Adversarial Multiview Clustering Networks With Adaptive Fusion

Qianqian Wang, Zhiqiang Tao, Wei Xia, *Graduate Student Member, IEEE*, Quanxue Gao, Xiaochun Cao, *Senior Member, IEEE*, and Licheng Jiao, *Fellow, IEEE*

***Abstract*— The existing deep multiview clustering (MVC) methods are mainly based on autoencoder networks, which seek common latent variables to reconstruct the original input of each view individually. However, due to the view-specific reconstruction loss, it is challenging to extract consistent latent representations over multiple views for clustering. To address this challenge, we propose adversarial MVC (AMvC) networks in this article. The proposed AMvC generates each view's samples conditioning on the fused latent representations among different views to encourage a more consistent clustering structure. Specifically, multiview encoders are used to extract latent descriptions from all the views, and the corresponding generators are used to generate the reconstructed samples. The discriminative networks and the mean squared loss are jointly utilized for training the multiview encoders and generators to balance the distinctness and consistency of each view's latent representation. Moreover, an adaptive fusion layer is developed to obtain a shared latent representation, on which a clustering loss and the $\ell_{1,2}$-norm constraint are further imposed to improve clustering performance and distinguish the latent space. Experimental results on video, image, and text datasets demonstrate that the effectiveness of our AMvC is over several state-of-the-art deep MVC methods.**

***Index Terms*— Adaptive fusion, adversarial training, multiview clustering (MVC).**

I. Introduction

CLUSTER analysis is a fundamental research problem in machine learning, computer vision, and data mining, which aims to divide the data into multiple groups composed of similar objects [1]. Due to the unsupervised nature, clustering is free from manually labeling data, and thus, is widely used in many practical applications, such as image segmentation [2], image retrieval [3], video summarization [4], and so on. Since multiview data that are characterized by different features [5], [6] are ubiquitous in the real world, multiview clustering (MVC) [7], [8] has emerged as a promising research direction, aiming to discover the intrinsic cluster structures by utilizing multiple views of a particular dataset.

Existing MVC methods can be roughly categorized into traditional MVC [9]–[11] and deep MVC [12]–[14]. Nowadays, traditional MVC has been widely studied. For instance, the multiview nonnegative matrix factorization method was proposed for multiview image processing, which emphasizes the structural inconsistency between different view's representations with a new regularization term [15]. Nie *et al.* [16] developed a multiview neighborhood learning method with adaptive similarity matrix learning from raw data, which achieves impressive clustering performance on multiview data. Wang *et al.* [17] introduced position-aware proprietary items to take full advantage of the complementary information embedded in various view representations, based on which a novel multiview subspace clustering model is designed. The proposed model employs consistency constraints to make these complementary representations have common features. However, the traditional MVC methods extract the inherent data structure with shallow and linear embedding functions and, to some extent, cannot model the deeper characteristics of complex data.

Recently, many research efforts have been made on developing deep multiview neural networks for clustering. For example, inspired by CCA and data reconstruction, Wang *et al.* [18] integrated graph and discriminative learning into deep multiview subspace clustering and solved the problems of jointly combining multiview features in sparse subspace clustering. For unsupervised multimodal subspace learning, deep multimodal subspace clustering (DMSC) networks [19] and multiview deep subspace clustering network (MvDSCN) [20] were proposed based on convolutional autoencoders and end-to-end multiview self-representation learning, respectively. Zhao *et al.* [15] employed graph regularized seminonnegative matrix factorization to study the deep MVC algorithm. Nevertheless, the existing autoencoder-based deep MVC methods have some limitations: 1) they only use reconstruction loss to learn the consistency information

Manuscript received December 15, 2020; revised July 4, 2021 and November 8, 2021; accepted January 14, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0100600, in part by the Initiative Postdocs Supporting Program under Grant BX20190262, in part by the China Postdoctoral Science Foundation under Grant 2019M663642, in part by the National Natural Science Foundation of Shaanxi Province under Grant 2020JZ-19, in part by the National Natural Science Foundation of China under Grant 62176203, and in part by the Fundamental Research Funds for the Central Universities. *(Corresponding author: Quanxue Gao.)*

Qianqian Wang is with the State Key Laboratory of Integrated Services Networks and the Key Laboratory of Ministry of Education of Intellisense and Image Understanding, Xidian University, Xi'an 710071, China.

Zhiqiang Tao is with the Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA 95053 USA (e-mail: ztao@scu.edu).

Wei Xia and Quanxue Gao are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: qxgao@xidian.edu.cn).

Xiaochun Cao is with the School of Cyber Science and Technology, Shenzhen Campus, Sun Yat-sen University, Shenzhen 518107, China.

Licheng Jiao is with the Key Laboratory of Ministry of Education of Intellisense and Image Understanding, School of Artificial Intelligence, Xidian University, Xi'an 710071, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3145048.

Digital Object Identifier 10.1109/TNNLS.2022.3145048

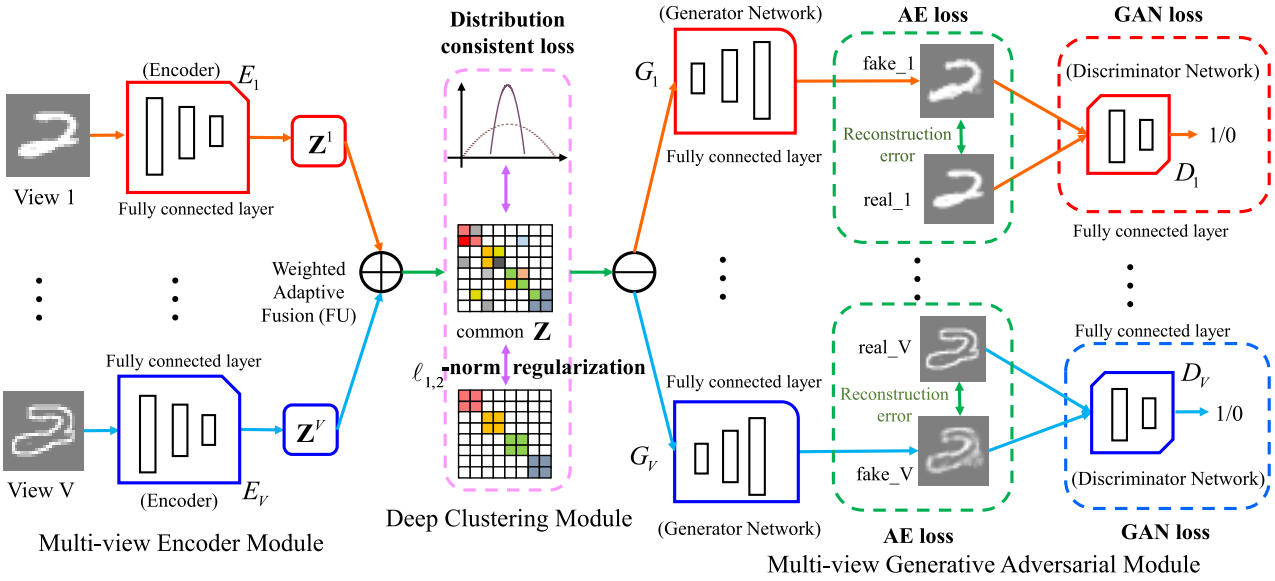


Fig. 1. Illustration of AMvC network. AMvC consists of multiview encoder E, multiview generator G, multiview discriminator D, and an adaptive fusion layer and an embedding clustering layer on the top of our encoder. Multiview encoder network E outputs a low-dimensional latent layer feature $\mathbf{Z}^v$ for each view. For each $\mathbf{Z}^v$, multiview generator network G generates reconstructed samples. A Discriminator network is used to distinguish generated sample or a real one. Adaptive fusing layer fuses V latent representation $\mathbf{Z}^v$ to a common representation $\mathbf{Z}$. Clustering layer can improve clustering performance with distribution consistent loss and $\ell_{1,2}$-norm regularization.

between reconstructed samples and original ones, whereas reconstruction loss is view-specific, and it is difficult to extract consistent latent representations over multiple views for clustering. 2) The shared representation might not be discriminative enough for clustering. 3) Various fusion methods are used in these methods, but they ignore that different views usually differ in terms of importance.

To overcome these challenges, we develop a novel adversarial MVC (AMvC) model to extract the internal structure embedded in multiview data (see Fig. 1). Our model adopts adversarial training [21] as a regularizer to guide the training of encoders and generators based on reconstruction loss, ensuring the consistent relationship between the reconstructed samples and the original input. In addition, AMvC learns an effective mapping by developing weight-shared multiview encoder networks, which can map the raw data features into a shared and low-dimensional embedding space. The latent feature representation is extracted from each view, and a multiview generator is used to generate each view from the latent feature. Compared with traditional algorithms, the proposed AMvC can reveal the nonlinear characteristics of multiview data. Moreover, a learnable weighting mechanism is designed to adaptively fuse each view to obtain the shared latent representation, and we define the clustering loss with the relative entropy between the distributions of the ideal label and the predefined label, to constrain the shared latent representation and improve the clustering results. We also perform a $\ell_{1,2}$-norm constraint on the shared latent representation to make the extracted features more discriminative.

This article is an extended version of our previous work [22]. Compared with [22], the substantial differences are as follows: 1) we further design a new adaptive fusion layer that can integrate the complementary view-specific information for better representations. 2) We add a discriminative feature learning loss to learn a more discriminative embedding. 3) We provide more theoretical analyses, model discussions, and experimental evaluations for better effectiveness demonstration. The contributions of this work are highlighted in three-folds.

1) We propose a novel AMvC network. It uses a multiview encoder to extract latent descriptions from multiple views and a multiview generator to generate samples of each view, which encourages the uniqueness and sharing of the latent features extracted from each view and effectively improves the clustering performance.
2) The proposed model introduces an adaptive fusion that adaptively learns a weight for the latent feature of each view and then obtains the optimal shared representation. It is beneficial to obtain a better feature representation to improve the clustering performance.
3) We introduce $\ell_{1,2}$-norm as a regularization term in AMvC to select discriminative feature representation in feature learning and increase the discriminative capability of the shared representation.

We conduct several experiments on video, image, and text datasets to demonstrate the superiority of AMvC to other MVC methods.

II. Related Work

A. Multiview Clustering

Traditional MVC algorithms can be divided into five categories. Some methods [15] build MVC based on matrix factorization that uses nonnegative matrix factorization to seek common latent factors. Some methods use the multikernel learning [23] strategy to solve this problem. Since multiple kernels naturally correspond to multiple views in multikernel

learning, many MVC methods adopt multikernel learning by using different kernels to process different views and data-fuse them into a unified kernel in a linear or nonlinear manner for clustering [24]. From the perspective of subspace learning, subspace clustering is also a relatively common method to solve this problem. The MVC methods based on subspace learning generally assume that all input views are generated from a latent subspace shared by multiple views. Since the latent subspace has a lower dimension than that of all input views, subspace learning can effectively reduce the dimension of the feature. The obtained low-dimensional latent subspace should also capture the most likely consistency shared by all views [25]. Then, we can obtain the final clustering results by conducting any clustering algorithm on this subspace. Canonical correlation analysis (CCA) [26] and kernel CCA (KCCA) [27] have long been the main force of multiview feature learning and dimensionality reduction. They can obtain the multiview data's consistent representation by maximizing the correlation of the subspaces in the two views and make the two view data similar to each other. The learned consistent representation can be used for MVC and regression with some extra design [28]. Recently, graph clustering attracts considerable attention due to its simple implementation, convenient calling, and excellent performance, which is applied in MVC and achieves good performance. Nie *et al.* [29] developed autoweighted multiple graph learning (AMGL) that can learn the weight set of all graphs automatically without additional parameters.

B. Deep Multiview Clustering

Compared with traditional shallow models, deep neural networks (DNNs) can learn a better feature representation [30], [31]. Numerous deep multiview learning has attracted much attention and is widely used in classification and clustering [32], [33]. A representative method for MVC is based on CCA [34]–[36]. By maximizing the correlation of the subspaces in two views, the CCA method makes the two view data similar to each other to learn a consistent representation. According to this idea, Andrew *et al.* [34] proposed an extended version of CCA (DCCA) by using DNN to learn the complex nonlinear transformation of bimodal data so that the resulting representation presents a high linear correlation. Inspired by CCA and reconstruction-based goals, Wang *et al.* [37] developed deep canonically correlated autoencoders (DCCAEs). Unlike DCCA, DCCAE not only makes the resulting representation highly linearly related but also ensures the reliability of the learned representation through reconstruction. However, all of the above-mentioned methods can only be applied to two view data. To solve this challenge, Benton *et al.* [35] proposed deep generalized CCA (DGCCA), which is a nonlinear transformation used to learn arbitrary multiview data forms method so that the resulting representations have the most similar information to each other.

In addition, scholars have also proposed a variety of other multiview embedding clustering methods [38], [39]. For example, Abavisani and Patel [19] applied convolutional neural networks to unsupervised DMSC. Although this method has shown encouraging results, it is challenging to perform the DMSC algorithm on large-scale datasets due to self-expression constraints. For another example, Xie *et al.* [36] proposed a novel joint deep MVC (DMJC) framework that is able to learn multiple deep embedding features simultaneously. DMJC studies different fusion mechanisms and cluster allocation goals. In addition, Huang *et al.* [40] first explored the application of spectral clustering in deep multiview learning. The proposed method considers both the local invariant information within every single view and the consistent information shared by all views. In addition, it stacked multiple orthogonal constraint layers on the embedded network. Sun *et al.* [41] integrated self-supervised learning and spectral clustering into a deep MVC framework, which leverages the clustering results and leverages classification and spectrum clustering loss to supervise the latent representation learning and the common latent subspace learning of multiple views, respectively. Wang *et al.* [42], [43] designed a GAN-based model for partial MVC, which employs the GAN model to recover the missing multiview data. Although deep MVC algorithms have developed rapidly, it is still under exploration to learn discriminative low-dimensional latent spaces among multiview data through DNNs.

III. Adversarial Multiview Clustering Network

A. Motivation

Existing autoencoder-based deep MVC methods incorporate the mean square error to learn the latent representation that can retain the structural characteristic of each view. Relying on this constraint makes the latent representation contain much view-specific information, resulting in insufficient exploration of shared descriptions among views and poor clustering results. In response to this problem, the proposed AMvC network is proposed. AMvC uses a multiview generator to generate each view's samples conditioning on the fused latent representations among different views to encourage a consistent clustering structure. Moreover, the discriminative networks and reconstruction loss are used to balance the distinctness and consistency of each view's latent representation. Thus, the extracted features of one view not only contain its unique information but also contain the shared information by all the views.

The existing fusion methods for deep MVC are various. These methods treat all views equally in the fusion process. However, different views may have different roles when fusing to a shared representation. Thus, AMvC uses an adaptive fusion layer to gradually learn the weights for each view's latent feature and then obtain the optimal shared representation. In addition, in order to increase the discriminative capability of the shared representation, we introduce $\ell_{1,2}$-norm constraint to select discriminative feature representation in feature learning. The $\ell_{1,2}$-norm regularization makes the shared latent representation more distinguishable, as shown in Fig. 2, where the features with $\ell_{1,2}$-norm have less similar part with others in different clusters.

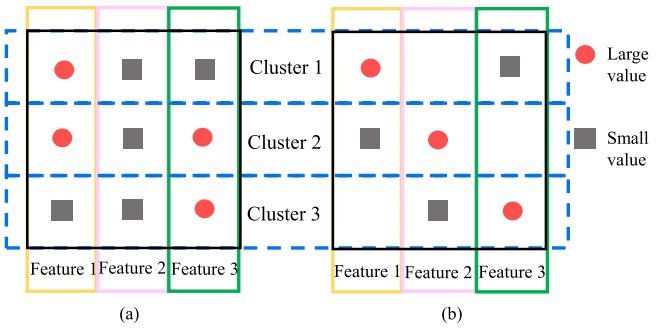


Fig. 2. Illustration of $\ell_{1,2}$-norm discriminability. (a) Features that are learned without $\ell_{1,2}$-norm regularization. (b) Features that are learned with $\ell_{1,2}$-norm regularization.

B. Notations

Given a multiview dataset $\mathbf{X} = \{\mathbf{X}^1, \ldots, \mathbf{X}^v, \ldots, \mathbf{X}^V\}$, where V is the number of views, $\mathbf{X}^v = \{x_1^{(v)}, x_2{}^{(v)}, \ldots, x_n^{(v)}\} \in R^{d_v \times n}$ denotes the n samples of dimension d_v from the vth view, we construct the AMvC networks with three submodules: multiview encoder module E, multiview generative adversarial module (contains generators G and discriminators D), and deep clustering module (contains weighted adaptive fusion layer and deep embedding clustering layer). For multiview data, corresponding to each view, our model has V encoders, one fusion layer, one clustering layer, V generators, and V discriminators. Table I gives the description for main network parts in detail, in which d_v is the feature dimension of vth view, d_h is the dimension of output/input of view-specific fully connected layer (FC layer) in encoder/generator network. m is the dimension of output/input of shared fully connected layer in the encoder/generator network. Fig. 1 shows AMvC network for the V-view case.

C. Network Module

The detailed design of the proposed AmVC network is introduced as follows.

1) Multiview Encoder Module: In our multiview encoder network $E = \{E_1, \ldots, E_v, \ldots, E_V\}$, for each view, there are M-layer independent fully connected networks and N-layer fully connected networks with shared parameters. The independent layers are used to handle the different feature dimensions of each view. For vth view $\mathbf{X}^v$, the encoder E_v aims at learning a latent representation $\mathbf{Z}^v = \{z_1^{(v)}, z_2^{(v)}, \ldots, z_n^{(v)}\}$ ($\mathbf{Z}^v \in R^{m \times n}$). Specifically, it maps the d_v-dimensional input data $x_i^{(v)}$ to a low-dimensional representation $z_i^{(v)}$. This mapping could be represented as $\mathbf{Z}^v = E_v(\mathbf{X}^v; \theta_{E_v})$, where E_v represents the vth view's encoding network whose parameters are θ_{E_v}.

2) Multiview Generative Adversarial Module: This module has multiview generator $G = \{G_1, \ldots, G_v, \ldots, G_V\}$ and multiview discriminator $D = \{D_1, \ldots, D_v, \ldots, D_V\}$. Our multiview generator network has a symmetric architecture to our multiview encoder E. It consists of N-layer fully connected networks with the shared parameters and M-layer independent fully connected networks for each view, which can generate all visual reconstructed samples with the latent representations corresponding to each view. Specifically, we suppose $\hat{\mathbf{X}}^v = G_v(\mathbf{Z}^v)$, where $\hat{\mathbf{X}}^v$ represents the vth view's reconstructed sample matrix.

The discriminator network consists of V fully connected layer networks. Each discriminator D_v is composed of three fully connected layers, and we should note that $x_i^{(v)}$ is a real instance and $\hat{x}_i^{(v)}$ is a generated sample. D_v feeds discriminated the result back to the generator G_v to update its parameters. By this means, the discriminator works as a regularizer to guide the training of our multiview encoder network E for better robustness of embedding representations and effectively solves the overfitting problem.

3) Deep Clustering Module: To obtain a shared latent representation $\mathbf{Z}$, we introduce a weighted adaptive fusing layer FU in our model, which adaptively fuses V latent representation $\mathbf{Z}^v$ to a common representation $\mathbf{Z} = f(\{\mathbf{Z}^v\}_{v=1}^V; \beta)$, where $f(\cdot; \beta)$ represents the fusion function. In order to seek for a clustering-friendly latent space, we develop a unique deep embedding clustering layer CU in the network. The embedded clustering layer contains the new clustering centroids after each iteration. We use the shared representation $\mathbf{Z}$ and the cluster centroids $\{\mu_j\}_{j=1}^k$ to obtain the current data distribution and target data distribution. Furthermore, we employ the Kullback–Leibler (KL) divergence of the current data distribution and the target data distribution as the objective function to iteratively update the shared representation $\mathbf{Z}$ and the cluster centroids $\{\mu_j\}_{j=1}^k$.

D. Overall Objective Function

The total loss function of AMvC includes four parts: the autoencoder (AE) loss $\mathcal{L}_{\text{AE}}$, the GAN loss $\mathcal{L}_{\text{GAN}}$, the clustering loss $\mathcal{L}_{\text{CLU}}$, and the $\ell_{1,2}$-norm regularization $\mathcal{L}_{1,2}$. The overall objective function is presented as follows:

$$\min_{E,G,\beta,\mu} \max_{D} \mathcal{L}_{\text{AE}} + \lambda_1 \mathcal{L}_{\text{GAN}} + \lambda_2 \mathcal{L}_{\text{CLU}} + \lambda_3 \mathcal{L}_{12} \tag{1}$$

where λ_1, λ_2, and λ_3 are used to balance the impact of $\mathcal{L}_{\text{GAN}}$, $\mathcal{L}_{\text{CLU}}$, and $\mathcal{L}_{12}$. β denotes learning weight of adaptive fusion layer. μ represents the cluster centroids to be updated.

1) Autoencoder Loss: We minimize the AE loss to optimize our multiview encoders E. $\mathcal{L}_{\text{AE}}$ is measured by the mean square error between the generated samples and real samples as

$$\mathcal{L}_{\text{AE}} = \sum_{v=1}^{V} ||\mathbf{X}^v - \hat{\mathbf{X}}^v||_{\text{F}}^2. \tag{2}$$

However, the mean square error may lead to blurred reconstructed results and cannot model the data distribution of each view. To alleviate this issue, we adopt adversarial training to generate (recover) more realistic results and further enhance the model generalization.

2) Generative Adversarial Network Loss: Our model has a multiview generator G and a multiview discriminator D. Generator G continuously learns the probability distribution of real data in each view. Its goal is to convert the latent representations into reconstructed data of each view. The reconstructed data are fake data. The discriminator D determines whether an input is a real data. Suppose that the real data distribution of vth view is $x^v \sim P(\mathbf{X}^v)$, and the generated data distribution

TABLE I
DESCRIPTION FOR MAIN NETWORK ARCHITECTURE

Model components		Notation	Input→ Output Shape	Layer Information
Multi-view Encoder Module	Encoders	E_v	$(d_v \to d_h)$ $(d_h \to m)$	View-specific FC layers of E_v project d_v to $d_h, \forall v$ The shared FC layer among multiple views
Multi-view Generative	Generators	G_v	$(m \to d_h)$ $(d_h \to d_v)$	The shared FC layer among multiple views. View-specific FC layers of G_v project d_h to $d_v, \forall v$.
Adversarial Module	Discriminators	D_v	$(d_v \to 1)$	View-specific FC layers of D_v project d_v to $1, \forall v$.
Deep Clustering	Weighted adaptive fusion layer	FU	$([d_h, \ldots, d_h] \to d_h)$	V latent representations are fused to one.
Module	Deep embedding clustering layer	CU	$(d_h \to d_h)$	Update the shared representation $\mathbf{Z}$.

of vth view is $\hat{x}^v \sim P(\hat{\mathbf{X}}^v)$. Therefore, the GAN loss in our model can be described as

$$\mathcal{L}_{\text{GAN}} = \sum_{v=1}^{V} (\mathbb{E}_{x^v \sim P(\mathbf{X}^v)}[\log D_v(x^v)] + \mathbb{E}_{\hat{x}^v \sim P(\hat{\mathbf{X}}^v)}[\log(1 - D_v(\hat{x}^v))]). \quad (3)$$

Denoting the number of samples as N, $\mathbb{E}$ represents: $\mathbb{E}_{x \sim P(X)}[f(x)] = (1/N)\sum_{i=1}^{N} f(x^i)$. During the training process, the generator network and the discriminator network play a min–max game until converged. To be specific, the multiview generator together with the multiview encoder is trained to generate fake data similar to real data of each view, and meanwhile, we train the discriminators so that they can distinguish the fake data of each view effectively. However, GAN loss alone can only ensure that the same input is mapped to some random permutation of samples from a target data distribution, while it cannot guarantee the desired output at an instance level which is not suitable for a clustering task. In light of this, we incorporate GAN loss with the AE loss to achieve high reliability of data reconstruction.

3) Distribution Consistent Loss: The AE loss and the GAN loss enable our multiview generator to generate fake samples that are more similar to real ones, which encourages our embedding representations to contain original feature information as much as possible. However, they cannot guarantee that the encoded low-dimensional space has a good cluster structure. To seek a discriminative embedding, we encapsulate the clustering loss measured by KL-divergence, i.e., distribution consistent loss, in our AMvC network. Specifically, we learn V latent representations $\mathbf{Z}^v$ for the V views. Then, we can get a common latent representation based on these V views with an adaptive fusing layer, which is shown as follows:

$$\mathbf{Z} = f\left(\{\mathbf{Z}^v\}_{v=1}^{V}; \beta\right) = \sum_{v=1}^{V} \beta_v \mathbf{Z}^v \Big/ \sum_{v=1}^{V} \beta_v \quad (4)$$

where $\beta = \{\beta_1, \ldots, \beta_V\}$ are learnable parameters, and $f(\cdot; \beta)$ represents the fusion function.

Given the initial cluster centroids $\{\mu_j\}_{j=1}^{k}$, we refer to [44] and measure the similarity between common latent representation point z_i and centroid μ_j by using the student's t-distribution as a kernel

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'} (1 + ||z_i - \mu_{j'}||^2/\alpha)^{-\frac{\alpha+1}{2}}} \quad (5)$$

where α is freedom degree of the student's t-distribution, q_{ij} is interpreted as the probability that the sample i is assigned to the cluster j, which can be also named soft assignment. In the experiments, we set $\alpha = 1$. With the help of an auxiliary target distribution p_i, we iteratively refine the clusters by learning from their high confidence assignments. During the training process, we update model parameters by matching the soft assignment to the target distribution p_{ij}. Therefore, our objective is defined as the KL divergence loss between the auxiliary distribution p_{ij} and the soft assignment q_{ij} as follows:

$$\mathcal{L}_{\text{CLU}} = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \quad (6)$$

To calculate p_i, we raise q_i to its second power and normalize it with the frequency per cluster as

$$p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j'} q_{ij'}^2 / f_{j'}} \quad (7)$$

where $f_j = \sum_i q_{ij}$ are soft cluster frequencies. By introducing the squared q_{ij}, p_{ij} can enlarge the distance between points of the same cluster and reduce the distance between the points of different clusters, which helps obtain more discriminative and sparser results.

4) $\ell_{1,2}$-Norm Regularization: The common latent representation feature $\mathbf{Z}$ may have a large amount of redundant information, which results in that some samples are incorrectly clustered. To learn a more discriminative representation, we utilize the $\ell_{1,2}$-norm term to constrain the common latent representation feature matrix. The loss function is given by

$$\mathcal{L}_{12} = ||\mathbf{Z}||_{1,2} = \sqrt{\sum_i \left(\sum_j |z_{ij}|\right)^2} \quad (8)$$

where $||\cdot||_{1,2}$ is the $\ell_{1,2}$-norm. For a matrix $\mathbf{Z}$, the $\ell_{1,2}$ norm is defined as $\|\mathbf{Z}\|_{1,2} = (\sum_{j=1}^{M} (\sum_{i=1}^{N} |z_{ij}|)^2)^{(1/2)}$. Therefore,

the $\ell_{1,2}$-norm first does ℓ_1-norm on the column, and then smooths the ℓ_2-norm of the row. We choose $\ell_{1,2}$-norm on the common latent representation due to $\ell_{1,2}$-norm is widely known to learn robust and discriminative representation [45]. $\mathcal{L}_{12}$ selects a subset of features which are most correlated with each class separately and obtains features that generally perform better in many real-world datasets, including text, image and bio-microarray data.

E. Optimization

We jointly optimize the parameters of multiview encoder E, multiview generator G, multiview discriminator D, adaptive weights β, and the cluster centroids μ using the Adam optimizer [46]. The gradients of the AE loss, the GAN loss, and the $\mathcal{L}_{12}$ loss are easy to calculate; therefore, we focus on the calculation of the clustering loss gradient. We compute the gradients of $\mathcal{L}_{\mathrm{CLU}}$ with respect to common authentication information of each cluster centroid μ_j and each data point z_i as

$$\frac{\partial L}{\partial z_i} = \frac{\alpha+1}{\alpha}\sum_j \left(1+\frac{||z_i-\mu_j||^2}{\alpha}\right)^{-1}(p_{ij}-q_{ij})(z_i-\mu_j) \tag{9}$$

$$\frac{\partial L}{\partial \mu_j} = -\frac{\alpha+1}{\alpha}\sum_i \left(1+\frac{||z_i-\mu_j||^2}{\alpha}\right)^{-1}(p_{ij}-q_{ij}) \times (z_i-\mu_j). \tag{10}$$

The gradients of z_i w.r.t. the latent representation $z_i^{(v)}$ for each vth view data and fusion parameter β_v are computed by

$$\frac{\partial z_i}{\partial z_i^{(v)}} = \beta_v \Big/ \sum_{v=1}^{V}\beta_v \tag{11}$$

$$\frac{\partial z_i}{\partial \beta_v} = \frac{z_i^{(v)}\sum_{v=1}^{V}\beta_v - \sum_{v=1}^{V}\left(\beta_v z_i^{(v)}\right)}{\left(\sum_{v=1}^{V}\beta_v\right)^2}. \tag{12}$$

According to the above-mentioned formula, the gradients of $\mathcal{L}_{\mathrm{CLU}}$ can be obtained with respect to the clustering center μ_j, the multiview encoder E, and the fusion parameters β_v. These gradients then update the fully connected network and other parameters in a standard backpropagation manner. In order to discover cluster assignments, we stop our procedure when the proportion of points that change the cluster assignment between two consecutive iterations is less than tol% of points.

F. Training Procedure

It is generally difficult to ensure the convergence of training multiple GAN models. Thus, we follow previous work [42], [47] to treat GAN as a regularizer in our model and mainly adopt the clustering loss/objective as the convergence criterion. We pretrain the encoder and generator with the AE loss to facilitate the following GAN training under the assistance of an adaptive weight fusion layer to ensure the convergence of GAN training for multiple views. The following is our training procedure.

1) **Pretraining** multiview encoder E and multiview generator G by minimizing the AE loss. We take $\{x^1, x^2, \ldots, x^V\}$ as input for multiview encoder E and get V latent layer feature $\{z^1, z^2, \ldots, z^V\}$. Then, we take $\{z^1, z^2, \ldots, z^V\}$ as the input of multiview generator G and get V outputs. For any latent layer feature z^v, it can generate reconstruction samples of V views. We update multiview encoder E and multiview generator G by minimizing the AE loss. After these, we get common representation $\mathbf{Z}$, then we save the clustering centroids $\{\mu_j\}_{j=1}^k$ for the following training by performing the k-means algorithm on $\mathbf{Z}$.
2) **Pretraining** multiview encoder E, multiview generator G, and multiview discriminator D by optimizing the sum of the AE loss and GAN loss. As with the last step, we get V outputs corresponding to V views by multiview encoder E and multiview generator G. Then, we send these generated samples and corresponding real samples to the discriminative networks D, respectively. After that, we iteratively update the multiview encoder-generator network and the discriminative networks by optimizing the sum of the AE loss and GAN loss.
3) **Training** multiview encoder E, adaptive fusion parameters β, multiview generator G, multiview discriminator D, and embedded clustering layer.

Our embedded clustering layer contains the new clustering centroids after each iteration. In the beginning, we leverage the clustering centroids $\{\mu_j\}_{j=1}^k$ from step 1 and the common representation $\mathbf{Z}$ to calculate the clustering loss. Then, we use the total loss function to train the entire network. In each iteration, we update the clustering centroids μ and the adaptive fusion parameters β. After the training is completed, we use the obtained common representation to perform spectral clustering [48] to obtain the final clustering result.

Algorithm 1 AMvC Network

Input:
Multi-view data set $\{\mathbf{X}^v\}_{v=1}^V$, $\lambda_1, \lambda_2, \lambda_3$.
Procedure:
$learningrate : lr$=0.0001,
$optimizer : Adam$
$epoch = 30$.
Whlie pre-training not converged **do:**
Update E and G by minimize $\mathcal{L}_{AE}$.
End pre-training.
While training not converged **do:**
Update E, G and D by minimize $\mathcal{L}_{AE} + \mathcal{L}_{GAN}$.
End training
While training not converged **do:**
Update E, G, D, β and μ by minimize the total loss $\mathcal{L}$.
End training
return E, G, D, β and μ.
Obtain $\mathbf{Z}$ according to E and β.
Use $\mathbf{Z}$ as the similarity matrix and then do spectral clustering on it.
End Procedure.

IV. Experiments

A. Experimental Settings

1) Datasets: We evaluate the performance of AMvC on five multiview datasets to demonstrate its superiority. A brief introduction is given as follows.

a) Image dataset: The handwritten numerals (HW) dataset [49] consists of ten-digit classes from 0 to 9, and each digit class includes 200 data samples. In our experiment, we construct a multiview dataset from the HW dataset by extracting two types of features, i.e., 216 profile correlations, and 76 Fourier coefficients of the character shapes.

b) Image and text dataset: BDGP [50] is a dataset of two views, which consists of 2500 images about drosophila embryos of five categories. Each image includes a visual modality represented by a 1750-D visual vector and a textual modality represented by a 79-D feature vector. We validate our method and baselines on the entire BDGP dataset and evaluate their performance on both types of features.

c) Video dataset: The Columbia consumer video (CCV) dataset [51] contains 9317 YouTube videos with 20 diverse semantic categories. In our experiment, we use the subset (6773 videos) of CCV provided by [51], along with three hand-crafted features: STIP feature with 5000-D bag-of-words (BoWs) representation, SIFT feature extracted every two seconds with 5000-D BoWs representation, and MFCC feature with 4000-D BoWs representation.

d) Face dataset: The Notting-hill (NH) dataset [52] is a video-based face database constructed from the movie "NH" and includes 4660 face pictures of five main actors. In our experiments, 110 pictures of each actor are randomly selected, and LBP features, Gabor features, and intensity features are extracted to construct a multiview dataset.

e) Large-scale dataset: Mixed National Institute of Standards and Technology database (MNIST) consists of handwritten digit images with 28×28 pixels and is a widely used benchmark dataset. In our experiment, we employ its two-view version (70 000 samples) provided by [47], which takes the original gray images as the first view and images only highlighting the digit edge as the second view. Caltech UCSD Birds-2011 (CUB) [53] includes a total of 11 788 bird images of 200 species categories. There is an article from Wikipedia for each species, and all species are organized according to scientific classification (order, family, genus, and species). The species name list is obtained from an online field guide.

Table II shows the statistics of these five multiview datasets. Note that, we utilize both the training and testing samples in each dataset for the unsupervised clustering.

2) Comparison Algorithms: We choose spectral clustering [48] and 11 state-of-the-art MVC algorithms as baselines. 1) Feature concatenation spectral clustering (ConSC) [54] performs spectral clustering on the feature representation obtained by concatenating the features of each view. 2) Robust multiview spectral clustering (RMSC) [55] employs a low-rank constraint to recover a latent transition probability matrix from pairwise similarity matrices of each view. 3) AMGL [29] constructs a graph from every single view, and for each graph, it automatically learns an optimal weight without introducing additive parameters. 4) MVC and semisupervised classification with adaptive neighbors (MLAN) [16] simultaneously conducts local manifold structure learning and clustering, and it automatically allocates each view a weight. 5) MVSC [56] first obtains the subspace representation of each view and then conducts clustering on them simultaneously. It adopts a common clustering structure so that the consistency among various views is ensured. 6) Self-weighted MVC (SwMC) [57] proposes a self-weighted fusion scheme to address MVC. 7) Consistent and specific multiview subspace clustering (CSMSC) [52] can fit the real-world datasets better, which leverages a shared consistent representation to formulate the consistency and a set of specific representations to exploit the complementary property of multiple views. 8) Deep CCA (DCCA) [34] maps two-view data into a subspace with two learned nonlinear transformations, such that the representations are highly linear. 9) Locality adaptive latent MVC (LALMVC) [58] learns a latent representation, which is shared by different views via linear transformation, and coefficient matrix that well characterizes the latent representation neighbor relationship by manifold learning. 10) Scalable multiview subspace clustering (SMVSC) with unified anchors [59] combines anchor learning and graph construction into a unified optimization framework. 11) Deep multimodal subspace clustering (DMSC) [19] achieves multimodal subspace clustering based on CNN. 12) Deep AMvC network [22] uses GAN for MVC.

TABLE II

Statistics of Five Image Multimodal Image Datasets. Note That the Training and Testing Images in Each Dataset Are Jointly Utilized for Clustering

Dataset	#Sample	#Class	#View	#Dimension
HW	2000	10	2	216/76
BDGP	2500	5	2	1750/79
CCV	6773	20	3	4000/5000/5000
NH	550	5	3	2000/3304/6750
MNIST	70000	10	2	784/784

3) Evaluation Metrics: We adopt six standard metrics for clustering performance evaluation, i.e., accuracy (ACC) [60], normalized mutual information (NMI) [60], purity [61], F-score [62], precision [62], and recall [62], whose calculation formulas are given as follows.

Given a database A_i, suppose that $\{a_i\}$ and $\{b_i\}$ are the set of obtained labels the set of labels provided by the corpus, respectively, and then ACC can be calculated with the following formula:

$$\mathrm{ACC} = \frac{\sum_{i=1}^{n} \delta(b_i \mathrm{map}(a_i))}{n}$$

where n is the sample number, $\delta(a, b)$ is the delta function whose result is 1 if $a = b$ and is 0, otherwise; $\mathrm{map}(b_i)$ is the permutation mapping function that maps each cluster label b_i to the equivalent label from the data corpus. We can use the Kuhn–Munkres algorithm [63] to find the optimal mapping.

TABLE III
EXPERIMENTAL RESULTS OF EACH METHOD BY ACC, NMI, AND PURITY METRICS ON THE FOUR DATASETS. BEST RESULTS ARE HIGHLIGHTED IN **BOLD**

Method	BDGP			HW			CCV			NH		
	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity
SC1	0.4824	0.2863	0.4824	0.7057	0.6661	0.7144	0.1620	0.1219	0.1978	0.7065	0.6241	0.7664
SC2	0.9488	0.8894	0.9488	0.6540	0.6592	0.6938	0.1952	0.1765	0.2209	0.7998	0.7391	0.8302
SC3	-	-	-	-	-	-	0.1966	0.1384	0.2218	0.7011	0.7265	0.8124
ConSC [54]	0.6112	0.4067	0.6112	0.8208	0.7820	0.8363	0.1523	0.1143	0.1969	0.7861	0.7184	0.8274
RMSC [55]	0.8452	0.6739	0.8452	0.8225	0.7267	0.8225	0.2160	0.1799	0.2407	0.8291	0.7827	0.8545
AMGL [29]	0.9580	0.9037	0.9580	0.6545	0.6699	0.6785	0.1776	0.1405	0.2058	0.8291	0.7569	0.8400
MLAN [16]	0.6806	0.4881	0.6806	0.8434	0.8334	0.8650	-	-	-	0.7927	0.7432	0.8400
MVSC [56]	0.9728	0.9253	0.9728	0.6835	0.6849	0.7275	0.1814	0.1239	0.1993	0.8054	0.7480	0.8273
SwMC [57]	0.9580	0.9035	0.9580	0.7945	0.7793	0.8165	0.2091	0.1557	0.2350	0.7982	0.7803	0.8400
CSMSC [52]	0.9680	0.9112	0.9680	0.7740	0.7519	0.8040	0.2397	0.1871	0.2787	0.9255	0.8306	0.9255
DCCA [34]	0.7022	0.6061	0.7119	0.8112	0.7695	0.8260	0.1939	0.1593	0.2195	0.7963	0.6826	0.8073
DMSC [19]	0.9520	0.8694	0.9520	0.9160	0.8516	0.9160	0.2111	0.1764	0.2318	0.8654	0.7944	0.8654
LALMVC [58]	0.9500	0.8806	0.9500	0.9605	0.9169	0.9605	0.2308	0.1725	0.2458	0.8364	0.8169	0.8600
SMVSC [59]	0.6056	0.4442	0.6056	0.8460	0.7628	0.8460	0.2182	0.1684	0.2439	0.9145	0.8276	0.9145
DAMC [22]	0.9820	0.946	0.9820	0.9650	0.9320	0.9650	0.2560	0.2250	0.2860	0.9236	0.8412	0.9236
AMvC	**0.9900**	**0.9696**	**0.9900**	**0.9660**	**0.9347**	**0.9660**	**0.2912**	**0.2474**	**0.3067**	**0.9309**	**0.8498**	**0.9309**

Denote the ground truth of the clusters as C and the label output by the algorithm as C', and we formulate NMI as

$$\mathrm{NMI}(C, C') = \frac{\mathrm{MI}(C, C')}{\max(H(C), H(C'))}$$

where $H(C)$ and $H(C')$ are the entropies of c_i and c'_j. $\mathrm{MI}(C, C')$ is mutual information metric and the calculation formula is defined as

$$\mathrm{MI}(C, C') = \sum_{c_i \in C, c'_j \in C'} p\left(c_i, c'_j\right) \cdot \log_2 \frac{p\left(c_i, c'_j\right)}{p(c_i) \cdot p\left(c'_j\right)}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that arbitrarily selected documents from the corpus belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected document belongs to the clusters c_i and c'_j at the same time. It is obvious that $\mathrm{NMI}(C, C')$ ranges from 0 to 1, and NMI equals 1 when two sets of clusters are identical and equals 0 when two sets are independent.

Purity can be calculated by

$$\mathrm{Purity} = \frac{n_1}{n_1 + n_2}$$

where n_1 is the number of pairs that are classified together, both in the "real" classification and in the classification obtained by the algorithm. n_2 is the number of pairs that are classified together in the algorithm's classification, but not in the correct classification.

The definition of F-score is as

$$\text{F-score} = \frac{(1 + \lambda^2) \times \text{Precision} \times \text{Recall}}{\lambda^2 \times (\text{Prcision} + \text{Recall})}$$

where λ is used to balance the importance of precision and recall, with $\text{Precision} = (\text{TP}/(\text{TP} + \text{FP}))$ and $\text{Recall} = (\text{TP}/(\text{TP} + \text{FN}))$. In this article, $\lambda = 1$. True positive (TP) means assigning two samples of the same object to the same category, true negative (TN) means assigning samples of two different objects to different categories, False positive (FP) assigns samples of two different objects to the same category, and false negative (FN) is assigning two samples of the same object to different categories.

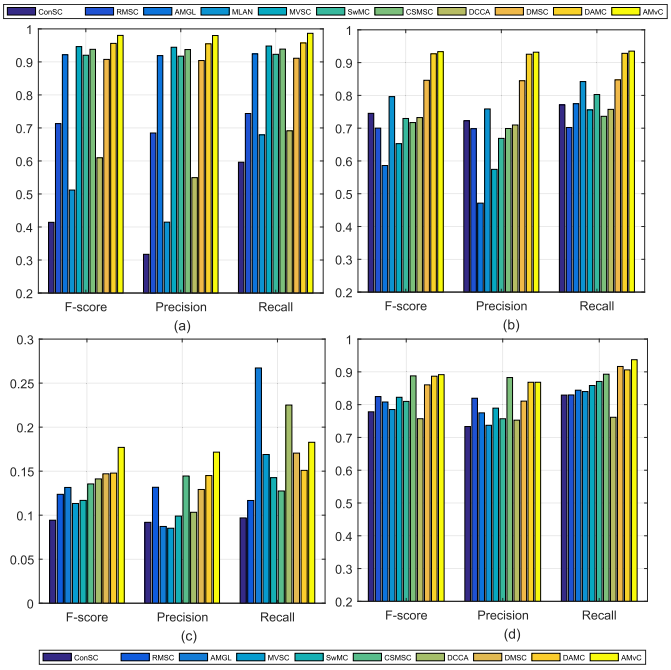


Fig. 3. Experimental results of each method by F-score, precision, and recall metrics on the four datasets: (a) BDGP, (b) HW, (c) CCV, and (d) NH.

4) Implementation Details: We implemented our methods and other nonlinear methods using the public library of PyTorch for deep learning. All the experiments are conducted on the platform running Ubuntu Linux 16.04 and equipped with 64-GB DDR3 memory and NVIDIA Titan Xp graphics processing units (GPUs). We train our model utilizing the Adam [46] optimizer with default parameter settings, and the learning rate was fixed as 0.0001. For each training step, we conducted 30 epochs, and we test the other linear methods one the same environment using MATLAB.

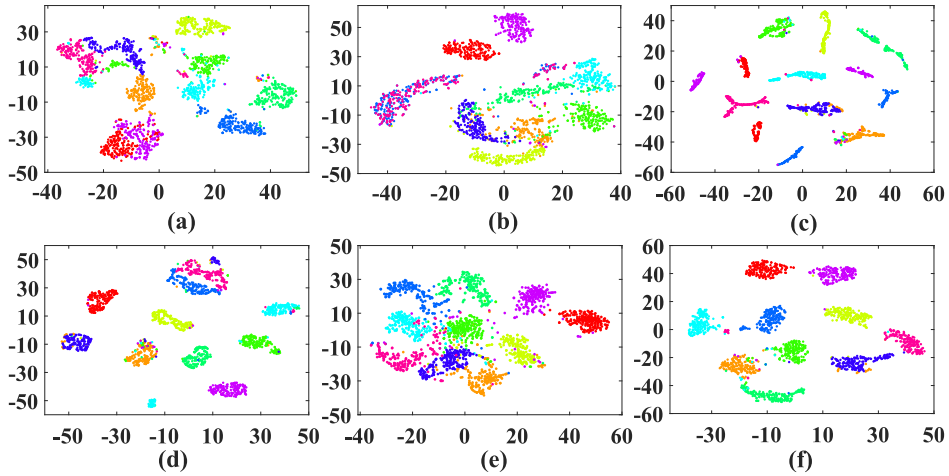


Fig. 4. Visualization of original features for each view and the common latent representations given by different methods with t-SNE [64] on the HW datasets, where (a) original data of first view, (b) original data of second view, (c) MLAN, (d) SwMC, (e) DMSC, and (f) AMvC.

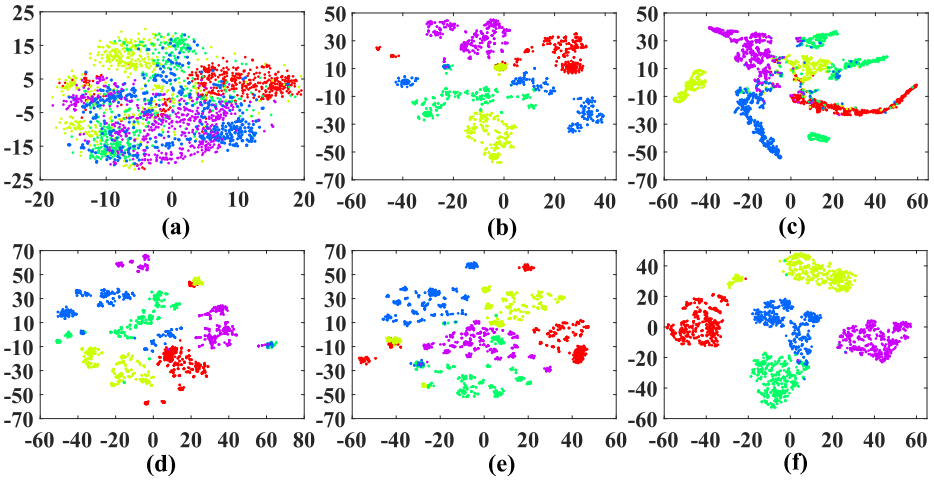


Fig. 5. Visualization of original features for each view and the common latent representations given by different methods with t-SNE [64] on the BDGP datasets, where (a) original data of first view, (b) original data of second view, (c) MLAN, (d) SwMC, (e) DMSC, and (f) AMvC.

Our source code has been uploaded to the Github website: https://github.com/IMKBLE/AmVC.

Since DCCA is designed for two-view data, we first select two views according to the performance in our model as two branches for DCCA and obtain the embedding features of them. Then, we conduct K-means on the concatenated two-branch features.

B. Experimental Results

Table III and Figs. 3–5 show the clustering performance on the first four datasets. Several important observations could be made as follows.

1) *Compared With the Traditional MVC Method:* Traditional MVC algorithms mainly employ linear methods to get the common representations shared by multiple views, and thus, they cannot handle high-dimensional and complex data due to its nonlinear nature. In light of this, our approach fully exploits the discriminative feature by introducing $\ell_{1,2}$-norm regularization, and the adaptive fusion strategy helps our method to further capture the data distribution of each view. As shown in Table III and Fig. 3, our method significantly outperforms baseline methods with a clear improvement, which demonstrates the superiority of our algorithm. There, MLAN is not available and DMSC can only process one view data on the CCV dataset due to the limited memory.
2) *Compared With Deep MVC Methods:* DNNs have shown superior performance on learning feature representations for image/video data recently. However, for the clustering task, deep MVC methods are limited to grid data, which are not straightforward to handle generic features. For example, DMSC [19] is specifically designed for image data and cannot be directly used with irregular data features (e.g., textual features in BDGP). In our experiment, we adopt zero padding to make DMSC available on the BDGP, HW, and CCV datasets, which, however, lowers its performance inevitably. Different from deep MVC methods, our approach builds on the top

TABLE IV
ABLATION STUDY OF AMVC ON THE FOUR DATASETS

Model	$\mathcal{L}_{AE}$	$\mathcal{L}_{GAN}$	$\mathcal{L}_{CLU}$	FU	$\mathcal{L}_{12}$	BDGP			HW			CCV			NH		
						ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity
M_1	✓					0.8092	0.6936	0.8092	0.9400	0.8843	0.9400	0.2455	0.2144	0.2714	0.9164	0.8135	0.9164
M_2	✓	✓				0.9468	0.8796	0.9468	0.9560	0.9085	0.9560	0.2702	0.2348	0.2884	0.9236	0.8349	0.9236
M_3	✓	✓	✓			0.9820	0.9460	0.9820	0.9650	0.9320	0.9650	0.2561	0.2250	0.2860	0.9236	0.8412	0.9236
M_4	✓	✓	✓	✓		0.9856	0.9531	0.9856	0.9655	0.9327	0.9655	0.2851	0.2426	0.3031	0.9291	0.8353	0.9291
AMvC	✓	✓	✓	✓	✓	**0.9900**	**0.9696**	**0.9900**	**0.9660**	**0.9347**	**0.9660**	**0.2912**	**0.2474**	**0.3067**	**0.9309**	**0.8498**	**0.9309**

of a fully connected network, and thus, achieves higher flexibility and generalizability for MVC. In addition, DCCA, DMSC, and DAMC all focus on consistent subspace learning. However, they ignore that the discriminative feature learning of common subspace is important to MVC tasks. Therefore, the proposed AmVC introduces $\ell_{1,2}$-norm regularization to learn discriminative features and improve the MVC performance.

3) *Different Performances on Different Datasets:* From Table III, we can see that all methods on the CCV dataset have poor clustering performances. Considering the characteristics of these datasets, we find that CCV is a video dataset with semantic categories, while other datasets are image datasets. Therefore, the reason for the above-mentioned phenomenon may be that video samples are much more complex than image samples, and the labels indicating data semantic category result in the diversity in data of the same category. In addition, video data contains too many irrelevant objects and noise. Thus, it is difficult to extract useful features from the CCV dataset.
4) *Visualization Results:* Figs. 4 and 5 show the visualization clustering results on the BDGP and HW datasets. Fig. 4(a) provides a t-distributed stochastic neighbor embedding (t-SNE) [64] visualization for feature embeddings in terms of every single view, three competitive compared methods, and our proposed AMvC on the HW dataset. In detail, we apply t-SNE on the common-view feature representations (e.g., the latent layer features in AMvC) given by different methods, respectively. As can be seen, our approach exhibits a clearer and more compact cluster structure than all the other methods and original data. A similar observation could be found on the BDGP dataset, as shown in Fig. 5(b). This clearly shows the nice cluster-structured property given by our deep embedded clustering layer, as it explicitly guides our feature learning process with a clustering purpose.

C. Ablation Study

The purpose of the ablation study in this section is to study the influence of the AE loss $\mathcal{L}_{\text{AE}}$, the GAN loss $\mathcal{L}_{\text{GAN}}$, the clustering loss $\mathcal{L}_{\text{CLU}}$, the weighted adaptive fusion layer FU, and the $\ell_{1,2}$-norm loss $\mathcal{L}_{12}$ on the clustering performance. We report experimental results on the BDGP, HW, CCV, and NH datasets with different ablated models in Table IV.

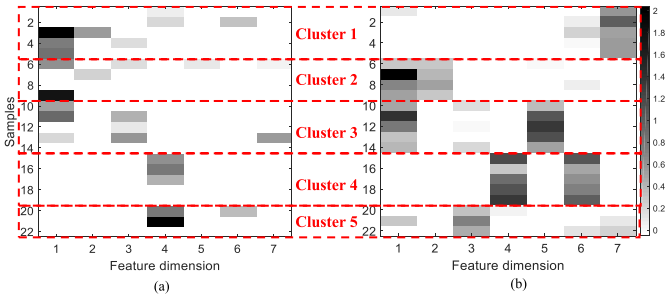


Fig. 6. Illustration of $\ell_{1,2}$-norm discriminability. (a) Feature without $L_{1,2}$-norm regularization. (b) Feature with $L_{1,2}$-norm regularization.

On the one hand, M2 achieves significantly better results than M1 in terms of ACC, NMI, and purity, by utilizing the GAN loss. Specifically, there is a 13.76% improvement, a 1.60% improvement, a 2.47% improvement, and a 0.72% improvement in the BDGP dataset, HW dataset, CCV dataset, and NH dataset by ACC, respectively. This clearly shows the effectiveness of using discriminator networks for each view, which could further capture the data distribution and disentangle the latent space. On the other hand, M3 and M4 consistently outperform M2, especially in the BDGP dataset by ACC, which demonstrates the clustering loss is useful to guide $\mathbf{Z}$ for a better clustering performance. The discriminator network and the deep embedding clustering layer empower the proposed method to handle the hard sample pairs. Moreover, M4 outperforms M3 by leveraging the adaptive fusion layer FU to enhance the complementary information across different views further. Finally, our full model improves M4 further by using a $\mathcal{L}_{12}$ loss, where the $\ell_{1,2}$-norm can make the extracted features more distinguishable. To sum up, the ablation study demonstrates that our proposed discriminative networks, the adaptive fusion layer, and the $\ell_{1,2}$-norm are effective for the MVC task.

D. Discussion of the $\ell_{1,2}$-Norm

We randomly select 22 samples and seven dimensional features from five clusters on the CUB dataset. Each class is circled by a red box. Fig. 6 shows the feature extraction results without/with $\ell_{1,2}$-norm regularization. From Fig. 6(a), according to the feature without $\ell_{1,2}$-norm regularization, samples of clusters 1–3 are similar with each other, and samples of cluster 4 are similar with those of cluster 5. However, from Fig. 6(b), the feature with $\ell_{1,2}$-norm regularization, and samples from different clusters have fewer similar values in each

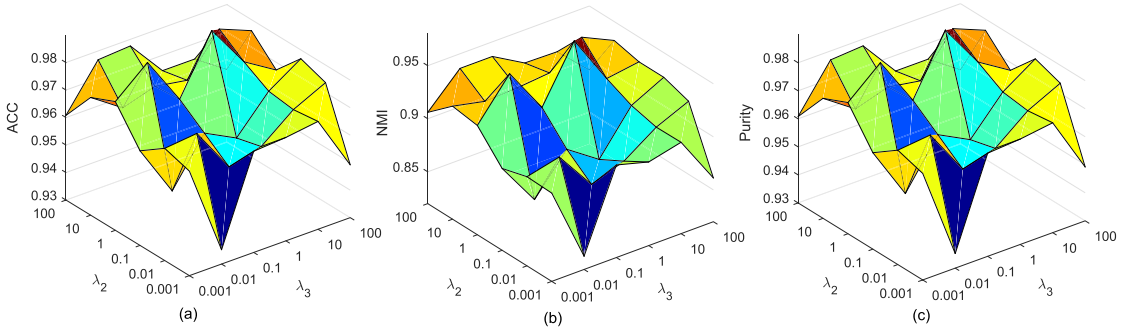


Fig. 7. Impact of parameters on clustering performance on BDGP dataset. (a) ACC. (b) NMI. (c) Purity.

TABLE V
CLUSTERING PERFORMANCE OF AVAILABLE DEEP MVC METHODS ON LARGE-SCALE DATASET

Method	ACC	NMI	Purity
DCCA	0.468	0.426	0.505
DAMC	0.651	0.562	0.659
AMvC	**0.665**	**0.573**	**0.665**

TABLE VI
INFLUENCE OF HYPERPARAMETER α ON BDGP DATASET

α	0.01	0.1	1	10	100
ACC	**99.00**	**99.00**	**99.00**	97.84	96.24
NMI	**97.06**	96.96	96.96	94.04	91.10
Purity	**99.00**	**99.00**	**99.00**	97.84	96.24

feature dimension, which means the discriminability of the feature with $\ell_{1,2}$-norm regularization is significantly increased. Thus, through the constraint of $\ell_{1,2}$-norm, the extracted feature could improve the clustering performance.

E. Clustering on Large-Scale Dataset

To demonstrate our approach can be applied to the large-scale dataset efficiently, we compare the proposed AMvC with DCCA and DAMC, on the two-view MNIST dataset provided by [47]. The MNIST dataset contains 70 000 samples. Traditional multiview methods might not be scalable to large-scale datasets due to their optimization methods. We cannot run these methods on MNIST because of the out-of-memory issue or an extremely high time complexity. DMSC adopts full-batch training since it develops a self-expressive layer. Therefore, it is nontrivial to test DMSC on MNIST due to the limited GPU memory. Differently, our model is based on DNNs and is trained with a stochastic mini-batch optimization strategy, making the proposed method scalable to large-scale datasets. As shown in Table V, our method consistently outperforms other methods with a clear improvement, which validates the effectiveness of AMvC on the large-scale dataset.

F. Parameter Analysis

1) Impact of α: The notion α is the freedom degree of the student's t-distribution. In this part, we discuss the impact of α on clustering performance. We conduct the experiment on the BDGP dataset. Table VI gives the clustering performance with different α. As can be seen, the proposed method is robust to a smaller α value.

2) Impact of λ: In our model, there are three regularization parameters λ_1, λ_2, and λ_3. According to our experience, we set the parameter λ_1 for the generative adversarial loss as 1. Then, we vary the regularization parameters λ_2 of the distribution consistent loss and λ_3 of the $\ell_{1,2}$-norm regularization in the range of $\{0.001, 0.01, 0.1, 1, 10, 100\}$. Since the strategies of setting parameters are the same on all four datasets, we only show the impact of parameters on the BDGP dataset for simplicity. From the results in Fig. 7, we can see our method can achieve the best ACC and NMI values on the BDGP dataset when $\lambda_2 = 1, \lambda_3 = 1$; the proposed model AmVC is stable since varying parameters has little influence on the clustering performance.

G. Computational Complexity

We theoretically analyze the computational complexity. For simplicity, we assume the outputs of all the layers have the same dimension of p, and the original features of all views are of the same dimension of d. Suppose V, L, and N are the number of views, layers, and samples, respectively, and the time complexity of the proposed model is $O(VNLpd)$.

V. CONCLUSION

In this article, we proposed a novel deep MVC model named AMvC. By using the shared weights and the adaptive fusion parameters, AMvC jointly embeds multiview data to a common low-dimensional subspace with nonlinear mappings. Upon the common subspace, we employ $\ell_{1,2}$-norm regularizer to make the feature representations more discriminative. Finally, the reported results demonstrate the superiority of our proposed method when compared with other outstanding methods. The work mainly focuses on MVC, and in the future, we will further consider supervised scenarios, aiming at partial multiview classification.

REFERENCES

[1] B. Wang, Y. Hu, J. Gao, Y. Sun, F. Ju, and B. Yin, "Learning adaptive neighborhood graph on Grassmann manifolds for video/image-set subspace clustering," *IEEE Trans. Multimedia*, vol. 23, pp. 216–227, 2021.

[2] Z. Tao, H. Liu, H. Fu, and Y. Fu, "Multi-view saliency-guided clustering for image cosegmentation," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4634–4645, Sep. 2019.

[3] W. W. Y. Ng, X. Tian, W. Pedrycz, X. Wang, and D. S. Yeung, "Incremental hash-bit learning for semantic image retrieval in nonstationary environments," *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 3844–3858, Nov. 2019.

[4] B. Zhao, X. Li, and X. Lu, "Property-constrained dual learning for video summarization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 3989–4000, Oct. 2020.

[5] C. Du, C. Du, and H. He, "Multimodal deep generative adversarial models for scalable doubly semi-supervised learning," *Inf. Fusion*, vol. 68, pp. 118–130, Apr. 2021.

[6] L. Nie, M. Liu, and X. Song, "Multimodal learning toward micro-video understanding," *Synth. Lect. Image, Video, Multimedia Process.*, vol. 9, no. 4, pp. 1–186, Sep. 2019.

[7] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proc. ICDM*, vol. 4, Nov. 2004, pp. 19–26.

[8] G. Sun, Y. Cong, Y. Zhang, G. Zhao, and Y. Fu, "Continual multiview task learning via deep matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 139–150, Jan. 2021.

[9] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2843–2849.

[10] M. Chen, L. Huang, C.-D. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *Proc. AAAI*, 2020, pp. 3513–3520.

[11] S. Huang, I. Tsang, Z. Xu, and J. C. Lv, "Measuring diversity in graph learning: A unified framework for structured multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, early access, Mar. 24, 2021, doi: 10.1109/TKDE.2021.3068461.

[12] Q. Wang, W. Xia, Z. Tao, Q. Gao, and X. Cao, "Deep self-supervised t-SNE for multi-modal subspace clustering," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1748–1755.

[13] S. Huang, Z. Kang, and Z. Xu, "Auto-weighted multi-view clustering via deep matrix decomposition," *Pattern Recognit.*, vol. 97, Jan. 2020, Art. no. 107015.

[14] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "Marginalized multiview ensemble clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 600–611, Feb. 2020.

[15] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proc. AAAI*, 2017, pp. 2921–2927.

[16] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *Proc. AAAI*, 2017, pp. 2408–2414.

[17] X. Wang, X. Guo, Z. Lei, C. Zhang, and S. Z. Li, "Exclusivity-consistency regularized multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 923–931.

[18] Q. Wang, J. Cheng, Q. Gao, G. Zhao, and L. Jiao, "Deep multi-view subspace clustering with unified and discriminative learning," *IEEE Trans. Multimedia*, vol. 23, pp. 3483–3493, 2020.

[19] M. Abavisani and V. M. Patel, "Deep multimodal subspace clustering networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1601–1614, Dec. 2018.

[20] P. Zhu, B. Hui, C. Zhang, D. Du, L. Wen, and Q. Hu, "Multi-view deep subspace clustering networks," 2019, *arXiv:1908.01978*.

[21] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[22] Z. Li, Q. Wang, Z. Tao, Q. Gao, and Z. Yang, "Deep adversarial multi-view clustering network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2952–2958.

[23] S. Yu *et al.*, "Optimized data fusion for kernel k-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, May 2012.

[24] S. Huang, Z. Kang, I. W. Tsang, and Z. Xu, "Auto-weighted multi-view clustering via kernelized graph learning," *Pattern Recognit.*, vol. 88, pp. 174–184, Apr. 2019.

[25] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, and X. Huang, "Robust subspace clustering for multi-view data by exploiting correlation consensus," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3939–3949, Nov. 2015.

[26] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.

[27] P. L. Lai and C. Fyfe, "Kernel and nonlinear canonical correlation analysis," *Int. J. Neural Syst.*, vol. 10, pp. 365–377, Oct. 2000.

[28] S. M. Kakade and D. P. Foster, "Multi-view regression via canonical correlation analysis," in *Proc. Int. Conf. Comput. Learn. Theory*. Berlin, Germany: Springer, 2007, pp. 82–96.

[29] Z. Kang, X. Lu, J. Yi, and Z. Xu, "Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1881–1887.

[30] M. Wu and N. Goodman, "Multimodal generative models for scalable weakly-supervised learning," in *Proc. NIPS*, Dec. 2018, pp. 5580–5590.

[31] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. ICML*, 2016, pp. 1558–1566.

[32] M. Chen and L. Denoyer, "Multi-view generative adversarial networks," in *Proc. ECML PKDD*, 2017, pp. 175–188.

[33] C. Du *et al.*, "Semi-supervised deep generative modelling of incomplete multi-modality emotional data," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 108–116.

[34] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. ICML*, 2013, pp. 1247–1255.

[35] A. Benton, H. Khayrallah, B. Gujral, D. A. Reisinger, S. Zhang, and R. Arora, "Deep generalized canonical correlation analysis," 2017, *arXiv:1702.02519*.

[36] Y. Xie *et al.*, "Joint deep multi-view learning for image clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 11, pp. 3594–3606, Nov. 2021.

[37] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning: Objectives and optimization," 2016, *arXiv:1602.01024*.

[38] C. Zhang *et al.*, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.

[39] S. Wang, Z. Chen, S. Du, and Z. Lin, "Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 21, 2021, doi: 10.1109/TPAMI.2021.3082632.

[40] Z. Huang, J. T. Zhou, X. Peng, C. Zhang, H. Zhu, and J. Lv, "Multi-view spectral clustering network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2563–2569.

[41] X. Sun, M. Cheng, C. Min, and L. Jing, "Self-supervised deep multi-view subspace clustering," in *Proc. ACML*, 2019, pp. 1001–1016.

[42] Q. Wang, Z. Ding, Z. Tao, Q. Gao, and Y. Fu, "Generative partial multi-view clustering with adaptive fusion and cycle consistency," *IEEE Trans. Image Process.*, vol. 30, pp. 1771–1783, 2021.

[43] Q. Wang, Z. Ding, Z. Tao, Q. Gao, and Y. Fu, "Partial multi-view clustering via consistent GAN," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 1290–1295.

[44] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. ICML*, 2016, pp. 478–487.

[45] D. Ming and C. Ding, "Robust flexible feature selection via exclusive L21 regularization," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3158–3164.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[47] C. Shang, A. Palmer, J. Sun, K.-S. Chen, J. Lu, and J. Bi, "VIGAN: Missing view imputation with generative adversarial networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 766–775.

[48] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. NIPS*, 2002, pp. 849–856.

[49] A. Asuncion and D. Newman, "UCI machine learning repository," Irvine, CA, USA, Tech. Rep., 2007.

[50] X. Cai, H. Wang, H. Huang, and C. Ding, "Joint stage recognition and anatomical annotation of Drosophila gene expression patterns," *Bioinformatics*, vol. 28, no. 12, pp. i16–i24, Jun. 2012.

[51] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, "Consumer video understanding: A benchmark database and an evaluation of human and machine performance," in *Proc. 1st ACM Int. Conf. Multimedia Retr. (ICMR)*, 2011, p. 29.

[52] S. Luo, C. Zhang, W. Zhang, and X. Cao, "Consistent and specific multi-view subspace clustering," in *Proc. AAAI*, 2018, pp. 3730–3737.

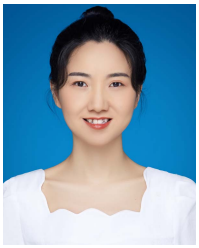
[53] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[54] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Proc. NIPS*, 2011, pp. 1413–1421.

[55] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. AAAI*, 2014, pp. 2149–2155.

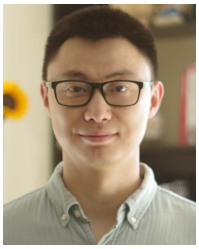
[56] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. AAAI*, 2015, pp. 2750–2756.

[57] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2564–2570.
[58] D. Xie, X. Zhang, Q. Gao, J. Han, S. Xiao, and X. Gao, "Multiview clustering by joint latent representation and similarity learning," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4848–4854, Nov. 2020.
[59] M. Sun *et al.*, "Scalable multi-view subspace clustering with unified anchors," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1–9.
[60] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.
[61] R. Varshavsky, M. Linial, and D. Horn, "Compact: A comparative package for clustering assessment," in *Proc. Int. Symp. Parallel Distrib. Process. Appl.* Berlin, Germany: Springer, 2005, pp. 159–167.
[62] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman, "Incremental hierarchical clustering of text documents," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2006, pp. 357–366.
[63] L. Lovász and M. D. Plummer, *Matching Theory*, vol. 5. Providence, RI, USA: AMS, 2009, pp. 42–46.
[64] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.



Qianqian Wang received the B.Eng. degree in communication engineering from Lanzhou University of Technology, Lanzhou, China, in 2014, the Ph.D. degree from Xidian University, Xi'an, China, in 2019.

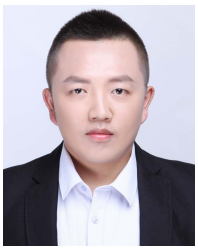
She was a Visiting Scholar with Northeastern University, Boston, MA, USA, from 2017 to 2018. She is currently a Lecturer with the School of Telecommunications Engineering, Xidian University. Her research interests include pattern recognition, principal component analysis, multiview clustering, and partial multiview clustering.



Zhiqiang Tao received the B.Eng. degree in software engineering from the School of Computer Software, Tianjin University, Tianjin, China, in 2012, the M.Eng. degree in computer science from the School of Computer Science and Technology, Tianjin University, in 2015, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA, in 2020.

He has been a tenure-track Assistant Professor with the Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA, USA, since 2020. His research interests include representation learning, sequence mining, user modeling, and interpretability.

Dr. Tao has served as a Reviewer for several IEEE TRANSACTIONS, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS (TCYB). He has served as a Program Committee Member for conferences, including NeurIPS, International Conference on Machine Learning (ICML), ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), International Conference on Learning Representations (ICLR), Association for the Advancement of Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), IEEE International Conference on Computer Vision and pattern Recognition (CVPR), and IEEE International Conference on Computer Vision (ICCV).



Wei Xia (Graduate Student Member, IEEE) received the B.Eng. degree in communication engineering from Lanzhou University of Technology, Lanzhou, China, in 2018. He is currently pursuing the Ph.D. degree in communication and information system with Xidian University, Xi'an, China.

His research interests include multiview clustering, low-rank representation, and deep neural networks.

Quanxue Gao received the B.Eng. degree from Xi'an Highway University, Xi'an, China, in 1998, the M.S. degree from Lanzhou University of Technology, Lanzhou, China, in 2001, and the Ph.D. degree from Northwestern Polytechnical University, Xi'an, in 2005.

He was an Associate Researcher with the Biometrics Center, The Hong Kong Polytechnic University, Hong Kong, from 2006 to 2007. From 2015 to 2016, he was a Visiting Scholar with the Department of Computer Science, The University of Texas at Arlington, Arlington, TX, USA. He is currently a Professor with the School of Telecommunications Engineering, Xidian University, Xi'an, and also a Key Member of the State Key Laboratory of Integrated Services Networks. He has authored 60 technical articles in refereed journals and proceedings, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEM (TNNLS), IEEE TRANSACTIONS ON CYBERNETICS (TC), IEEE International Conference on Computer Vision and pattern Recognition (CVPR), Association for the Advancement of Artificial Intelligence (AAAI), and International Joint Conference on Artificial Intelligence (IJCAI). His current research interests include pattern recognition and artificial intelligence.

Xiaochun Cao (Senior Member, IEEE) received the B.E. and M.E. degrees in computer science from Beihang University, Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, FL, USA, in 2006.

He was a Research Scientist with Object Video, Inc., Reston, VA, USA, for about three years. From 2008 to 2012, he was a Professor with Tianjin University, Tianjin, China.

Dr. Cao is on the Editorial Boards of the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON MULTIMEDIA. His Ph.D. dissertation was nominated for the university-level Outstanding Dissertation Award.

Licheng Jiao (Fellow, IEEE) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

From 1990 to 1991, he was a Post-Doctoral Fellow with the National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an. He has been a Professor with the School of Electronic Engineering, Xidian University, where he is currently the Director of the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education of China. He has led 40 major scientific research projects. He has authored or coauthored more than 20 monographs and 100 articles in international journals and conferences. His current research interests include image processing, natural computation, machine learning, and intelligent information processing.

Dr. Jiao is a member of the IEEE Xi'an Section Executive Committee, the Chairman of the Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, a Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, and an Expert of the Academic Degrees Committee of the State Council.