Differentiable Bi-Sparse Multi-View Co-Clustering

Shide Du, Zhanghui Liu, Zhaoliang Chen, Wenyuan Yang, and Shiping Wang

***Abstract*—Deep multi-view clustering utilizes neural networks to extract the potential peculiarities of complementarity and consistency information among multi-view features. This can obtain a consistent representation that improves clustering performance. Although a multitude of deep multi-view clustering approaches have been proposed, most lack theoretic interpretability while maintaining the advantages of good performance. In this paper, we propose an effective differentiable network with alternating iterative optimization for multi-view co-clustering termed differentiable bi-sparse multi-view co-clustering (DBMC) and an extension named elevated DBMC (EDBMC). The proposed methods are transformed into equivalent deep networks based on the constructed objective loss functions. They have the advantages of strong interpretability of the classical machine learning methods and the superior performance of deep networks. Moreover, DBMC and EDBMC can learn a joint and consistent collaborative representation from multi-source features and guarantee sparsity between multi-view feature space and single-view sample space. Meanwhile, they can be converted into deep differentiable network frameworks with block-wise iterative training. Correspondingly, we design two three-step iterative differentiable networks to resolve resultant optimization problems with theoretically guaranteed convergence. Extensive experiments on six multi-view benchmark datasets demonstrate that the proposed frameworks outperform other state-of-the-art multi-view clustering methods.**

***Index Terms*—Deep learning, multi-view clustering, co-clustering, sparse representation, differentiable blocks.**

I. Introduction

CLUSTERING is a technique used in machine learning [1], deep learning [2], [3], computer vision [4], [5], and related fields. It aims to allocate data with different structures but similar characteristics into identical groups. Numerous traditional algorithms have been extensively utilized in clustering, such as k-means, spectral clustering, and non-negative matrix factorization. Many emerging one-sided clustering methods have also been proposed and implemented. For instance, self-supervised convolutional subspace clustering networks [6] have been combined with a convolutional network, a self-indicting module, and a spectral clustering method to form joint optimization frameworks for clustering. Xu *et al.* [7] utilized dual autoencoder networks to extract potential feature relationships to improve clustering performance. A method for directly discovering cluster groups in samples was also presented in [8].

However, most of the above algorithms exclusively consider one-sided clustering, that is, clustering is performed only from the sample or feature space, which is not applicable to many scenarios [9], [10]. For example, clustering tasks involving term-document matrices or user-item matrices need to consider the dual relationship between samples and features. Co-clustering is a precise approach that is developed for such situation. For instance, based on non-negative matrix factorization, the method of orthogonal non-negative matrix tri-factorization [11] has been proposed for co-clustering. A new co-clustering bipartite graph strategy [12] was developed with promising performance. The bilateral k-means method [13] extended the one-sided algorithm to co-clustering tasks.

All of the above methods are representative single-view methods. Nevertheless, in large-scale real-world applications, data originates from multifarious sources and multiple modalities [14], [15]. These multi-view features are highly heterogeneous due to different information representations, making it challenging to explore the consistency of these features [16], [17]. The single-view clustering methods may fail to deal with actual multimedia data scenarios. Hence, multi-view and multimedia clustering tasks have become a focus of mainstream research [18], [19] in recent years. For example, Xie *et al.* [20] jointly learned multi-view correlations and local geometrical structures in a unified tensor space and a view-specific self-representation feature space, respectively. Yin *et al.* [21] proposed a tensor construction method to organize multi-view tensorial data, to which the tensor-tensor product could be applied. Chen *et al.* [22] proposed a nonlinear method to learn the kernel representation tensor and affinity matrix jointly. Although the above methods maintain strong interpretability, they also have the disadvantages of the fixed solution space and insufficient data fitting abilities. Deep learning adopts back propagation and gradient descent, which can be tuned end-to-end to solve the above problems. Therefore, deep learning has been incorporated into multi-view clustering tasks and become a research hotspot in multimedia tasks.

A common characteristic of most existing deep multi-view clustering methods is that they adopt networks to learn a comprehensive latent representation from various features [23]. For example, deep canonically correlated autoencoders [24], [25]

Manuscript received October 18, 2020; revised July 1, 2021; accepted July 16, 2021. Date of publication August 6, 2021; date of current version August 20, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was E. Aboutanios. This work was supported in part by the National Natural Science Foundation of China under Grant U1705262 and in part by the Natural Science Foundation of Fujian Province under Grants 2020J01130193 and 2018J07005. *(Corresponding author: Shiping Wang.)*

Shide Du, Zhanghui Liu, Zhaoliang Chen, and Shiping Wang are with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China, and also with the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China (e-mail: dushidems@gmail.com; lzh@fzu.edu.cn; chenzl23@outlook.com; shipingwangphd@163.com).

Wenyuan Yang is with the Fujian Key Laboratory of Granular Computing and Applications, Minnan Normal University, Zhangzhou 363000, China (e-mail: yangwycn@gmail.com).

Digital Object Identifier 10.1109/TSP.2021.3101979

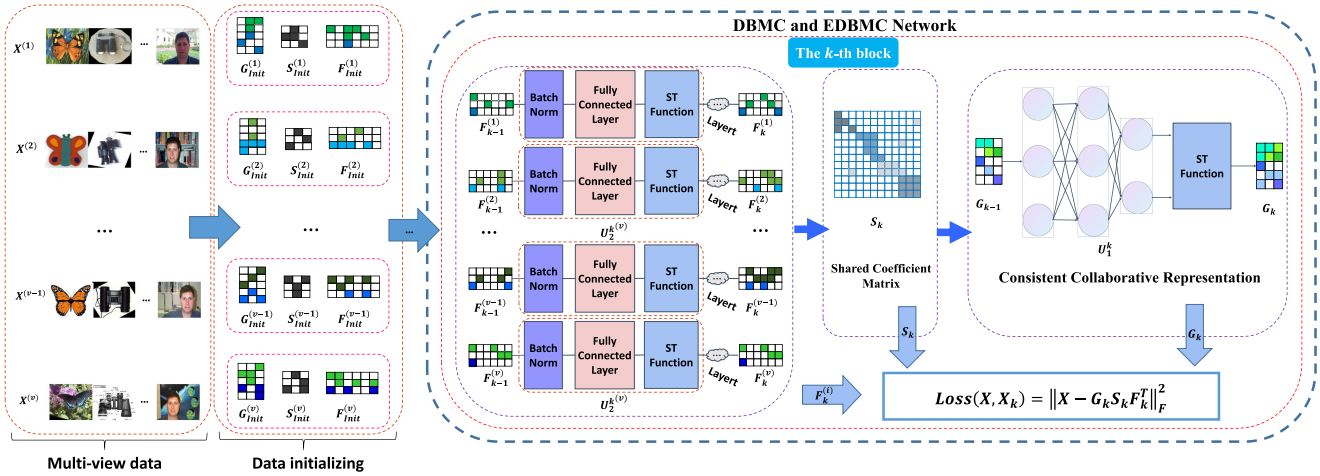


Fig. 1. An overview of the proposed frameworks. After initialization, we obtain a consistent collaborative representation $\mathbf{G}$ by DBMC and EDBMC block-wise networks, where $\mathbf{U}_i$ corresponds to the fully connected layers. The ST represents a soft-threshold function, differing between DBMC and EDBMC. The proximal operators and differentiable proximal operators are parameterized by θ_1 and θ_2, respectively. Finally, we obtain the clustering results according to the consistent collaborative representation $\mathbf{G}$.

proposed to maximize the canonical correlation of the features, thereby learning a two-view low-dimensional representation. Deep generalized canonical correlation analysis (DGCCA) [26] searched for the interconnectedness among multi-view features to explore a complete potential representation. Gao *et al.* [27] extended deep canonically correlated autoencoders to those combining relevant constraints with a self-expression layer and made full use of the information among the multi-view features. The above four networks utilized deep learning and canonical correlation analysis to learn a unified latent correlation representation to perform multi-view clustering. Moreover, Zhao *et al.* [28] used deep networks by simulating non-negative matrix factorization to excavate common latent feature semantics. Autoencoder in autoencoder networks [29] integrated autoencoder and degradation networks to project multiple sample spaces onto a shared subspace. Multi-view spectral clustering networks (MvSCN) [30] aimed to fuse the spectral clustering method into deep networks to explore potential representations of local invariance. The above methods have demonstrated the feasibility and effectiveness of multi-view clustering methods on deep networks to obtain multi-view consistent representations.

Although the above methods have showed encouraging performance, some problems still remain to be solved. First, some existing multi-view clustering networks are difficult to be interpretable. We may need to construct deep networks based on optimization methods and make the networks more interpretable. Second, most methods do not consider the dual relationship between the multi-view feature space and its sample space, which is an associative structure between samples and features. Finally, limited research is devoted to exploring the sparsity of the dual space and the non-linear data relationship using learnable and interpretable deep networks.

To solve the above problems, we propose a network framework for multi-view clustering called DBMC and an extension named EDBMC. DBMC and EDBMC are equivalent deep networks of proposed optimizing loss functions, which provides favorable theoretical interpretability. The introduction of differentiable learnable blocks allows the corresponding solutions to be implicitly learned by an end-to-end way in the network. Furthermore, the proposed networks consider a multi-view dual space relationship and keep the corresponding space sparse to learn an interpretable and consistent collaborative representation. The overall framework is shown in Fig. 1. The main contributions of this paper can be summarized as follows:

- DBMC and EDBMC are transformed into deep differentiable network frameworks on differentiable iterative optimization of the proposed objective loss functions.
- DBMC and EDBMC learn interpretable and consistent collaborative representations from multi-view features and maintain sparsity in the dual space of features and samples.
- The performance of the proposed frameworks is compared with that of nine state-of-the-art multi-view clustering methods to demonstrate their superiority.

The rest of the paper is structured as follows. In Section II, we review related research on multi-view co-clustering and sparse representation. In Section III, DBMC and EDBMC methods are proposed and their convergences are verified. Section IV describes extensive experiments with real-world datasets to validate the clustering performance of the proposed frameworks. Finally, the paper is concluded in Section V.

II. Related Works

In this section, we present the progress about multi-view co-clustering and sparse representation.

A. Multi-View Co-Clustering

In many real-world applications, data originates from various sources. For instance, information retrieval is supported by diverse languages with descriptions or webpages containing both contents and citation linkages [31]. Understanding how to exploit the complementarity and consistency in multi-view data has

become a popular research direction. Multi-view co-clustering is based on the duality between samples and multi-view features. Samples are grouped according to their feature distributions, and features are grouped by building on their distributions with data points. Unlike one-sided multi-view clustering, multi-view co-clustering captures the immediate effects of both in the sample and feature space simultaneously. Moreover, multi-view co-clustering is a method of grouping two types of entities simultaneously by the similarities of their pairwise interactions. Furthermore, multi-view co-clustering methods have captured increasing attention in recent years. For example, [32] added the maximum entropy into the loss function to control the weights of different views. An auto-weighted method [33] was introduced into multi-view co-clustering to assign the weight of each view feature. Moreover, Hu *et al.* [34] proposed a dynamic multi-view co-clustering algorithm with mutual information. These works all consider the spatial structure of the feature and sample space into multi-view clustering tasks.

B. Sparse Representation

Sparse representation can handle redundant information to maintain the processed features more interpretable [35]. Several algorithms with such characteristics have already been proposed. For instance, Beck *et al.* [36] designed a fast iterative shrinkage and thresholding algorithm (ISTA), which solved linear sparse inverse problems. Learned ISTA (LISTA) [37] was an extension of ISTA that combined sparse representation with neural networks. Recently, differentiable deep neural networks [38] possessed the same conceptual peculiarities as above and have been applied to compressive sensing reconstruction models. Chen *et al.* [39] introduced a differentiable weighted sparse structure and obtained a theoretical linear convergence.

We can observe that the above algorithms employ the idea of sparse representation and have achieved encouraging performance in their respective fields. Next, we focus on ISTA algorithm and prepare the way for the proposed frameworks. The loss function of solving sparse coding $\mathbf{W}$ in the original ISTA is described as

$$\min_{\mathbf{W}} \frac{1}{2} \|\mathbf{X} - \mathbf{D}_d\mathbf{W}\|_2^2 + \alpha\|\mathbf{W}\|_1. \tag{1}$$

Problem (1) is discussed in [37], which can be solved by the following iterative updating rules

$$u = x_{k-1} - \frac{1}{L}\nabla\mathcal{L}\left(x_{k-1}\right), x_k = \mathbf{Prox}_{\frac{\alpha}{L}}(u). \tag{2}$$

Herein, ISTA defines a constant L, which is an upper bound of the largest eigenvalue of $\mathbf{D}_d^T\mathbf{D}_d$. $\nabla\mathcal{L}(x_{k-1})$ of (1) is computed as

$$\nabla\mathcal{L}\left(\mathbf{W}\right) = \mathbf{D}_d^T(\mathbf{D}_d\mathbf{W} - \mathbf{X}). \tag{3}$$

TABLE I
SYMBOLIC NORMALIZATION NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
$\{\mathbf{X}_v\}_{v=1}^V$	$\mathbf{X}_v \in \mathbb{R}^{n\times d_v}$ is the multi-view training data from the v-th view.
$\mathbf{G}$	$\mathbf{G} \in \mathbb{R}^{n\times c_1}$ is a consistent collaborative representation.
$\mathbf{S}$	$\mathbf{S} \in \mathbb{R}^{c_1\times c_2}$ is a shared coefficient matrix.
$\{\mathbf{F}_v\}_{v=1}^V$	$\mathbf{F}_v \in \mathbb{R}^{d_v\times c_2}$ is the feature indicator matrix from the v-th view.
$\mathbf{L}_{\mathbf{G}}^v$	$\mathbf{L}_{\mathbf{G}}^v \in \mathbb{R}^{n\times n}$ is a graph Laplacian matrix of the v-th sample space.
$\mathbf{I}$, $\mathbf{U}_i$	A unit matrix and a layer learned by deep networks.
$\mathbf{Prox}_\theta(\cdot)$	A proximal operator, where θ is a threshold.
$\xi_{(\theta_1)}(\cdot)$, $\eta_{(\theta_2)}(\cdot)$	Non-linear functions parameterized by θ_1, and θ_2.

(3) can be substituted with (2) to obtain the algorithmic forms as

$$\begin{aligned}\mathbf{W}_k &= \mathbf{Prox}_{(\alpha/L)}\left(\mathbf{W}_{k-1} - \frac{1}{L}\mathbf{D}_d^T\left(\mathbf{D}_d\mathbf{W}_{k-1} - \mathbf{X}\right)\right)\\ &= \mathbf{Prox}_{(\alpha/L)}\left(\left(\mathbf{I} - \frac{1}{L}\mathbf{D}_d^T\mathbf{D}_d\right)\mathbf{W}_{k-1} + \frac{1}{L}\mathbf{D}_d^T\mathbf{X}\right).\end{aligned} \tag{4}$$

Hence, these variables in (4) are interpreted as

$$\begin{aligned}&\mathbf{D}_e = \frac{1}{L}\mathbf{D}_d^T, \mathbf{S} = \mathbf{I} - \frac{1}{L}\mathbf{D}_d^T\mathbf{D}_d,\\ &\left[\mathbf{Prox}_\theta(\mathbf{V})\right]_{ij} = \operatorname{sign}\left(\mathbf{V}_{ij}\right)\left(|\mathbf{V}_{ij}| - \theta_{ij}\right)_+.\end{aligned} \tag{5}$$

(5) is incorporated into (4) to obtain the canonical form of the LISTA algorithm, calculated as

$$\mathbf{W}_k = \mathbf{Prox}_\theta\left(\mathbf{D}_e\mathbf{X} + \mathbf{S}\mathbf{W}_{k-1}\right). \tag{6}$$

The above formula shows that the sparse solution $\mathbf{W}_k$ can be obtained by training learnable layers $\{\mathbf{D}_e, \mathbf{S}\}$ and proximal operator $\mathbf{Prox}_\theta(\cdot)$. The LISTA algorithm provides a direction for transforming existing sparsity based learning methods into equivalent deep networks.

III. PROBLEM FORMULATION AND OPTIMIZATION

In this section, we start with a primitive loss function, propose a loss function for DBMC, and provide an optimization plan. In advance, we replace two proximal steps in DBMC to propose an elevated DBMC. First, we define the required notations in Table I.

A. Problem Formulation

Denote given data be $\mathbf{X} \in \mathbb{R}^{n\times d}$, where n is the number of samples, and d is the number of features. The clustering problem is to divide data points into c clusters $\{C_i\}_{i=1}^c$. So as to convert discrete k-means problems into continuous ones, the cluster indicator matrix $\mathbf{G} \in \mathbb{R}^{n\times c}$ are added to eventually obtain clustering results, where $\mathbf{G}_{ij} = \frac{1}{\sqrt{C_i}}$. If sample x_i pertains to cluster C_j, and $\mathbf{G}_{ij} = 0$ otherwise. It can be validated that $\mathbf{G}^T\mathbf{G} = \mathbf{I}$. Based on the above analysis, the optimization problem is expressed analogously as

$$\min_{\mathbf{G},\mathbf{S}} \frac{1}{2}\|\mathbf{X} - \mathbf{G}\mathbf{S}\|_F^2, \text{s.t. } \mathbf{G} \geq 0, \mathbf{G}^T\mathbf{G} = \mathbf{I}, \tag{7}$$

where $\mathbf{S} \in \mathbb{R}^{c\times d}$ is a shared coefficient matrix. The above loss function only considers clustering from the data side $\mathbf{G}$. The collaborative clustering problem is to group the data $\mathbf{G}$ and features $\mathbf{F}$ simultaneously. Analogously, the single-view co-clustering problem can be written as

$$\min_{\mathbf{G},\mathbf{S},\mathbf{F}} \frac{1}{2}\left\|\mathbf{X}-\mathbf{G}\mathbf{S}\mathbf{F}^T\right\|_F^2, \quad \text{s.t. } \mathbf{G}\geq 0, \mathbf{F}\geq 0, \mathbf{G}^T\mathbf{G}=\mathbf{I}, \mathbf{F}^T\mathbf{F}=\mathbf{I}, \tag{8}$$

where $\mathbf{G} \in \mathbb{R}^{n\times c_1}$ and $\mathbf{F} \in \mathbb{R}^{d\times c_2}$ are a consistent collaborative representation and a feature indicator matrix, and $\mathbf{S} \in \mathbb{R}^{c_1\times c_2}$ is a shared coefficient matrix. The collaborative clustering problem is a prolongation of k-means. However, the aforementioned clustering problem does not preserve the topological structure of the sample space. Accordingly, the term $\mathbf{L_G}$ is introduced to learn a local geometric structure of the input training samples. Moreover, the optimization problem does not consider the sparse representation in the space of samples and features. Although the orthogonal constraints on $\mathbf{G}$ and $\mathbf{F}$ already have high sparsity, the sparse norm can enhance this property, so that some rows of $\mathbf{G}$ and $\mathbf{F}$ can be equal to zero. Therefore, two terms are added to learn a sparse representation, expressed as

$$\begin{aligned}\min_{\mathbf{G},\mathbf{S},\mathbf{F}} \frac{1}{2}\left\|\mathbf{X}-\mathbf{G}\mathbf{S}\mathbf{F}^T\right\|_F^2 &+ \alpha\,\mathrm{Tr}(\mathbf{G}^T\mathbf{L_G}\mathbf{G}) + \beta\left\|\mathbf{F}\right\|_1 \\ &+ \gamma\|\mathbf{G}\|_1, \text{s.t. } \mathbf{G}\geq 0, \mathbf{F}\geq 0, \mathbf{G}^T\mathbf{G}=\mathbf{I}, \mathbf{F}^T\mathbf{F}=\mathbf{I}.\end{aligned} \tag{9}$$

Considering the above single-view idea as a multi-view one, multi-view co-clustering obtains the consistent collaborative representation $\mathbf{G}$, the global coefficient matrix $\mathbf{S}$, and the single-view feature indicator matrix $\mathbf{F}_v$. Thus, the multi-view co-clustering problem is extended as

$$\begin{aligned}\min_{\mathbf{G},\mathbf{S},\{\mathbf{F}_v\}_{v=1}^V} \frac{1}{2}\sum_{v=1}^{V}(&\left\|\mathbf{X}_v-\mathbf{G}\mathbf{S}\mathbf{F}_v^T\right\|_F^2 + \alpha\,\mathrm{Tr}(\mathbf{G}^T\mathbf{L}_\mathbf{G}^v\mathbf{G}) \\ &+ \frac{\beta}{2}\left\|\mathbf{F}_v^T\mathbf{F}_v-\mathbf{I}\right\|_F^2 + \gamma\left\|\mathbf{F}_v\right\|_1) \\ &+ \frac{\delta}{4}\left\|\mathbf{G}^T\mathbf{G}-\mathbf{I}\right\|_F^2 + \zeta\|\mathbf{G}\|_1, \\ &\text{s.t. } \mathbf{G}\geq 0, \mathbf{F}_v\geq 0, 1\leq v\leq V.\end{aligned} \tag{10}$$

Problem (10) can be resolved into three sub-problems and transformed into the corresponding sub-block networks. $\mathbf{G}$ and $\mathbf{S}$ are the global optimization variables learned from multi-view features, and $\mathbf{F}_v$ is an optimization variable obtained from the single-view data. $\mathbf{L}_\mathbf{G}^v = \mathbf{D}_\mathbf{G}^v - \mathbf{P}_\mathbf{G}^v$ is a graph Laplacian matrix, constructed from a pre-chosen pairwise similarity matrix $\mathbf{P}_\mathbf{G}^v = \kappa(\mathbf{X}_v, \mathbf{X}_v)$, where $\kappa(\cdot,\cdot)$ is the RBF kernel function.

B. Problem Optimization and Its Block Network

Step 1. Minimize the objective function over $\mathbf{G}$ *with fixed* $\mathbf{S}$ *and* $\mathbf{F}_v$ *values from Problem (10):* Optimizing Problem (10) with respect to $\mathbf{G}$ is equivalent to optimizing

$$\begin{aligned}\min_{\mathbf{G}} \frac{1}{2}\sum_{v=1}^{V}\Big(&\left\|\mathbf{X}_v-\mathbf{G}\mathbf{S}\mathbf{F}_v^T\right\|_F^2 + \alpha\,\mathrm{Tr}\left(\mathbf{G}^T\mathbf{L}_\mathbf{G}^v\mathbf{G}\right)\Big) \\ &+ \frac{\delta}{4}\left\|\mathbf{G}^T\mathbf{G}-\mathbf{I}\right\|_F^2 + \zeta\|\mathbf{G}\|_1, \text{s.t. } \mathbf{G}\geq 0.\end{aligned} \tag{11}$$

Consequently, based on the analysis in Subsection II-B, the sub-problem (11) is differentiable without ℓ_1-norm. The Lagrange function $\mathcal{L}_1$ ignoring ℓ_1-norm is constructed as

$$\begin{aligned}\mathcal{L}_1(\mathbf{G}) = \min_{\mathbf{G}} \frac{1}{2}\sum_{v=1}^{V}(&\left\|\mathbf{X}_v-\mathbf{G}\mathbf{S}\mathbf{F}_v^T\right\|_F^2 \\ &+ \alpha\,\mathrm{Tr}(\mathbf{G}^T\mathbf{L}_\mathbf{G}^v\mathbf{G})) + \frac{\delta}{4}\left\|\mathbf{G}^T\mathbf{G}-\mathbf{I}\right\|_F^2, \text{ s.t. } \mathbf{G}\geq 0.\end{aligned} \tag{12}$$

Taking derivative of $\mathcal{L}_1$ with respect to $\mathbf{G}$, a single-view gradient of $\nabla\mathcal{L}_1(\mathbf{G})$ is computed as

$$\begin{aligned}\frac{\partial\mathcal{L}_1}{\partial\mathbf{G}} = \Bigg(&-\mathbf{X}_v\mathbf{F}_v\mathbf{S}^T + \mathbf{G}\mathbf{S}\mathbf{F}_v^T\mathbf{F}_v\mathbf{S}^T + \alpha\mathbf{L}_\mathbf{G}^v\mathbf{G} \\ &+ \delta\Big(\mathbf{G}\mathbf{G}^T\mathbf{G}-\mathbf{G}\Big)\Bigg).\end{aligned} \tag{13}$$

While a solving strategy is applied to non-smooth ℓ_1-regularized problems, the process of calculating $\mathbf{G}$ is written as

$$\begin{aligned}\mathbf{G} &= \mathbf{Prox}_{(\zeta/L_1)}\Big(\mathbf{G}-\frac{1}{L_1}\Big(-\mathbf{X}_v\mathbf{F}_v\mathbf{S}^T + \mathbf{G}\mathbf{S}\mathbf{F}_v^T\mathbf{F}_v\mathbf{S}^T \\ &+ \alpha\mathbf{L}_\mathbf{G}^v\mathbf{G} + \delta\left(\mathbf{G}\mathbf{G}^T\mathbf{G}-\mathbf{G}\right)\Big)\Big) \\ &= \mathbf{Prox}_{(\zeta/L_1)}(\mathbf{G} + \frac{1}{L_1}\mathbf{X}_v\mathbf{F}_v\mathbf{S}^T - \frac{1}{L_1}\mathbf{G}\mathbf{S}\mathbf{F}_v^T\mathbf{F}_v\mathbf{S}^T \\ &- \frac{\alpha}{L_1}\mathbf{L}_\mathbf{G}^v\mathbf{G} - \frac{\delta}{L_1}(\mathbf{G}\mathbf{G}^T\mathbf{G}-\mathbf{G})) \\ &= \mathbf{Prox}_{(\zeta/L_1)}\left(\left(\mathbf{I}-\frac{1}{L_1}\mathbf{S}\mathbf{F}_v^T\mathbf{F}_v\mathbf{S}^T\right)\mathbf{G} + \frac{1}{L_1}\mathbf{X}_v\mathbf{F}_v\mathbf{S}^T\right. \\ &\left.-\frac{\alpha}{L_1}\mathbf{L}_\mathbf{G}^v\mathbf{G} - \frac{\delta}{L_1}\left(\mathbf{G}\mathbf{G}^T\mathbf{G}-\mathbf{G}\right)\right).\end{aligned} \tag{14}$$

(14) illustrates to calculate the process of iterative form $\mathbf{G}$ in a single-view case. Here, we employ a learnable layer $\mathbf{U}_1$ to represent $(\mathbf{I}-\frac{1}{L_1}\mathbf{S}\mathbf{F}_v^T\mathbf{F}_v\mathbf{S}^T)$. Then, the network of updating $\mathbf{G}_k$ as a multi-view case can be rewritten as

$$\begin{aligned}\mathbf{G}_k = \mathbf{Prox}_{(\zeta/L_1)}\Bigg(\frac{1}{V}\sum_{v=1}^{V}\Bigg(&\mathbf{U}_1\mathbf{G}_{k-1} + \frac{1}{L_1^{(v)}}\mathbf{X}_v\mathbf{F}_{k-1}^{(v)}\mathbf{S}_{k-1}^T \\ &- \frac{\alpha}{L_1^{(v)}}\mathbf{L}_\mathbf{G}^v\mathbf{G}_{k-1} - \frac{\delta}{L_1^{(v)}}\Big(\mathbf{G}_{k-1}\mathbf{G}_{k-1}^T\mathbf{G}_{k-1}-\mathbf{G}_{k-1}\Big)\Bigg)\Bigg).\end{aligned} \tag{15}$$

The consistent collaborative representation $\mathbf{G}$ can be obtained in (15). The proximal operator guarantees the constraint $\mathbf{G}\geq 0$. $L_1^{(v)}$ is the Frobenius norm of $\mathbf{S}_{k-1}(\mathbf{F}_{k-1}^{(v)})^T$.

Step 2. Minimize the objective function over $\mathbf{F}_v$ *with fixed* $\mathbf{S}$ *and* $\mathbf{G}$ *values from Problem (10):* Optimizing Problem (10) with respect to $\mathbf{F}_v$ is equivalent to optimizing

$$\min_{\{\mathbf{F}_v\}_{v=1}^V} \frac{1}{2}\Big(\left\|\mathbf{X}_v - \mathbf{G}\mathbf{S}\mathbf{F}_v^T\right\|_F^2 + \frac{\beta}{2}\left\|\mathbf{F}_v^T\mathbf{F}_v - \mathbf{I}\right\|_F^2 + \gamma\left\|\mathbf{F}_v\right\|_1\Big),\ \text{s.t. } \mathbf{F}_v \geq 0. \tag{16}$$

The Lagrange function $\mathcal{L}_2$ without ℓ_1-norm is constructed as

$$\mathcal{L}_2(\mathbf{F}_v) = \min_{\{\mathbf{F}_v\}_{v=1}^V} \frac{1}{2}(\left\|\mathbf{X}_v - \mathbf{G}\mathbf{S}\mathbf{F}_v^T\right\|_F^2 + \frac{\beta}{2}\left\|\mathbf{F}_v^T\mathbf{F}_v - \mathbf{I}\right\|_F^2), \text{s.t.}\mathbf{F}_v \geq 0. \tag{17}$$

Taking derivative of $\mathcal{L}_2$ with respect to $\mathbf{F}_v$, we obtain a single-view gradient of $\nabla\mathcal{L}_2(\mathbf{F}_v)$,

$$\frac{\partial\mathcal{L}_2}{\partial\mathbf{F}_v} = \left(-\mathbf{X}_v^T\mathbf{G}\mathbf{S} + \mathbf{F}_v\mathbf{S}^T\mathbf{G}^T\mathbf{G}\mathbf{S} + \beta\left(\mathbf{F}_v\mathbf{F}_v^T\mathbf{F}_v - \mathbf{F}_v\right)\right). \tag{18}$$

With the above analysis, the single-view sub-problem (16) is analogously solved as

$$\begin{aligned}\mathbf{F}_v &= \mathbf{Prox}_{(\gamma/L_2)}\Bigg(\mathbf{F}_v - \frac{1}{L_2}(-\mathbf{X}_v^T\mathbf{G}\mathbf{S} + \mathbf{F}_v\mathbf{S}^T\mathbf{G}^T\mathbf{G}\mathbf{S} \\ &\quad + \beta(\mathbf{F}_v\mathbf{F}_v^T\mathbf{F}_v - \mathbf{F}_v))\Bigg) \\ &= \mathbf{Prox}_{(\gamma/L_2)}\Bigg(\mathbf{F}_v + \frac{1}{L_2}\mathbf{X}_v^T\mathbf{G}\mathbf{S} - \frac{1}{L_2}\mathbf{F}_v\mathbf{S}^T\mathbf{G}^T\mathbf{G}\mathbf{S} \\ &\quad - \frac{\beta}{L_2}(\mathbf{F}_v\mathbf{F}_v^T\mathbf{F}_v - \mathbf{F}_v)\Bigg) \\ &= \mathbf{Prox}_{(\gamma/L_2)}\Bigg(\left(\mathbf{I} - \frac{1}{L_2}\mathbf{S}^T\mathbf{G}^T\mathbf{G}\mathbf{S}\right)\mathbf{F}_v + \frac{1}{L_2}\mathbf{X}_v^T\mathbf{G}\mathbf{S} \\ &\quad - \frac{\beta}{L_2}(\mathbf{F}_v\mathbf{F}_v^T\mathbf{F}_v - \mathbf{F}_v)\Bigg).\end{aligned} \tag{19}$$

The network structure of the v-th view $\mathbf{F}_k^{(v)}$ updating process is written as

$$\mathbf{F}_k^{(v)} = \mathbf{Prox}_{(\gamma/L_2)}\Bigg(\mathbf{U}_2\mathbf{F}_{k-1}^{(v)} + \frac{1}{L_2}\mathbf{X}_v^T\mathbf{G}_{k-1}\mathbf{S}_{k-1} - \frac{\beta}{L_2}\left(\mathbf{F}_{k-1}^{(v)}(\mathbf{F}_{k-1}^{(v)})^T\mathbf{F}_{k-1}^{(v)} - \mathbf{F}_{k-1}^{(v)}\right)\Bigg), \tag{20}$$

where $\mathbf{U}_2 = (\mathbf{I} - \frac{1}{L_2}\mathbf{S}^T\mathbf{G}^T\mathbf{G}\mathbf{S})$, and L_2 is the Frobenius norm of $\mathbf{G}_{k-1}\mathbf{S}_{k-1}$. The proximal operator guarantees the constraint $\mathbf{F}_v \geq 0$.

Step 3. Minimize the objective function over $\mathbf{S}$ *with fixed* $\mathbf{F}_v$ *and* $\mathbf{G}$ *values from Problem (10):* Optimizing Problem (10) with respect to $\mathbf{S}$ is equivalent to optimizing

$$\min_{\mathbf{S}} \frac{1}{2}\sum_{v=1}^V\left(\left\|\mathbf{X}_v - \mathbf{G}\mathbf{S}\mathbf{F}_v^T\right\|_F^2\right). \tag{21}$$

Let $\mathcal{L}_3$ represent the sub-problem (21), and take derivative for $\mathcal{L}_3$ to obtain gradient of $\nabla\mathcal{L}_3(\mathbf{S})$ as

$$\frac{\partial\mathcal{L}_3}{\partial\mathbf{S}} = \sum_{v=1}^V\left(\mathbf{F}^T\mathbf{F}\mathbf{S}^T\mathbf{G}^T\mathbf{G} - \mathbf{F}^T\mathbf{X}_v^T\mathbf{G}\right). \tag{22}$$

Let the derivative be equal to zero, the network form of $\mathbf{S}$ updating process is formulated as

$$\mathbf{S}_k = \frac{1}{V}\sum_{v=1}^V\left((\mathbf{G}_{k-1}^T\mathbf{G}_{k-1})^{-1}\mathbf{G}_{k-1}^T\mathbf{X}_v\mathbf{F}_{k-1}^{(v)}(\mathbf{F}_{k-1}^{(v)}\mathbf{F}_{k-1}^{(v)})^{-1}\right). \tag{23}$$

Equations (15), (20), and (23) are the core of the entire network. It can be seen that DBMC considers both multi-view co-clustering and differentiable sparse coding to learn a consistent collaborative representation.

C. Network Structures of DBMC and EDBMC

The specific programming implementation is that we generate $\mathbf{G}$ from optimizing and training Problem (10) corresponding to differentiable block networks, and then $\mathbf{G}$ is utilized to instruct the final clustering results. In the above-mentioned learnable block networks, each block is differentiable and reusable. The one-block network structure of DBMC is summarized as

$$\begin{cases}\mathbf{G}_k = \mathbf{Prox}_{(\zeta/L_1)}\Bigg(\frac{1}{V}\sum_{v=1}^V\Bigg(\mathbf{U}_1\mathbf{G}_{k-1} + \frac{1}{L_1^{(v)}}\mathbf{X}_v\mathbf{F}_{k-1}^{(v)}\mathbf{S}_{k-1}^T \\ \qquad - \frac{\alpha}{L_1^{(v)}}\mathbf{L}_{\mathbf{G}}^v\mathbf{G}_{k-1} - \frac{\delta}{L_1^{(v)}}\left(\mathbf{G}_{k-1}\mathbf{G}_{k-1}^T\mathbf{G}_{k-1} - \mathbf{G}_{k-1}\right)\Bigg)\Bigg), \\ \mathbf{F}_k^{(v)} = \mathbf{Prox}_{(\gamma/L_2)}\Bigg(\mathbf{U}_2\mathbf{F}_{k-1}^{(v)} + \frac{1}{L_2}\mathbf{X}_v^T\mathbf{G}_{k-1}\mathbf{S}_{k-1} \\ \qquad - \frac{\beta}{L_2}\left(\mathbf{F}_{k-1}^{(v)}(\mathbf{F}_{k-1}^{(v)})^T\mathbf{F}_{k-1}^{(v)} - \mathbf{F}_{k-1}^{(v)}\right)\Bigg), \\ \mathbf{S}_k = \frac{1}{V}\sum_{v=1}^V\Bigg((\mathbf{G}_{k-1}^T\mathbf{G}_{k-1})^{-1}\mathbf{G}_{k-1}^T\mathbf{X}_v\mathbf{F}_{k-1}^{(v)} \\ \qquad (\mathbf{F}_{k-1}^{(v)}\mathbf{F}_{k-1}^{(v)})^{-1}\Bigg),\end{cases} \tag{24}$$

where $\mathbf{\Theta} = \{(\mathbf{U}_1)_k, (\mathbf{U}_2)_k\}_{k=0}^K$ is a learnable set. The i-th block computes the output with (24). Algorithm 1 describes the main procedures of the proposed method DBMC.

In the spirit of learning-based optimization, we propose an enhanced learnable network named elevated DBMC (EDBMC). Specifically, we retain the updating rule for $\mathbf{S}_k$ and replace two proximal steps in (24) with differentiable proximal operators, summarized as

$$\widehat{\mathbf{G}}_k = \xi_{(\theta_1)_k}\Bigg(\frac{1}{V}\sum_{v=1}^V\Bigg(\mathbf{U}_1\widehat{\mathbf{G}}_{k-1} + \frac{1}{L_1^{(v)}}\mathbf{X}_v\widehat{\mathbf{F}}_{k-1}^{(v)}\mathbf{S}_{k-1}^T - \frac{\alpha}{L_1^{(v)}}\mathbf{L}_{\mathbf{G}}^v\widehat{\mathbf{G}}_{k-1} - \frac{\delta}{L_1^{(v)}}\left(\widehat{\mathbf{G}}_{k-1}\widehat{\mathbf{G}}_{k-1}^T\widehat{\mathbf{G}}_{k-1} - \widehat{\mathbf{G}}_{k-1}\right)\Bigg)\Bigg), \tag{25}$$

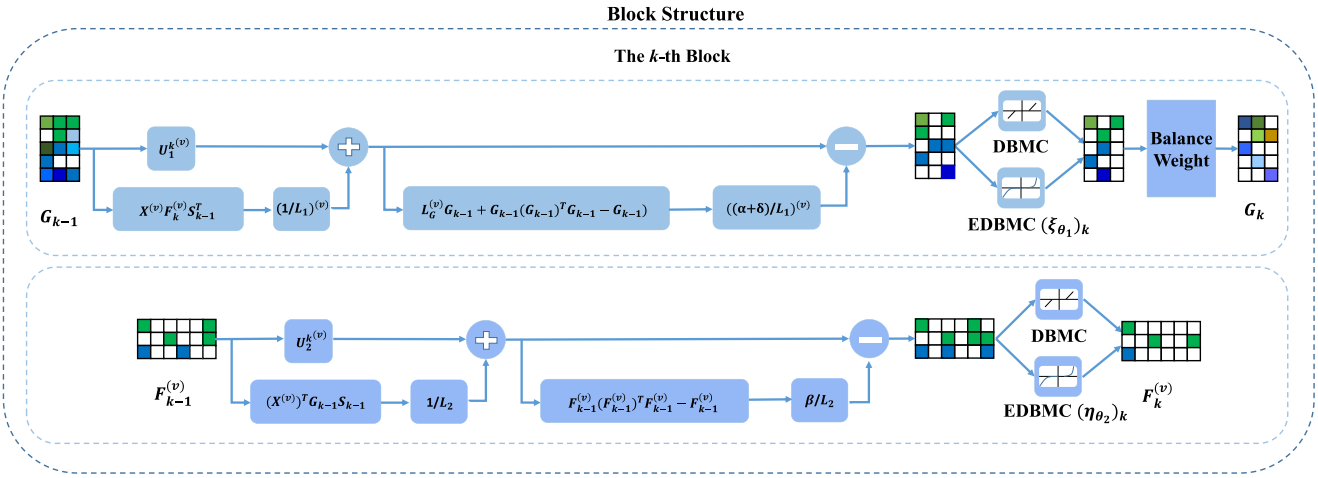


Fig. 2. One-block structure of DBMC and EDBMC.

Algorithm 1: Differentiable Bi-Sparse Multi-View Co-Clustering (DBMC).

Input: Multi-view features $\{\mathbf{X}_v\}_{v=1}^V$, cluster numbers c_1 and c_2, regularization parameters α, β, γ, δ and ζ, number of blocks t, training epochs e, and learning rate lr.

Output: Consistent collaborative representation $\mathbf{G}$.

Initialize $\mathbf{G}$, $\mathbf{S}$, $\mathbf{F}_v$, learnable parameters $\mathbf{\Theta}$, and graph Laplacian matrices $\{\mathbf{L}_{\mathbf{G}}^v\}_{v=1}^V$;
while not convergent **do**
  for $k = 1 \to t$ **do**
    Compute $\mathbf{S}$ by the (23);
    Update $\mathbf{F}_v$ by the (20);
    Update $\mathbf{G}$ by the (15);
  end for
  Update $\mathbf{\Theta}$ with back propagation;
  Update counter $k = k + 1$;
end while
return Consistent collaborative representation $\mathbf{G}$.

Algorithm 2: Elevated Differentiable Bi-Sparse Multi-View Co-Clustering (EDBMC).

Input: Multi-view features $\{\mathbf{X}_v\}_{v=1}^V$, cluster numbers c_1 and c_2, regularization parameters α, β, and ζ, number of blocks t, training epochs e, and learning rate lr.

Output: Consistent collaborative representation $\widehat{\mathbf{G}}$.

Initialize $\widehat{\mathbf{G}}$, $\mathbf{S}$, $\widehat{\mathbf{F}}_v$, learnable parameters $\widehat{\mathbf{\Theta}}$, and graph Laplacian matrices $\{\mathbf{L}_{\mathbf{G}}^v\}_{v=1}^V$;
while not convergent **do**
  for $k = 1 \to t$ **do**
    Compute $\mathbf{S}$ by the (27);
    Update $\widehat{\mathbf{F}}_v$ by the (26);
    Update $\widehat{\mathbf{G}}$ by the (25);
  end for
  Update $\widehat{\mathbf{\Theta}}$ with back propagation;
  Update counter $k = k + 1$;
end while
return Consistent collaborative representation $\widehat{\mathbf{G}}$.

$$\widehat{\mathbf{F}}_k^{(v)} = \eta_{(\theta_2)_k}\left(\mathbf{U}_2\widehat{\mathbf{F}}_{k-1}^{(v)} + \frac{1}{L_2}\mathbf{X}_v^T\widehat{\mathbf{G}}_{k-1}\mathbf{S}_{k-1} - \frac{\beta}{L_2} \times \left(\widehat{\mathbf{F}}_{k-1}^{(v)}(\widehat{\mathbf{F}}_{k-1}^{(v)})^T\widehat{\mathbf{F}}_{k-1}^{(v)} - \widehat{\mathbf{F}}_{k-1}^{(v)}\right)\right), \tag{26}$$

$$\mathbf{S}_k = \frac{1}{V}\sum_{v=1}^{V}\left((\widehat{\mathbf{G}}_{k-1}^T\widehat{\mathbf{G}}_{k-1})^{-1}\widehat{\mathbf{G}}_{k-1}^T\mathbf{X}_v\widehat{\mathbf{F}}_{k-1}^{(v)}(\widehat{\mathbf{F}}_{k-1}^{(v)}\widehat{\mathbf{F}}_{k-1}^{(v)})^{-1}\right), \tag{27}$$

where $\widehat{\mathbf{\Theta}} = \{(\mathbf{U}_1)_k, (\mathbf{U}_2)_k, (\theta_1)_k, (\theta_2)_k\}_{k=0}^K$ is the learnable set. In addition, $\xi_{(\theta_1)}(\cdot)$ and $\eta_{(\theta_2)}(\cdot)$ are differentiable proximal operators parameterized by the self-learning parameters θ_1 and θ_2, consisting of some non-linear functions (e.g. RELU) to replace the two proximal steps of entering manually parameters γ and ζ. DMBC and EDMBC are comprised of t differentiable blocks, which contain the learnable parameters $\mathbf{\Theta}$ and $\widehat{\mathbf{\Theta}}$. One-block network structure of the proposed DBMC and EDBMC is summarized in Fig. 2, and Algorithm 2 illustrates the proposed method in detail. The training objective function of the DBMC and EDBMC networks computes each view of reconstruction loss for $(\mathbf{X}_v^*)_k = \mathbf{G}_k\mathbf{S}_k(\mathbf{F}_v^T)_k$, or $\widehat{\mathbf{G}}_k\mathbf{S}_k(\widehat{\mathbf{F}}_v^T)_k$ in each epoch, represented as

$$\mathcal{J}(\mathbf{X}_v, (\mathbf{X}_v^*)_k; \mathbf{\Theta}; \widehat{\mathbf{\Theta}}) = \sum_{v=1}^{V}\|\mathbf{X}_v - (\mathbf{X}_v^*)_k\|_F^2. \tag{28}$$

D. Computational Complexity Analysis

In this subsection, we summarize and analyze the convergence and computational complexity of the networks. Problem (10) is not a resultant convex problem of all variables, so we design two three-step iterative networks to resolve the joint optimization problem (i.e., Algorithm 1). If each sub-problem is convergent,

the synthesis problem is also convergent. The convergence of each sub-problem is shown as follows.

Update S. Sub-problem (21) is a convex function, and we provide a closed-form solution in (23).

Update G and $\mathbf{F}_v$. Sub-problems (11) and (16) have the same optimization sub-problem form. In general, it is difficult to analyze whether learnable layers of $\mathbf{U}_i$ used in Algorithms 1-2 may improve upon the fixed parameter methods. However, a number of studies have proved the effectiveness of this strategy [40], [41]. The proximal step theories of ℓ_1-norm regularized problems [36] and approximate replacements [37], [39] guarantee the correctness and convergence of replacements. According to the theoretical analysis in [36], sub-problems (11) and (16) can have an analytical solution by using the proximal steps. As stated by [37], [39], using the learnable layers instead of fixed parameters can obtain the iterative solution. Therefore, sub-problems (11) and (16) can guarantee the convergence. Moreover, the time complexity of the proposed framework is $\mathcal{O}(\sum_{v=1}^{V}(\max(n^2 d_v, d_v^3) + n^3))$ for each block network, where n is the number of samples, and d_v is the number of the v-th view feature.

IV. Experiment and Study

In this section, we describe a comprehensive experimental evaluation of six classical multi-view datasets. First, the datasets, evaluation indicators, and evaluation methods are introduced. Second, we present the overall performance compared with several state-of-the-art multi-view methods. Third, we perform ablation study experiments and analyze the parameter sensitivity of the related parameters. Finally, we make conclusions from the experimental analyses. The proposed frameworks are implemented with Pytorch on a standard Ubuntu-16.04 operating system with four NVIDIA Tesla P100 GPUs. In brief, we build t-block differentiable networks with batch normalization layers, fully connected layers, and non-linear functions.

A. Datasets

Six publicly available datasets are used to verify the effectiveness of the proposed methods, including ALOI, Caltech101-7, Caltech101-20, Notting-Hill, NUS-WIDE, and WebKB-texas.

ALOI is derived from the Amsterdam Library of Object Images,[1] which consists of 1079 images of 10 small objects. Each image is represented by four types of features: RGB, HSV, color correlogram, and Haralick features.

Caltech101-7 and Caltech101-20 are based on the Caltech-101[2] dataset, and contain 1474 images in seven classes and 2386 images in 20 classes, respectively. Each image is represented by six types of features: Gabor, wavelet moments, centrist, HoG, GIST, and LBP.

Notting-Hill is a video-based dataset of face images collected from the movie *Notting-Hill*. It contains 4660 face images of the five main actors in 76 tracks. Intensity, LBP, and Gabor features are extracted for representations.

[1]https://elki.dbs.ifi.lmu.de/wiki/DataSets/MultiView

[2]https://www.vision.caltech.edu/Image_Datasets/Caltech101

TABLE II
A Brief Description of the Tested Datesets

Datasets	# Samples	# Views	# Total Features	# Classes
ALOI	1,079	4	218	10
Caltech101-7	1,474	6	3,766	7
Caltech101-20	2,386	6	3,766	20
Notting-Hill	4,660	3	12,054	5
NUS-WIDE	1,600	6	1,134	8
WebKB-texas	187	2	1,890	5

NUS-WIDE is selected from the NUS-WIDE-Object[3] dataset and contains 1600 images of the first eight classes. Each image is represented by six types of features: color histogram, color correlogram, edge direction histogram, wavelet moments, block-wise color moments, and SIFT.

WebKB-texas is selected from WebKB[4] and contains webpages collected from one of four universities: Cornell, Texas, Washington, and Wisconsin. The webpages are distributed over five clusters and described by two views: content and citation.

All tested datasets are derived from real-world applications, ranging from object images to web images. Details of the numbers of samples, views, total features, and classes are presented in Table II.

B. Experimental Setup

In this subsection, to validate the effectiveness of the proposed methods, several state-of-the-art multi-view clustering methods are used for comparison: k-means, MVKSC [42], MSC-IAS [43], MCGC [44], BMVC [45], ETMLSC [10], DMF-MVC [28], DGCCA [26], and MvSCN [30].

MVKSC: Multi-view kernel spectral clustering was formulated as a weighted kernel canonical correlation analysis in a primal-dual typical optimization method of least squares support vector machines.

MSC-IAS: Multi-view subspace clustering with intactness-aware similarity adopted Hilbert-Schmidt Independence to learn a space by encoding complementary information.

MCGC: Multi-view consensus graph clustering learned an exact consensus graph by minimizing the imparity among different views and constraining the rank of the Laplacian matrix.

BMVC: Binary multi-view clustering was based on a joint learning framework that simultaneously addressed compact collaborative discrete representation and binary structure learning.

ETLMSC: Essential tensor learning for multi-view spectral clustering proposed a novel essential tensor learning method for Markov chain based spectral clustering.

Moreover, some parameters of the compared methods should be clarified in advance. All methods are tuned using their default settings if feasible. For other open hyper-parameters, we adopt the following settings. For k-means, all parameters are run as defaults. For MVKSC, the kernel type uniformly selects the RBF

[3]https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html

[4]https://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/

TABLE III
EXPERIMENTAL PARAMETER SETTINGS OF DBMC AND EDBMC, WHERE α, β, γ, δ, AND ζ ARE HYPER-PARAMETERS

Datasets \ Parameter Name		Block	Learning Rate	Epoch	α	β	γ	δ	ζ
ALOI	DBMC	11	10^{-4}	45	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
	EDBMC	9	10^{-4}	75	10^{-3}	10^{-3}	-	10^{-3}	-
Caltech101-7	DBMC	6	10^{-4}	65	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
	EDBMC	5	10^{-3}	10	10^{-3}	10^{-3}	-	10^{-3}	-
Caltech101-20	DBMC	2	10^{-4}	40	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
	EDBMC	3	10^{-3}	15	10^{-3}	10^{-3}	-	10^{-3}	-
Notting-Hill	DBMC	2	10^{-3}	10	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
	EDBMC	4	10^{-3}	15	10^{-3}	10^{-3}	-	10^{-3}	-
NUS-WIDE	DBMC	2	10^{-4}	45	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
	EDBMC	4	10^{-4}	15	10^{-3}	10^{-3}	-	10^{-3}	-
WebKB-texas	DBMC	3	10^{-4}	30	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
	EDBMC	6	10^{-3}	5	10^{-3}	10^{-3}	-	10^{-3}	-

kernel, and the kernel parameters of t and d are tuned as $[1, 10]$. For instance, the number of the nearest neighbors for MSC-IAS is fixed as 3, and the intact space dimension is fixed as 500. For MCGC, the regularization parameter β is tuned as $\beta = 0.1$. For BMVC, we randomly generate 10% training multi-view data points for the non-linear anchor embedding. For ETLMSC, the random seed is set to 100. For DMF-MVC, the number of the nearest neighbors k is fixed at 5. For DGCCA, the number of hidden layers is set to 128. For MvSCN, the total number of pairs for Siamese networks is set to 600 000. Experimental parameter settings of the proposed networks are summarized in Table III. Particularly, cluster numbers c_1 and c_2 are set as the number of clusters in advance, respectively.

C. Evaluation Metrics

Several evaluation indicators employed in the experiments of the proposed networks are introduced. We compare DBMC and EDBMC with several state-of-the-art multi-view clustering methods. Seven well-known clustering evaluation metrics are applied to the experiments: clustering accuracy (ACC), normalized mutual information (NMI), Purity, adjusted rand index (ARI), F-score, Precision, and Recall. The higher the values of these metrics are, the better the performance is. All experiments of the proposed frameworks are run ten times, with means and standard deviations recorded as the final results.

D. Overall Performance

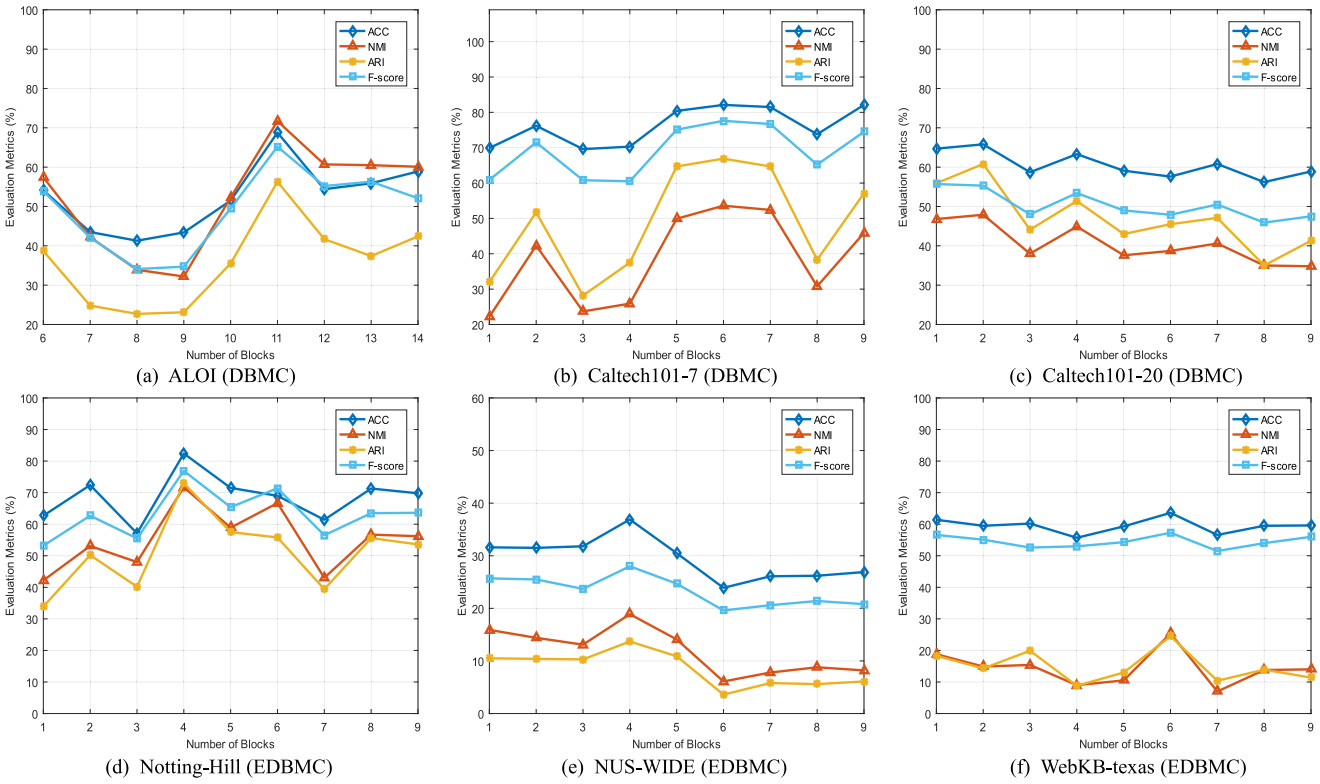
In this subsection, comprehensive experiments are described to evaluate different multi-view clustering methods. The experimental results verify the superiority of the proposed framework on six real-world multi-view datasets with nine compared algorithms and networks. From Table IV, we can obtain the following observations. On one hand, the bi-sparse multi-view co-clustering provides better performance than other state-of-the-art multi-view clustering algorithms on most tested datasets. The main reason is that the bi-sparse co-clustering keeps the space of samples and features sparse simultaneously. Furthermore, DBMC of parameter self-learning is more progressive in learning an effective consistent collaborative representation. In general, adopting deep networks to learn a latent representation is often more satisfactory than using shallow models. Specifically, DBMC and EDBMC achieve better performance than the other state-of-the-art multi-view algorithms on most tested datasets. The proposed frameworks exceed other algorithms significantly on all datasets in terms of ACC. On the other hand, the proposed frameworks demonstrate their superiority over other state-of-the-art methods. Notably, the advantage of the proposed frameworks is considerable on all six datasets. DBMC and EDBMC significantly outperform the sub-problem optimization models (such as BMVC, MCGC) and deep learning networks (such as MvSCN) in most indicators on all datasets. It can also be observed that the performance of EDBMC is better than that of DBMC, demonstrating the effectiveness of the designed learnable parameters. These results validate the feasibility and effectiveness of the proposed methods.

E. Ablation Study

In this subsection, we describe ablation experiments on the proposed models. The detailed results of the ablation studies are shown in Table V, and the parameters used here are the same as those in Table III. It can be observed that the methods that consider the topology learning matrices $\{\mathbf{L}_{\mathbf{G}}^{v}\}_{v=1}^{V}$, bi-sparse strategy, and orthogonality constraints have better performance than none of these terms are considered, while considering bi-sparse strategy performs better than only considering the topology learning and orthogonality constraints. Moreover, the performance is improved on most datasets when pairwise strategies are combined. In particular, the combination of the bi-sparse strategy and orthogonality constraints is more effective than only considering bi-sparse strategy or orthogonality constraints, which proves the necessity and effectiveness of adding the two constraints in the proposed models. When the three methods are combined simultaneously, an advantageous synergetic representation is learned, which improves the multi-view clustering performance. Furthermore, it can be seen that EDBMC automatically learns a better threshold. This value makes DBMC learn a consistent representation that is in line with the characteristics of the samples.

F. Parameter Sensitivity

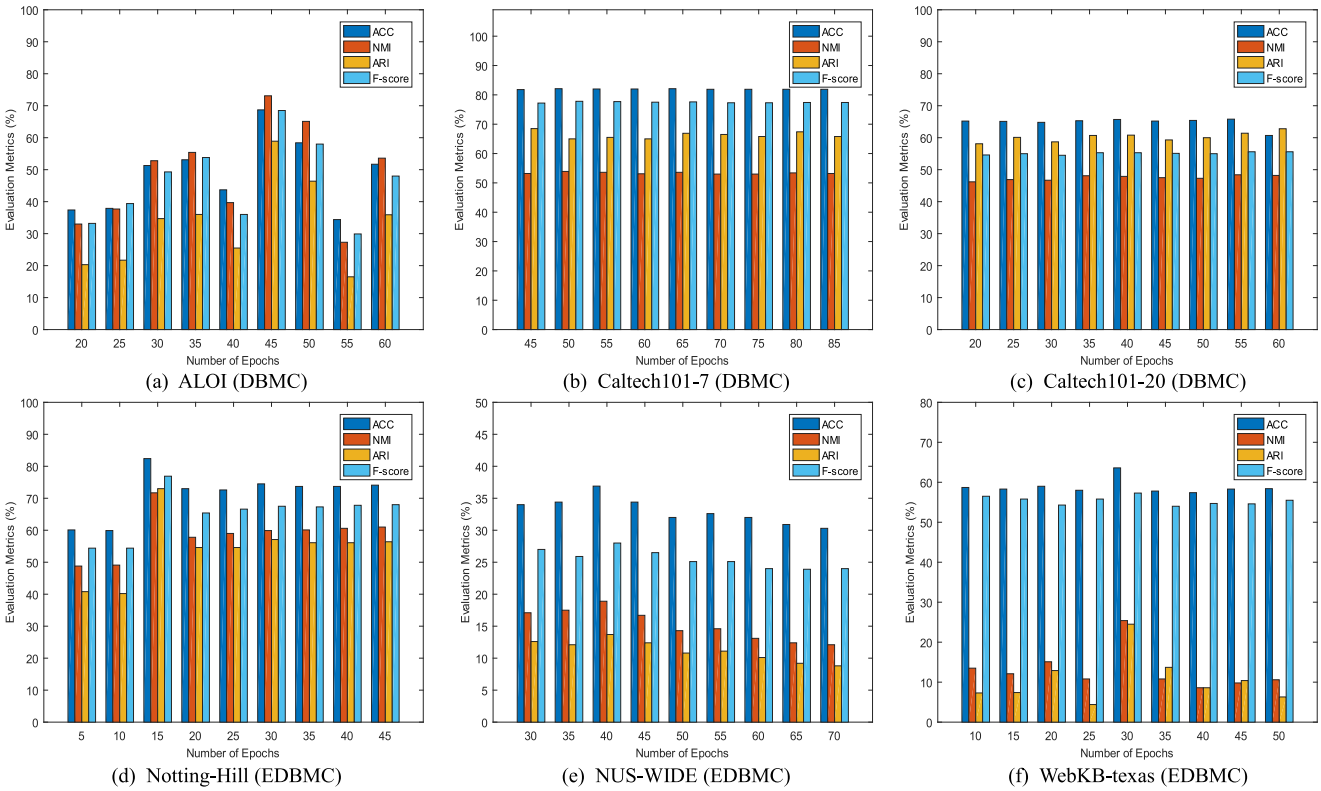
In this subsection, the parameter sensitivity of the proposed frameworks is analyzed to determine the validity of the parameter settings. DBMC shows the parameters of ALOI, Caltech101-7, and Caltech101-20, while EDBMC selects the related parameters of Notting-Hill, NUS-WIDE, and WebKB-texas for display. Notice that all parameter sensitivity analyses are carried out in the settings given in Table III. The numbers of blocks and epochs affecting on the six datasets are reported in Figs. 3–4. The other parameters are fixed to the values while tuning the numbers of blocks and epochs. We select ACC, NMI, ARI, and F-score as evaluation metrics. The numbers of blocks and epochs are searched by the grid $\{1, 2, \ldots, 14\}$ and $\{5, 10, \ldots, 85\}$, respectively. It can be observed that the overall performance increases when the number of blocks is less than 12. This enables



(a) ALOI (DBMC) (b) Caltech101-7 (DBMC) (c) Caltech101-20 (DBMC)

(d) Notting-Hill (EDBMC) (e) NUS-WIDE (EDBMC) (f) WebKB-texas (EDBMC)

Fig. 3. Parameter sensitivity analysis of DBMC and EDBMC in terms of block numbers on six datasets.



(a) ALOI (DBMC) (b) Caltech101-7 (DBMC) (c) Caltech101-20 (DBMC)

(d) Notting-Hill (EDBMC) (e) NUS-WIDE (EDBMC) (f) WebKB-texas (EDBMC)

Fig. 4. Parameter sensitivity analysis of DBMC and EDBMC in terms of epoch numbers on six datasets.

TABLE IV
CLUSTERING PERFORMANCE OF ALL COMPARED MULTI-VIEW CLUSTERING METHODS, WHERE THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINED (MEAN% AND STANDARD DEVIATION%). DMF-MVC RUNNING ON ALOI ENCOUNTERS SVD PROBLEMS

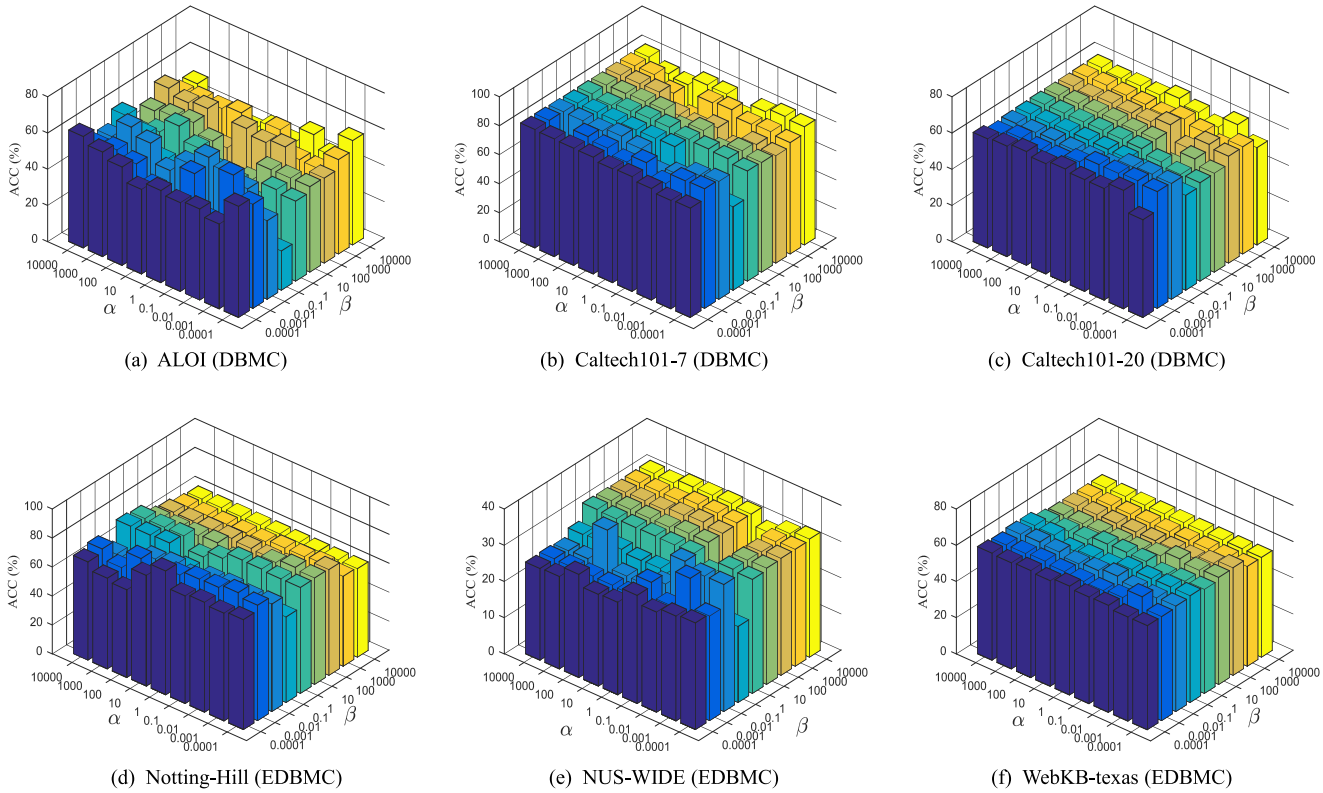
Datasets \ Methods		k-means	MVKSC	MSC-IAS	MCGC	BMVC	ETMLSC	DMF-MVC	DGCCA	MvSCN	DBMC	EDBMC
ALOI	ACC	47.5 (3.3)	60.4 (0.0)	59.4 (4.3)	52.4 (0.0)	59.6 (0.0)	72.2 (5.3)	-	57.3 (1.2)	56.0 (2.5)	68.8 (3.4)	**73.7 (2.9)**
	NMI	47.3 (2.1)	58.4 (0.0)	70.1 (1.8)	52.5 (0.0)	54.7 (0.0)	73.5 (2.1)	-	48.4 (0.4)	60.2 (2.7)	71.7 (1.4)	**74.8 (1.4)**
	Purity	48.6 (2.3)	64.4 (0.0)	67.6 (2.6)	56.5 (0.0)	60.0 (0.0)	72.2 (3.6)	-	59.9 (1.2)	57.5 (2.5)	69.0 (3.4)	**73.9 (3.9)**
	ARI	33.0 (2.9)	43.8 (0.0)	53.2 (3.5)	25.9 (0.0)	40.8 (0.0)	61.1 (4.8)	-	34.7 (1.0)	40.9 (3.1)	56.3 (3.9)	**62.9 (3.0)**
	F-score	41.1 (2.4)	49.9 (0.0)	58.5 (2.9)	37.0 (0.0)	47.4 (0.0)	68.8 (4.2)	-	46.0 (0.8)	57.5 (2.3)	65.2 (2.2)	**70.7 (1.6)**
	Precision	34.0 (2.9)	45.6 (0.0)	51.5 (5.0)	24.4 (0.0)	42.3 (0.0)	63.5 (6.5)	-	42.7 (0.7)	49.0 (2.5)	58.7 (3.0)	**65.8 (2.6)**
	Recall	52.1 (0.3)	55.0 (0.0)	68.2 (2.8)	76.1 (0.0)	53.8 (0.0)	75.4 (1.5)	-	49.9 (1.6)	69.8 (1.6)	73.5 (2.7)	**76.4 (1.0)**
Caltech101-7	ACC	49.6 (5.8)	34.2 (0.0)	71.3 (4.3)	53.6 (0.0)	35.9 (1.2)	39.4 (4.2)	56.3 (0.3)	76.3 (0.7)	58.1 (0.0)	82.1 (0.3)	**84.3 (0.3)**
	NMI	32.7 (1.9)	30.7 (0.0)	49.5 (3.8)	42.9 (0.0)	36.1 (2.0)	37.4 (0.9)	35.0 (1.0)	41.2 (2.6)	57.7 (0.0)	53.4 (1.5)	**63.5 (1.1)**
	Purity	80.4 (1.4)	78.3 (0.0)	84.7 (0.0)	83.5 (0.0)	83.3 (1.5)	81.9 (1.6)	81.5 (0.8)	76.3 (0.7)	**86.8 (0.0)**	82.2 (0.4)	85.5 (0.6)
	ARI	30.2 (4.1)	23.1 (0.0)	52.1 (6.7)	38.4 (0.0)	23.1 (2.1)	30.0 (2.4)	31.3 (0.4)	51.1 (1.9)	44.7 (0.0)	66.9 (3.8)	**80.7 (1.6)**
	F-score	49.6 (4.5)	39.5 (0.0)	69.1 (4.8)	56.5 (0.0)	40.6 (1.3)	45.4 (2.3)	52.1 (0.2)	70.8 (1.3)	66.0 (0.0)	77.6 (0.6)	**81.5 (0.8)**
	Precision	68.6 (1.5)	71.5 (0.0)	75.1 (3.5)	73.9 (0.0)	68.8 (3.6)	79.8 (1.8)	66.2 (0.3)	61.7 (1.2)	**82.0 (0.0)**	69.5 (1.8)	77.4 (1.2)
	Recall	39.0 (5.2)	27.3 (0.0)	64.0 (5.9)	45.8 (0.0)	28.8 (0.7)	31.7 (2.0)	43.0 (0.2)	83.0 (1.4)	55.2 (0.0)	**88.0 (2.6)**	86.2 (2.3)
Caltech101-20	ACC	31.3 (2.5)	49.5 (0.0)	41.9 (2.7)	62.9 (0.0)	37.9 (0.7)	45.5 (4.1)	54.8 (0.9)	59.1 (0.5)	38.3 (0.8)	65.7 (0.8)	**68.3 (0.4)**
	NMI	34.5 (1.1)	18.2 (0.0)	36.8 (2.5)	54.6 (0.0)	30.3 (0.3)	43.2 (1.4)	53.0 (0.5)	41.1 (0.9)	49.5 (0.8)	47.9 (1.1)	**54.7 (0.6)**
	Purity	59.0 (1.0)	49.5 (0.0)	56.6 (2.0)	64.4 (0.0)	59.5 (0.3)	55.4 (2.8)	63.2 (0.5)	59.1 (0.5)	68.3 (0.4)	65.8 (0.7)	**68.8 (0.5)**
	ARI	18.9 (1.8)	25.4 (0.0)	16.9 (3.0)	36.6 (0.0)	32.3 (0.7)	33.6 (2.7)	39.8 (1.3)	55.3 (1.9)	25.8 (0.5)	60.8 (2.9)	**69.0 (2.7)**
	F-score	27.2 (1.8)	43.7 (0.0)	33.7 (2.1)	49.1 (0.0)	39.0 (0.7)	39.0 (2.6)	47.5 (1.0)	51.1 (0.6)	39.5 (0.1)	55.3 (0.8)	**60.2 (0.5)**
	Precision	43.0 (2.6)	28.9 (0.0)	27.3 (2.5)	40.4 (0.0)	**60.4 (3.7)**	26.0 (2.0)	17.0 (0.0)	53.5 (0.0)	58.1 (0.6)	49.5 (1.5)	55.7 (1.4)
	Recall	19.9 (1.5)	33.6 (0.0)	44.2 (3.7)	62.6 (0.0)	28.2 (0.6)	58.1 (3.4)	39.3 (1.3)	64.0 (1.7)	29.9 (0.0)	62.8 (1.0)	**65.6 (1.4)**
Notting-Hill	ACC	63.7 (0.7)	65.5 (11.8)	69.3 (0.0)	52.3 (0.0)	69.6 (0.0)	80.2 (2.6)	73.8 (0.0)	75.3 (3.8)	71.0 (0.0)	78.1 (4.1)	**82.4 (4.1)**
	NMI	58.0 (4.9)	51.2 (10.1)	63.7 (0.0)	60.1 (0.0)	57.1 (0.0)	69.8 (1.7)	65.0 (0.0)	61.2 (2.6)	63.3 (0.0)	65.3 (1.6)	**71.7 (2.2)**
	Purity	70.1 (5.6)	68.1 (10.3)	79.0 (0.0)	69.0 (0.0)	71.8 (0.0)	83.2 (2.6)	78.8 (0.0)	75.7 (3.9)	71.3 (0.0)	78.6 (4.2)	**86.0 (4.1)**
	ARI	49.7 (6.9)	49.3 (14.4)	56.6 (0.0)	41.3 (0.0)	52.1 (0.1)	72.1 (2.1)	60.1 (0.0)	56.8 (0.4)	59.6 (0.0)	66.0 (3.4)	**73.0 (2.8)**
	F-score	61.7 (4.8)	62.1 (9.8)	65.9 (0.0)	56.8 (0.0)	62.3 (0.0)	68.2 (2.0)	69.0 (0.0)	68.8 (2.3)	68.1 (0.0)	71.8 (1.6)	**76.9 (2.4)**
	Precision	57.4 (7.1)	56.4 (13.2)	67.6 (0.0)	47.1 (0.0)	64.4 (0.0)	64.0 (1.4)	68.1 (0.0)	62.3 (4.4)	65.2 (0.0)	66.3 (3.2)	**73.0 (4.9)**
	Recall	67.2 (5.2)	70.8 (7.2)	64.2 (0.0)	71.7 (0.0)	60.3 (0.0)	**93.9 (3.8)**	69.9 (0.0)	77.2 (0.1)	71.2 (0.0)	78.7 (2.3)	81.4 (0.3)
NUS-WIDE	ACC	32.0 (0.7)	32.4 (1.8)	31.1 (0.0)	25.7 (0.0)	25.2 (0.5)	24.9 (1.2)	15.6 (0.4)	29.6 (0.0)	30.2 (1.4)	35.5 (0.7)	**36.9 (0.8)**
	NMI	17.5 (0.7)	14.4 (0.0)	**21.1 (0.6)**	14.7 (0.0)	19.0 (0.4)	12.8 (0.7)	13.3 (0.3)	10.0 (0.3)	20.1 (1.5)	18.5 (1.2)	18.9 (1.0)
	Purity	34.4 (1.2)	32.3 (0.0)	37.0 (1.2)	29.3 (0.0)	**39.4 (0.7)**	26.5 (1.3)	28.9 (0.4)	26.6 (0.2)	34.2 (1.7)	36.3 (0.7)	37.8 (1.7)
	ARI	9.00 (0.7)	10.4 (0.0)	11.4 (0.7)	5.70 (0.0)	13.5 (0.4)	6.80 (0.5)	5.90 (0.2)	7.20 (1.6)	9.50 (1.0)	12.9 (0.4)	**13.7 (0.5)**
	F-score	24.3 (0.6)	22.1 (0.0)	24.9 (0.5)	25.0 (0.0)	24.9 (0.4)	18.4 (0.5)	19.4 (0.2)	23.0 (0.4)	26.4 (1.1)	**28.4 (1.5)**	28.0 (1.1)
	Precision	18.0 (0.6)	21.1 (0.0)	20.7 (0.5)	15.2 (0.0)	23.4 (0.4)	18.5 (0.5)	16.8 (0.2)	16.6 (0.1)	**27.2 (0.7)**	24.3 (1.1)	24.6 (1.4)
	Recall	37.5 (2.6)	23.1 (0.0)	30.0 (0.8)	**71.1 (0.0)**	26.3 (0.4)	18.3 (0.4)	22.9 (0.3)	37.4 (2.3)	25.6 (1.4)	34.3 (2.9)	32.8 (3.2)
WebKB-texas	ACC	53.1 (9.1)	47.8 (3.0)	54.6 (0.0)	50.3 (0.0)	52.4 (0.0)	26.6 (1.1)	54.7 (0.6)	58.6 (0.4)	55.3 (0.0)	61.3 (0.9)	**63.6 (2.5)**
	NMI	7.10 (5.2)	9.70 (2.3)	8.70 (0.0)	16.0 (0.0)	17.0 (0.0)	2.80 (0.5)	2.10 (0.4)	13.3 (0.4)	23.0 (0.0)	17.8 (2.0)	**25.4 (4.9)**
	Purity	59.8 (5.9)	57.5 (1.9)	57.2 (0.0)	59.4 (0.0)	57.7 (0.3)	27.7 (1.4)	55.9 (0.3)	**72.5 (4.6)**	68.1 (0.0)	61.3 (0.9)	66.0 (2.8)
	ARI	7.00 (16.6)	12.0 (3.4)	14.8 (0.0)	13.8 (0.0)	17.9 (0.0)	0.10 (0.0)	2.20 (0.8)	13.1 (2.5)	20.0 (0.0)	17.2 (1.6)	**24.5 (6.1)**
	F-score	48.4 (5.0)	42.6 (0.4)	53.2 (0.0)	45.3 (0.0)	43.0 (0.0)	25.9 (0.4)	53.3 (0.4)	53.5 (1.1)	49.5 (0.0)	56.2 (0.9)	**57.3 (2.0)**
	Precision	40.4 (7.5)	45.3 (2.4)	43.7 (0.0)	45.9 (0.0)	**52.1 (0.0)**	20.0 (0.4)	36.9 (0.3)	42.1 (0.5)	50.7 (0.0)	44.0 (0.8)	48.4 (3.1)
	Recall	75.6 (1.6)	40.4 (5.4)	67.9 (0.0)	44.9 (0.0)	36.6 (0.0)	37.1 (0.6)	**95.6 (0.6)**	73.8 (6.5)	48.4 (0.0)	77.5 (1.0)	70.9 (6.6)

TABLE V
ABLATION STUDY OF SEVERAL ABLATION METHODS ON THE TESTED DATASETS, WHERE THE BEST AND SECOND BEST ACC PERFORMANCE ARE HIGHLIGHTED IN BOLD AND UNDERLINED (MEAN% AND STANDARD DEVIATION%). THE ✓ MEANS THAT THIS STRATEGY IS CONSIDERED, AND × OTHERWISE

Datasets	Graph Laplacian	×	✓	×	×	✓	✓	×	✓	✓
	Bi-Sparsity	×	×	✓	×	✓	×	✓	✓	✓
	Orthogonality Constraints	×	×	×	✓	×	✓	✓	✓	✓
ALOI		53.1 (4.0)	53.9 (0.1)	54.4 (0.9)	55.5 (2.9)	60.6 (0.7)	55.4 (3.0)	48.6 (1.9)	68.8 (3.4)	**73.7 (2.9)**
Caltech101-7		74.0 (2.3)	74.6 (2.2)	77.6 (1.4)	75.0 (0.5)	78.2 (0.3)	81.9 (0.6)	79.8 (0.3)	82.1 (0.3)	**84.3 (0.3)**
Caltech101-20		57.3 (0.6)	58.0 (3.2)	59.9 (0.7)	58.7 (0.9)	60.1 (0.6)	64.6 (0.6)	65.7 (0.5)	65.7 (0.8)	**68.3 (0.4)**
Notting-Hill		50.6 (2.0)	64.8 (0.9)	69.3 (1.1)	60.6 (1.6)	75.4 (0.3)	52.3 (2.4)	73.7 (3.6)	78.1 (4.1)	**82.4 (4.1)**
NUS-WIDE		25.6 (0.3)	27.3 (1.2)	28.5 (0.9)	26.2 (0.5)	32.0 (0.4)	27.8 (0.7)	30.0 (0.5)	35.5 (0.7)	**36.9 (0.8)**
WebKB-texas		57.8 (0.8)	58.0 (1.2)	58.5 (0.4)	60.0 (2.6)	60.3 (0.1)	58.9 (0.5)	59.4 (0.2)	61.3 (0.9)	**63.6 (2.5)**

us to consider this critical point when selecting the number of blocks. Moreover, we can also observe that the performance is relatively encouraging within 50 epochs and degrades beyond this value. This means that a larger number of epochs does not imply better performance. Generally, the number of blocks maintains good robustness within a fixed range (such as ALOI 8-13, Notting-Hill 3-5) with different ranges for each dataset. If the number is smaller or larger than this range, the performance of the models is not stable enough, and remarkable performance cannot be obtained. Meanwhile, the number of epochs is kept within 70, which can achieve a desirable multi-view clustering effect. Therefore, the numbers of blocks and training epochs are indicated in Table III.

Fig. 5 shows the parameter sensitivity of α, β, and δ in DBMC and EDBMC for the six datasets. The abscissa of the values α, β, and δ are searched by the grid $\{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$,



(a) ALOI (DBMC) (b) Caltech101-7 (DBMC) (c) Caltech101-20 (DBMC)

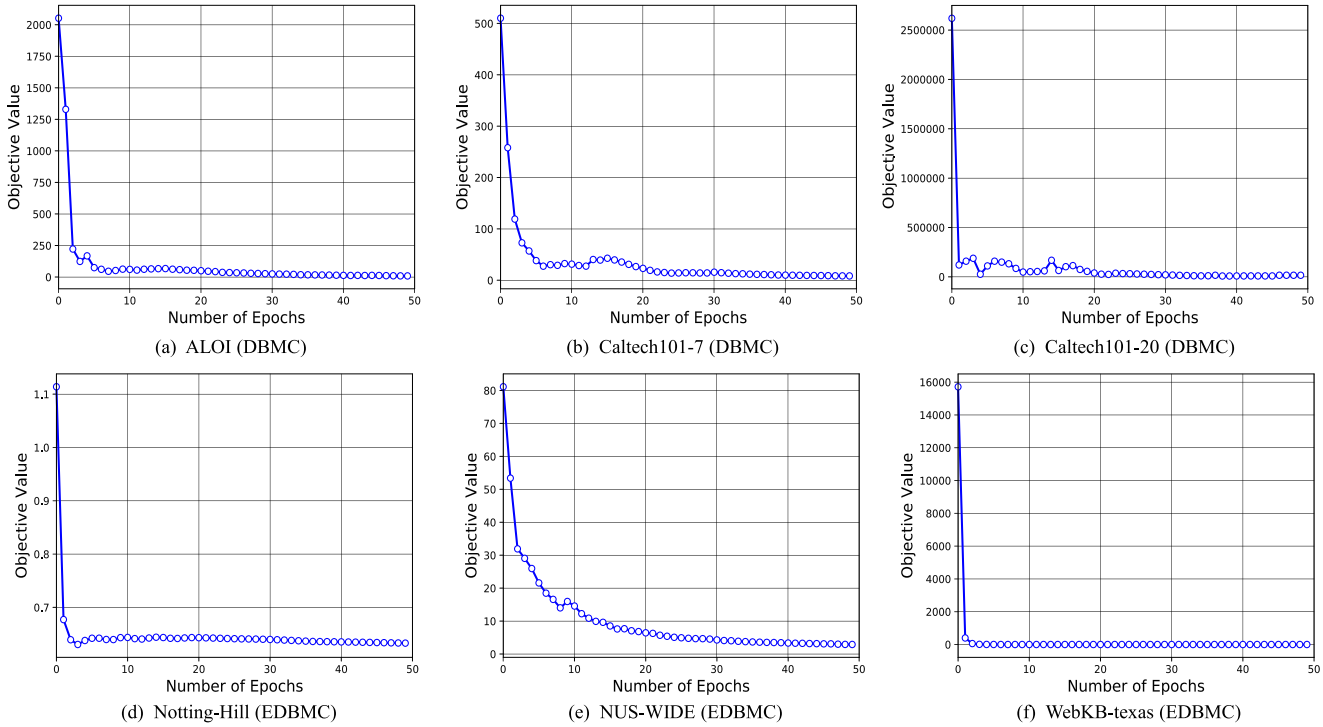
(d) Notting-Hill (EDBMC) (e) NUS-WIDE (EDBMC) (f) WebKB-texas (EDBMC)

Fig. 5. Parameter sensitivity analysis of α, β, and δ of DBMC and EDBMC on six datasets, where the values of α, β, and δ range within $\{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$, and $\beta = \delta$.

TABLE VI

PARAMETER SENSITIVITY ANALYSIS OF PROVIDING γ AND ζ VALUES IN DBMC AND USING DIFFERENTIABLE PROXIMAL OPERATORS TO REPLACE TWO PROXIMAL STEPS IN EDBMC ON SIX DATASETS, WHERE γ AND ζ RANGE WITHIN $\{10^{-4}, 10^{-3}, \ldots, 10^{2}\}$ IN DBMC. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINED (MEAN% AND STANDARD DEVIATION%)

Methods		DBMC							EDBMC
Datasets \ γ and ζ		10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}	Learned γ and ζ
ALOI	NMI	38.7 (0.3)	71.7 (1.4)	60.4 (2.1)	10.0 (0.6)	34.9 (1.0)	48.1 (0.8)	45.7 (0.6)	**74.8 (1.4)**
	Purity	47.5 (0.7)	69.0 (3.4)	53.7 (5.6)	25.3 (0.6)	48.1 (2.5)	47.1 (0.7)	42.7 (0.6)	**73.9 (3.9)**
	F-score	37.8 (0.5)	65.2 (2.2)	56.2 (2.4)	20.6 (1.2)	33.7 (1.0)	42.9 (0.9)	40.8 (0.7)	**70.7 (1.6)**
Caltech101-7	NMI	33.9 (0.9)	53.4 (1.5)	45.3 (1.9)	13.5 (0.1)	47.9 (0.3)	40.7 (1.6)	35.5 (0.9)	**63.5 (1.1)**
	Purity	47.1 (3.7)	82.2 (0.4)	79.9 (0.4)	62.6 (0.4)	78.0 (0.1)	77.8 (1.2)	75.8 (0.6)	**85.5 (0.6)**
	F-score	59.4 (1.7)	77.6 (0.6)	73.1 (0.6)	52.8 (0.8)	74.3 (0.1)	71.0 (1.3)	67.7 (0.9)	**81.5 (0.8)**
Caltech101-20	NMI	48.4 (1.1)	47.9 (1.1)	46.9 (0.9)	26.1 (1.6)	26.2 (1.3)	32.9 (0.7)	39.7 (1.3)	**54.7 (0.5)**
	Purity	65.5 (0.6)	65.8 (0.7)	64.3 (0.4)	54.1 (1.4)	54.2 (0.9)	57.7 (0.3)	60.8 (1.7)	**68.8 (0.5)**
	F-score	54.3 (0.9)	55.3 (0.8)	54.8 (0.7)	41.7 (1.5)	**63.2 (0.6)**	46.4 (0.4)	51.4 (0.8)	60.2 (0.5)
Notting-Hill	NMI	61.5 (0.3)	65.3 (1.6)	51.4 (1.7)	31.8 (0.6)	59.3 (1.0)	62.1 (0.0)	52.1 (0.0)	**71.7 (2.2)**
	Purity	75.0 (0.2)	78.6 (4.2)	73.5 (1.9)	59.8 (2.3)	74.2 (0.1)	76.3 (0.0)	54.1 (0.0)	**86.0 (4.1)**
	F-score	68.7 (0.3)	71.8 (1.6)	62.2 (0.5)	48.7 (0.4)	66.3 (0.6)	67.8 (0.0)	57.8 (0.0)	**76.9 (2.4)**
NUS-WIDE	NMI	12.8 (0.5)	18.5 (1.2)	18.2 (1.1)	4.80 (0.4)	15.5 (0.4)	15.3 (0.4)	14.9 (0.8)	**18.9 (1.0)**
	Purity	30.8 (0.8)	36.3 (0.7)	36.2 (0.5)	26.9 (0.6)	32.9 (0.2)	33.1 (0.2)	33.0 (0.6)	**37.8 (1.7)**
	F-score	23.6 (0.3)	**28.4 (1.5)**	27.7 (1.4)	21.0 (0.5)	26.6 (0.4)	26.5 (0.4)	25.9 (0.5)	28.0 (1.1)
WebKB-texas	NMI	9.10 (0.9)	17.8 (2.0)	8.20 (2.0)	11.5 (1.5)	8.80 (1.1)	9.50 (0.6)	9.60 (0.5)	**25.4 (4.9)**
	Purity	59.6 (0.3)	61.3 (0.9)	58.2 (0.8)	60.6 (0.8)	60.4 (2.2)	56.7 (2.0)	57.0 (1.5)	**66.0 (2.8)**
	F-score	53.5 (0.1)	56.2 (0.9)	55.1 (0.5)	52.5 (1.2)	51.1 (1.1)	52.3 (1.7)	54.3 (0.5)	**57.3 (2.0)**



(a) ALOI (DBMC) (b) Caltech101-7 (DBMC) (c) Caltech101-20 (DBMC)

(d) Notting-Hill (EDBMC) (e) NUS-WIDE (EDBMC) (f) WebKB-texas (EDBMC)

Fig. 6. The convergent curves of the proposed frameworks on the six tested datasets.

and $\beta = \delta$. It can be seen that the methods learn a satisfactory geometric or topological structure of the associated features and the good orthogonality of the consistent representations. When α, β, and δ are fixed as 10^{-3}, the models obtain good overall performance. In contrast, if the regularization terms do not select a definitely suitable parameter, it will lead to learning a deformed topology and a twisted orthogonality, resulting in invalid clustering performance. Thus, the values of α, β, and δ are consistent in Table III.

Table VI shows the influence of providing γ and ζ values in DBMC and using differentiable proximal operators to replace two proximal steps in EDBMC. The values of γ and ζ are searched by the grid $\{10^{-4}, 10^{-3}, \ldots, 10^{2}\}$. We select NMI, Purity, and F-score of the proposed frameworks as evaluation metrics. On one hand, it can be observed that the three indicators illustrate the superiority of DBMC when γ and $\zeta = 10^{-3}$. Therefore, the values of γ and ζ used in DBMC are presented in Table III. On the other hand, the differentiable proximal operators of the self-learning parameters in EDBMC provide better performance than fixing γ and ζ values in the two proximal steps, and EDBMC performs better than DBMC. This demonstrates that we have learned a more advantageous sparse representation with learnable parameters for consistent collaborative representations.

The loss values of DBMC and EDBMC with increasing numbers of epochs on different datasets are shown in Fig. 6. It can be seen that the loss values of DBMC and EDBMC gradually decrease as the number of epochs increases. Eventually, it will converge to a stable value and fluctuate slightly when the epoch number is large enough, which indicates convergence.

V. Conclusion

In this paper, we proposed an effective differentiable network called DBMC and an improved version named EDBMC, which were transformed into deep networks based on the proposed objective loss functions. Each block of the networks was differentiable and reusable. Moreover, DBMC and EDBMC learned a joint and consistent multi-view collaborative representation from multi-view features and guaranteed sparsity between the multi-view feature space and consistent collaborative representation space. Correspondingly, two effective frameworks of convergence were proved. Finally, comprehensive experiments on six real-world datasets validated the effectiveness of the proposed frameworks. From the experimental results, it is concluded that DBMC provided improved clustering results, while EDBMC primarily enhanced the capability of parameter self-learning and obtained even better performance than DBMC. Deep differentiable networks based on optimization methods could guide the multi-view clustering tasks to obtain more interpretable performance. In future work, we will attempt to extend the proposed networks to more realistic applications by combining semi-supervised methods, subspace learning, and other techniques, and explore more interpretable deep networks.

References

[1] L. Bai and J. Liang, "A three-level optimization model for nonlinearly separable clustering," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 3211–3218.

[2] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong, and C. C. Loy, "Online deep clustering for unsupervised representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6688–6697.

[3] T.-E. Lin, H. Xu, and H. Zhang, "Discovering new intents via constrained deep adaptive clustering with cluster refinement," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 8360–8367.
[4] A. Bansal, M. Vo, Y. Sheikh, D. Ramanan, and S. Narasimhan, "4D visualization of dynamic events from unconstrained multi-view videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5366–5375.
[5] S. M. Ahmed and C. M. Chew, "Density-based clustering for 3D object detection in point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10608–10617.
[6] J. Zhang *et al.*, "Self-supervised convolutional subspace clustering network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5473–5482.
[7] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, "Deep spectral clustering using dual autoencoder network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4066–4075.
[8] S. Sarfraz, V. Sharma, and R. Stiefelhagen, "Efficient parameter-free clustering using first neighbor relations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8934–8943.
[9] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, and Y. Qu, "On unifying multi-view self-representations for clustering by tensor multi-rank minimization," *Int. J. Comput. Vis.*, vol. 126, no. 11, pp. 1157–1179, 2018.
[10] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5910–5922, Dec. 2019.
[11] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2006, pp. 126–135.
[12] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Proc. 31st Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4129–4138.
[13] J. Han, K. Song, F. Nie, and X. Li, "Bilateral k-means algorithm for fast co-clustering," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1969–1975.
[14] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and M. Hong, "Structured sumcor multiview canonical correlation analysis for large-scale data," *IEEE Trans. Signal Process.*, vol. 67, no. 2, pp. 306–319, Jan. 2019.
[15] Y. Chen, S. Wang, C. Peng, Z. Hua, and Y. Zhou, "Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 30, pp. 4022–4035, 2021.
[16] S. Li, W.-T. Li, and W. Wang, "Co-GCN for multi-view semi-supervised learning," in *Proc. 33d AAAI Conf. Artif. Intell.*, 2020, pp. 4691–4698.
[17] Y. Chen, X. Xiao, C. Peng, G. Lu, and Y. Zhou, "Low-rank tensor graph learning for multi-view subspace clustering," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2021.3055625.
[18] 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.*, to be published, doi: 10.1109/TPAMI.2021.3082632.
[19] Y. Chen, X. Xiao, Z. Hua, and Y. Zhou, "Adaptive transition probability matrix learning for multiview spectral clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2021.3059874.
[20] Y. Xie, W. Zhang, Y. Qu, L. Dai, and D. Tao, "Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 572–586, Feb. 2020.
[21] M. Yin, J. Gao, S. Xie, and Y. Guo, "Multiview subspace clustering via tensorial t-product representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 851–864, Mar. 2019.
[22] Y. Chen, X. Xiao, and Y. Zhou, "Jointly learning Kernel representation tensor and affinity matrix for multi-view clustering," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 1985–1997, Aug. 2020.
[23] Q. Lyu and X. Fu, "Nonlinear multiview analysis: Identifiability and neural network-assisted implementation," *IEEE Trans. Signal Process.*, vol. 68, pp. 2697–2712, 2020.
[24] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1247–1255.
[25] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1083–1092.
[26] A. Benton, H. Khayrallah, B. Gujral, D. A. Reisinger, S. Zhang, and R. Arora, "Deep generalized canonical correlation analysis," in *Proc. Workshop Representation Learn.*, 2019, pp. 1–6.
[27] Q. Gao, H. Lian, Q. Wang, and G. Sun, "Cross-modal subspace clustering via deep canonical correlation analysis," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 3938–3945.
[28] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2921–2927.
[29] C. Zhang, Y. Liu, and H. Fu, "AE^2-Nets: Autoencoder in autoencoder networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2577–2585.
[30] 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.*, 2019, pp. 2563–2569.
[31] K. Davila, S. Setlur, D. Doermann, U. K. Bhargava, and V. Govindaraju, "Chart mining: A survey of methods for automated chart analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2020.2992028.
[32] P. Xu, Z. Deng, K.-S. Choi, L. Cao, and S. Wang, "Multi-view information-theoretic co-clustering for co-occurrence data," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 379–386.
[33] F. Nie, S. Shi, and X. Li, "Auto-weighted multi-view co-clustering via fast matrix factorization," *Pattern Recognit.*, vol. 102, 2020, Art. no. 107207.
[34] S. Hu, X. Yan, and Y. Ye, "Dynamic auto-weighted multi-view co-clustering," *Pattern Recognit.*, vol. 99, 2020, Art. no. 107101.
[35] L. Sun, C. H. Nguyen, and H. Mamitsuka, "Multiplicative sparse feature decomposition for efficient multi-view multi-task learning," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3506–3512.
[36] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[37] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 399–406.
[38] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1828–1837.
[39] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds," in *Proc. 32nd Conf. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9061–9071.
[40] J. Li, C. Fang, and Z. Lin, "Lifted proximal operator machines," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 4181–4188.
[41] X. Xie, J. Wu, G. Liu, Z. Zhong, and Z. Lin, "Differentiable linearized ADMM," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6902–6911.
[42] L. Houthuys, R. Langone, and J. A. K. Suykens, "Multi-view kernel spectral clustering," *Inf. Fusion*, vol. 44, pp. 46–56, 2018.
[43] X. Wang, Z. Lei, X. Guo, C. Zhang, H. Shi, and S. Z. Li, "Multi-view subspace clustering with intactness-aware similarity," *Pattern Recognit.*, vol. 88, pp. 50–63, 2019.
[44] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1261–1270, Mar. 2019.
[45] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

Shide Du received the B.S. degree in 2019 from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, where he is currently working toward the M.S. degree. His current research interests include machine learning, deep learning, and differentiable programming.

Zhanghui Liu received the M.S. degree from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, in 1997. He is currently an Associate Professor with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His current research interests include big data technology, intelligent computing, and information security.

Zhaoliang Chen received the B.S. degree in 2019 from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, where he is currently working toward the M.S. degree. His current research interests include machine learning and differentiable programming.

Wenyuan Yang received the M.S. degree in computer science from Fuzhou University, Fuzhou, China, in 2005 and the Ph.D. degree in nuclear science and engineering from Xiamen University, Xiamen, China, in 2014. He is currently a Full Professor with the Fujian Key Laboratory of Granular Computing and Application, Minnan Normal University, Zhangzhou, China. His research interests include computer vision, pattern recognition, and machine learning.

Shiping Wang received the Ph.D. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China in 2014. From August 2015 to August 2016, he was a Research Fellow with Nanyang Technological University, Singapore. He is currently a Full Professor and Qishan Scholar with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His research interests include machine learning, computer vision, and granular computing.