including the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, TRANSACTIONS ON IMAGE PROCESSING, TRANSACTIONS ON CYBERNETICS, TRANSACTIONS ON MULTIMEDIA, TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, the International Conference on Computer Vision, Computer Vision and Pattern Recognition, the International Conference on Machine Learning, Neural Information Processing Systems, the International Joint Conference on Artificial Intelligence, and the Association for Advancement of Artificial Intelligence. His current research interests include computer vision, pattern recognition, and information hiding.

Zhiyuan Dang received the B.E. degree in electronic and information engineering from Xidian University, China, in 2017, where he is currently pursuing the Ph.D. degree with the School of Electronic Engineering. His current research interests include machine learning and pattern recognition.

Dacheng Tao (Fellow, IEEE) is currently a Professor of computer science and an ARC Laureate Fellow with the School of Computer Science and the Faculty of Engineering and the inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney, Sydney, NSW, Australia. His research results in artificial intelligence have expounded in one monograph and more than 200 publications at prestigious journals and prominent conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the International Journal of Computer Vision, the Journal of Machine Learning Research, Artificial Intelligence, the AAAI Conference on Artificial Intelligence, the International Joint Conference on Artificial Intelligence, the Conference on Neural Information Processing Systems, the International Conference on Machine Learning, the IEEE Conference on Computer Vision and Pattern Recognition, the International Conference on Computer Vision, the European Conference on Computer Vision, the IEEE International Conference on Data Mining, and the ACM Knowledge Discovery and Data Mining.

Dr. Tao is a fellow of the American Association for the Advancement of Science, the Association for Computing Machinery, and the Australian Academy of Science. He received several best paper awards. He received the 2015 Australian Museum Scopus-Elsevier Prize and the 2018 IEEE ICDM Research Contributions Award.