Self-Supervised Deep Multiview Spectral Clustering

Linlin Zong, Faqiang Miao, Xianchao Zhang, Wenxin Liang, *Member, IEEE*, and Bo Xu

***Abstract*—Multiview spectral clustering has received considerable attention in the past decades and still has great potential due to its unsupervised integration manner. It is well known that pairwise constraints boost the clustering process to a great extent. Nevertheless, the constraints are usually marked by human beings. To ameliorate the performance of multiview spectral clustering and alleviate the consumption of human resources, we propose self-supervised multiview spectral clustering with a small number of automatically retrieved pairwise constraints. First, the fused multiple autoencoders are used to extract the latent consistent feature of multiple views. Second, the pairwise constraints are achieved based on the commonality among multiple views. Then, the pairwise constraints are propagated through the neural network with historical memory. Finally, the propagated constraints are used to optimize the fused affinity matrix of spectral clustering. Our experiments on four benchmark datasets show the effectiveness of our proposed approach.**

***Index Terms*—Constraint propagation network, deep multiview, self-supervised, spectral clustering.**

I. Introduction

CLUSTERING, which partitions a set of data into several groups, is a fundamental task of data mining. With the rapid development of information technology, most of data is presented in different views, such as text, image, videos *et al.* These data with multiple views are known as multiview data. The task of clustering multiview data is known as multiview clustering [1].

Multiview clustering makes full use of the compatible and complementary information of multiple views and improves the clustering performance of multiview data. The main challenge of multiview clustering is to integrate the knowledge among multiple views. The existing algorithms integrate multiple representations to identify clusters in an unsupervised manner. The unsupervised learning pattern limits the deep integration of knowledge from multiple views. Fortunately, the constrained multiview clustering [2], in which limited domain knowledge is available, extends the classical unsupervised setting of the multiview clustering. Nevertheless, the constraints are usually marked by human beings and are not guaranteed to be accurate.

Recently, self-supervised clustering has alleviated the consumption of human resources in achieving constraints. Self-supervised clustering trains a model without the need for human annotation. In the multiview clustering literatures, one self-supervised multiview clustering has been proposed [3]. It integrates spectral clustering and affinity learning into a deep learning framework. The affinity matrix is constructed according to the high-level and cluster-driven representation, in which the pseudo label and clustering label of each data are constrained to be consistent. However, the label of a cluster may not permanent in each iteration, e.g., one cluster may be tagged as 1 in current iteration but are tagged as 2 in the next iteration, which will lead to chaotic consistency between the pseudo label and clustering label. Moreover, when the data is hard to be correctly grouped, the label consistency of all the data may not work.

In light of the limitations of the existing multiview clustering methods, we propose self-supervised deep multiview spectral clustering with a small number of automatically retrieved pairwise constraints. Different from the labels, the pairwise constraints only require to learn whether two instances belonging to the same cluster or not rather than the exact labels. In this article, we first propose the deep multiview spectral clustering. On account of the high effectiveness of deep clustering for feature extraction tasks, we use deep autoencoders to extract the low-dimensional latent feature representations from multiple views and fuse them simultaneously. After that, the orthogonal representation of the fusion features is obtained by Cholesky decomposition. Based on the orthogonal representation, the affinity matrix containing multiple knowledge are constructed for spectral clustering. Furthermore, we propose the self-supervised method to optimize the affinity matrix further. We achieve the pairwise constraints by utilizing the commonality among multiview data. After that, we propose the pairwise constraint propagation network consists of neuron-like nodes organized into successive layers. The output of the network is a complete pairwise constraint matrix used to optimize the affinity matrix further. Experimental results on four multiview datasets demonstrate the effectiveness of the proposed method. In short summary, the contributions of this article are three-folds:

Manuscript received 9 August 2021; revised 12 March 2022 and 28 May 2022; accepted 23 July 2022. Date of publication 9 August 2022; date of current version 1 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62006034, Grant 61806034, Grant 61876028, and Grant 61972065; in part by the Natural Science Foundation of Liaoning Province under Grant 2021-BS-067; in part by the Fundamental Research Funds for the Central Universities under Grant DUT21RC(3)015 and Grant DUT19RC(3)048; in part by the State Key Laboratory of Novel Software Technology, Nanjing University under Grant KFKT2022B41; and in part by the Dalian High-level Talent Innovation Support Plan under Grant 2021RQ056. *(Corresponding author: Bo Xu.)*

Linlin Zong is with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian 116620, China, and also with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: llzong@dlut.edu.cn).

Faqiang Miao is with the Institute of Farmland Irrigation of CAAS, Xinxiang 453003, China (e-mail: fqmiao@yeah.net).

Xianchao Zhang and Wenxin Liang are with the School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: xczhang@dlut.edu.cn; wxliang@dlut.edu.cn).

Bo Xu is with the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China (e-mail: xubo@dlut.edu.cn).

Digital Object Identifier 10.1109/TNNLS.2022.3195780

2162-237X © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 02,2024 at 02:20:00 UTC from IEEE Xplore. Restrictions apply.

1) We design a view fusion module which integrates the orthogonal constraint to learn the latent consistent feature friendlier for spectral clustering.
2) We obtain the pairwise constraints by mining multiview consistency information, which improves the quality of constraints and makes the pairwise constraint propagation feasible.
3) We design a sequential pairwise constraint propagation network to diffuse the pairwise constraint and enhance the affinity matrix of spectral clustering.

II. Related Work

A. Unsupervised Multiview Clustering

From the perspective of basic algorithm exploration, multiview clustering algorithms can roughly be separated into two groups: traditional clustering-based approaches and deep clustering-based approaches. Traditional clustering-based approaches usually minimize the diversity of different views [4]–[7]. On account of high effectiveness of deep clustering for feature extraction, we focus on deep clustering-based approaches in this article. The deep multiview clustering is dedicated to extracting common features from multiview data using deep neural networks (DNNs).

On the one hand, autoencoder-based approaches demonstrate its effectiveness in wide range of areas. For example, Ngiam *et al.* [8] proposed a series of frameworks for deep multimodal learning based on autoencoders. Wang *et al.* [9] extended deep canonical correlation analysis (DCCA) [10] via adding an autoencoder regularization term to DCCA. Deep multimodal subspace clustering networks (DMSCN) [11] applied convolutional neural network (CNN)-based autoencoder that contains three main stages: multiview encoder, self-expressive layer, and multiview decoder. And then learned the affinity matrix that is used to cluster the multiview datas through self-expressive layer. Nousi and Tefas [12] manipulate the reconstruction space of an autoencoder (AE) to guide the low-dimensional representation to form more compact and meaningful clusters. Adversarial incomplete multi-view clustering (AIMC) [13] used the autoencoder to seek the common latent space of multiview data, meanwhile, AIMC performs missing data inference through the generative adversarial network.

On the other hand, Zhao *et al.* [14] adopted semi-nonnegative matrix factorization to learn the hierarchical semantics of multiview data and enforced the nonnegative representation of each view in the final layer to maximize the mutual information. Wang *et al.* [15] used the common representation encoded by one view to generate the missing data of the corresponding view by generative adversarial networks. multiview spectral clustering network (MvSCN) [16] incorporated the local invariance defined by a deep metric learning network within each view and the consistency across different views into an objective function to obtain the features of view sharing. Zhang *et al.* [17] proposed an end-to-end multiview fusion clustering framework which using neural networks to simulate nonnegative matrix decomposition.

B. Constrained Multiview Clustering

Constrained multiview clustering partition multiview data with the help of constrained information, e.g., the manually labeled data and the pairwise constraints of pairs of data. Eric *et al.* [18] used local similarity measures to propagate pairwise constraints to instances that can be mapped to other views and allowed the propagated constraints to be transmitted across views via partial mappings. Constrained non-negative matrix factorization based multi-view clustering (CMVNMF) [19] solved the problem of multiview clustering of unmapped data by introducing interview constraints to establish disagreement between each pair of views for guiding nonnegative matrix factorization. MultisSC [20] proposed an alternate minimization method that utilized constrained prior information to achieve a sparse representation of each high-dimensional data point relative to other data points in the same view and to maximize the correlation between the representations of different viewpoints. Zong *et al.* [21] provides an active internal view constraint selection strategy that extends the CMVNMF by querying the relationship between the most influential sample and the sample farthest from the existing constraint set. Cai *et al.* [22] introduced an auxiliary matrix for each view to learn the representation and constructed a label constraint matrix shared by all views to ensure that the label information is fused into the new fused representation, then used auxiliary matrix to integrate the complementary information from different views. Cai *et al.* [2] proposed an orthogonal constraint to obtain a representation of each view and used coregularization to integrate complementary information from different views.

C. Self-Supervised Clustering

Self-supervised clustering is an emerging method, which accomplishes the clustering task with the assist of classification task and the labels for classification are obtained automatically rather than human annotation. Zhang *et al.* [23] introduced a dual self-supervision that exploits the output of spectral clustering to supervise the training of the feature learning module and the self-expression module. Sharma *et al.* [24] proposed a self-supervised Siamese network that can be trained without the need for video/track-based supervision. Alwassel *et al.* [25] get the pseudo-labels through clustering and present three different approaches for training video models from self-supervised audio–visual information. With the development of research, self-supervised learning is introduced into multiview learning, Sun *et al.* [3] constructed affinity matrix according to the high-level and cluster-driven representation, in which the pseudo label and clustering label of each data are constrained to be consistent.

III. Review on Spectral Clustering

Spectral clustering partitions data into clusters according to their similarities. Given a set of data $X = \{x_1, x_2, \ldots, x_n\}$ belonging to k clusters, Ng *et al.* [26] proposed a normalized spectral clustering algorithm. The brief flow of the algorithm is summarized as follows:

1) Construct the affinity matrix W;

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 02,2024 at 02:20:00 UTC from IEEE Xplore. Restrictions apply.

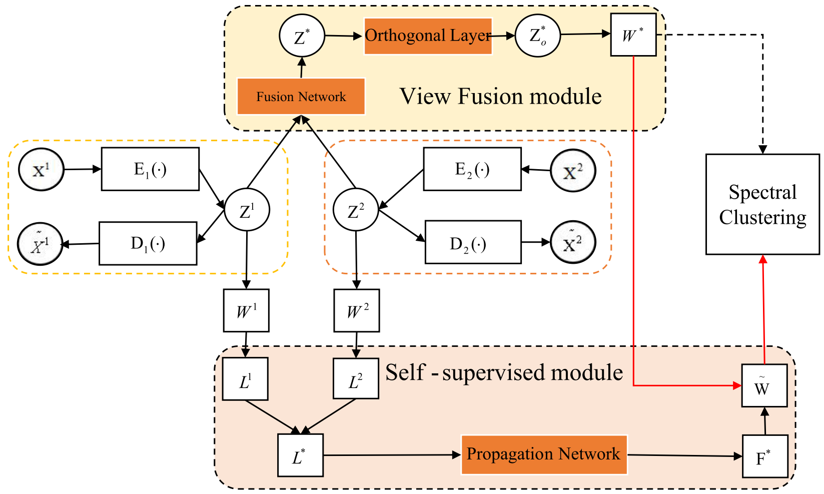

Fig. 1. Flow of the proposed method. The middle and upper parts are the deep multiview spectral clustering. The middle part is the autoencoders of each view data and the upper part is the view fusion module to extract view shared features. The bottom part is the self-supervised module to optimize the affinity matrix.

2) Compute the normalized Laplacian matrix $S = D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix and $D_{ii} = \sum_{j=1}^{n} W_{ij}$;
3) Compute the first k eigenvectors of S and construct a matrix with the eigenvectors as column;
4) Cluster the eigenvector matrix with k-means.

IV. Proposed Method

Given multiview data $X = \{X^1, X^2, \ldots, X^m\}$, where m is the number of views and $X^m = \{x_1^m, x_2^m, \ldots, x_n^m\}$ is the set of instances in the mth view, where n is the number of instances and $x_i^m \in R^{d_m}$ is the ith instance in the mth view. We are committed to clustering the multiview data X into k clusters $C = \{c_1, c_2, \ldots, c_k\}$ and each view has the same ground truth label. In the following, we first introduce a deep multiview spectral clustering that can explore affinity matrix (as shown in the middle and upper parts of Fig. 1). And then introduce a self-supervised module that can collect automatically pairwise constraints by mining multiview consistency and a constraint propagation network that propagate paired constraint information for optimizing affinity matrix. (as shown in the bottom part of Fig. 1).

A. Deep Multiview Spectral Clustering

One of the core steps of spectral clustering is to obtain an affinity matrix. The proposed Deep Multi-view Spectral Clustering (DMvSC) (short for deep multiview spectral clustering) algorithm first extracts the deep latent feature of multiple views and constructs the affinity matrix containing multiple knowledge. After that, the affinity matrix is subsequently applied to traditional spectral clustering.

1) Deep Latent Feature Extraction: For multiview data, we use multiple autoencoders to learn the low-dimensional latent representation of each view. The feature extraction process consists of two parts: autoencoder learning stage (as shown in the middle part of Fig. 1) and fusion stage (as shown in the upper part of Fig. 1).

For the autoencoder stage, first, we make use of encoder to extract the feature representation of each multiview. Define $E_j(\cdot)$ as the encoder mapping function, we can get $E_j(X^j) = Z^j$, where Z^j is the latent features of the jth view. Second, we take advantage of decoder to reconstruct X^j by decoding Z^j. Define $\tilde{X}^j$ as the reconstructed data of the instance X^j, and $D_j(\cdot)$ is the decoder mapping function, we can obtain $D_j(Z^j) = \tilde{X}^j$. To obtain better latent feature, we use the mean squared error (MSE) loss $\mathcal{L}_r$ to define the reconstruction loss to optimize the parameters of the autoencoders

$$\mathcal{L}_r = \frac{1}{m}\sum_{j=1}^{m} \|X^j - D_j(E_j(X^j)))\|^2. \tag{1}$$

In the fusion layer, to extract the latent features shared by multiview more effectively, we chose to fuse the features of each view while encoding. Denote Fusion$(\cdot)$ as the fusion process and Z^* is the fused feature of multiview, then, we have $Z^* = \text{Fusion}(Z^1, Z^2, \ldots, Z^m)$. We define the fusion loss $\mathcal{L}_f$ for optimizing parameters of the fusion network

$$\mathcal{L}_f = \frac{1}{m}\sum_{j=1}^{m} \|Z^* - Z^j\|^2. \tag{2}$$

The reconstruction loss $\mathcal{L}_r$ constantly optimizes Z^j to get $\tilde{X}$ similar to X, while the fusion loss $\mathcal{L}_f$ also optimizes Z^j to get a better Z^*. Therefore, the two losses promote each other in the training process. Therefore, we jointly optimize the reconstruction loss $\mathcal{L}_r$ and the fusion loss $\mathcal{L}_f$

$$\min\ \mathcal{L}_r + \beta\mathcal{L}_f. \tag{3}$$

2) Construct Affinity Matrix: The multiple views share the same ground-truth affinity matrix. We calculate the matrix based on Z^*. In addition, an affinity matrix for spectral clustering has the following characteristics: symmetry, nonnegativity, and the rank approaches to the number of clusters k.

To meet the above characteristics, we first add an orthogonal layer to calculate the orthogonal representation Z_o^* of Z^*. The orthogonal constraint is implemented by Cholesky decomposition [27]

$$Z_o^* = Z^*((((Z^*)^{\mathrm{T}}Z^*)^{-1})^{\mathrm{T}}). \tag{4}$$

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 02,2024 at 02:20:00 UTC from IEEE Xplore. Restrictions apply.

Note that, the full rankness of $(Z_o^*)^{\mathrm{T}} Z_o^*$ could be easily guaranteed by adding a sufficiently small number (e.g., 10^{-5}) at the diagonal elements without loss of generality. By integrating the orthogonal constraint, the learned latent consistent representation is friendlier for spectral clustering.

Then, the gaussian kernel function is used to construct the affinity matrix W^* of Z_o^*

$$W^* = \exp\left(-\frac{d\left(Z_o^*, Z_o^*\right)}{2\sigma^2}\right) \tag{5}$$

where $d(\cdot,\cdot)$ is a distance measure (e.g., Euclidean distance) and σ is the length scale parameter.

B. Self-Supervised DMvSC

The DMvSC introduced above is an unsupervised clustering model. On the basis of DMvSC, we further propose the self-supervised DMvSC (named as SDMvSC) which optimizes the affinity matrix using pairwise constraints that is obtained automatically from existing knowledge (as shown in the bottom part of Fig. 1).

1) Capture Pairwise Constraints: To introduce the self-supervised learning model, we need to show some knowledge about pairwise constraints. We denote a set of must-link pairwise constraints as $\mathcal{M} = \{(x_a^j, x_b^j) : l_a^j = l_b^j\}$ and a set of cannot-link pairwise constraints as $\mathcal{C} = \{(x_a^j, x_b^j) : l_a^j \neq l_b^j\}$, where l_a^j is the label of data x_a^j. Here, we use the commonality among multiple views to define pairwise constraints.

First, in each view, we can calculate the affinity value $W_{a,b}^j$ between any pair of features z_a^j and z_b^j through the gaussian kernel function

$$W_{a,b}^j = \exp\left(-\frac{d\left(z_a^j, z_b^j\right)}{2\sigma^2}\right) \tag{6}$$

where z_a^j and z_b^j are the latent features of x_a^j and x_b^j separately.

Second, we specify that a pair of instances is must-link when the affinity between them is higher than a threshold δ_m, and a pair of instances is cannot-link when the affinity between them is lower than a threshold δ_c. The remaining parts is set as 0 indicating unknown pairwise constraints. Denote L^j as the pairwise constraints matrix of the jth view, then we can obtain

$$L_{ab}^j = \begin{cases} 1, & \text{if } W_{ab}^j > \delta_m \\ -1, & \text{if } W_{ab}^j < \delta_c \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Finally, the intersection of the constraint matrices of all the views is selected as the final constraint matrix L^*. This means that the pairwise constraints contained in L^* are extracted in either L^j. For example, L_{ab}^* is equal to 1 or -1 if and only if all L_{ab}^j is equal to 1 or -1, otherwise L^* is equal to 0

$$L^* = \bigcap_{j=1}^{m} L^j. \tag{8}$$

In general, the more correct constraint information contained in the pairwise constraint matrix, the better the clustering effect will be. The fusion method of intersection ensures the accuracy (ACC) of constraint information but limits the quantity of constraint information. Next, we will diffuse the constraint information.

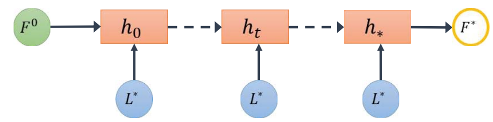

Fig. 2. Pairwise constraints propagation network.

2) Pairwise Constraints Propagation Network: In the last subsection, we obtain a pairwise constraint matrices fused by multiple views. On the one hand, the ACC of constraint information is ensured by the fusion method of finding the intersection, and on the other hand, the quantity of constraint information is limited. In general, the more correct constraint information contained in the pairwise constraint matrix, the better the clustering effect will be. In this subsection, we aim to spread the initial pairwise constraints as well as possible to the entire constraint matrix. To achieve this goal, we construct a propagation network to propagate pairwise constraints. The structure of the network is shown in the Fig. 2.

The network is of neuron-like nodes organized into successive layers. The output of the given layer is the input of the next layer. At each layer of the constraint propagation network, we continuously input L^* as supervisory information to ensure effective diffusion of the constraint information. Denote $F(t-1)$ as the output of the $t-1$ layer, the detailed flow of a network neuron is

$$F(t) = \phi(UL^* + VF(t-1) + b) \tag{9}$$

where U and V are the parameter of propagation network, b is the bias, and $\phi(\cdot)$ is the activation function.

The input of the network is the pairwise constraint matrix L^* and the initial matrix F^0. The intermediate output variable $F(t)$ continuously diffuses the constraint information with the assistance of L^*, and finally obtains the network output result F^* that contains the original constraint information and the propagated constraint information for optimizing W^*. Therefore, to train the network, we design to ensure the smoothness of F^* by minimizing $\mathrm{tr}((F^*)^{\mathrm{T}}\mathrm{S}F^*)$ with W^* and minimizing $\|F^* - L^*\|_2^2$ to ensure that F^* can diffuse and absorb initial constraint information. The objective function is shown as follows:

$$\min_{F^*} \; \gamma \,\mathrm{tr}((F^*)^{\mathrm{T}}\mathrm{S}F^*) + (1-\gamma)\|F^* - L^*\|_2^2 \tag{10}$$

where $\gamma > 0$ denotes the hyperparameter. $S = D^{-(1/2)}(W^*)D^{-(1/2)}$ denotes the normalized Laplacian matrix, D is a diagonal matrix with its $(i,i)-$element equal to the sum of the $i-$th row of W^*, $\mathrm{tr}(\cdot)$ denotes the trace of a matrix.

3) Optimize Affinity Matrix: The sequential pairwise constraint propagation network have diffused the pairwise constraint, and the results could be used to enhance the affinity matrix of spectral clustering. We use $F^* = (F^* + (F^*)^{\mathrm{T}})/2$ to make sure that F^* is symmetric. Then we will use F^* to adjust the affinity matrix W^* of the fusion multiview features.

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 02,2024 at 02:20:00 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Training Process of SDMvSC

Input: The number of multiview m; The dataset of each view j, $X^j = \{X^1, X^2, \ldots, X^m\}$;
Output: The Propagated affinity Matrix $\tilde{W}$

```
while Eq.(3) doesn't converge do
    Sampling a minibatch of instances from each multiview;
    for j = 1 to m do
        Z^j = E_j(X^j)
        X̃^j = D_j(Z^j)
        Z* = Fusion(Z^1, Z^2, ..., Z^m)
    end for
end while
Z*_o = Z*((((Z*)^T Z*)^{-1})^T)
Calculate W* using Eq.(5)
Calculate W^1, W^2, ..., W^m using Eq.(6)
Calculate L^1, L^2, ..., L^m using Eq.(7)
L* = Intersect(L^1, L^2, ..., L^m)
S = D^{-1/2}(W*)D^{-1/2}
F_0 = I − S
while Eq.(10) doesn't converge do
    F(t) = φ(UL* + VF(t − 1) + b)
end while
Calculate W̃ using Eq.(11)
```

We think of each term $|F^*_{ab}|$ in F^* as a confident value of the pairwise constraints. We adjust W^* using the formula (11)

$$\tilde{W}_{ab} = \begin{cases} 1 - \left(1 - F^*_{ab}\right)\left(1 - w^*_{ab}\right), & F^*_{ab} \geq 0 \\ \left(1 + F^*_{ab}\right) w^*_{ab}, & F^*_{ab} < 0 \end{cases} \tag{11}$$

where $\tilde{W}$ is nonnegative and symmetric [28], and then will be used for spectral clustering.

C. Algorithm Descriptions

In this subsection, we describe the training process of the proposed SDMvSC in algorithm 1. The SDMvSC is divided into two steps.

1) In the first step, we use (3) to train the autoencoders and fusion network to extract multiview fusion feature Z^*, then orthogonal feature Z^*_o is obtained by orthogonal layer.
2) In the second step, we train the constraint propagation network through (10) and optimize the affinity matrix W^* to obtain $\tilde{W}$ through (11). The input F^0 is initialized as $I - S$.

After that, $\tilde{W}$ is applied to spectral clustering to obtain the final clustering results of multiview data. The existing model in our method can directly add new data to obtain the fused affinity matrix and extract pairwise constraints. If the size of the newly added dataset is equal to that of the trained propagation network, it does not need to retrain the propagation network. If the size of the newly added dataset is not equal to that of the trained propagation network, we can leave out the propagation network, and approximately calculate the optimized affinity matrix of the whole dataset using the optimized affinity matrix of the existing data. The propagation network can also be retrained to obtain the optimized affinity matrix of the whole dataset.

D. Complexity Analysis

In this subsection, we will examine the complexity of the algorithm. Suppose the maximum number of neurons in each layer of the encoder/decoder is $\widetilde{D}_1/\widetilde{D}_2$, the maximum number of neurons in each layer of the view fusion network is $\widetilde{D}_3$, the maximum number of neurons in each layer of the propagation network is $\widetilde{D}_4$ and maximum epochs for deep latent feature extraction and fusion is T_1, maximum epochs for propagation network training is T_2. Then the time complexity of the feature extraction and fusion stage is $\mathcal{O}(T_1 nm \widetilde{D}_1^2 + T_1 nm \widetilde{D}_2^2 + T_1 nm \widetilde{D}_3^2)$ and the time complexity of the pairwise constraints propagation network stage is $\mathcal{O}(T_2 \widetilde{D}_4^2)$. So the total time of SDMvSC is $\mathcal{O}((T_1 \widetilde{D}_1^2 + T_1 \widetilde{D}_2^2 + T_1 \widetilde{D}_3^2)nm + T_2 \widetilde{D}_4^2)$, which is polynomial order to the number of examples n and the number of modalities m.

V. Experiment

A. Datasets

In this section, we verify the performance of the proposed method on four multiview datasets:

1) *Amsterdam Library of Object Images (ALOI)-100*[1]*:* We select to experiment on 100 classes with three views: $77D$ RGB color histograms, 13-D Hue-Saturation-Value/Hue-Saturation-Brightness (HSV/HSB) color histograms, and $64D$ Color affinity. 1080 instances are selected for each category, a total of 10800 instances were used.
2) *Land Use (LUse)-21*[2]*:* There are three views including $254D$ Local Binary Pattern (LBP), $512D$ Generalized Search Tree (GIST), and $256D$ CENsus TRansform hISTogram (CENTRIST) descriptors.
3) *Scene*-15 *[29]:* We extract $254D$ LBP, $512D$ GIST, and $256D$ CENTRIST descriptors from these datasets as three views.
4) *Caltech-UCSD Birds (CUB)-Ho*[3]*:* We use googlenet to extract $1024D$ middle crop, $1024D$ upper crop,and $1024D$ horizontally flipped from one image as three views, and select 50 classes containing 2889 instances.

More detailed introductions about the datasets are summarized in Table I. As shown in Table I, column m represents the number of views in each dataset. Column k represents the number of clusters in each dataset. Column n represents the

TABLE I
DESCRIPTION OF DATASETS

Dataset	# of m	# of k	# of n	$D_j(j = 1, \ldots, m)$
ALOI-100	3	100	10800	77,13,64
LUse-21	3	21	2100	254,512,256
Scene-15	3	15	3000	254,512,256
CUB-Ho	3	50	2889	1024,1024,1024

[1] http://elki.dbs.ifi.lmu.de/wiki/DataSets/MultiView
[2] http://weegee.vision.ucmerced.edu/datasets/landuse.html
[3] http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 02,2024 at 02:20:00 UTC from IEEE Xplore. Restrictions apply.

number of data points in each dataset. Column $D_j (j = 1, \ldots, m)$ represents the dimension of each view in each dataset.

B. Comparing Methods

To illustrate the significant improvement of our SDMvSC compared to single-view clustering algorithms, we test several state-of-the-art single-view clustering methods on four datasets.

1) Unsupervised deep embedding for clustering analysis (DEC) [30].
2) Improved deep embedded clustering with local structure preservation (IDEC) [31].
3) Joint unsupervised learning of deep representations and image clusters (JULE) [32].
4) *SpectralNet:* Spectral Clustering using Deep Neural Networks(SpectralNet) [27].

We compare our SDMvSC algorithm against the following traditional multiview clustering methods:

1) Heterogeneous image feature integration via multimodal spectral clustering (MMSC) [33].
2) Low-rank tensor constrained multiview subspace clustering (LT-MSC) [34].

We compare our SDMvSC algorithm against the following deep multiview clustering methods:

1) Deep multimodal subspace clustering networks (DMSCN) [11].
2) Multi-view Spectral Clustering Network (MvSCN) [16].
3) End-To-End Deep Multimodal Clustering (DMMC) [17].
4) Deep embedded multiview clustering with collaborative training (DEMVC) [35].

We compare our SDMvSC algorithm against the following self-supervised deep multiview clustering algorithm:

1) Self-Supervised Deep Multiview Subspace Clustering (S2DMVSC) [3].

C. Model Settings

All the parameter settings of the comparing methods are based on the original articles. For the proposed method, we tune the parameter as follows.

1) *Autoencoders:* Each view contains an autoencoder, and the encoder and decoder are composed of a full connection layer, which is connected by the $RelU$ activation function. To effectively retain the information, no activation function is added to the last layer. The structure of encoder and decoder is symmetrical. we set the encoder dimensions as $d-100-80-60-40-k$ for experiment, where d is the dimension of input feature and k is the number of cluster. The number of autoencoder network layers can make appropriate adjustments according to the data dimension for different dataset.
2) *Fusion Network:* In the fusion network, the input features are trained by a full connection layer, all the outputs are stacked, and then the dimension is gradually reduced through the network until the output dimension is the same as the number of cluster.

TABLE II
CLUSTERING ACC (%), THE BEST RESULT ARE IN BOLD

	ALOI-100	LUse-21	Scene-15	CUB-Ho	Average
DEC	28.18	12.34	16.97	15.66	18.29
IDEC	28.42	25.14	25.03	20.35	24.74
JULE	51.06	27.14	38.10	41.99	39.57
SpectralNet	22.96	28.59	46.74	23.25	30.39
MMSC	43.21	20.38	18.20	19.11	25.23
LT-MSC	52.64	30.86	43.93	44.48	42.98
DMSCN	40.47	29.81	24.73	48.43	35.86
MvSCN	57.83	31.05	45.00	32.25	41.53
DMMC	59.54	31.95	44.07	45.06	45.16
DEMVC	18.16	23.76	34.17	19.04	23.78
S2DMVSC	39.93	30.76	43.23	48.98	40.73
SDMvSC	**62.60**	**33.62**	**49.53**	**50.22**	**48.99**

3) *Orthogonal Layer:* The orthogonal layer is only used to output the orthogonal representation of the feature, and the parameters of the network are calculated manually by Cholesky decomposition.
4) *Pairwise Constraints Acquisition:* The threshold δ_m and δ_c are selected through setting the ratio δ of constraints, i.e., $n_1/n^2 \geq \delta$, $n_2/n^2 \geq \delta$, n_1 is the number of pair instances whose affinity is greater than δ_m, and n_2 is the number of pair instances whose affinity is less than δ_c.
5) *Constraint Propagation Network:* Entries of F need to be in $[-1, 1]$. $tanh$ activation function is used between layers of the propagation network. In the experiment, we use a three-layer constraint propagation network.

All network parameters except the orthogonal layer are obtained by initialization, and the learning rate α of Adam algorithm is set to 0.001. Through a large number of experimental results, we set β and γ as 0.9 and 0.7. δ is set as 0.01. Our implementation is based on Pytorch.[4]

D. Experiment Result

We use three widely used metrics to measure the clustering performance: ACC [36], normalized mutual information (NMI) [37] and purity (PUR) [38]. A larger value of ACC/NMI/PUR indicates a better clustering result. All the reported results are the average values of five independent runs.

The clustering results of the three metrics are presented in the Tables II–IV, respectively. The last column shows the average performance of every algorithm over the four datasets. In each column of the three tables, we used bold to mark the best experimental results.

By observing the experimental results, we reached the following conclusions:

1) SDMvSC is superior to the single-view algorithm, and the three metrics in each dataset are higher compared to the single-view algorithm. Taking the best single-view clustering algorithm for example, on the ALOI-100 dataset, the three indicators increased by

[4]https://pytorch.org/

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 02,2024 at 02:20:00 UTC from IEEE Xplore. Restrictions apply.

TABLE III
CLUSTERING NMI(%), THE BEST RESULT ARE IN BOLD

	ALOI-100	LUse-21	Scene-15	CUB-Ho	Average
DEC	59.27	15.18	16.10	28.37	29.73
IDEC	51.74	28.02	19.92	35.19	33.72
JULE	71.32	34.99	35.84	54.79	49.24
SpectralNet	50.09	33.78	42.60	38.35	41.21
MMSC	66.66	26.56	14.44	37.24	36.23
LT-MSC	60.85	35.62	41.36	47.69	46.38
DMSCN	50.60	37.21	27.59	62.99	44.60
MvSCN	69.67	35.75	**49.65**	47.50	50.64
DMMC	67.28	37.42	42.67	**63.03**	52.60
DEMVC	49.00	31.54	35.95	40.80	39.32
S2DMVSC	55.68	**38.06**	46.59	62.99	50.83
SDMvSC	**75.60**	36.31	42.55	61.04	**53.88**

TABLE IV
CLUSTERING PUR (%), THE BEST RESULT ARE IN BOLD

dataset	ALOI-100	LUse-21	Scene-15	CUB-Ho	Average
DEC	30.80	14.29	17.65	17.44	20.05
IDEC	31.05	27.52	26.43	21.64	26.66
JULE	55.14	30.57	38.97	44.27	42.24
SpectralNet	25.59	31.33	44.87	25.12	31.73
MMSC	43.59	21.81	18.60	19.97	26.00
LT-MSC	57.18	32.24	44.70	47.69	45.45
DMSCN	14.57	33.23	25.37	47.59	30.19
MvSCN	32.96	33.24	**51.07**	34.74	38.00
DMMC	62.35	32.46	45.57	47.04	46.86
DEMVC	19.19	26.38	34.57	19.18	24.83
S2DMVSC	34.07	33.86	48.47	51.85	42.06
SDMvSC	**65.27**	**34.19**	50.57	**51.99**	**50.51**

11.54%, 4.28%, and 10.13%, respectively. The results show that multiview clustering has great advantages over single-view clustering.

2) Our algorithm is superior to the multiview clustering algorithm in performance. Compared with LT-MSC algorithm, the ACC of SDMvSC raises 9.96% on the ALOI-100 dataset, raises 2.76% on the LUse-21 dataset, raises 5.60% on the Scene-15 dataset, and raises 5.74% on the CUB-Ho dataset. Compared to DEMVC, the NMI of SDMvSC improves 26.60%, 4.77%, 6.60%, and 20.24% on four dataset, respectively. The PUR increases by 0.96%, 0.95%, and 1.73% compared to DMSCN, MvSCN, and DMMC on LUse-21 dataset. In CUB-Ho dataset, the NMI of SDMvSC decreases 1.99% compared to DMMC, but the ACC and PUR increase 5.16% and 4.95%, respectively. Although the NMI is 7.10% lower than MvSCN in Scene-15 datasets, the ACC are better compared to MvSCN. The reason may be that the extracted pairwise constraints have a greater influence on the near nodes in the propagation process, resulting in a slight imbalance in the standard distribution. The average performance of SDMvSC is the best over the four datasets. In general, our algorithm is better than the multiview algorithms.
3) Compared with self-supervised deep multiview clustering algorithm, the PUR of SDMvSC increases 31.20% compared to S2DMVSC on ALOI-100 dataset. Although the NMI is 1.75%, 4.04%, and 1.95% lower than S2DMVSC in LUse-21, Scene-15, and CUB-Ho datasets, the ACC and PUR are better compared to S2DMVSC. The average performance of SDMvSC is superior to that of the self-supervised deep multiview algorithms over the four datasets. In general, our algorithm is better than self-supervised deep multiview algorithms.

TABLE V
CLUSTERING ACC (%), THE BEST RESULT ARE IN BOLD

	ALOI-100	LUse-21	Scene-15	CUB-Ho	Average
DMvSC	58.01	27.81	45.77	49.53	45.28
PDMvSC	58.13	27.29	47.50	48.74	45.42
SDMSC	55.02	28.52	44.15	46.91	43.65
SDMvSC	**62.60**	**33.62**	**49.53**	**50.22**	**48.99**

TABLE VI
CLUSTERING NMI(%), THE BEST RESULT ARE IN BOLD

	ALOI-100	LUse-21	Scene-15	CUB-Ho	Average
DMvSC	73.88	29.50	40.13	60.09	50.90
PDMvSC	73.80	28.76	40.25	60.80	50.90
SDMSC	66.96	30.14	35.56	55.28	46.99
SDMvSC	**75.60**	**36.31**	**42.55**	**61.04**	**53.88**

TABLE VII
CLUSTERING PUR (%), THE BEST RESULT ARE IN BOLD

dataset	ALOI-100	LUse-21	Scene-15	CUB-Ho	Average
DMvSC	61.35	29.76	47.43	51.64	47.55
PDMvSC	61.66	29.14	48.67	50.29	47.44
SDMSC	58.70	27.90	45.35	39.63	42.90
SDMvSC	**65.27**	**34.19**	**50.57**	**51.99**	**50.51**

E. Ablation Study

To verify the effectiveness of the proposed pairwise constrained propagation network and the view fusion module, we use three algorithms for ablation study. The experimental results are shown in Tables V–VII.

SDMSC removes the orthogonal layer to verify the validity of the view fusion module. For dataset ALOI-100, the three indicators of SDMvSC are 7.58%, 8.64%, and 6.57% higher than that of SDMSC. The performance of SDMvSC is significantly higher than that of SDMSC, indicating that the orthogonal layer plays a positive role in feature extraction and the learned latent consistent features are friendlier for spectral clustering.

We use two algorithms DMvSC and Propagate Deep Multiview Spectral Clustering (PDMvSC) to verify the effectiveness of the proposed pairwise constrained propagation network. DMvSC is a deep multiview clustering algorithm, which does not contain pairwise constraint propagation network. PDMvSC uses a mathematical method [28] to propagate pairwise constraint instead of the pairwise constraint propagation network proposed in this article. Self-supervised Deep Multiview Spectral Clustering (SDMvSC) outperforms DMvSC and

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 02,2024 at 02:20:00 UTC from IEEE Xplore. Restrictions apply.

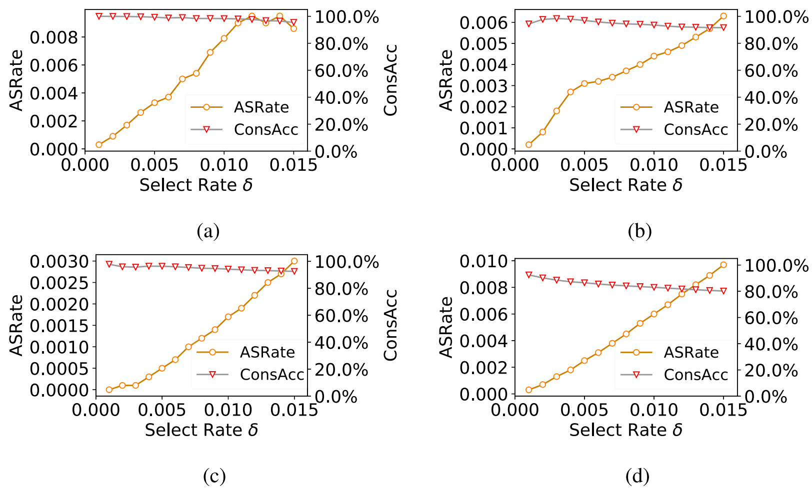

Fig. 3. Pairwise constraint ACC analysis. (a) ALOI-100. (b) Land-21. (c) Scene-15. (d) CUB-Ho.

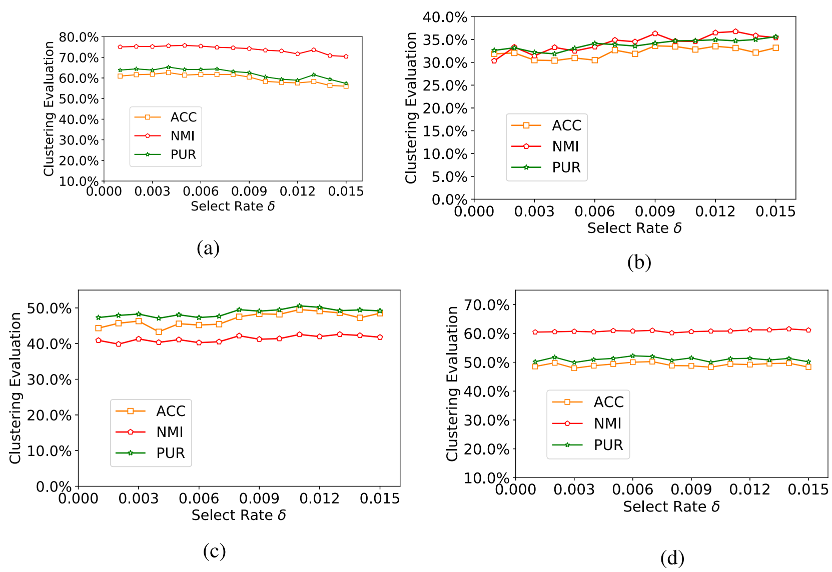

Fig. 4. Effect of constraints on results. (a) ALOI-100. (b) LUse-21. (c) Scene-15. (d) CUB-Ho.

PDMvSC in terms of ACC, NMI and PUR. This means that the proposed constraint propagation network diffuses the pairwise constraints better and make the clustering performance better.

F. Constraints Analysis

We investigate the correctness of the selected pairwise constraints and the effect of constraints on the clustering results.

Fig. 3 shows the statistic of the pairwise constraint. The horizontal axis is the selection rate δ vary in $[0.001 - 0.015]$, the left vertical axis is the actual selection rate of the constraints in L^*, and the right vertical axis is the ACC ratio of the actual selected constraints in L^*. We find that in general, through obtained the intersection of all view pairwise constraint matrices, the ratio of pairwise constraints actually obtained is far lower than the set threshold δ, so it is helpful to fitter incorrect constraint and improve the correctness of the extracted pairwise constraint information.

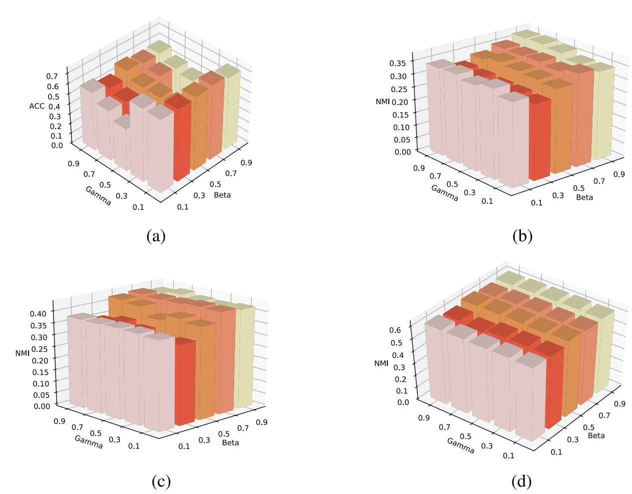

Fig. 5. Parameter study. (a) ALOI-100. (b) LUse-21. (c) Scene-15. (d) CUB-Ho.

With the increase of threshold δ, the granularity of filtering information of each view increases, and the selected pairwise

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 02,2024 at 02:20:00 UTC from IEEE Xplore. Restrictions apply.

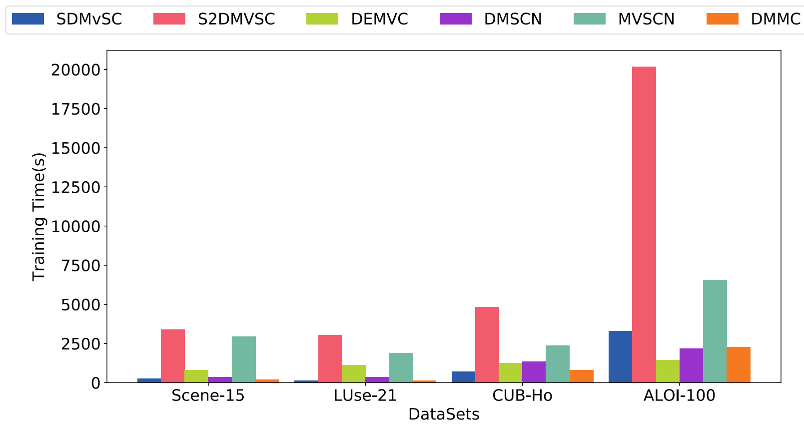

Fig. 6. Training time study.

constraint information increases. Even after the common filtering of multiple views, a certain amount of incorrect pairwise constraint information will still be retained. Thus, the ACC of the selected pairwise constraints decrease in gradual.

Fig. 4 shows the effect of constraints on the clustering results. The horizontal axis is the selection rate δ, the vertical axis is clustering evaluation (e.g., ACC/NMI/PUR) for four dataset. Generally, as the number of constraints increases, the performance of the proposed algorithm first increase and then gradually decrease, e.g., the performance of ALOI-100, LUse-21, Scene-15 and CUB-Ho dataset decrease when the selection rate exceeds 0.004, 0.013, 0.011 and 0.006.

Pairwise constraints as constraint information can improve the performance of clustering, but with the increase of threshold, the proportion of error constraints increases, and the ability of propagation network is inhibited. Therefore, the ability of multiview clustering can be improved by setting a reasonable threshold δ.

From Figs. 3 and 4, we conclude that the selected pairwise constraints are not all right, the wrong constraints may inhibit the propagation to some extent. But with a small number of constrains, the performance of the multiview clustering is improved.

G. Hyperparameter Setting Analysis

Fig. 5 analyzes the effect of the hyperparameters β and γ for the performance of SDMvSC. We present the results on four dataset. β and γ vary in the range $[0.1, 0.3, 0.5, 0.7, 0.9]$. SDMvSC achieves nearly stable good performance when $\beta \in [0.7, 0.9]$ and $\gamma \in [0.5, 0.7, 0.9]$. Finally, we set $\beta = 0.9$ and $\gamma = 0.7$ as the hyperparameter values of the SDMvSC model in the above experiment.

H. Training Time Analysis

In Fig. 6, we study the training time of SDMvSC algorithm and all the other five deep multiview clustering algorithms used in this article. From the results, we find that S2DMVSC consumed the most time on the four datasets. This is because the training time of S2DMVSC includes two stages: pretraining and fine-tuning. MVSCN also requires more training time than the other algorithms since MvSCN needs to train a SiameseNet network for each view. The training time of DEMVC, DMSCN, DMMC, and SDMvSC present similar trends on Scene-15, LUse-21, and CUB-Ho datasets. On the ALOI-100 dataset, SDMvSC requires more time than DEMVC, DMSCN, and DMMC, but requires less time than S2DMVSC and MvSCN. As the sample size increases, the increase in training time of SDMvSC is greater than that of DEMVC, DMSCN, DMMC, and MvSCN, but less than that of S2DMVSC. Considering the improvement of our method in terms of performance, the increase of the training time of our method is comparable with most baseline models.

VI. Conclusion

In this article, we proposed a novel deep self-supervised multiview clustering method, termed as self-supervised deep multiview spectral clustering (SDMvSC). We first extract the fusion features of multiview through joint optimization of autoencoders and feature fusion network. Then, the orthogonal representation of the fusion features is obtained by Cholesky decomposition. Meanwhile, the pairwise constraints are achieved by utilizing the commonality among multiview data. Then we optimize the affinity matrix of fusion features through the propagation network. In this work, propagation network is designed to propagate the pairwise constraints for optimizing the affinity matrix. Experiments on four benchmark datasets show the effectiveness of our proposed approach.

References

[1] X. Cai, F. Nie, and H. Huang, “Multi-view K-means clustering on big data,” in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1–7.

[2] H. Cai, B. Liu, Y. Xiao, and L. Lin, “Semi-supervised multi-view clustering based on orthonormality-constrained nonnegative matrix factorization,” *Inf. Sci.*, vol. 536, pp. 171–184, Oct. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025520304904

[3] X. Sun, M. Cheng, C. Min, and L. Jing, “Self-supervised deep multi-view subspace clustering,” in *Proc. Asian Conf. Mach. Learn.*, 2019, pp. 1001–1016.

[4] L. Zong, X. Zhang, X. Liu, and H. Yu, “Weighted multi-view spectral clustering based on spectral perturbation,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4621–4628.

[5] S. Luo, C. Zhang, W. Zhang, and X. Cao, “Consistent and specific multi-view subspace clustering,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[6] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, “COMIC: Multi-view clustering without parameter selection,” in *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA, vol. 97, Jun. 2019, pp. 5092–5101.

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 02,2024 at 02:20:00 UTC from IEEE Xplore. Restrictions apply.

[7] Z. Huang, Y. Ren, X. Pu, L. Pan, D. Yao, and G. Yu, "Dual self-paced multi-view clustering," *Neural Netw.*, vol. 140, pp. 184–192, Aug. 2021.

[8] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. ICML*, 2011, pp. 1–8.

[9] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "On deep multi-view representation learning," in *Proc. ICML*, 2015, pp. 1–10.

[10] G. Andrew, R. Arora, J. A. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. ICML*, 2013, pp. 1–9.

[11] 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.

[12] P. Nousi and A. Tefas, "Self-supervised autoencoders for clustering and classification," *Evolving Syst.*, vol. 11, no. 3, pp. 453–466, 2020.

[13] C. Xu, Z. Guan, W. Zhao, H. Wu, Y. Niu, and B. Ling, "Adversarial incomplete multi-view clustering," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3933–3939.

[14] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[15] 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.

[16] 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. (IJCAI)*, Aug. 2019, pp. 2563–2569.

[17] X. Zhang, J. Mu, L. Zong, and X. Yang, "End-to-end deep multimodal clustering," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6.

[18] E. Eaton, M. desJardins, and S. Jacob, "Multi-view clustering with constraint propagation for learning with an incomplete mapping between views," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2010, pp. 389–398.

[19] X. Zhang, L. Zong, X. Liu, and H. Yu, "Constrained NMF-based multi-view clustering on unmapped data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 29, no. 1, 2015, pp. 3174–3180.

[20] Q. Yin, S. Wu, R. He, and L. Wang, "Multi-view clustering via pairwise sparse subspace representation," *Neurocomputing*, vol. 156, pp. 12–21, May 2015.

[21] L. Zong, X. Zhang, and X. Liu, "Multi-view clustering on unmapped data via constrained non-negative matrix factorization," *Neural Netw.*, vol. 108, pp. 155–171, Dec. 2018.

[22] H. Cai, B. Liu, Y. Xiao, and L. Lin, "Semi-supervised multi-view clustering based on constrained nonnegative matrix factorization," *Knowl.-Based Syst.*, vol. 182, Oct. 2019, Art. no. 104798.

[23] J. Zhang *et al.*, "Self-supervised convolutional subspace clustering network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5473–5482.

[24] V. Sharma, M. Tapaswi, M. S. Sarfraz, and R. Stiefelhagen, "Video face clustering with self-supervised representation learning," *IEEE Trans. Biometrics, Behav., Identity Sci.*, vol. 2, no. 2, pp. 145–157, Apr. 2020.

[25] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran, "Self-supervised learning by cross-modal audio-video clustering," 2019, *arXiv:1911.12667*.

[26] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, 2002, pp. 849–856.

[27] U. Shaham, K. P. Stanton, H. Li, R. Basri, B. Nadler, and Y. Kluger, "SpectralNet: Spectral clustering using deep neural networks," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–21.

[28] Z. Lu and H. H. Ip, "Constrained spectral clustering via exhaustive and efficient constraint propagation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 1–14.

[29] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2005, pp. 524–531.

[30] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.

[31] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. IJCAI*, 2017, pp. 1753–1759.

[32] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5147–5156.

[33] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous image feature integration via multi-modal spectral clustering," in *Proc. CVPR*, Jun. 2011, pp. 1977–1984.

[34] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1582–1590.

[35] J. Xu, Y. Ren, G. Li, L. Pan, C. Zhu, and Z. Xu, "Deep embedded multi-view clustering with collaborative training," *Inf. Sci.*, vol. 573, pp. 279–290, Sep. 2021.

[36] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2003, pp. 267–273.

[37] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Jan. 2010.

[38] H. Schütze, C. D. Manning, and P. Raghavan, "Introduction to information retrieval," in *Proc. Int. Commun. Assoc. Comput. Mach. Conf.*, 2008, p. 260.

Linlin Zong received the scholar degree in software engineering and the Ph.D. degree in computer science from the Dalian University of Technology of China, Dalian, China, in 2011 and 2017, respectively.

She joined the Dalian University of Technology in 2017, where she is an Associate Professor. Her research interests include machine learning and data mining.

Faqiang Miao received the scholar degree in software engineering from Zhengzhou University, Zhengzhou, China, in 2019, and the master's degree in software engineering from the Dalian University of Technology, Dalian, China, in 2022.

He is a Professional Technician with the Institute of Farmland Irrigation of CAAS, Xinxiang, China. His research interests include machine learning and data mining.

Xianchao Zhang received the scholar and master's degrees in mathematics from the National University of Defense Technology, Changsha, China, in 1994 and 1998, respectively, and the Ph.D. degree in computer science from the University of Science and Technology of China, Anhui, China, in 2000.

From 2000 to 2003, he worked as a Research and Development Manager in some international companies. He joined the Dalian University of Technology in 2003, where he is a Full Professor. His research interests include design and analysis of algorithms, machine learning, data mining, and information retrieval.

Wenxin Liang (Member, IEEE) received the B.E. and M.E. degrees from Xi'an Jiaotong University, Xi'an, China, in 1998 and 2001, respectively, and the Ph.D. degree in computer science from the Tokyo Institute of Technology, Tokyo, Japan, in 2006.

He was a Postdoc Research Fellow, Japan Science and Technology Agency (JST) and a Guest Research Associate, Global Scientific Information and Computing Center (GSIC) of Tokyo Institute of Technology from October 2006 to March 2009. He is currently an Associate Professor with the School of Software, Dalian University of Technology, Dalian, China. His main research interests include data engineering, artificial intelligence, social networks, etc.

Dr. Liang is a Senior Member of the China Computer Federation (CCF) and a member of Chinese Association for Artificial Intelligence (CAAI), Association for Computing Machinery (ACM), International Society of Applied Intelligence (ISAI), the Association for Computing Machinery's Special Interest Group on Management of Data (ACM SIGMOD) Japan Chapter and Database Society of Japan (DBSJ).

Bo Xu received the B.Sc. and Ph.D. degrees from the Dalian University of Technology, Dalian, China, in 2011 and 2018, respectively.

He is currently an Associate Professor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. His current research interests include information retrieval and natural language processing.

Authorized licensed use limited to: Dalian University of Technology. Downloaded on March 02,2024 at 02:20:00 UTC from IEEE Xplore. Restrictions apply.