Information Sciences 626 (2023) 572–585

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

DBO-Net: Differentiable bi-level optimization network for multi-view clustering

Zihan Fang [a,b], Shide Du [a,b], Xincan Lin [a,b], Jinbin Yang [a,b], Shiping Wang [a,b,*], Yiqing Shi [c]

[a] College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China
[b] Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China
[c] College of Photonic and Electronic Engineering, Fujian Normal University, Fuzhou 350117, China

ARTICLE INFO

Article history:
Received 26 September 2022
Received in revised form 8 January 2023
Accepted 11 January 2023
Available online 16 January 2023

Keyword:
Multi-view clustering
Interpretable deep learning
Bi-level optimization
Differentiable network

ABSTRACT

Multi-view clustering on traditional optimization methods is derived from different theoretical frameworks, yet it may be inefficient in dealing with complex multi-view data compared to deep models. In contrast, deep multi-view clustering methods for implicit optimization have excellent feature abstraction ability but are inscrutable due to their black-box problem. However, very limited research was devoted to integrating the advantages of the above two types of methods to design an efficient method for multi-view clustering. Focusing on these problems, this paper proposes a differentiable bi-level optimization network (DBO-Net) for multi-view clustering, which is implemented by incorporating the traditional optimization method with deep learning to design an interpretable deep network. To enhance the representation capability, the proposed DBO-Net is constructed by stacking multiple explicit differentiable block networks to learn an interpretable consistent representation. Then all the learned parameters can be implicitly optimized through back-propagation, making the learned representation more suitable for the clustering task. Extensive experimental results validate that the strategy of bi-level optimization can effectively improve clustering performance and the proposed method is superior to the state-of-the-art clustering methods.

© 2023 Elsevier Inc. All rights reserved.

1. Introduction

Compared to single-view data, multi-view data can record descriptive data of objects from different perspectives with similar high-level semantics. In order to exploit multi-view data effectively and capture complementary and consistent information from all views, multi-view learning has emerged [1,2]. Currently, multi-view learning has been adopted in various domains, such as image classification [3], data mining [4], and transfer learning [5], proving its extraordinary potential. As deep learning improves by leaps and bounds, more researchers are inspired to employ flexible deep models for multi-view learning with promising results, such as auto-encoders [6], convolutional neural networks [7], and generative adversarial networks [8].

* Corresponding author at: College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China.
E-mail addresses: fzihan11@163.com (Z. Fang), dushidems@gmail.com (S. Du), xincanlinms@gmail.com (X. Lin), yangjinbinfzu@163.com (J. Yang), shipingwangphd@163.com (S. Wang), 417shelly@gmail.com (Y. Shi).

https://doi.org/10.1016/j.ins.2023.01.071
0020-0255/© 2023 Elsevier Inc. All rights reserved.

Among the multi-view learning technologies, numerous multi-view clustering methods have been extended from well-defined theoretic principles [9], such as subspace clustering [10,11], multiple kernel-based clustering [12,13], co-clustering [14,15], and graph-based clustering [16,17]. These multi-view learning methods can be broadly divided into two categories: one is optimization based learning methods for explicit solutions [18], and the other is deep learning based methods for implicit optimization [19]. The explicit optimization-based methods is designed with specific physical processes and solve the objective function by convex optimization methods, with a solid theoretical foundation and convincing results. But the ability to extract nonlinear relationships of complex data is relatively uncompetitive compared to deep networks, which is crucial to reveal the clustering distribution of the data. Instead of the handcrafted transformation parameters of the shallow learning model, the implicit optimization-based method is data-driven and learns complex mappings by implicitly updating deep network structures, allowing for superior performance over traditional shallow models [20]. However, only a small fraction of research has applied theoretical expertise to designing deep models [19,21], and the majority of studies are constrained from providing an explainable mechanism during reasoning processes for high stakes decisions in some security-sensitive fields.

Bi-level optimization has served as a powerful tool for many problems, such as hyperparametric optimization [22], meta-learning [23], and adversarial learning [24]. Motivated by these diverse applications, it is promising to design an interpretable deep network for multi-view clustering by introducing bi-level optimization as a bridge linking traditional models to deep learning. In this case, explicit optimization is performed as an upper-level task and implicit optimization as a lower-level task. Thus, it is able to maintain the powerful processing capability of deep networks for complex data while remaining to be interpretable.

For the above considerations, this paper proposes a differentiable bi-level optimization network (DBO-Net) for multi-view clustering that can simultaneously perform both explicit and implicit optimization. To be specific, the traditional optimization method learns a feature representation and optimizes it explicitly as the upper-level task and the deep network performs back-propagation to implicitly optimize parameters as the lower-level task. The overall framework of DBO-Net is demonstrated in Fig. 1. The explicit optimization module learns an interpretable representation by reformulating the iterative rule, then the implicit optimization module uses an implicit loss function for joint training and updates the parameters of the entire network by back propagation. The highlights of this paper are summarized as follows:

- We propose a differentiable bi-level optimization network for multi-view clustering with an interpretable mechanism while preserving the powerful feature abstraction capability of deep networks.
- An explicit optimization module is constructed from multiple differentiable blocks to learn an interpretable representation, and an implicit optimization module performs back-propagation to update the parameters to make the representation more cluster-friendly.
- The proposed framework is compared with state-of-the-art clustering algorithms on six multi-view benchmark datasets and experimental results demonstrate its superiority and effectiveness.

The following sections are arranged as follows. In Section 2, we briefly review related works about multi-view clustering and interpretable bi-level optimization network. In Section 3, DBO-Net is proposed to introduce the bi-level optimization loss function and the parameter optimization details. In Section 4, extensive experiments on six real-world datasets are conducted to validate the clustering performance of DBO-Net. In Section 5, we summarize the whole paper.

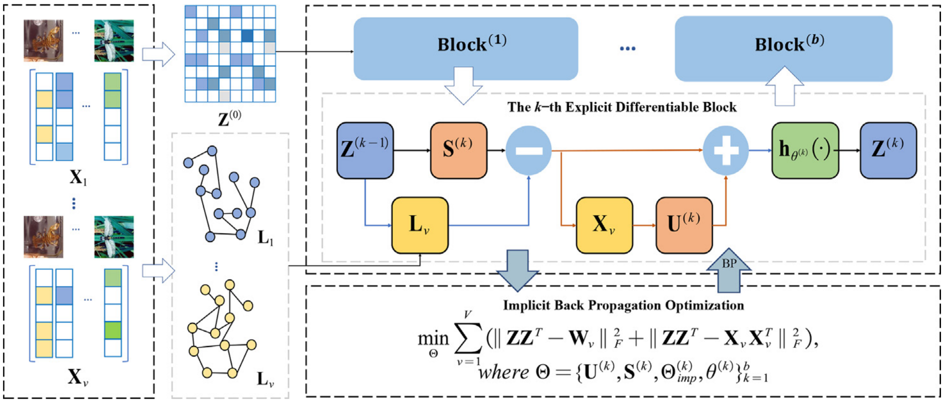

$\mathbf{X}_1$
$\mathbf{X}_v$
$\mathbf{Z}^{(0)}$
$\mathbf{L}_1$
$\mathbf{L}_v$
Block$^{(1)}$
Block$^{(b)}$
The k–th Explicit Differentiable Block
$\mathbf{Z}^{(k-1)}$
$\mathbf{S}^{(k)}$
$\mathbf{L}_v$
$\mathbf{X}_v$
$\mathbf{U}^{(k)}$
$\mathbf{h}_{\theta^{(k)}}(\cdot)$
$\mathbf{Z}^{(k)}$
BP
Implicit Back Propagation Optimization
$\min_{\Theta}\sum_{v=1}^{V}(\|\mathbf{ZZ}^T-\mathbf{W}_v\|_F^2+\|\mathbf{ZZ}^T-\mathbf{X}_v\mathbf{X}_v^T\|_F^2),$
$where\ \Theta=\{\mathbf{U}^{(k)},\mathbf{S}^{(k)},\Theta_{imp}^{(k)},\theta^{(k)}\}_{k=1}^{b}$

Fig. 1. The architecture of the proposed DBO-Net. DBO-Net is comprised of b differentiable blocks, where $\mathbf{U}$ and $\mathbf{S}$ are the fully connected layers and $h(\cdot)$ is a self-learning function with threshold θ. Adopting original data $\{\mathbf{X}_v\}_{v=1}^{V}$ and Laplacian matrix $\{\mathbf{L}_v\}_{v=1}^{V}$, we learn an interpretable representation $\mathbf{Z}$ with forward propagation of the network blocks, then perform back-propagation to optimize the parameters to make $\mathbf{Z}$ more cluster-friendly.

2. Related works

In this section, we focus on recent developments regarding two related topics, including multi-view clustering and interpretable bi-level optimization network.

2.1. Multi-view clustering

Multi-view clustering can be classified into the following two categories based on different optimization approaches. First, the explicit optimization-based multi-view clustering methods are extended from well-researched theoretical foundations, such as spectral graph theory [25,26], subspace learning [27], and matrix factorization [28,29]. The main principle is to optimize the objective function utilizing convex optimization methods to obtain an iterative form of the variables. For example, Tang et al. [30] introduced a unified graph jointly utilizing the information of view-specific graphs and embedding matrices to obtain the clustering indicator directly. Chen et al. [31] proposed an efficient orthogonal multi-view subspace clustering with a linear time complexity where the anchor selection, graph construction and partition are combined into a unified framework. For the problem of incomplete multi-view data, Lin et al. [32] analyzed it from the view of information theory, combining reconstruction loss, contrast learning across views, and dual prediction loss to achieve both missing view recovery and inter-view consistency enhancement.

The implicit optimization-based multi-view clustering can be regarded as applying various deep network models to exploit high-level information of multi-view data. Based on graph neural network, Fan et al. [33] designed an one2multi graph auto-encoder to learn node embeddings from attributed multi-view graph and reconstruct the graph structure from multiple views. Zhang et al. [6] proposed an encoding network based on an auto-encoder, in which joint internal and external networks learn to automatically balance consistency and complementarity between multi-view information. Li et al. [34] designed deep adversarial multi-view clustering network, which is used to capture data distributions to obtain potential representations shared by all views. Huang et al. [35] used a deep matrix factorization framework for iterative decomposition of multi-view data to learn the hierarchical semantics of multi-view data.

Although existing work on the explicit optimization-based multi-view clustering methods has achieved encouraging results, most of them are non-parametric shallow models that may perform inefficiently when they come to large and complex data distributions. The implicit optimization-based multi-view clustering methods use deep learning to possess reliable data fitting ability but the lack of interpretability becomes the bottleneck of its practical applications. Therefore, this paper aims to propose an efficient interpretable network for multi-view clustering tasks.

2.2. Interpretable bi-level optimization network

Bi-level optimization is a special type of optimization paradigm involving an upper-level task and a lower-level task in which the variables of the former task are constrained to be the optimal solution of the latter task, and the two tasks can be mutually reinforced [36,37]. The majority of the deep models mainly show the model explainability by empirical analyses such as experimental visualization, while there are few efforts towards designing deep models that are inherently interpretable [38]. Recently, researchers have started to investigate the intrinsic relationship between deep networks and classical machine learning models from various perspectives to design an interpretable bi-level optimization network. Huang et al. [39] designed a differentiable surrogate of the Hungarian algorithm which could be pluggable into the neural network to seek the correspondence of partially aligned multi-view data. Peng et al. [40] recast vanilla k-means into a neural network to learn a clustering-favorable representation in an end-to-end manner. Jin et al. [41] applied the bi-level model based on hyperparameter optimization to retinex-induced encoder-decoder architecture for fast adaptation to bridge the gap between low-light scenes. Liu et al. [42] proposed a bi-level model where the deformable registration is an upper-level task while maximizing a posterior for features is a lower-level task. To exploit the structural insights of traditional optimization-based methods, Zhang et al. [43] cast iterative shrinkage-thresholding algorithm (ISTA) into a novel structured deep network for image compressive sensing reconstruction. Motivated by the above observations, we construct an interpretable network by bi-level optimization, which transforms explicit optimization objective into a network model by theoretic principles and then performs implicit optimization.

3. Proposed model

This section discusses the proposed method, aiming to obtain a potentially consistent representation of multi-view data through an interpretable network with bi-level optimization. The specific issues of each optimization process will be specifically stated below. We first introduce some frequently used notations as shown in Table 1.

3.1. Bi-level optimization loss function

The proposed DBO-Net contains an explicit optimization module and an implicit optimization module. The bi-level optimization process first obtains a consistent representation through an explicit optimization function, then constrains the cor-

Table 1
Symbolic normalization notations with their descriptions.

Notations	Descriptions
$\mathbf{X}_v \in \mathbb{R}^{n\times d_v}$	The given multi-view training data.
$\mathbf{Z} \in \mathbb{R}^{n\times c}$	The latent consistent representation to be learned.
$\mathbf{G}_v \in \mathbb{R}^{c\times d_v}$	The indicator encoding matrix of $\mathbf{X}_v$.
$\mathbf{W}_v \in \mathbb{R}^{n\times n}$	The similarity matrix evaluated from $\mathbf{X}_v$.
$\mathbf{D}_v \in \mathbb{R}^{n\times n}$	The diagonal matrix constructed from $\mathbf{W}_v$.
$\mathbf{L}_v \in \mathbb{R}^{n\times n}$	The graph Laplacian matrix constructed from $\mathbf{W}_v$.
$h_\theta(\cdot)$	The self-learning function, with a learnable parameter θ.
$\mathbf{S}, \mathbf{U}$	The learnable layers in the proposed network.
$\mathbf{I}, \boldsymbol{\Theta}$	The identity matrix and the learnable parameter set.

responding variables through an implicit loss function to obtain a consistent representation, and finally uses back-propagation to update the entire bi-level optimization network parameters. The objective function can be represented by the following bi-level optimization

$$\begin{aligned} \mathbf{Z} = \min_{\mathbf{Z}} \frac{1}{2}\sum_{v=1}^{V}\left(\|\mathbf{X}_v - \mathbf{Z}\mathbf{G}_v\|_F^2 + \alpha \mathrm{Tr}\left(\mathbf{Z}^T\mathbf{L}_v\mathbf{Z}\right)\right) + \beta\|\mathbf{Z}\|_1, \\ \text{s.t.} \min_{\boldsymbol{\Theta}} \gamma \sum_{v=1}^{V}\left(\|\mathbf{Z}\mathbf{Z}^T - \mathbf{W}_v\|_F^2 + \|\mathbf{Z}\mathbf{Z}^T - \mathbf{X}_v\mathbf{X}_v^T\|_F^2\right). \end{aligned} \tag{1}$$

where $\boldsymbol{\Theta} = \{\mathbf{U}^{(k)}, \mathbf{S}^{(k)}, \boldsymbol{\Theta}_{Imp}^{(k)}, \theta^{(k)}\}_{k=0}^{b}$ is a learnable parameter set optimized by the proposed loss function, where b is the number of differentiable blocks. The details of the two optimization tasks and the derivation process are elaborated in the following subsections.

3.2. Explicit optimization

Suppose that $\mathbf{X}_v \in \mathbb{R}^{n\times d_v}(v = 1, 2, \cdots, V)$ is the v-th view feature, in which n is the number of samples and d_v is the number of features of the v-th view. $\mathbf{Z} \in \mathbb{R}^{n\times c}$ is a latent consistent representation learned from multi-view features $\{\mathbf{X}_v\}_{v=1}^{V}$, where c is the cluster number. To minimize the loss of the original multi-view features and to ensure $\mathbf{Z}$ to be sparse, the optimization function can be expressed as

$$\min_{\mathbf{Z}} \frac{1}{2}\sum_{v=1}^{V}\|\mathbf{X}_v - \mathbf{Z}\mathbf{G}_v\|_F^2 + \beta\|\mathbf{Z}\|_1. \tag{2}$$

It is expected that if the two samples x_i and x_j have high similarity, then their low-dimensional representations z_i and z_j should be also close to each other. Here, a weight matrix $\mathbf{W} = [\mathbf{W}_{ij}]_{n\times n}$ calculated with Heat kernel [44] is employed to construct the distribution of the samples, defined as $\mathbf{W}_{ij} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{\sigma^2}\right)$. Therefore, the representation of data space information captured under the manifold assumption is to minimize the following objective loss function

$$\begin{aligned} & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|z_i - z_j\|_2^2\mathbf{W}_{ij} \\ = & \sum_{i=1}^{n} z_i z_i^T \mathbf{G}_{ii} - \sum_{i=1}^{n}\sum_{j=1}^{n} z_i z_j^T \mathbf{W}_{ij} \\ = & \mathrm{Tr}\left(\mathbf{Z}^T\mathbf{D}\mathbf{Z}\right) - \mathrm{Tr}\left(\mathbf{Z}^T\mathbf{W}\mathbf{Z}\right) = \mathrm{Tr}\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right), \end{aligned} \tag{3}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix. $\mathbf{W}_v$ is the pairwise similarity matrix of $\mathbf{X}_v$, $\mathbf{D}_v$ is a diagonal matrix with $[\mathbf{D}_v]_{ii} = \sum_{j=1}^{n}\mathbf{W}_{ij}^{(v)}$, and $\mathbf{L}_v = \mathbf{D}_v - \mathbf{W}_v$ denotes the graph Laplacian matrix. In order for $\mathbf{Z}$ to preserve the flexible local manifold structure in the high-dimensional space of different views, we combine the manifold regularization (3) with Eq. (2). Taking $\mathscr{L}_{Exp}$ as the loss of the explicit optimization, the optimization problem can be interpreted as

$$\min_{\mathbf{Z}} \mathscr{L}_{Exp} = \frac{1}{2}\sum_{v=1}^{V}\left(\|\mathbf{X}_v - \mathbf{Z}\mathbf{G}_v\|_F^2 + \alpha \mathrm{Tr}(\mathbf{Z}^T\mathbf{L}_v\mathbf{Z})\right) + \beta\|\mathbf{Z}\|_1. \tag{4}$$

Using the proximal gradient descent method to solve the above optimization function, the iterative process of $\mathbf{Z}$ can be expressed in the iterative form of ISTA [45], we have

$$\mathbf{Z}^{(k)} = h_\theta\left(\mathbf{Z}^{(k-1)} - \rho\nabla\mathcal{L}_{Exp}\left(\mathbf{Z}^{(k-1)}\right)\right), \tag{5}$$

where $h_\theta(\cdot)$ is a self-learning function with a differentiable threshold θ and ρ is a step-size. Considering the consistent representations among all view features, the information of all view features should be learned, so $\nabla\mathcal{L}_{Exp}\left(\mathbf{Z}^{(k-1)}\right)$ can be calculated using the following formula

$$\nabla\mathcal{L}_{Exp}\left(\mathbf{Z}^{(k-1)}\right) = \sum_{v=1}^{V}\left(-\mathbf{X}_v\mathbf{G}_v^T + \mathbf{Z}^{(k-1)}\mathbf{G}_v\mathbf{G}_v^T + \alpha\mathbf{L}_v\mathbf{Z}^{(k-1)}\right). \tag{6}$$

Replacing Eq. (6) into Eq. (5) leads to

$$\begin{aligned}\mathbf{Z}^{(k)} &= h_{\frac{\beta}{L}}\left(\mathbf{Z}^{(k-1)} - \sum_{v=1}^{V}\left(\tfrac{1}{L^{(v)}}\left(-\mathbf{X}_v\mathbf{G}_v^T + \mathbf{Z}^{(k-1)}\mathbf{G}_v\mathbf{G}_v^T + \alpha\mathbf{L}_v\mathbf{Z}^{(k-1)}\right)\right)\right)\\ &= h_{\frac{\beta}{L}}\left(\sum_{v=1}^{V}\left(\mathbf{Z}^{(k-1)} + \tfrac{1}{L^{(v)}}\mathbf{X}_v\mathbf{G}_v^T - \tfrac{1}{L^{(v)}}\mathbf{Z}^{(k-1)}\mathbf{G}_v\mathbf{G}_v^T - \tfrac{\alpha}{L^{(v)}}\mathbf{L}_v\mathbf{Z}^{(k-1)}\right)\right)\\ &= h_{\frac{\beta}{L}}\left(\sum_{v=1}^{V}\left(\mathbf{Z}^{(k-1)}\left(\mathbf{I} - \tfrac{1}{L^{(v)}}\mathbf{G}_v\mathbf{G}_v^T\right) - \tfrac{\alpha}{L^{(v)}}\mathbf{L}_v\mathbf{Z}^{(k-1)} + \tfrac{1}{L^{(v)}}\mathbf{X}_v\mathbf{G}_v^T\right)\right),\end{aligned} \tag{7}$$

where $L^{(v)}$ is the Frobenius norm of $\mathbf{G}_v$. Here, we transform optimization variables and hyper-parameters into learnable variables for implicit optimization

$$\mathbf{S} = \mathbf{I} - \tfrac{1}{L^{(v)}}\mathbf{G}_v\mathbf{G}_v^T, \mathbf{U} = \tfrac{1}{L^{(v)}}\mathbf{G}_v^T, \theta = \tfrac{\beta}{L}. \tag{8}$$

Thus, Eq. (7) is transformed into the following explicitly differentiable block solution, where each iteration of $\mathbf{Z}$ feeds to two linear layers parameterized by $\mathbf{S} \in \mathbb{R}^{c\times c}$ and $\mathbf{U} \in \mathbb{R}^{d_v\times c}$, and a Laplacian multiplier operator, expressed as

$$\mathbf{Z}^{(k)} = h_\theta\left(\tfrac{1}{V}\sum_{v=1}^{V}\left(\mathbf{Z}^{(k-1)}\mathbf{S}^{(k)} - \mathbf{L}_v\mathbf{Z}^{(k-1)} + \mathbf{X}_v\mathbf{U}^{(k)}\right)\right). \tag{9}$$

We omit $\frac{\alpha}{L^{(v)}}$ to be learned implicitly. The first term and third term adopt learnable layers that accept the latent hidden representation of $\mathbf{Z}^{(k)}$ and the original multi-view features $\{\mathbf{X}_v\}_{v=1}^{V}$ for sufficient information exploitation. And the second term is designed to capture the hidden topological information of the sample space from the graph regularization. In particular, both trade-off parameters α and β in the objective function of Eq. (4) are optimally learned via the learning layers of the network. The potential representation is first obtained by the explicit optimization module, then the resulting consistent representation will be used in the implicit optimization process in the next subsection.

3.3. Implicit optimization

After obtaining a consistent representation through the explicit differentiable solving module, a joint optimization framework using the implicit loss function enables the representation more appropriate for the clustering task and implicitly updates the explicit module. To ensure that as much information as possible about the spatial distribution of the samples can be exploited, and to minimize the difference between the original features and the learned representation, the self-supervised loss can be described as

$$\mathcal{L}_{Imp} = \min_{\mathbf{\Theta}_{Imp}}\sum_{v=1}^{V}(\|\mathbf{Z}^{(k)}\mathbf{Z}^{(k)^T} - \mathbf{W}_v\|_F^2 + \|\mathbf{Z}^{(k)}\mathbf{Z}^{(k)^T} - \mathbf{X}_v\mathbf{X}_v^T\|_F^2), \tag{10}$$

where the first term is the reconstruction loss of the similarity matrix $\mathbf{W}_v$, and the second term is the reconstruction loss of the original multi-view features $\mathbf{X}_v$. The implicit optimization loss encourages the learned representation to explore complementary information from multiple views, enabling the potential consistent representation to contain both spatial topology and semantic information of multi-view features.

3.4. Back propagation and complexity analysis

In this subsection, the implicit learning part of the network is analyzed in detail. First, the learning parameter sets for explicit and implicit optimization are separately denoted as $\mathbf{\Theta}_{Exp}$ and $\mathbf{\Theta}_{Imp}$. All parameters can be optimized by back-propagation after the joint loss is obtained, the gradient of each loss item is updated by the following form

$$\begin{cases} \frac{\partial \mathscr{L}_{Imp}(\mathbf{\Theta}_{Imp})}{\partial \mathbf{\Theta}_{Imp}} & = \frac{\partial \mathbf{Z}}{\partial \mathbf{\Theta}_{Imp}} \frac{\partial \mathscr{L}_{Imp}(\mathbf{\Theta}_{Imp})}{\partial \mathbf{Z}} \\ & = \frac{\partial \mathbf{Z}}{\partial \mathbf{\Theta}_{Imp}} (8\mathbf{Z}\mathbf{Z}^T\mathbf{Z} - 4\mathbf{A}_v\mathbf{Z} - 4\mathbf{X}_v\mathbf{X}_v^T\mathbf{Z}), \\ \frac{\partial \mathscr{L}_{Exp}(\mathbf{\Theta}_{Exp})}{\partial \mathbf{\Theta}_{Exp}} & = \frac{\partial \mathbf{Z}}{\partial \mathbf{\Theta}_{Exp}} \frac{\partial \mathscr{L}_{Exp}(\mathbf{\Theta}_{Exp})}{\partial \mathbf{Z}}. \end{cases} \tag{11}$$

When the network weights are trained by the above formula, the back propagation form of the proposed network can be derived as

$$\begin{cases} \mathbf{\Theta}_{Imp}^{t+1} = \mathbf{\Theta}_{Imp}^{t} - \eta \frac{\partial \mathscr{L}_{Imp}(\mathbf{\Theta}_{Imp})}{\partial \mathbf{\Theta}_{Imp}}, \\ \mathbf{\Theta}_{Exp}^{t+1} = \mathbf{\Theta}_{Exp}^{t} - \eta \frac{\partial \mathscr{L}_{Exp}(\mathbf{\Theta}_{Exp})}{\partial \mathbf{\Theta}_{Exp}}, \end{cases} \tag{12}$$

where η is the learning rate. After each iteration to obtain the total loss, the network is back-propagated by Eq. (12). It is noticed that each block is differentiable and reusable, and a deep multi-view clustering network is composed of multiple differentiable blocks. We summarize the internal process of the proposed method in Algorithm 1. As to the computational complexity of the proposed framework, the complexities are $\mathcal{O}(c^2n + cn^2 + cd_vn)$ and $\mathcal{O}(cn^2 + \sum_{v=1}^{V} d_vn^2)$ for the explicit optimization and implicit optimization, respectively. Considering that the number c of clusters is much smaller than the number n of clusters and feature dimension d_v of the v-th view, the overall computational complexity is $\mathcal{O}(cn^2 + \sum_{v=1}^{V} d_vn^2)$ for each training epoch.

Algorithm 1: DBO-Net

```
Input: Multi-view data $\{\mathbf{X}_v\}_{v=1}^V$, number of network block $b$, number of training epoch $t$, trade-off factor $\gamma$, learning rate $\eta$.
Output: Latent consistent representation $\mathbf{Z}$.
Initialize consistent representation $\mathbf{Z}$, graph Laplacian matrices $\{\mathbf{L}_v\}_{v=1}^V$, and learnable parameter set $\mathbf{\Theta}$;
for $i = 1 \to t$ do
    for $k = 1 \to b$ do
        Explicitly update $\mathbf{Z}^{(k)}$ by Eq. (9);
    end for
    Calculate loss value by the implicit loss function (10);
    Update the learnable parameter set $\mathbf{\Theta}^{(k)}$ by Eq. (12);
end for
return Latent consistent representation $\mathbf{Z}^{(k)}$ as the optimal $\mathbf{Z}$.
```

4. Experiment and analysis

In this section, the effectiveness of the proposed model is verified through experimental results, using k-means clustering and seven other advanced multi-view clustering methods as comparisons. All the compared methods are tested on six mainstream multi-view datasets to analyze the results of the evaluation metrics and running time. Furthermore, we conduct an ablation study to analyze the model components, the parameter sensitivity and model convergence to verify the validity and feasibility of the proposed model. The proposed framework is implemented with Pytorch on a standard Ubuntu16.04 operation system with NVIDIA Tesla P100 GPU.

4.1. Experimental setup

In this subsection, the experimental setup is elaborated from the perspectives of test datasets and compared algorithms.

4.1.1. Datasets

In this subsection, six publicly available datasets are used to verify the effectiveness of the proposed method, summarized in Table 2. Single-view datasets can be transformed into multi-view datasets by using different feature extraction methods to acquire colors, textures and other information from the images. The details of these tested datasets are as follows.

ALOI[1]: A collection of substantial images containing objects was taken under various light conditions and rotation angles. Four commonly used features can be available for downstream tasks: 64-D RGB color histograms, 64-D HSV color histograms, 77-D color similarities, and 13-D Haralick features.

[1] https://elki-project.github.io/datasets/multi_view

Table 2
A brief description of the tested datasets.

Datasets	# Samples	# Views	# Features	# Classes
ALOI	1,079	4	64/64/77/13	10
Caltech101	9,144	6	48/40/254/1,984/512/928	102
MNIST	10,000	3	30/9/30	10
MITIndoor	5,360	4	3,600/1,770/1,240/4,096	67
NUS-WIDE	1,600	6	64/144/73/128/225/500	8
Scene15	4,485	3	1,800/1,180/1,240	15

Caltech101[2]: A popular object recognition dataset comes with 102 classes of images. Six extracted features are used in our experiments: 48-D Gabor features, 40-D wavelet moments features, 254-D Centrist features, 1,984-D histogram of oriented gradients features, 512-D GIST features and 928-D LBP features.

MITIndoor[3]: A scene dataset contains 5,360 images with 67 categories. In this dataset, we extract four types of features including 4,096-D PHOW, 3,600-D LBP, 1,770-D CENTRIST, and 1,240-D deep features.

MNIST[4]: This is a well-known dataset of handwritten digits. Three types of features are extracted from all test images: 30-D IsoProjection, 9-D Linear Discriminant Analysis, and 9-D Neighborhood Preserving Embedding features.

NUS-WIDE[5]: A web image dataset for object recognition, we select eight classes and six available representations: 64-D color histogram, 225-D block-wise color moments, 144-D color correlogram, 73-D edge direction histogram, 128-D wavelet texture and 500-D bag of words on SIFT descriptor.

Scene15[6]: An image dataset contains 15 scene categories with both indoor and outdoor environments, 4,485 images in total. 1,800-D LBP, 1,180-D PHOW, and 1,240-D CENTRIST features are utilized in the experiment.

4.1.2. Compared algorithms

In this subsection, several state-of-the-art multi-view clustering methods are compared to validate the effectiveness and efficiency of the proposed method. In all compared methods, the explicit optimization-based multi-view clustering methods consist of k-means [46], GMC [47], LMVSC [10], MCGC [25], and MVKSC [13], and the implicit optimization-based multi-view clustering methods include DBMC-Net [15], DGCCA-Net [48] and MvL-Net [19]. The details of compared multi-view clustering methods are as follows.

DBMC-Net: Differentiable Bi-sparse Multi-view Co-clustering is a differentiable network for multi-view co-clustering, which learned a collaborative representation while maintaining sparsity in the dual space of features and samples.

DGCCA-Net: Deep Generalized Canonical Correlation Analysis learned nonlinear transformations of an arbitrary view to maximize the correlation between the learned representations across views.

GMC: Graph-based Multi-view Clustering used manifold learning for multi-view data while learning a weight and a similarity matrix for each view in a mutually enhancing manner.

LMVSC: Large-scale Multi-view Subspace Clustering with linear order complexity employed an anchor strategy to reduce the size of the reconstruction matrix.

MCGC: Multiview Consensus Graph Clustering regularized graphs from different views leveraged the common consensus information and learned a consensus graph with a rank constraint on the Laplacian matrix.

MVKSC: Based on a weighted KCCA, Multi-View Kernel Spectral Clustering performed clustering that could exploit information from two or more views.

MvL-Net: Multi-view Laplacian Network reformulated the orthogonality constraint as a network layer on the embedding network to learn the consensus representation in a common space.

There are some parameters for the compared methods to be clarified in advance. All methods are tuned as their default settings if feasible. For other open hyper-parameters, we adopt the following settings. For k-means, experiments are performed on single-view data formed by the features of these multi-view data that are concatenated horizontally. For GMC, the value of γ is adjusted to 1. For LMVSC, the regularization parameter α is set to 0.01. For MCGC, the regularization parameter β is tuned as $\beta = 0.1$. For MVKSC, the kernel type uniformly selects the RBF kernel, in which the kernel parameters of t and d are tuned as $[1, 10]$. For DBMC-Net, all regularization parameters are set to 0.001. For DGCCA-Net, the number of hidden layers is set to 128. As to MvL-Net, the total number of pairs for Siamese networks is set to 600,000. The parameter setting of the proposed method is given in Table 3, suggesting that the epoch for larger datasets is set to 150, while for smaller datasets 50 epochs can result in satisfactory performance. In a summary, the experiments in SubSection 4.5 confirm that parameters of DBO-Net are relatively insensitive within the appropriate range. In particular, we use a batch-norm layer followed by a fully-connected layer in the network to form the proposed differentiable block-wise network.

[2] http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[3] http://web.mit.edu/torralba/www/indoor.html
[4] http://yann.lecun.com/exdb/mnist/
[5] https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html
[6] https://figshare.com/articles/dataset/15-Scene_Image_Dataset/7007177

Table 3
Details for parameter setting of the proposed method in experiments.

Datasets \ Parameters	Block	Epoch	Learning rate	γ	θ
ALOI	5	50	10^{-2}	1.0	10^{-2}
Caltech101	5	150	10^{-3}	1.0	10^{-1}
MITIndoor	5	150	10^{-2}	1.0	10^{-1}
MNIST	15	50	10^{-2}	1.0	10^{-1}
NUS-WIDE	5	50	10^{-2}	1.0	10^{-2}
Scene15	9	150	10^{-3}	1.0	10^{-2}

4.2. Experimental results

In this subsection, comprehensive experiments are performed on six real-world multi-view datasets to evaluate different multi-view clustering methods. Four well-known clustering evaluation metrics are applied to our experiments, including clustering accuracy (ACC), normalized mutual information (NMI), adjusted rand index (ARI), and F-score. ACC is defined as the number of correctly clustered samples divided by the number of samples. NMI is used to measure the nearness of the clustering labels to the ground truths. ARI reflects the extent of overlap between the clustering labels and the ground truths. F-score is a composite measure of Precision and Recall. The values of these metrics fall within the range of $[0, 1]$, where higher values indicate better performance. All experiments are run ten times, and we record the means and standard deviations as the final results. The parameter setting in experiments is shown in Table 3. Note that an error has occurred to MVKSC running on the Caltech101 and MNIST datasets due to insufficient memory.

Table 4 shows the clustering results for all methods, where we conclude the following observations. Overall, it is often more satisfactory to employ deep models to learn latent representations than shallow models that consider other strategies. A possible reason is that the deep model has strong data fitting capability, which can better perform in exploring nonlinear and complex relationships existing in multi-view features. Specifically, the proposed framework achieves competitive and steady performance on most datasets, outperforming other algorithms on all datasets in terms of ACC and F-score, and gaining significant results on the ALOI, MITIndoor, and Scene15 datasets.

Table 4
Clustering metrics of all compared multi-view clustering methods, where the best and runner-up performance are highlighted in bold and underlined respectively (mean% and standard deviation%).

Datasets \ Methods		*k*-means	GMC	LMVSC	MCGC	MVKSC	DBMC-Net	DGCCA-Net	MvL-Net	DBO-Net
ALOI	ACC	47.5 (3.3)	64.9 (0.0)	69.5 (0.0)	52.4 (0.0)	59.4 (4.3)	68.8 (3.4)	57.3 (1.2)	56.0 (2.5)	**80.1 (3.4)**
	NMI	47.3 (2.1)	61.8 (0.0)	72.5 (0.0)	52.5 (0.0)	70.1 (1.8)	71.7 (1.4)	48.4 (0.4)	60.2 (0.3)	**75.0 (2.6)**
	ARI	33.0 (2.9)	32.9 (0.0)	59.2 (0.0)	25.9 (0.0)	53.2 (3.5)	56.3 (3.9)	34.7 (1.0)	40.9 (0.3)	**64.5 (4.2)**
	F-score	41.1 (2.4)	42.1 (0.0)	63.7 (0.0)	37.0 (0.0)	58.5 (2.9)	65.2 (2.2)	46.0 (0.8)	57.5 (0.2)	**70.9 (3.3)**
Caltech101	ACC	13.4 (0.5)	19.5 (0.0)	22.1 (0.0)	23.6 (0.0)	–	35.0 (0.4)	36.1 (0.2)	20.0 (0.0)	**44.4 (0.3)**
	NMI	30.3 (0.2)	23.8 (0.0)	43.1 (0.0)	26.3 (0.0)	–	32.5 (0.6)	36.0 (0.3)	**47.2 (0.0)**	47.1 (0.5)
	ARI	8.00 (0.4)	0.42 (0.0)	16.5 (0.0)	0.40 (0.0)	–	33.8 (0.6)	37.7 (1.2)	13.6 (0.0)	**37.8 (3.2)**
	F-score	9.90 (0.3)	2.61 (0.0)	17.9 (0.0)	5.70 (0.0)	–	26.0 (1.1)	26.7 (0.2)	19.0 (0.0)	**39.4 (0.6)**
MITIndoor	ACC	7.30 (0.7)	10.7 (0.0)	15.7 (0.0)	23.6 (0.4)	5.00 (0.0)	21.5 (0.4)	20.4 (0.3)	7.40 (0.1)	**32.5 (0.9)**
	NMI	12.6 (1.3)	11.5 (0.0)	27.0 (0.0)	33.7 (0.5)	6.40 (0.0)	32.6 (0.6)	30.5 (0.3)	16.5 (0.0)	**42.7 (0.5)**
	ARI	1.60 (0.2)	0.20 (0.0)	5.30 (0.0)	9.20 (0.4)	1.40 (0.0)	10.5 (0.1)	7.60 (0.3)	0.90 (0.1)	**16.2 (0.3)**
	F-score	4.20 (0.2)	3.10 (0.0)	6.96 (0.0)	10.7 (0.4)	4.20 (0.0)	14.2 (0.5)	11.0 (0.2)	3.90 (0.0)	**21.3 (0.5)**
MNIST	ACC	65.2 (0.0)	90.3 (0.0)	51.8 (0.0)	61.0 (0.0)	–	79.5 (0.2)	29.5 (0.1)	73.3 (6.2)	**91.6 (0.0)**
	NMI	66.7 (0.0)	**84.3 (0.0)**	49.0 (0.0)	61.2 (0.0)	–	65.3 (0.6)	34.6 (1.6)	62.8 (3.0)	83.3 (0.1)
	ARI	56.8 (0.0)	82.4 (0.0)	34.9 (0.0)	43.5 (0.0)	–	50.5 (0.0)	17.9 (0.8)	56.0 (4.9)	**82.7 (0.1)**
	F-score	60.0 (0.0)	81.5 (0.0)	41.6 (0.0)	40.0 (0.0)	–	64.5 (0.1)	31.3 (0.9)	62.3 (3.9)	**84.7 (0.1)**
NUS-WIDE	ACC	32.0 (0.7)	20.1 (0.0)	12.1 (0.0)	25.7 (0.0)	31.1 (0.0)	35.5 (0.7)	24.5 (0.3)	30.2 (0.1)	**38.0 (0.5)**
	NMI	17.5 (0.7)	12.2 (0.0)	8.60 (0.0)	14.7 (0.0)	21.1 (0.6)	18.5 (1.2)	10.0 (0.3)	20.1 (0.2)	**24.0 (0.6)**
	ARI	9.00 (0.7)	4.20 (0.0)	2.40 (0.0)	5.70 (0.0)	11.4 (0.7)	12.9 (0.4)	7.20 (1.6)	9.50 (0.1)	**16.3 (0.2)**
	F-score	24.3 (0.6)	24.7 (0.0)	7.70 (0.0)	25.0 (0.0)	24.9 (0.5)	28.4 (1.5)	23.0 (0.4)	26.4 (0.1)	**32.5 (0.6)**
Scene15	ACC	30.8 (2.0)	38.1 (0.0)	40.6 (0.0)	42.3 (0.0)	22.1 (0.0)	45.6 (0.4)	15.8 (0.6)	35.4 (1.6)	**63.5 (1.3)**
	NMI	28.5 (1.7)	41.6 (0.0)	40.3 (0.0)	41.9 (0.0)	18.4 (0.0)	44.6 (3.3)	6.30 (0.8)	36.0 (0.2)	**59.3 (0.8)**
	ARI	14.8 (1.1)	19.1 (0.0)	24.8 (0.0)	24.7 (0.0)	10.9 (0.0)	23.6 (1.2)	2.90 (0.2)	19.9 (0.5)	**45.1 (1.2)**
	F-score	21.9 (0.9)	28.1 (0.0)	29.3 (0.0)	30.4 (0.0)	20.1 (0.0)	30.2 (2.2)	13.2 (0.7)	26.0 (0.6)	**51.8 (1.1)**

Among the shallow models, the proposed network outperforms all compared methods on all datasets, including GMC and MCGC based on spectral graph theory and LMVSC on subspace learning, as well as MVKSC for multiple kernel-based clustering. Moreover, the proposed network outperforms the three compared deep models, which employ different deep network frameworks to learn underlying feature structures between views. As expected, the proposed method significantly outperforms the baseline methods with a considerable improvement, thus demonstrating that the method on bi-level optimization network effectively improves the data fitting ability and clustering performance.

4.3. Running time

To compare the computational complexity of the abovementioned methods, we record the running time of deep multi-view clustering on all test datasets. It can be observed from Table 5 that the proposed method is more efficient than others, especially far more advanced than DGCCA-Net. Most deep methods design complex structures with high-dimensional parameters to improve the representation power of the model, driving up the computational effort. And DBO-Net is derived from traditional methods without constructing complex network structures to obtain the powerful data fitting capability of neural networks. As demonstrated, DBO-Net consistently achieves superior performance to the state-of-the-art methods while spending much less time running.

4.4. Ablation study

In this subsection, we adopt the ablation study on the proposed model by disabling each component separately to reveal the contribution of each model component to performance improvement. There are three components in the block network, we use $\mathscr{A}$ to denote the learnable layer $\mathbf{S}$, $\mathscr{B}$ for the Laplacian multiplier operator $\{\mathbf{L}_v\}_{v=1}^{V}$ and $\mathscr{C}$ for the learnable layer $\mathbf{U}$. The detailed results of the ablation studies are shown in Table 6, with the same parameter settings in Table 3. Overall, the proposed model works best, and its performance drops slightly as each component is disabled, confirming the effectiveness of the proposed model components. The proposed method on the MITIndoor dataset is most sensitive to whether the regularization terms of the graph are integrated into the proposed network structure. Moreover, on most of the datasets, the performance is the greatest improvement when the information of the original data $\{\mathbf{X}_v\}_{v=1}^{V}$ is considered.

4.5. Parameter sensitivity

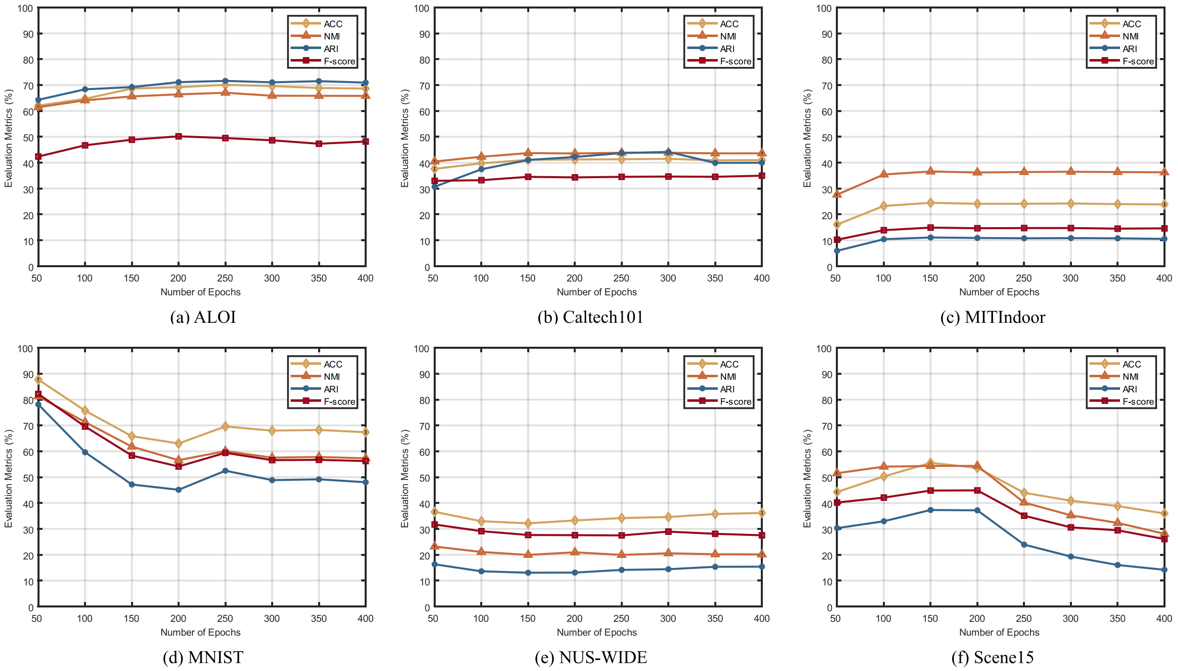
In this section, we conduct sensitivity analysis on five adjustable parameters of DBO-Net to examine the impact on the model performance. We use the grid search to find suitable parameter configurations and four clustering metrics to evaluate the performance of the proposed model to ensure the validity of the parameter set. We test the performance by altering the parameters and fixing other parameters in Table 3. The number of epochs is searched by the grid $\{50, 100, \cdots, 400\}$. The number of blocks is selected in the grid $\{1, 2, \cdots, 15\}$. Figs. 2,3 show the influence of the block number and epoch number on the tested datasets. Generally, it can be observed that the indicator achieves a desirable effect when the number of epochs in $[100, 200]$ generally stays at a high score. If the number of epochs is smaller or larger than this range, the performance of models will not be stable enough and the results are manifested to be unsatisfactory. Similarly, the performance improves on

Table 5
Running time (seconds) of all compared deep multi-view clustering methods, where the lowest time is highlighted in bold.

Datasets \ Time	DBMC-Net	DGCCA-Net	MvL-Net	DBO-Net
ALOI	24.5	226.4	132.4	**19.2**
Caltech101	3540.2	15529.6	4322.5	**2927.2**
MITIndoor	1533.4	7422.4	2344.2	**1158.3**
MNIST	2543.2	10054.0	3733.7	**1938.1**
NUS-WIDE	50.2	5929.2	420.3	**35.8**
Scene15	390.4	4844.3	1060.5	**382.9**

Table 6
Ablation study of the proposed method on the tested datasets (mean% and standard deviation%).

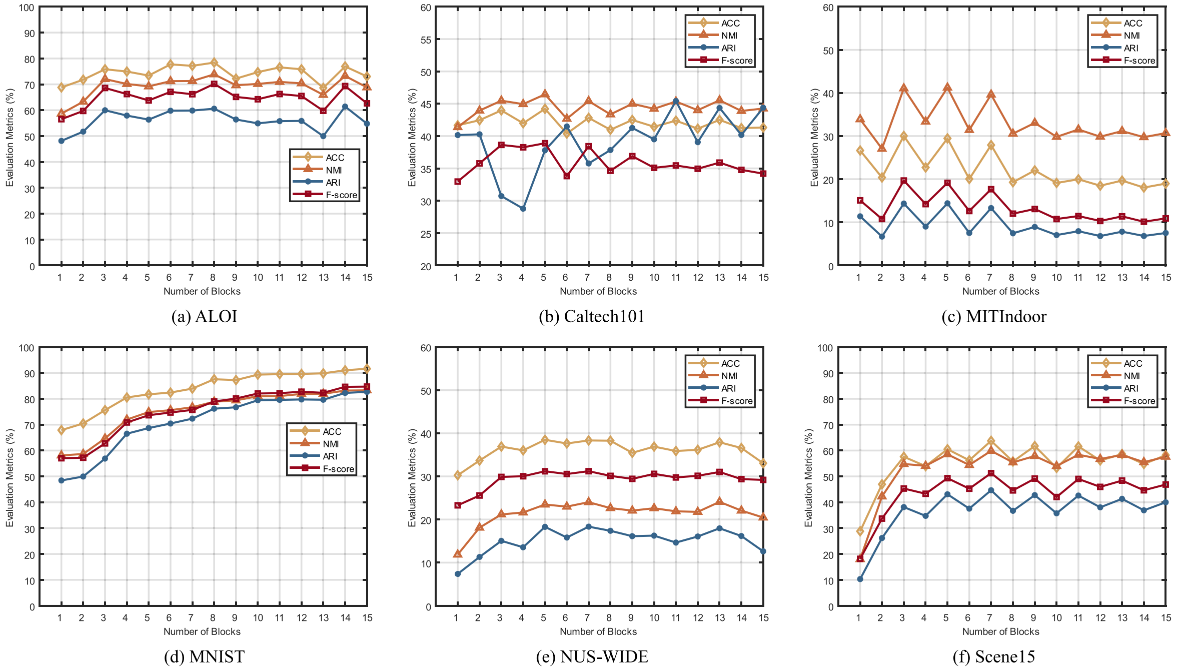
$\mathscr{A}$	$\mathscr{B}$	$\mathscr{C}$	ALOI	Caltech101	MITIndoor	MNIST	NUS-WIDE	Scene15
	✓	✓	77.6 (2.3)	44.0 (0.2)	31.4 (0.9)	90.1 (0.0)	37.2 (0.5)	62.6 (1.1)
✓		✓	74.0 (1.1)	44.3 (0.2)	24.8 (0.5)	91.0 (0.0)	36.7 (0.8)	53.6 (0.7)
✓	✓		60.0 (4.7)	41.1 (0.4)	30.3 (1.0)	53.0 (2.0)	31.5 (0.5)	36.3 (0.9)
✓	✓	✓	80.1 (3.4)	44.4 (0.3)	32.5 (0.9)	91.6 (0.0)	38.0 (0.5)	63.5 (1.3)



(a) ALOI (b) Caltech101 (c) MITIndoor

(d) MNIST (e) NUS-WIDE (f) Scene15

Fig. 2. Parameter sensitivity analysis of DBO-Net in terms of epoch numbers on six datasets.



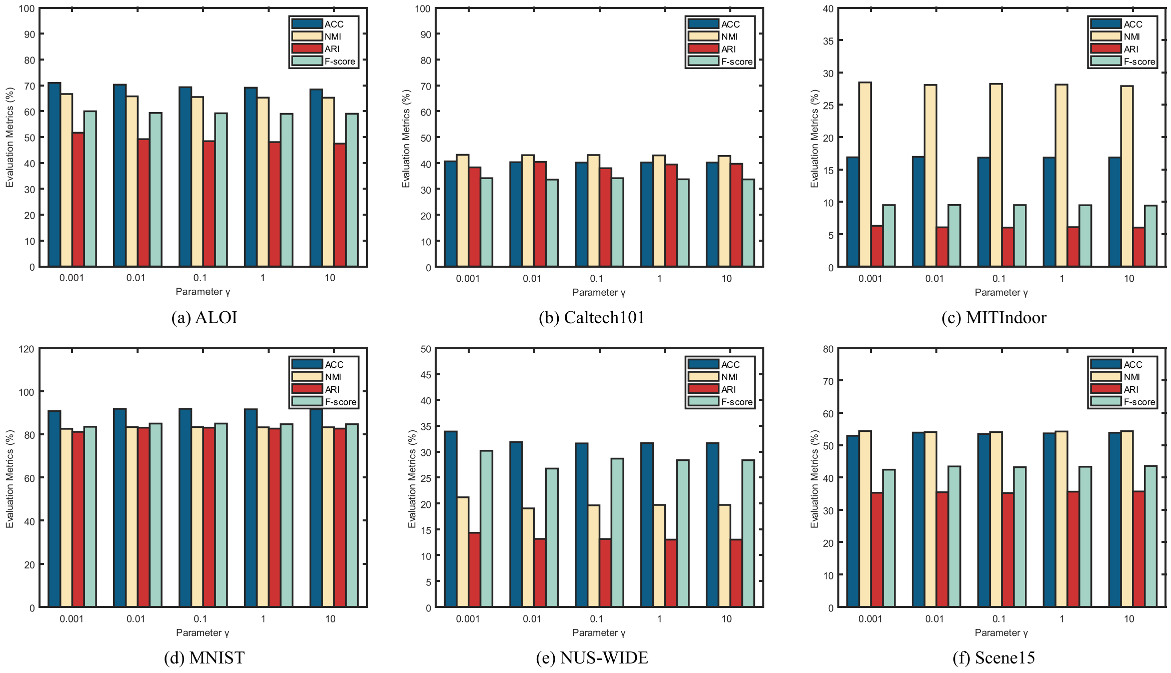
(a) ALOI (b) Caltech101 (c) MITIndoor

(d) MNIST (e) NUS-WIDE (f) Scene15

Fig. 3. Parameter sensitivity analysis of DBO-Net in terms of block numbers on six datasets.

most datasets as the number of blocks grows from one, when the number of blocks falls within a suitable range, it maintains good robustness. Although the performance scores fluctuate somewhat, the overall impact is insignificant and this range varies slightly for different datasets.

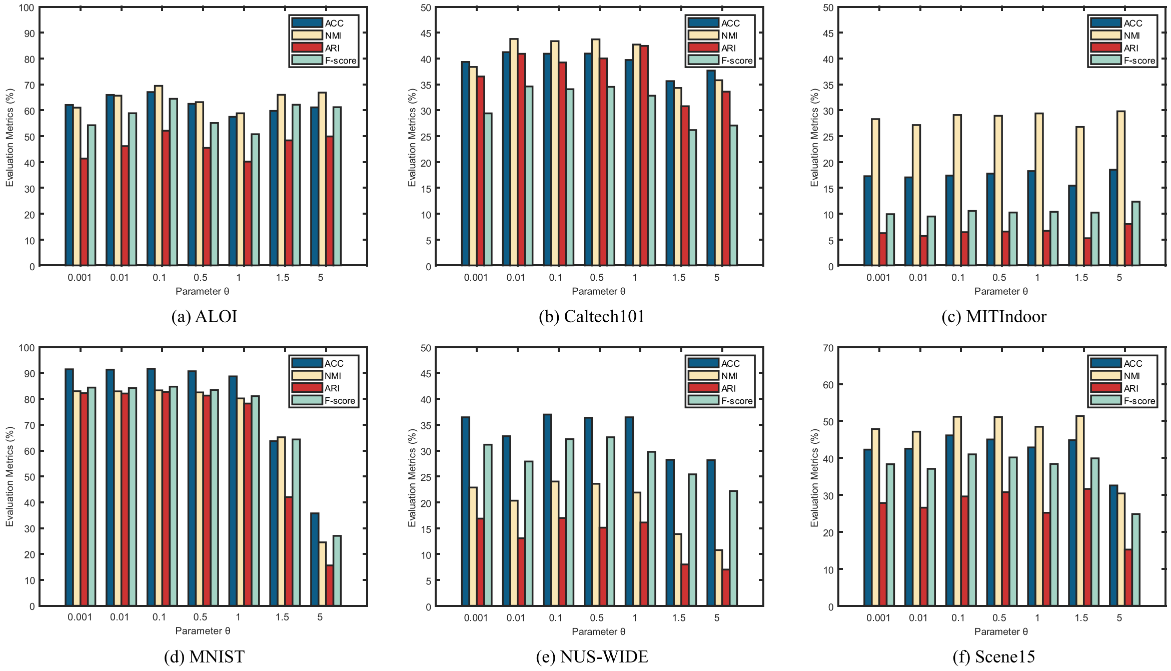
Fig. 4 explores the effect of the parameter γ on the model performance metrics. We evaluate γ within the set $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$. It can be seen that the performance evaluation is relatively stable, so we can conclude that the parameter γ has a minor impact on the model results, and the model is not highly sensitive to the hyper-parameter



(a) ALOI (b) Caltech101 (c) MITIndoor

(d) MNIST (e) NUS-WIDE (f) Scene15

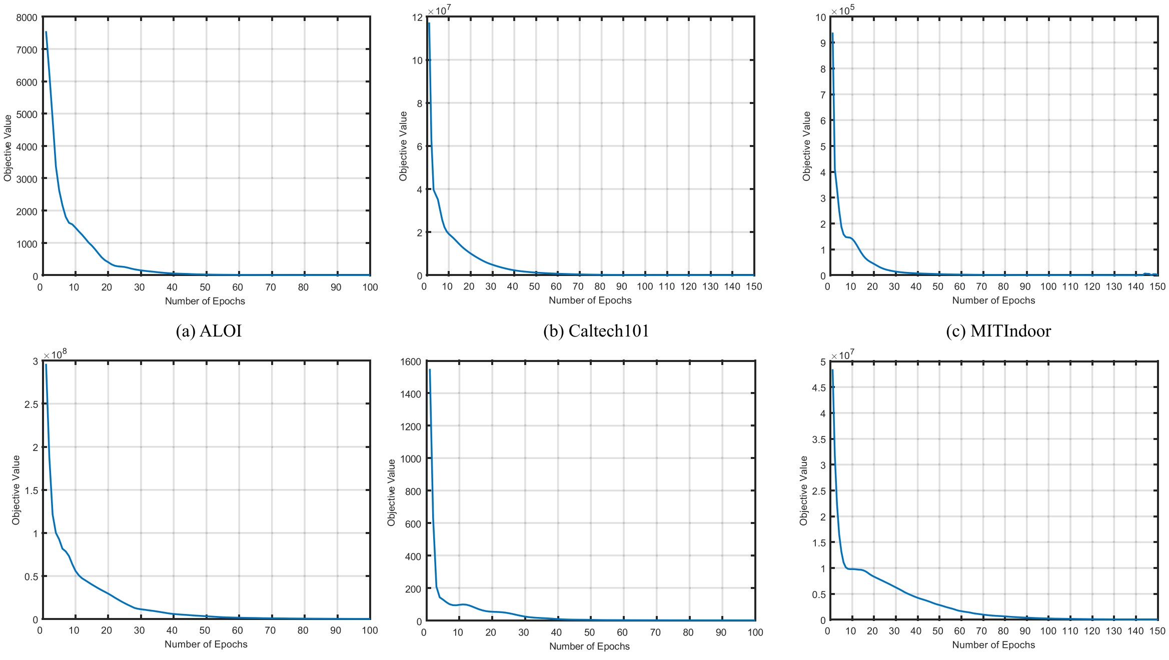
Fig. 4. Parameter sensitivity analysis of DBO-Net with respect to γ on six datasets.



(a) ALOI (b) Caltech101 (c) MITIndoor

(d) MNIST (e) NUS-WIDE (f) Scene15

Fig. 5. Parameter sensitivity analysis of DBO-Net with respect to θ on six datasets.



(a) ALOI

(b) Caltech101

(c) MITIndoor

(d) MNIST

(e) NUS-WIDE

(f) Scene15

Fig. 6. Convergence curves of DBO-Net on six tested datasets.

γ. For the initial value of threshold analysis, θ is selected from $\{10^{-3}, \cdots, 10^{1}, 10^{2}\}$. As shown in Fig. 5, it can be noticed that the performance degrades slightly when the threshold is greater than one, while a range of values from 10^{-3} to 10^{1} always gives rise to desirable results.

4.6. Convergence analysis

As depicted in Fig. 6, we observe that the curve of loss function for the proposed method decreases from steep to gentle with increasing epochs on different datasets. Specifically, the loss value drops sharply at the beginning for a certain number of epochs, while after a sufficient number of epochs, the loss value will reach a stable value with slight bumps. Except for the ALOI dataset, the loss function of the model on other datasets largely converges after less than 100 rounds of training, indicating that the loss function of the model is convergent and ensures effective learning of the network.

5. Conclusion

In this paper, we built an interpretable optimization network for multi-view clustering, termed DBO-Net. Concretely, we first derived the iterative form of the variables based on optimization theory, then reformulated them as learnable network layers to obtain a block structure, and stacked them to eventually form a deep network. The proposed DBO-Net was composed of two modules, including an explicit module and an implicit module. The former explicit optimization module learned a consistent latent representation of multi-view data on sparse coding and introduced manifold regularization to preserve the local structure of the data. The implicit optimization module learned both the spatial topology of the sample space and the semantic information of the features to make the representation more cluster-friendly. Finally, the conducted experiments verified the effectiveness and feasibility of DBO-Net on the clustering task. In future work, we will continue to explore this strategy, combining traditional optimization methods with neural networks to design an interpretable deep network and extend it to other learning tasks.

CRediT authorship contribution statement

Zihan Fang: Software, Visualization, Validation, Writing - original draft. **Shide Du:** Conceptualization, Methodology, Validation, Supervision. **Xincan Lin:** Investigation, Validation. **Jinbin Yang:** Investigation, Validation. **Shiping Wang:** Conceptualization, Methodology, Writing - review & editing. **Yiqing Shi:** Investigation, Validation.

Data availability

No data was used for the research described in the article.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Shiping Wang reports financial support was provided by National Natural Science Foundation of China.

Acknowledgments

This work is in part supported by the National Natural Science Foundation of China under Grants U21A20472 and 62276065, and the National Key Research and Development Plan of China under Grant 2021YFB3600503.

References

[1] C. Tang, X. Zheng, X. Liu, W. Zhang, J. Zhang, J. Xiong, L. Wang, Cross-view locality preserved diversity and consensus learning for multi-view unsupervised feature selection, IEEE Trans. Knowl. Data Eng. 34 (10) (2022) 4705–4716.

[2] C. Tang, X. Zheng, W. Zhang, X. Liu, X. Zhu, E. Zhu, Unsupervised feature selection via multiple graph fusion and feature weight learning, Sci. China Inform. Sci. (2022), https://doi.org/10.1007/s11432-022-3579-1.

[3] X. Ji, J.F. Henriques, A. Vedaldi, Invariant information clustering for unsupervised image classification and segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9865–9874.

[4] W. Tang, B. Hui, L. Tian, G. Luo, Z. He, Z. Cai, Learning disentangled user representation with multi-view information fusion on social networks, Inform. Fusion 74 (2021) 77–86.

[5] P. Yang, W. Gao, Multi-view discriminant transfer learning, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2013, pp. 1848–1854.

[6] C. Zhang, Y. Liu, H. Fu, AE^2-nets: Autoencoder in autoencoder networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2577–2585.

[7] Y. Feng, Z. Zhang, X. Zhao, R. Ji, Y. Gao, GVCNN: Group-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 264–272.

[8] Y. Sun, S. Wang, T.-Y. Hsieh, X. Tang, V. Honavar, MEGAN: A generative adversarial network for multi-view network embedding, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019, pp. 3527–3533.

[9] B.-Y. Liu, L. Huang, C.-D. Wang, J.-H. Lai, P.S. Yu, Multi-view consensus proximity learning for clustering, IEEE Trans. Knowl. Data Eng. 34 (7) (2022) 3405–3417.

[10] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, Z. Xu, Large-scale multi-view subspace clustering in linear time, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 4412–4419.

[11] J. Lv, Z. Kang, B. Wang, L. Ji, Z. Xu, Multi-view subspace clustering via partition fusion, Inf. Sci. 560 (2021) 410–423.

[12] Y. Chen, X. Xiao, Y. Zhou, Jointly learning kernel representation tensor and affinity matrix for multi-view clustering, IEEE Trans. Multimedia 18 (10) (2019) 2115–2126.

[13] L. Houthuys, R. Langone, J.A.K. Suykens, Multi-view kernel spectral clustering, Inform. Fusion 44 (2018) 46–56.

[14] S. Hu, X. Yan, Y. Ye, Dynamic auto-weighted multi-view co-clustering, Pattern Recogn. 99 (2020) 107101.

[15] S. Du, Z. Liu, Z. Chen, W. Yang, S. Wang, Differentiable bi-sparse multi-view co-clustering, IEEE Trans. Signal Process. 69 (2021) 4623–4636.

[16] X.-L. Li, M.-S. Chen, C.-D. Wang, J.-H. Lai, Refining graph structure for incomplete multi-view clustering, IEEE Trans. Neural Networks Learn. Syst. (2022), https://doi.org/10.1109/TNNLS.2022.3189763.

[17] S. Wang, S. Xiao, W. Zhu, Y. Guo, Multi-view fuzzy clustering of deep random walk and sparse low-rank embedding, Inf. Sci. 586 (2022) 224–238.

[18] P. Jing, Y. Su, Z. Li, L. Nie, Learning robust affinity graph representation for multi-view clustering, Inf. Sci. 544 (2021) 155–167.

[19] Z. Huang, J.T. Zhou, H. Zhu, C. Zhang, J. Lv, X. Peng, Deep spectral representation learning from multi-view data, IEEE Trans. Image Process. 30 (2021) 5352–5362.

[20] S. Wang, Z. Chen, S. Du, Z. Lin, Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (9) (2021) 5042–5055.

[21] Z. Chen, P. Lin, Z. Chen, D. Ye, S. Wang, Diversity embedding deep matrix factorization for multi-view clustering, Inf. Sci. 610 (2022) 114–125.

[22] T. Okuno, A. Takeda, A. Kawana, M. Watanabe, On lp-hyperparameter learning via bilevel nonsmooth optimization, J. Mach. Learn. Res. 22 (245) (2021) 1–47.

[23] A. Shaban, C.-A. Cheng, N. Hatch, B. Boots, Truncated back-propagation for bilevel optimization, in: Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, 2019, pp. 1723–1732.

[24] Y. Li, L. Song, X. Wu, R. He, T. Tan, Learning a bi-level adversarial network with global and local perception for makeup-invariant face verification, Pattern Recogn. 90 (2019) 99–108.

[25] K. Zhan, F. Nie, J. Wang, Y. Yang, Multiview consensus graph clustering, IEEE Trans. Image Process. 28 (3) (2019) 1261–1270.

[26] J. Yao, R. Lin, Z. Lin, S. Wang, Multi-view clustering with graph regularized optimal transport, Inf. Sci. 612 (2022) 563–575.

[27] C. Tang, X. Zhu, X. Liu, M. Li, P. Wang, C. Zhang, L. Wang, Learning a joint affinity graph for multiview subspace clustering, IEEE Trans. Multimedia 21 (7) (2018) 1724–1736.

[28] H. Zhao, Z. Ding, Y. Fu, Multi-view clustering via deep matrix factorization, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 2921–2927.

[29] G.A. Khan, J. Hu, T. Li, B. Diallo, H. Wang, Multi-view data clustering via non-negative matrix factorization with manifold regularization, Int. J. Mach. Learn. Cybern. 13 (3) (2022) 677–689.

[30] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, E. Zhu, Unified one-step multi-view spectral clustering, IEEE Trans. Knowl. Data Eng. (2022), https://doi.org/10.1109/TKDE.2022.3172687.

[31] M.-S. Chen, C.-D. Wang, D. Huang, J.-H. Lai, P.S. Yu, Efficient orthogonal multi-view subspace clustering, in: Proceedings of the Twenty-Eighth ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 127–135.

[32] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, X. Peng, Completer: incomplete multi-view clustering via contrastive prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 11174–11183.

[33] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, B. Wang, One2multi graph autoencoder for multi-view graph clustering, in: Proceedings of the International World Wide Web Conference, 2020, pp. 3070–3076.

[34] Z. Li, Q. Wang, Z. Tao, Q. Gao, Z. Yang, others, Deep adversarial multi-view clustering network, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019, pp. 2952–2958.

[35] S. Huang, Z. Kang, Z. Xu, Auto-weighted multi-view clustering via deep matrix decomposition, Pattern Recogn. 97 (2020) 107015.

[36] R. Liu, J. Liu, Z. Jiang, X. Fan, Z. Luo, A bilevel integrated model with data-driven layer ensemble for multi-modality image fusion, IEEE Trans. Image Process. 30 (2020) 1261–1274.

[37] X. Xie, Q. Wang, Z. Ling, X. Li, G. Liu, Z. Lin, Optimization induced equilibrium networks: An explicit optimization perspective for understanding equilibrium models, IEEE Trans. Pattern Anal. Mach. Intell. (2022), https://doi.org/10.1109/TPAMI.2022.3181425.

[38] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Mach. Intell. 1 (5) (2019) 206–215.

[39] Z. Huang, P. Hu, J.T. Zhou, J. Lv, X. Peng, Partially view-aligned clustering, Adv. Neural Inform. Process. Syst. 33 (2020) 2892–2902.

[40] X. Peng, Y. Li, I.W. Tsang, H. Zhu, J. Lv, J.T. Zhou, XAI beyond classification: nterpretable neural clustering, J. Mach. Learn. Res. 23 (6) (2022) 1–28.

[41] D. Jin, L. Ma, R. Liu, X. Fan, Bridging the gap between low-light scenes: Bilevel learning for fast adaptation, in: Proceedings of the ACM International Conference on Multimedia, 2021, pp. 2401–2409.

[42] R. Liu, Z. Li, Y. Zhang, X. Fan, Z. Luo, Bi-level probabilistic feature learning for deformable image registration, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 723–730.

[43] J. Zhang, B. Ghanem, ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1828–1837.

[44] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Adv. Neural Inform. Process. Syst. 14 (2001) 585–591.

[45] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202.

[46] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[47] H. Wang, Y. Yang, B. Liu, GMC: Graph-based multi-view clustering, IEEE Trans. Knowl. Data Eng. 32 (6) (2019) 1116–1129.

[48] A. Benton, H. Khayrallah, B. Gujral, D.A. Reisinger, S. Zhang, R. Arora, Deep generalized canonical correlation analysis, in: Proceedings of the Workshop on Representation Learning for NLP, 2019, pp. 1–6.