A Differentiable Perspective for Multi-View Spectral Clustering With Flexible Extension

Zhoumin Lu, Feiping Nie, *Senior Member, IEEE*, Rong Wang, and Xuelong Li, *Fellow, IEEE*

Abstract—Multi-view clustering aims to discover common patterns from multi-source data, whose generality is remarkable. Compared with traditional methods, deep learning methods are data-driven and have a larger search space for solutions, which may find a better solution to the problem. In addition, more considerations can be introduced by loss functions, so deep models are highly reusable. However, compared with deep learning methods, traditional methods have better interpretability, whose optimization is relatively stable. In this paper, we propose a multi-view spectral clustering model, combining the advantages of traditional methods and deep learning methods. Specifically, we start with the objective function of traditional spectral clustering, perform multi-view extension, and then obtain the traditional optimization process. By partially parameterizing this process, we further design corresponding differentiable modules, and finally construct a complete network structure. The model is interpretable and extensible to a certain extent. Experiments show that the model performs better than other multi-view clustering algorithms, and its semi-supervised classification extension also has excellent performance compared to other algorithms. Further experiments also show the stability and fewer iterations of the model training.

Index Terms—Multi-view learning, spectral clustering, semi-supervised classification, flexible extension, differentiable programming

1 Introduction

As a fundamental issue in machine learning, clustering continues to receive attention because of its wide-ranging applications in visual, medical and other fields. In the case of unknown categories, it aims to divide similar samples into the same cluster and dissimilar ones into different clusters. This behavior is clearly consistent with reality. If a small number of intra-cluster and/or inter-cluster relationships are given, unsupervised clustering can be extended to semi-supervised clustering for more accurate cluster partitioning. If a small number of labels are given, semi-supervised clustering can be extended to semi-supervised classification, and the class to which each cluster belongs can be estimated. The community's current focus mainly lies in high-dimensional nonlinear data and multi-view learning.

To efficiently handle high-dimensional nonlinear data, two types of methods have been proposed: traditional and deep learning methods. Traditional ones aim to construct a graph through features, and then use the graph structure to find cluster divisions, such as spectral clustering [1], subspace clustering [2], [3], [4] and kernel clustering [5]. Deep learning ones aim at non-linear dimensionality reduction of features and then clustering with new representations such as DEC [6], DSC [7], CC [8] and GCC [9].

Multi-view learning focuses on discovering a common pattern from different perspectives, which helps improve performance. For a video, its text, audio and images can be regarded as three perspectives. For an image, its color features and texture features can be viewed as multiple perspectives. For a document, its textual descriptions from multiple sources can be seen as several perspectives. Even various dimensionality reduction ways and numerous graph construction means can be treated as different perspectives. Multi-view clustering is not only an important branch of multi-view learning, but also a significant extension of single-view clustering.

In recent years, deep clustering has been greatly favored. Because of its large parameter space, a good low-dimensional representation can be searched. Since it utilizes loss functions to guide learning, deep models are remarkably reusable. Even so, traditional clustering still has great research value and is favored by many fields. Since traditional clustering has significant interpretability and theoretical support, its output reliability is guaranteed. Because its optimization direction keeps relatively fixed, it remains relatively stable during training and is not prone to unacceptable situations.

In this paper, considering the characteristics of traditional and deep learning methods, we propose a multi-view spectral clustering model with flexible extensions. Using traditional optimization as the basis, partial parameterization as the means, network structure as the subject, and loss functions as the guidance, the model becomes an improved

- *Zhoumin Lu and Feiping Nie are with the School of Computer Science, School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Key Laboratory of Intelligent Interaction and Applications, Ministry of Industry and Information Technology, Northwestern Polytechnical University, Xi'an 710072, China. E-mail: {walker.zhoumin.lu, feipingnie}@gmail.com.*
- *Rong Wang and Xuelong Li are with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Key Laboratory of Intelligent Interaction and Applications, Ministry of Industry and Information Technology, Northwestern Polytechnical University, Xi'an 710072, China. E-mail: wangrong07@tsinghua.org.cn, li@nwpu.edu.cn.*

Manuscript received 12 June 2022; revised 30 October 2022; accepted 16 November 2022. Date of publication 28 November 2022; date of current version 5 May 2023.
This work was supported in part by the National Natural Science Foundation of China under Grants 62236001, 62276212, and 62176212.
(Corresponding authors: Feiping Nie and Xuelong Li.)
Recommended for acceptance by C. Zhang.
This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2022.3224978, provided by the authors.
Digital Object Identifier no. 10.1109/TPAMI.2022.3224978

version of the vanilla spectral clustering with the advantages of neural networks. The contributions of this work are summarized as follows:

- From the perspective of multi-view learning, we propose a new multi-view clustering model. It can also be applied to multi-view semi-supervised classification problems with only one additional loss function. Theoretical and experimental analyses demonstrate that our model has many good properties: fewer iterations, more stable training, better performance, lower label dependence, etc. Furthermore, the model can be more easily used on different datasets due to the few and insensitive hyperparameters.
- From the perspective of model reuse, by simply modifying the activation function and/or loss function, our model can also solve problems other than multi-view clustering, such as semi-supervised classification, non-negative matrix factorization, principal component analysis, deep clustering, etc. Due to its modular nature, it can be easily embedded in other applications. In addition, the modularity allows modification of the activation function without traversing the entire model.
- From the perspective of explainable model, we directly construct a transparent model instead of explaining a black box. The model builds on a traditional optimization process and is therefore naturally interpretable. In addition, we also give a multi-faceted theoretical analysis on the partial parameterization, making each step well-founded. The ablation studies further demonstrate the rationality and effectiveness of our model.

2 Related Work

The related work is elaborated in six parts: spectral clustering, multi-kernel clustering, multi-view clustering, multi-view classification, model explainability and interpretable model.

2.1 Spectral Clustering

Spectral clustering is well known for good clustering power, such as ratio cut [10], normalized cut [11], and min-max cut [12]. In addition, learning a good graph is of great benefit to spectral clustering, resulting in many algorithms. SSC [3] constructs similarity via self-expressiveness property. CAN [13] and CLR [14] learn a structured graph through rank constraints, while SGL [15] learns by spectral constraints. FGNSC [16] explores the good neighbors of each sample.

2.2 Multi-Kernel Clustering

Through reasonable relaxation and transformation, the objective function of kernel k-means can be simplified into $\max_{\mathbf{F}^T\mathbf{F}=\mathbf{I}}\mathrm{Tr}(\mathbf{K}\mathbf{F}\mathbf{F}^T)$, whose form is consistent with spectral clustering. Even $\mathbf{K}$ can be viewed as a weighted self-loop graph. Hence, there are some commonalities between spectral and kernel clustering. Further, multi-kernel clustering attempts to find a desired kernel by fusing multiple kernels. In a sense, multi-kernel clustering is also a special kind of multi-view clustering. Inspired by SimpleMKKM [17], Localized SimpleMKKM [18] and IMKAMC [19] further focus on locality and incomplete kernels, respectively.

2.3 Multi-View Clustering

Nowadays, numerous multi-view clustering algorithms emerge in an endless stream. SwMC [20], [21] fuses multiple graphs, MLAN [22] learns consensus graph, and GMC [23] considers the above two. MCGC [24] forces the learned graph to be a diagonal matrix. CDMGC [25] focuses both diversity and consistency between views. BMVC [26] utilizes binary coding to accelerate k-means. OPLFMC [27] develops a one-pass late fusion strategy. DSRL [28] proposes a deep sparse regularizer for performance. EOMSC [29] uses anchors for efficient subspace clustering.

2.4 Multi-View Classification

In some cases, classification tasks are also required. KNN, SVM and AdaBoost are all well-known classic algorithms. Under the need of multi-view classification, AMGL [30], MLAN and DSRL are all extended accordingly. MVAR [31] utilizes adaptive regression to achieve multi-view classification. TMC [32] is aimed at the problem of trusted multi-view classification in neural networks.

2.5 Model Explainability

In recent years, deep learning has generally achieved promising performance on many tasks. Relying on some intuitions and assumptions, rich loss functions are designed. But deep architectures are stacked wider and deeper with ad-hoc modules, making it seem difficult to understand their working mechanisms. Therefore, much work has been devoted towards exploring how existing models work or providing connections to traditional models, i.e., the explainability of neural networks. In general, these works utilize visualization techniques [33], [34], [35] or specific experiments [36], [37] for post-hoc explanation.

2.6 Interpretable Model

For high-stakes decisions, Rudin [38] advocates stopping explaining black-box models and use interpretable models instead. As a good tool, differentiable programming bridges traditional machine learning and deep neural networks. The earliest work should be traced back to the LISTA [39] proposed by Gregor and LeCun, which transformed the ISTA algorithm into a simple RNN structure. Inspired by LISTA, some studies [40], [41], [42] solve the LASSO problem through RNN, while others focus on compressed sensing [43], [44], sparse coding [45] and clustering [46], [47].

3 Method

The proposed method is elaborated in five parts: problem formulation, differentiable framework, loss function, flexible extension and interpretability.

3.1 Problem Formulation

Symmetric spectral clustering [48] can be written in the following form:

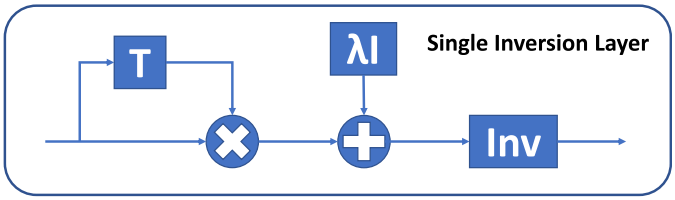

Fig. 1. Single inversion layer for inferring $\mathbf{G}_k^{(i)}$ from $\mathbf{F}_{k-1}$.

$$\min_{\mathbf{F}\geq\mathbf{0}} \left\|\mathbf{W}-\mathbf{F}\mathbf{F}^T\right\|_F^2, \tag{1}$$

where $\mathbf{W}\in\mathbb{R}^{n\times n}$ indicates a graph, $\mathbf{F}\in\mathbb{R}^{n\times c}$ represents a cluster indicator matrix, n and c are the number of samples and clusters, respectively. This is a relaxed form of subgraph partitioning, which usually performs better, but is relatively cumbersome to solve. Asymmetric spectral clustering can be written as

$$\min_{\mathbf{F}\geq\mathbf{0},\mathbf{G}\geq\mathbf{0}} \left\|\mathbf{W}-\mathbf{F}\mathbf{G}^T\right\|_F^2, \tag{2}$$

where $\mathbf{G}\in\mathbb{R}^{n\times c}$ can be regarded as a cluster center matrix, each column of which represents a cluster center. Essentially this is generalized k-means by adjacency relations, whose performance is usually lower than the former, but solution is relatively easy. A simple compromise is as follows.

$$\min_{\mathbf{F}\geq\mathbf{0},\mathbf{G}\geq\mathbf{0}} \left\|\mathbf{W}-\mathbf{F}\mathbf{G}^T\right\|_F^2+\lambda\|\mathbf{F}-\mathbf{G}\|_F^2. \tag{3}$$

When λ is large enough, the above equation approaches the former. When λ is small enough, the above formula tends to the latter. Extending it to the case of multiple views, there is the following objective function.

$$\min_{\mathbf{F}\geq\mathbf{0},\mathbf{G}^{(i)}\geq\mathbf{0}} \sum_{i=1}^{n_v}\left[\left\|\mathbf{W}^{(i)}-\mathbf{F}\mathbf{G}^{(i)T}\right\|_F^2+\lambda_i\left\|\mathbf{F}-\mathbf{G}^{(i)}\right\|_F^2\right]. \tag{4}$$

All views share the same cluster indicator matrix. When $\lambda_i=0$, the ith view hardly affects $\mathbf{F}$ because $\mathbf{F}$ tends to maintain symmetry over views where λ_i is large. And for any $\mathbf{F}$, if $\mathbf{F}=\mathbf{G}^{(i)}$ is not enforced, there is always a suitable $\mathbf{G}^{(i)}$ such that the residual of $\|\mathbf{W}^{(i)}-\mathbf{F}\mathbf{G}^{(i)T}\|_F^2$ is small enough. Therefore, λ_i can be directly regarded as the current-perspective weight.

3.2 Differentiable Framework

Ignoring the non-negative constraint temporarily, Eq. (4) is expanded as follows.

$$\begin{aligned}\mathcal{L}=\sum_{i=1}^{n_v}\Big[&\mathrm{Tr}\left(\mathbf{W}^{(i)T}\mathbf{W}^{(i)}-2\mathbf{W}^{(i)T}\mathbf{F}\mathbf{G}^{(i)T}+\mathbf{G}^{(i)}\mathbf{F}^T\mathbf{F}\mathbf{G}^{(i)T}\right)\\&+\lambda_i\mathrm{Tr}\left(\mathbf{F}^T\mathbf{F}-2\mathbf{F}^T\mathbf{G}^{(i)}+\mathbf{G}^{(i)T}\mathbf{G}^{(i)}\right)\Big].\end{aligned} \tag{5}$$

Taking the partial derivative with respect to $\mathbf{G}^{(i)}$, we get

$$\frac{\partial\mathcal{L}}{\partial\mathbf{G}^{(i)}}=2\left(\mathbf{G}^{(i)}\mathbf{F}^T\mathbf{F}-\mathbf{W}^{(i)T}\mathbf{F}\right)+2\lambda_i\left(\mathbf{G}^{(i)}-\mathbf{F}\right). \tag{6}$$

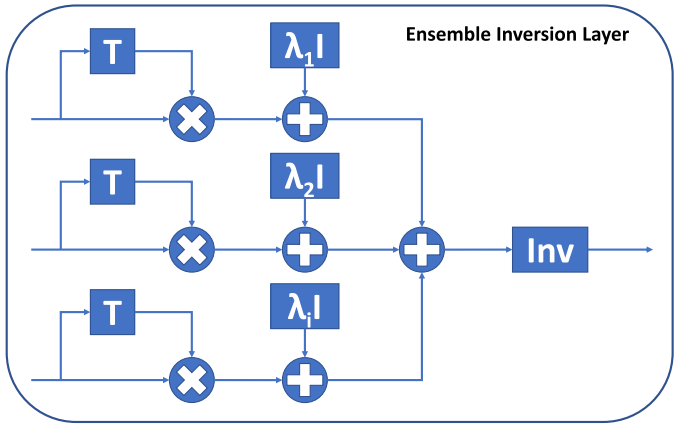

Fig. 2. Ensemble inversion layer for updating $\mathbf{F}_k$ by aggregating $\mathbf{G}_k^{(i)}$.

The solution to $\mathbf{G}^{(i)}$ under unconstrained conditions is

$$\mathbf{G}^{(i)}=\left(\mathbf{W}^{(i)T}+\lambda_i\mathbf{I}\right)\mathbf{F}\left(\mathbf{F}^T\mathbf{F}+\lambda_i\mathbf{I}\right)^{-1}. \tag{7}$$

Projecting it onto the feasible region of non-negative constraints, we have

$$\mathbf{G}^{(i)}=\max\left\{\left(\mathbf{W}^{(i)T}+\lambda_i\mathbf{I}\right)\mathbf{F}\left(\mathbf{F}^T\mathbf{F}+\lambda_i\mathbf{I}\right)^{-1},0\right\}. \tag{8}$$

Replacing $(\mathbf{W}^{(i)T}+\lambda_i\mathbf{I})\mathbf{F}$ with a learnable parameter $\mathbf{U}^{(i)}$, the following update rule can be obtained.

$$\mathbf{G}_k^{(i)}=\mathbf{ReLU}\left(\mathbf{U}_k^{(i)}\left(\mathbf{F}_{k-1}^T\mathbf{F}_{k-1}+\lambda_i\mathbf{I}\right)^{-1}\right). \tag{9}$$

Based on the above rule, we construct a single inversion layer as shown in Fig. 1.

Taking the partial derivative with respect to $\mathbf{F}$, we get

$$\frac{\partial\mathcal{L}}{\partial\mathbf{F}}=2\sum_{i=1}^{n_v}\left[\left(\mathbf{F}\mathbf{G}^{(i)T}\mathbf{G}^{(i)}-\mathbf{W}^{(i)}\mathbf{G}^{(i)}\right)+\lambda_i\left(\mathbf{F}-\mathbf{G}^{(i)}\right)\right]. \tag{10}$$

The solution to $\mathbf{F}$ under unconstrained conditions is

$$\mathbf{F}=\left[\sum_{i=1}^{n_v}\left(\mathbf{W}^{(i)}+\lambda_i\mathbf{I}\right)\mathbf{G}^{(i)}\right]\left[\sum_{i=1}^{n_v}\left(\mathbf{G}^{(i)T}\mathbf{G}^{(i)}+\lambda_i\mathbf{I}\right)\right]^{-1}. \tag{11}$$

Projecting it onto the feasible region of non-negative constraints, we have

$$\mathbf{F}=\max\left\{\left[\sum_{i=1}^{n_v}\left(\mathbf{W}^{(i)}+\lambda_i\mathbf{I}\right)\mathbf{G}^{(i)}\right]\left[\sum_{i=1}^{n_v}\left(\mathbf{G}^{(i)T}\mathbf{G}^{(i)}+\lambda_i\mathbf{I}\right)\right]^{-1},0\right\}. \tag{12}$$

Replacing $\sum_{i=1}^{n_v}(\mathbf{W}^{(i)}+\lambda_i\mathbf{I})\mathbf{G}^{(i)}$ with a learnable parameter $\mathbf{V}$, the following update rule can be obtained.

$$\mathbf{F}_k=\mathbf{ReLU}\left(\mathbf{V}_k\left[\sum_{i=1}^{n_v}\left(\mathbf{G}_k^{(i)T}\mathbf{G}_k^{(i)}+\lambda_i\mathbf{I}\right)\right]^{-1}\right). \tag{13}$$

Based on the above rule, we construct an ensemble inversion layer as shown in Fig. 2.

The full rules are summarized in Eq. (14). According to Eq. (14), the complete network structure can be constructed as depicted in Fig. 3. It should be pointed out that λ_i also

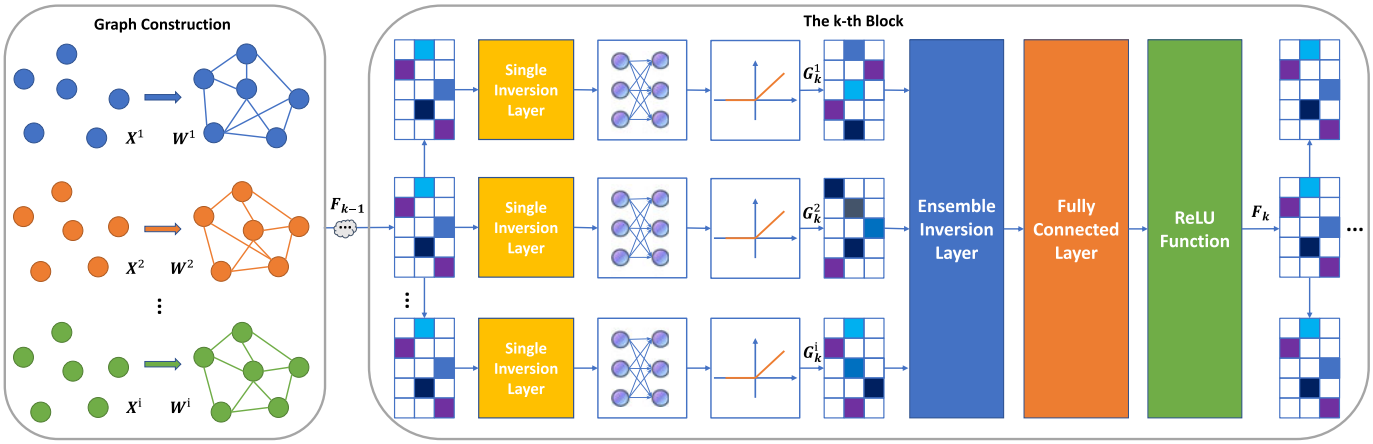

Fig. 3. The k-th block in our network architecture, and the complete network is formed by stacking multiple such blocks.

serves as a learnable parameter.

$$\begin{cases} \mathbf{G}_k^{(i)} = \mathbf{ReLU}\left(\mathbf{U}_k^{(i)}\left(\mathbf{F}_{k-1}^T\mathbf{F}_{k-1} + \lambda_i\mathbf{I}\right)^{-1}\right) \\ \mathbf{F}_k = \mathbf{ReLU}\left(\mathbf{V}_k\left[\sum_{i=1}^{n_v}\left(\mathbf{G}_k^{(i)T}\mathbf{G}_k^{(i)} + \lambda_i\mathbf{I}\right)\right]^{-1}\right) \end{cases}. \tag{14}$$

3.3 Loss Function

Considering that each view should fit its corresponding graph structure, there is an L_1 loss as

$$L_1 = \left\| \sum_{i=1}^{n_v} (\mathbf{W}^{(i)} - \mathbf{G}_k^{(i)}\mathbf{G}_k^{(i)T})/n_v \right\|_F^2. \tag{15}$$

On the other hand, the sharing matrix should fit the graph structure containing all the view information, so there is an L_2 loss as

$$L_2 = \left\| \sum_{i=1}^{n_v} \mu_i \mathbf{W}^{(i)} - \mathbf{F}_k\mathbf{F}_k^T \right\|_F^2. \tag{16}$$

In summary, the complete clustering loss L_{clu} is as follows.

$$L_{clu} = L_1 + \alpha L_2. \tag{17}$$

In order to find a suitable $\boldsymbol{\mu}$, we first give the following definition [49].

Definition 1. *Let $\mathbf{P}_c = [\mathbf{p}_1, \ldots, \mathbf{p}_c]$ and $\tilde{\mathbf{P}}_c = [\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_c]$ be orthogonal eigenvector spanned subspaces, and let $\sigma_1 \le \sigma_2 \le \cdots \le \sigma_c$ be the singular values of $\mathbf{P}_c^T\tilde{\mathbf{P}}_c$. Then the values $\omega_i = \arccos\sigma_i$ are called canonical angles between $\mathbf{P}_c$ and $\tilde{\mathbf{P}}_c$.*

If $\mathbf{P}_c$ and $\tilde{\mathbf{P}}_c$ are composed of the first c eigenvectors of $\mathbf{W}$ and $\tilde{\mathbf{W}}$ respectively, the canonical angles can naturally capture the difference between $\mathbf{W}$ and $\tilde{\mathbf{W}}$. Specifically, the smaller the largest canonical angle is, the closer $\mathbf{W}$ and $\tilde{\mathbf{W}}$ are. Denote the desirable graph by $\mathbf{W} = \sum_{i=1}^{n_v}\mu_i\mathbf{W}^{(i)}$. The following theory holds [49].

Theorem 1. *Let $q_i^{(v)}$ and q_i be the ith eigenvalue of $\mathbf{W}^{(v)}$ and $\mathbf{W}$ respectively, where $q_1^{(v)} \ge \cdots \ge q_n^{(v)}$ and $q_1 \ge \cdots \ge q_n$. Let $\mathbf{p}_i^{(v)}$ and $\mathbf{p}_i$ be the ith eigenvector of $\mathbf{W}^{(v)}$ and $\mathbf{W}$ respectively. Denote $\boldsymbol{\Omega}$ $\operatorname{diag}\ \omega_1, \ldots, \omega_c$ as the canonical angles between $\mathbf{P}_c^{(v)}$ and $\mathbf{P}_c$. If there exists a gap $\delta > 0$, such that $q_c^{(v)} \ge \delta$ and $|q_c^{(v)} - q_{c+1}| \ge \delta$. Then $\|\sin\boldsymbol{\Omega}\|_F \ll 1/\delta\|\mathbf{W}\mathbf{P}_c^{(v)} - \mathbf{P}_c^{(v)}\mathbf{Q}_c^{(v)}\|_F$, where $\mathbf{Q}_c^{(v)} = \operatorname{diag}(q_1^{(v)}, \ldots, q_c^{(v)})$.*

From the above, it can be seen that the following relationship is required.

$$\begin{aligned} &\min_{\boldsymbol{\mu}} \sum_{v=1}^{n_v} \|\mathbf{W}\mathbf{P}_c^{(v)} - \mathbf{P}_c^{(v)}\mathbf{Q}_c^{(v)}\|_F^2 \\ &\text{s.t. } \mathbf{W} = \sum_{v=1}^{n_v}\mu_v\mathbf{W}^{(v)},\ \boldsymbol{\mu}^T\mathbf{1} = 1,\ \boldsymbol{\mu} \ge \mathbf{0}. \end{aligned} \tag{18}$$

Let $\mathbf{M}$ and $\mathbf{y}$ satisfy the following equalities.

$$M_{ij} = \sum_{v=1}^{n_v} \mathrm{Tr}(\mathbf{W}^{(i)}\mathbf{P}_c^{(v)}\mathbf{P}_c^{(v)T}\mathbf{W}^{(j)T}) \tag{19}$$

$$y_i = \sum_{v=1}^{n_v} \mathrm{Tr}(\mathbf{W}^{(i)}\mathbf{P}_c^{(v)}\mathbf{Q}_c^{(v)T}\mathbf{P}_c^{(v)T}). \tag{20}$$

Eq. (18) can be further organized into Eq. (21).

$$\begin{aligned} &\min_{\boldsymbol{\mu}} \boldsymbol{\mu}^T\mathbf{M}\boldsymbol{\mu} - 2\boldsymbol{\mu}^T\mathbf{y} \\ &\text{s.t. } \boldsymbol{\mu}^T\mathbf{1} = 1,\ \boldsymbol{\mu} \ge \mathbf{0}. \end{aligned}$$

This is obviously a standard quadratic programming problem. We provide Algorithm 1 to solve it. The derivation details of Algorithm 1 are described in Appendix A, available online.

Algorithm 1. Solution to Problem (21)

Input: Matrix $\mathbf{M}$, vector $\mathbf{y}$ and penalty parameter ρ.
Output: Weight $\boldsymbol{\mu}$ for each view.

```
Initialize μ_i = 1/n_v and η = 0.
Define f(ξ̄) = (1/n_v) Σ_{i=1}^{n_v} (ξ̄ − z_i)_+ − ξ̄.
while non-convergence do
    Compute ν = μ + (η − M^T μ)/ρ.
    Compute e = (ρν − η − Mν − 2y)/ρ.
    Compute z = e − (11^T/n_v) e + 1/n_v.
    Obtain ξ̄* by iterating ξ̄_{t+1} = ξ̄_t − f(ξ̄_t)/f'(ξ̄_t) 2-4 times.
    Compute μ_i = (z_i − ξ̄*)_+.
    Get a more accurate multiplier by η = η + ρ(μ − ν).
    Increase the penalty parameter by ρ = τρ.
end while
```

3.4 Flexible Extension

In fact, due to the particularity of the network structure, our model can be flexibly extended to solve other problems, by changing activation functions and loss functions.

3.4.1 Semi-Supurvised Classification

For semi-supervised classification, the correct labels need to be assigned while clustering. Therefore, the correct cluster indicator matrix $\mathbf{F}$ should approximate the empirically estimated $\mathbf{F}_{emp}$. There is an L_3 loss as

$$L_3 = \left\|\mathbf{F}_k - \mathbf{F}_{emp}\right\|_F^2. \tag{22}$$

Considering both clustering and label correctness, the complete classification loss L_{cla} is as follows.

$$L_{cla} = L_1 + \alpha L_2 + \beta L_3. \tag{23}$$

Rearrange the sample order so that the labeled samples come first, and $\mathbf{F}_{emp}$ is as follows.

$$\mathbf{F}_{emp} = \begin{bmatrix} \mathbf{F}_l \\ \mathbf{F}_u \end{bmatrix}, \tag{24}$$

where $\mathbf{F}_l$ and $\mathbf{F}_u$ are the cluster indicator matrices for labeled samples and unlabeled samples, respectively. Although common, it is not reasonable to treat the part of unlabeled samples as 0. Further, we konw the following relationship holds.

$$\mathbf{W} = \mathbf{F}_{emp}\mathbf{F}_{emp}^T. \tag{25}$$

Hence, we have

$$\begin{bmatrix} \mathbf{F}_l \\ \mathbf{F}_u \end{bmatrix} \begin{bmatrix} \mathbf{F}_l^T & \mathbf{F}_u^T \end{bmatrix} = \begin{bmatrix} \mathbf{F}_l\mathbf{F}_l^T & \mathbf{F}_l\mathbf{F}_u^T \\ \mathbf{F}_u\mathbf{F}_l^T & \mathbf{F}_u\mathbf{F}_u^T \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{ll} & \mathbf{W}_{lu} \\ \mathbf{W}_{ul} & \mathbf{W}_{uu} \end{bmatrix}. \tag{26}$$

Then $\mathbf{F}_u$ is estimated as

$$\mathbf{F}_u = \mathbf{W}_{ul}\mathbf{F}_l(\mathbf{F}_l^T\mathbf{F}_l)^{-1}. \tag{27}$$

3.4.2 Nonnegative Matrix Factorization

When the input is single-view data, the loss function L_{clu} can be rewritten as

$$L_{SymNMF} = \left\|\mathbf{W} - \mathbf{G}_k\mathbf{G}_k^T\right\|_F^2 + \alpha\left\|\mathbf{W} - \mathbf{F}_k\mathbf{F}_k^T\right\|_F^2. \tag{28}$$

This is obviously a symmetric nonnegative matrix factorization problem with $\alpha = 1$. Without loss of generality, $\mathbf{W}$ is replaced by $\mathbf{X}$. If symmetry is not required, the above loss function L_{SymMNF} can be further written as

$$L_{NMF} = \left\|\mathbf{X} - \mathbf{F}_k\mathbf{G}_k^T\right\|_F^2. \tag{29}$$

This is a standard non-negative matrix factorization problem with a wide range of applications. If manifold preserving is required, the following loss function can be adopted.

$$L_{GNMF} = \left\|\mathbf{X} - \mathbf{F}_k\mathbf{G}_k^T\right\|_F^2 + \alpha\mathrm{Tr}(\mathbf{F}_k^T\mathbf{L}_F\mathbf{F}_k). \tag{30}$$

This is a graph-regularized nonnegative matrix factorization form. In addition, a hierarchical variant of NMF is widely recognized, and learned with the following loss function.

$$L_{DeepNMF} \quad \mathbf{X}^T \quad \mathbf{G}_1\mathbf{G}_2 \quad \mathbf{G}_k\mathbf{F}_k^{T\,2}_F. \tag{31}$$

Here each block operation is treated as a decomposition rather than an iteration. The idea is feasible because the update rules of $\mathbf{F}$ and $\mathbf{G}$ do not depend on $\mathbf{X}$ due to the local parameterization.

3.4.3 Principal Component Analysis

The original principal component analysis can be rewritten as follows.

$$\min_{\mathbf{F},\mathbf{G}} \left\|\mathbf{X} - \mathbf{F}\mathbf{G}^T\right\|_F^2 \quad \text{s.t. } \mathbf{G}^T\mathbf{G} = \mathbf{I}. \tag{32}$$

Therefore, we have the following loss function

$$L_{PCA} = \left\|\mathbf{X} - \mathbf{F}_k\mathbf{G}_k^T\right\|_F^2 + \alpha\left\|\mathbf{G}_k^T\mathbf{G}_k - \mathbf{I}\right\|_F^2, \tag{33}$$

where α should be set large enough.

Let the activation functions corresponding to $\mathbf{F}$ and $\mathbf{G}$ be $\mathrm{Activation}_F(x)$ and $\mathrm{Activation}_G(x)$. If other constraints are not imposed on $\mathbf{F}$ and $\mathbf{G}$, let $\mathrm{Activation}_F(x) = x$ and $\mathrm{Activation}_G(x) = x$. If $\mathbf{F}$ is required to be non-negative or non-linear, let $\mathrm{Activation}_F(x) = \mathbf{ReLU}(x)$. If $\mathbf{F}$ is required to be sparse, let $\mathrm{Activation}_F(x) = \mathrm{sgn}(x)(|x| - \theta)_+$.

3.4.4 Joint Learning

Taking DEC as an example, deep clustering usually consists of two main parts: a feature learning module and a clustering module. Joint learning of the two modules is a key to this type of approach. On the one hand, we can learn features $\mathbf{Z}$ using the reconstruction loss of the autoencoder. On the other hand, we can utilize the proposed module to perform generalized k-means on $\mathbf{Z}$. Since the proposed module is differentiable, it can be learned jointly with the autoencoder and has the following loss function:

$$L_{JL} = \left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_F^2 + \alpha\left\|\mathbf{Z} - \mathbf{F}_k\mathbf{G}_k^T\right\|_F^2, \tag{34}$$

where $\hat{\mathbf{X}}$ is the data reconstructed by an autoencoder. Besides, $\mathbf{F}$ should be non-negative, so let $\mathrm{Activation}_F(x) = \mathbf{ReLU}(x)$ while $\mathrm{Activation}_G(x) = x$.

3.5 Interpretability

Compared with traditional optimization algorithms, deep learning models often have better performance. But deep learning is generally difficult to explain, which greatly limits the reliability assessment of results. While much work helps to understand networks by designing experiments or visualizing intermediate processes, many fields still tend to adopt traditional models. This is because the network structure of deep models is a complex non-convex mapping, resulting in unstable training. On the contrary, in order to stabilize the output, a series of additional operations are proposed to further increase the network complexity.

Different from explaining a black box, we use a traditional optimization algorithm to construct the network structure. Such a network is naturally interpretable, and narrows the search range of solutions to ensure the correct direction of optimization. Since the update rule is partially parameterized, it can be learned more flexibly through various loss functions, which is called learning-based optimization. This is equivalent

to adding computational perturbations to traditional optimization, thereby expanding the search range of solutions.

Essentially our method searches for a convergent sequence $\{\mathbf{F}_t\}$ like traditional ones. So data is propagated through a block, similar to one iteration in traditional optimization. In traditional optimization, $\mathbf{F}_t$ does not change once it is determined, and the sequence length is unknown. In learning-based optimization, the sequence length is fixed to the number of blocks, and $\mathbf{F}_t$ can be updated.

In addition, for partial parameterization, we have the following considerations.

(1) *Substitutability*. $\sum_{i=1}^{n_v}(\mathbf{W}^{(i)}+\lambda_i\mathbf{I})\mathbf{G}^{(i)}$ is a compound linear operation that can be easily replaced. $[\sum_{i=1}^{n_v}(\mathbf{G}^{(i)T}\mathbf{G}^{(i)}+\lambda_i\mathbf{I})]^{-1}$ contains an inversion operation and is symmetric, which is difficult to replace.

(2) *Complexity*. After $\sum_{i=1}^{n_v}(\mathbf{W}^{(i)}+\lambda_i\mathbf{I})\mathbf{G}^{(i)}$ is replaced, the forward/backward propagation complexity is $O(nc^2)$. After $[\sum_{i=1}^{n_v}(\mathbf{G}^{(i)T}\mathbf{G}^{(i)}+\lambda_i\mathbf{I})]^{-1}$ is replaced, the forward/backward propagation complexity is $O(n^2c)$.

(3) *Stability*. Eq. (35) shows, the larger λ_i is, the larger the spectral norm of $\sum_{i=1}^{n_v}(\mathbf{W}^{(i)}+\lambda_i\mathbf{I})\mathbf{G}^{(i)}$ is, and the more severe the space transformation is. Eq. (36) shows, the larger λ_i is, the smaller the spectral norm of $[\sum_{i=1}^{n_v}(\mathbf{G}^{(i)T}\mathbf{G}^{(i)}+\lambda_i\mathbf{I})]^{-1}$ is, and the more gentle the space transformation is.

$$\begin{aligned}&\left\|\sum_{i=1}^{n_v}\left(\mathbf{W}^{(i)}+\lambda_i\mathbf{I}\right)\mathbf{G}^{(i)}\right\|_2\\&\geq\left\|\sum_{i=1}^{n_v}\lambda_i\mathbf{G}^{(i)}\right\|_2=\sum_{i=1}^{n_v}\lambda_i\sigma_{\max}(\mathbf{G}^{(i)})\end{aligned}\tag{35}$$

$$\begin{aligned}&\left\|\left[\sum_{i=1}^{n_v}\left(\mathbf{G}^{(i)T}\mathbf{G}^{(i)}+\lambda_i\mathbf{I}\right)\right]^{-1}\right\|_2\\&\leq\sum_{i=1}^{n_v}\left\|\left(\mathbf{G}^{(i)T}\mathbf{G}^{(i)}+\lambda_i\mathbf{I}\right)^{-1}\right\|_2\leq\sum_{i=1}^{n_v}\frac{1}{\lambda_i}.\end{aligned}\tag{36}$$

(4) *Robustness*. Eq. (37) shows, when λ_i is large enough, the condition number of $\sum_{i=1}^{n_v}(\mathbf{W}^{(i)}+\lambda_i\mathbf{I})\mathbf{G}^{(i)}$ is very large, meaning that a small disturbance will cause a huge difference in the solution. Eq. (38) shows, when λ_i is large enough, the condition number of $[\sum_{i=1}^{n_v}(\mathbf{G}^{(i)T}\mathbf{G}^{(i)}+\lambda_i\mathbf{I})]^{-1}$ is always 1, meaning that the solution is not easily affected by disturbances.

$$\begin{aligned}&\lim_{\lambda_i\to\infty}\mathrm{Cond}\left[\sum_{i=1}^{n_v}\left(\mathbf{W}^{(i)}+\lambda_i\mathbf{I}\right)\mathbf{G}^{(i)}\right]\\&=\lim_{\lambda_i\to\infty}\frac{\sum_i\sigma_{\max}(\mathbf{W}^{(i)}\mathbf{G}^{(i)})+\lambda_i\sigma_{\max}(\mathbf{G}^{(i)})}{\sum_i\sigma_{\min}(\mathbf{W}^{(i)}\mathbf{G}^{(i)})+\lambda_i\sigma_{\min}(\mathbf{G}^{(i)})}\\&=\frac{\sum_i\sigma_{\max}(\mathbf{G}^{(i)})}{\sum_i\sigma_{\min}(\mathbf{G}^{(i)})}\end{aligned}\tag{37}$$

$$\lim_{\lambda_i\to\infty}\mathrm{Cond}\left\{\left[\sum_{i=1}^{n_v}\left(\mathbf{G}^{(i)T}\mathbf{G}^{(i)}+\lambda_i\mathbf{I}\right)\right]^{-1}\right\}\tag{38}$$

$$=\lim_{\lambda_i\to\infty}\frac{\sum_i\sigma^2_{\min}(\mathbf{G}^{(i)})+\lambda_i}{\sum_i\sigma^2_{\max}(\mathbf{G}^{(i)})+\lambda_i}=1.$$

TABLE 1
Description for Multi-View Datasets

Dataset	Type	#Views	#Instances	#Classes
MSRC	Image	5	210	7
MNIST	Image	3	2,000	10
ALOI	Image	4	1,080	10
NUS-WIDE	Image	6	1,600	8
Caltech7	Image	6	1,474	7
100Leaves	Image	3	1,600	100
3Sources	Document	3	169	6
BBCNews	Document	4	685	5
BBCSport	Document	2	544	5
Youtube	Video	6	2,000	10

4 Experiment

This section is divided into four aspects: experimental details, multi-view clustering, multi-view classification and ablation study.

4.1 Experimental Details

The experimental details are elaborated in three parts: datasets, baselines and settings.

Datasets. In the experiments, 6 image datasets, 3 document datasets and 1 video dataset are adopted. Their statistics are described in Table 1, and the rest are as follows.

- *MSRC*[1] is an image dataset containing 7 distinct subjects. Its feature descriptors Color Moment (CM), Gist, Histogram of Oriented Gradient (HOG), CENsus TRansform hISTogram (CENTRIST) and Local Binary Pattern (LBP) are seen as five views.
- *MNIST*[2] is a handwritten digit dataset. Its original features are processed by three methods Isometric Projection, Linear Discriminant Analysis and Neighborhood Preserving Embedding. Each method is seen as one view.
- *ALOI*[3] is an object image dataset under varied light conditions and rotation angles. Its feature descriptors RGB Color Histogram, HSV Color Histogram, Color Similarity and Haralick are seen as four views.
- *NUS-WIDE*[4] is an object image dataset collected from web, whose feature descriptors contain Color Histogram, block-wise CM, Color Correlogram, Edge Direction Histogram (EDH), Wavelet Texture and Bag of Words (BoW) from Scale-Invariant Feature Transform (SIFT). Each feature descriptor is treated as one view.
- *Caltech7*[5] is a subset of object dataset Caltech101, whose feature descriptors contain Gabor, Wavelet Moment (WM), CENTRIST, HOG, Gist and LBP. Each feature descriptor is treated as one view.

1. http://yacvid.hayko.at/task.php?did=35
2. http://yann.lecun.com/exdb/mnist/
3. https://aloi.science.uva.nl/
4. https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html
5. http://www.vision.caltech.edu/Image_Datasets/Caltech101/

TABLE 2
Accuracy (ACC% ± Std%), Normalized Mutual Information (NMI% ± Std%) and Adjusted Rand Index (ARI% ± Std%) for Multi-View Clustering Algorithm Comparison

Dataset	Metric	SwMC	MLAN	GMC	MCGC	CDMGC	BMVC	OPLFMC	DSRL	EOMSC	Ours
MSRC	ACC	78.8±8.3	85.7±0.0	83.3±0.0	81.9±0.0	85.1±6.6	70.5±0.0	83.8±0.0	81.4±2.6	56.2±0.0	**92.0±0.3**
	NMI	71.7±7.5	74.9±0.0	79.6±0.0	71.6±0.0	77.1±4.1	62.3±0.0	70.1±0.0	74.1±2.8	45.7±0.0	**86.1±0.4**
	ARI	62.7±11.9	70.4±0.0	72.7±0.0	69.3±0.0	71.0±7.6	53.5±0.0	67.1±0.0	65.5±3.5	33.4±0.0	**83.1±0.6**
MNIST	ACC	84.5±3.5	81.9±0.0	90.2±0.0	86.5±0.0	87.0±0.4	77.5±1.2	80.0±0.0	82.9±2.6	88.2±0.0	**91.7±0.1**
	NMI	75.0±1.2	74.3±0.0	79.9±0.0	76.9±0.0	76.3±0.5	69.4±0.1	74.6±0.0	74.2±1.6	76.4±0.0	**82.7±0.2**
	ARI	72.9±3.2	71.4±0.0	81.0±0.0	75.2±0.0	73.9±1.0	64.7±0.6	70.6±0.0	68.1±4.4	77.6±0.0	**84.2±0.1**
ALOI	ACC	65.3±3.7	77.0±1.2	66.7±0.0	72.5±0.0	69.2±0.3	68.4±0.0	70.9±0.0	78.1±2.6	50.5±0.0	**97.9±0.4**
	NMI	64.2±2.3	69.6±1.1	62.6±0.0	72.0±0.0	67.7±0.4	68.2±0.0	72.3±0.0	78.8±1.8	57.7±0.0	**96.2±0.5**
	ARI	36.4±3.3	50.1±2.2	34.2±0.0	55.0±0.0	41.7±0.7	54.1±0.0	59.1±0.0	61.3±4.1	39.3±0.0	**95.4±0.8**
NUS-WIDE	ACC	20.1±3.2	36.1±0.0	20.1±0.0	31.1±0.0	22.4±1.4	33.0±0.2	29.4±0.0	**41.0±0.7**	31.5±0.0	39.4±0.3
	NMI	8.7±3.7	24.1±0.1	12.2±0.0	15.7±0.0	13.8±0.8	18.3±0.3	13.7±0.0	**26.4±0.3**	15.2±0.0	25.1±0.2
	ARI	3.0±2.1	15.6±0.0	4.1±0.0	10.5±0.0	4.9±0.4	13.1±0.3	8.5±0.0	16.0±1.3	11.5±0.0	**17.5±0.1**
Caltech7	ACC	71.3±4.9	72.7±0.0	69.2±0.0	64.5±0.0	72.1±3.2	58.6±4.0	39.1±0.0	84.0±1.3	60.7±0.0	**84.5±1.7**
	NMI	54.4±1.8	59.2±0.0	60.6±0.0	49.2±0.0	54.5±0.7	53.7±3.9	25.8±0.0	62.4±2.8	46.5±0.0	**74.0±2.1**
	ARI	46.5±4.8	51.5±0.0	59.4±0.0	49.4±0.0	46.6±3.1	45.4±5.1	20.5±0.0	61.7±2.9	43.6±0.0	**83.6±3.7**
100Leaves	ACC	77.6±12.0	87.2±0.6	88.9±0.0	85.8±0.0	88.4±1.2	67.7±0.1	76.5±0.0	89.0±0.7	35.9±0.0	**92.1±0.3**
	NMI	86.2±7.5	93.8±0.1	94.4±0.0	89.8±0.0	94.9±1.0	83.4±0.1	87.1±0.0	95.0±0.3	62.2±0.0	**95.3±0.1**
	ARI	46.4±17.8	80.1±0.7	71.8±0.0	57.3±0.0	79.1±8.2	56.2±0.5	66.5±0.0	80.9±1.2	16.4±0.0	**87.7±0.5**
3Sources	ACC	70.8±4.4	76.3±0.0	65.1±0.0	46.8±0.0	64.3±1.3	65.7±0.0	56.2±0.0	71.0±2.9	37.9±0.0	**80.0±0.3**
	NMI	55.5±5.1	67.0±0.0	46.6±0.0	54.2±0.0	46.9±0.6	59.4±0.0	56.4±0.0	57.6±3.5	16.6±0.0	**78.6±0.3**
	ARI	45.3±6.8	58.0±0.0	32.9±0.0	35.2±0.0	33.2±1.6	54.8±0.0	38.5±0.0	52.1±5.1	7.9±0.0	**73.1±0.7**
BBCNews	ACC	70.6±0.3	83.7±0.0	69.1±0.0	79.4±0.0	71.8±2.1	85.4±8.3	56.8±0.0	47.6±0.7	41.6±0.0	**93.1±0.2**
	NMI	49.8±0.9	64.9±0.0	47.9±0.0	62.2±0.0	54.9±2.6	69.5±5.4	36.7±0.0	22.5±1.3	6.9±0.0	**80.3±0.6**
	ARI	48.2±1.4	68.0±0.0	47.5±0.0	57.0±0.0	52.7±2.5	73.1±9.3	29.2±0.0	8.7±1.3	9.2±0.0	**83.5±0.5**
BBCSport	ACC	90.4±6.8	97.1±0.0	97.8±0.0	79.4±0.0	73.6±0.1	85.8±5.9	62.0±0.0	63.3±2.7	36.0±0.0	**98.6±0.1**
	NMI	82.3±5.5	90.0±0.0	92.6±0.0	76.6±0.0	69.7±0.4	74.7±5.3	59.5±0.0	59.3±3.5	6.0±0.0	**95.1±0.5**
	ARI	80.7±8.7	92.1±0.0	93.4±0.0	75.5±0.0	59.5±0.3	78.0±7.8	51.4±0.0	36.4±3.1	5.8±0.0	**96.5±0.5**
Youtube	ACC	16.3±4.4	17.0±0.3	11.1±0.0	18.5±0.0	11.2±0.4	22.2±0.1	20.7±0.0	41.1±0.8	14.3±0.0	**47.8±0.1**
	NMI	7.1±4.7	6.9±0.1	1.5±0.0	7.3±0.0	1.6±0.4	11.1±0.1	7.1±0.0	26.8±0.6	3.5±0.0	**32.5±0.3**
	ARI	2.1±1.9	2.3±0.1	0.1±0.0	4.5±0.0	0.1±0.0	6.2±0.2	3.6±0.0	18.5±0.8	0.7±0.0	**25.8±0.2**

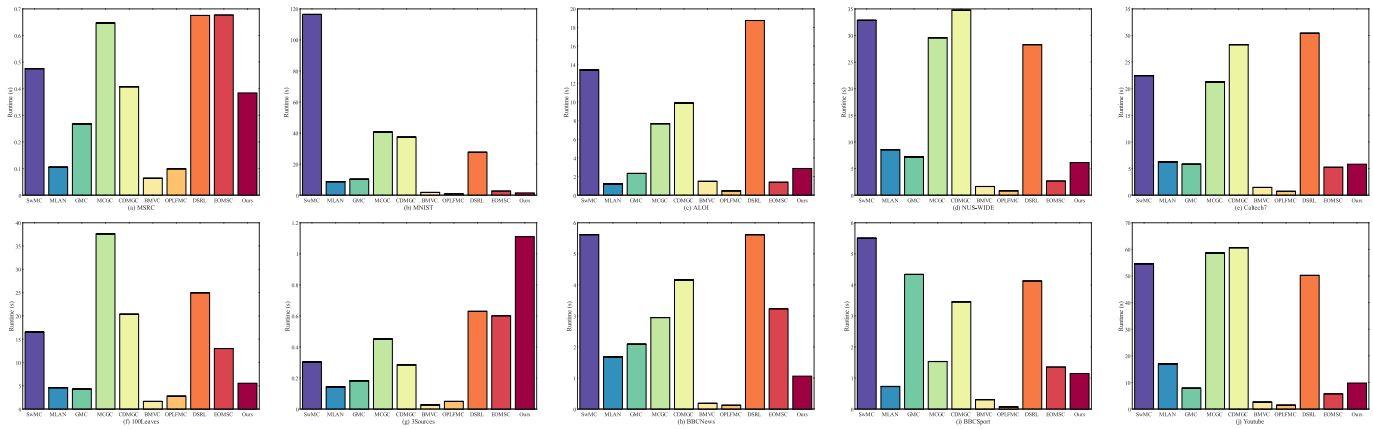

Fig. 4. Runtime for multi-view clustering algorithm comparison.

- *100Leaves*[6] is a dataset with 100 species of leaves, described by color, bag-of-words and texture features. Each feature is treated as one view.
- *3Sources*[7] is a news dataset extracted from sources BBC, Reuters and Guardian, where each source is regarded as one view.
- *BBCNews*[8] is a news dataset extracted from 3Sources. It is divided into four segments, where each segment is regarded as one view.
- *BBCSport*[8] is a sports news dataset extracted from BBCNews. It is divided into two segments, where each segment is regarded as one view.

6. https://archive.ics.uci.edu/ml/datasets.php
7. http://mlg.ucd.ie/datasets/3sources.html
8. http://mlg.ucd.ie/datasets/segment.html

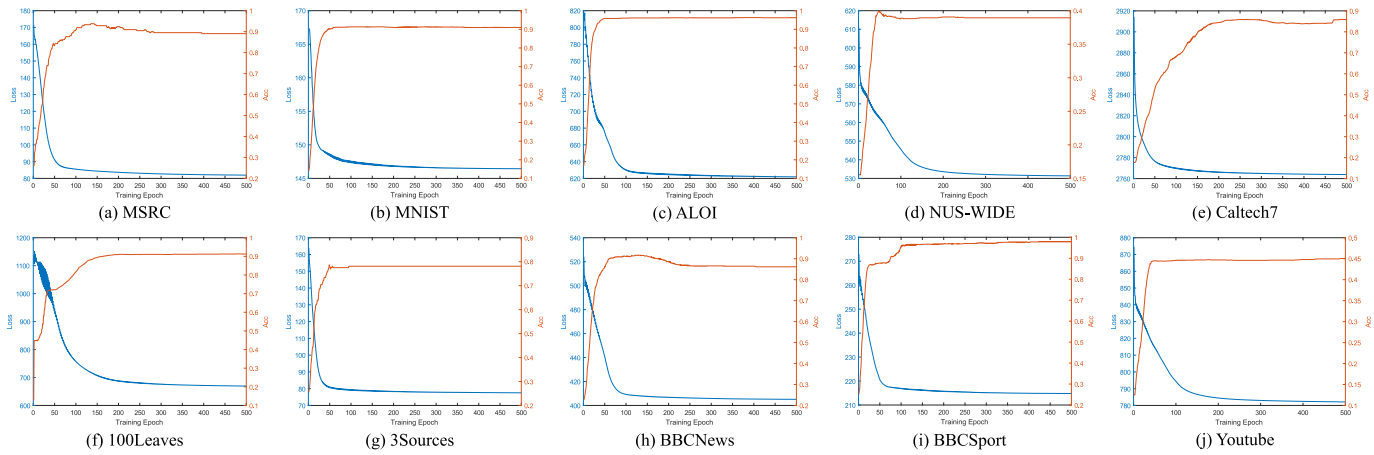

Fig. 5. Convergence curves of our model during training for multi-view clustering.

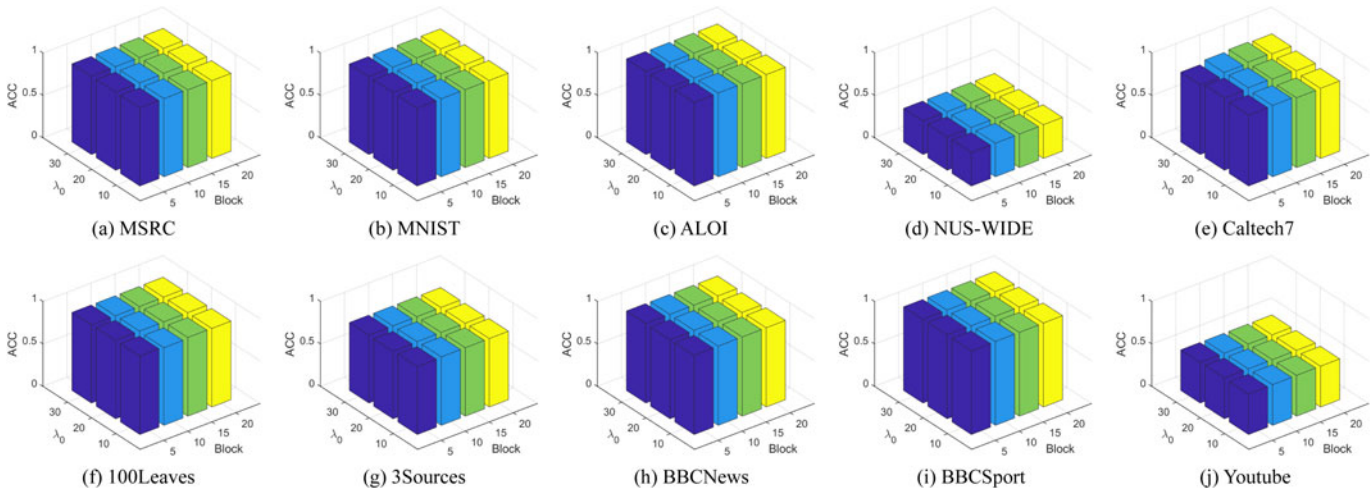

Fig. 6. Parameter sensitivity of our model for multi-view clustering.

TABLE 3
Accuracy (ACC% ± Std%) for Multi-View Classification Algorithm Comparison Under 10% Labeled Data

Dataset	KNN	SVM	AdaBoost	AMGL	MLAN	MVAR	TMC	DSRL	Ours
MSRC	68.7±4.3	49.2±7.8	41.1±6.5	75.3±8.4	83.7±4.2	64.8±9.7	74.1±6.6	76.1±6.6	**94.7±0.4**
MNIST	88.6±0.9	88.2±1.0	65.2±5.5	90.8±0.2	91.0±0.7	84.7±2.4	85.3±1.6	89.5±0.6	**91.9±0.2**
ALOI	88.0±0.3	85.7±1.8	63.5±7.6	85.1±2.7	89.6±1.1	84.2±6.9	95.3±1.0	91.6±0.7	**98.8±0.2**
NUS-WIDE	44.9±1.3	12.1±0.2	31.8±2.6	41.0±1.2	46.5±2.1	36.2±2.1	43.2±1.2	43.7±1.6	**51.0±0.3**
Caltech7	85.0±1.4	54.2±0.6	87.1±1.3	89.0±2.6	92.3±1.7	87.9±1.9	90.4±2.1	92.8±1.1	**94.2±0.6**
100Leaves	56.4±1.7	40.2±1.5	5.4±1.2	63.3±2.4	62.6±2.7	28.6±2.8	66.1±1.6	61.6±1.7	**96.8±0.1**
3Sources	70.1±6.6	30.4±1.4	34.5±7.7	74.3±8.1	70.3±7.9	66.6±5.3	69.0±3.8	76.8±5.4	**92.9±0.6**
BBCNews	84.6±1.3	33.0±0.2	32.8±0.6	84.7±3.8	86.8±2.4	83.1±4.1	79.9±4.4	88.0±1.6	**93.7±0.2**
BBCSport	89.7±1.8	32.0±7.7	35.6±0.4	93.4±2.8	95.1±3.6	78.1±7.5	80.5±6.0	95.3±3.4	**98.9±0.1**
Youtube	39.0±1.3	37.2±2.4	31.2±3.4	32.4±7.3	37.7±2.3	36.0±2.3	47.0±2.2	48.8±1.4	**71.0±0.2**

- *Youtube*[6] is a video dataset described by visual features and audio features, including Cuboid Histogram, Motion Estimate Histogram, HOG, Mel Frequency Cepstral Coefficent (MFCC), Volume Streams, and Spectrogram Streams. Each feature is regarded as one view.

Baselines. In multi-view clustering, our model is compared with the following nine state-of-the-art algorithms: SwMC, MLAN, GMC, MCGC, CDMGC, BMVC, OPLFMC, DSRL and EOMSC. In multi-view classification, our model is compared with the following eight classic or state-of-the-art algorithms: KNN, SVM, AdaBoost, AMGL, MLAN, MVAR, TMC and DSRL. All algorithms have been described in Section 2.

Settings. In both multi-view clustering and classification, our method retains the same settings on each dataset. Specifically, the number of blocks is set to 5 and the initial λ is set to 20. The balance coefficients $\alpha = 1$ and $\beta = 3$. The initial learning rate is set to 0.3, and the learning rate decays every two epochs with a decay rate of 0.99. Full training

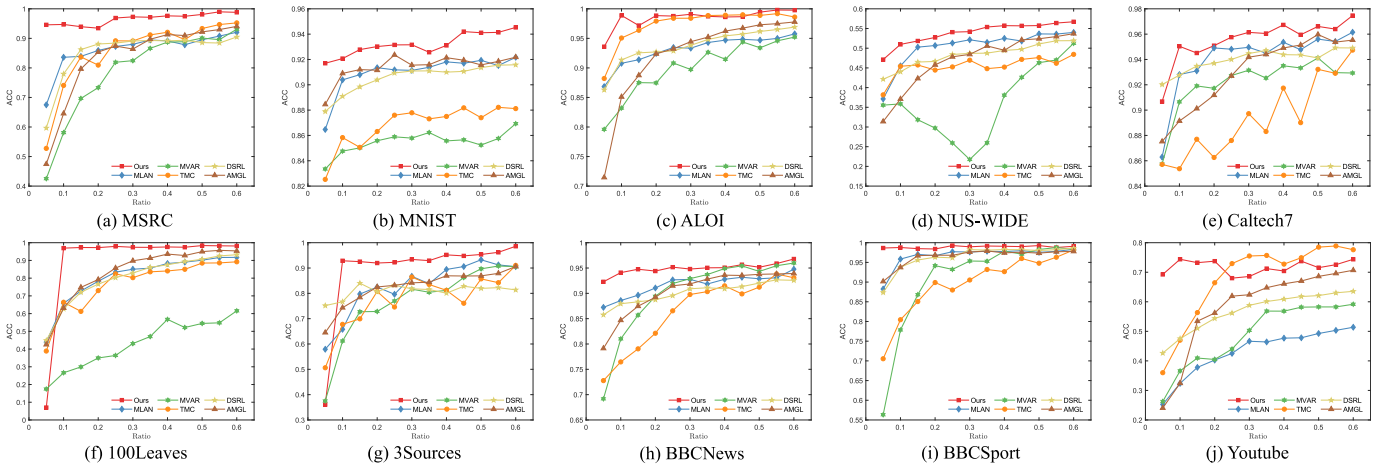

Fig. 7. Accuracy variation trend of multi-view classification algorithms under different proportions of labeled data.

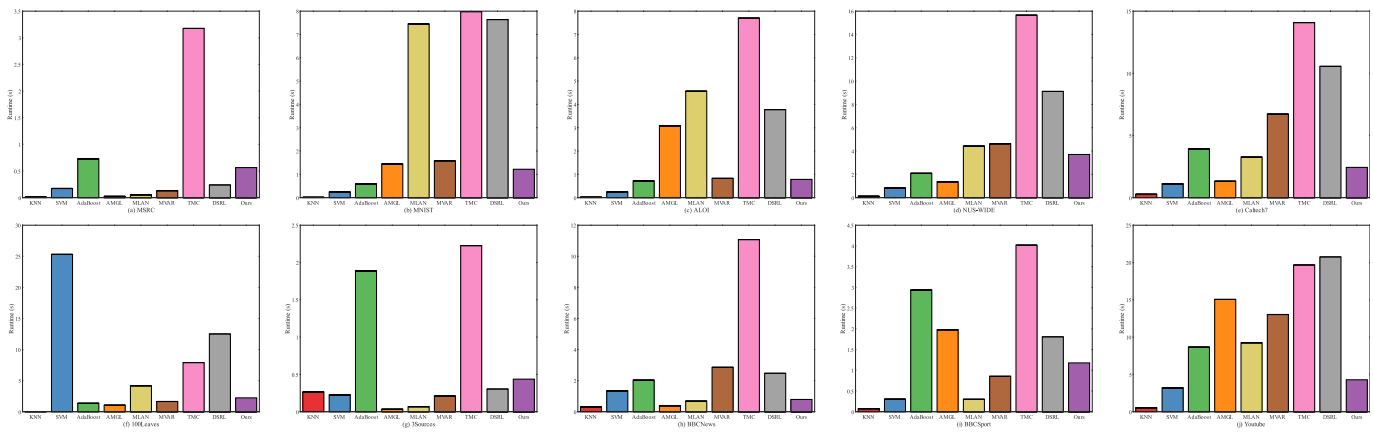

Fig. 8. Runtime for multi-view classification algorithm comparison.

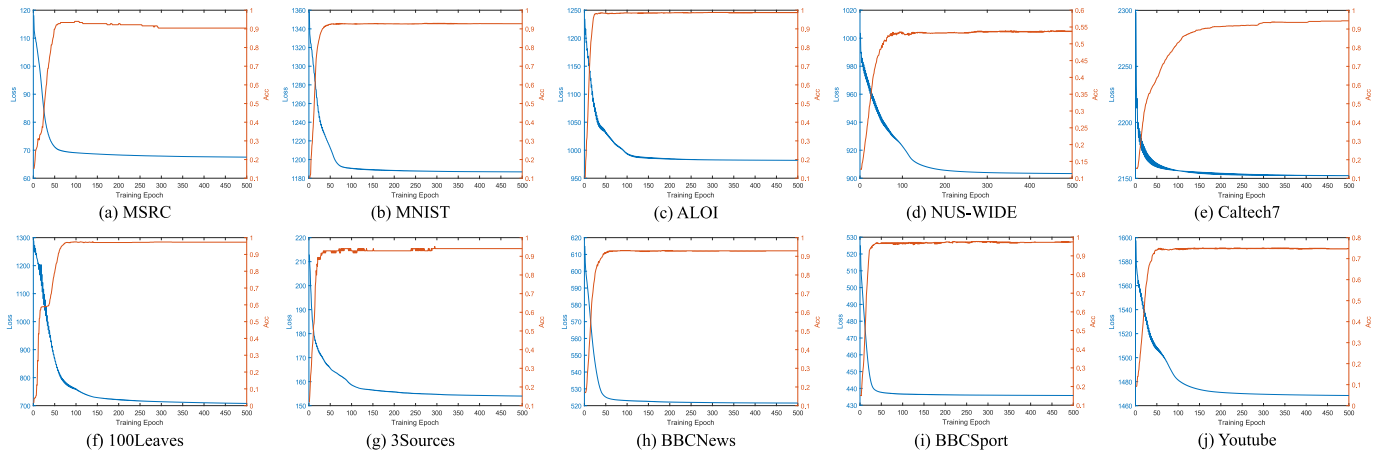

Fig. 9. Convergence curves of our model during training for multi-view classification.

requires 500 epochs with the Adam optimizer. The source code of all comparison algorithms comes from the release of the corresponding authors, and their parameters are set or searched according to the corresponding papers.

4.2 Multi-View Clustering

In multi-view clustering, the experimental results are elaborated in four parts: performance, runtime, convergence and parameter sensitivity.

Performance. Our method is evaluated against 9 state-of-the-art algorithms, by Accuracy (ACC), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). As shown in Table 2, compared to other algorithms, our method almost always has better performance, or even a significant improvement. Although the ACC and NMI of our algorithm on NUS-WIDE are slightly lower than DSRL, it is still significantly better than others.

Runtime. The running time of all multi-view clustering algorithms is shown in Fig. 4. Compared to other methods,

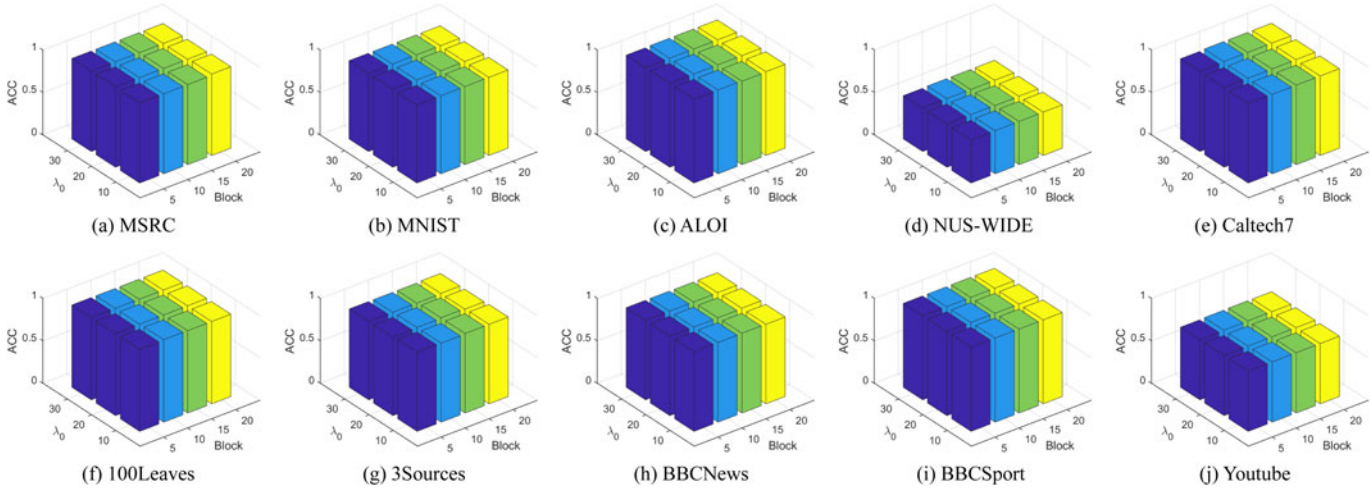

Fig. 10. Parameter sensitivity of our model for multi-view classification.

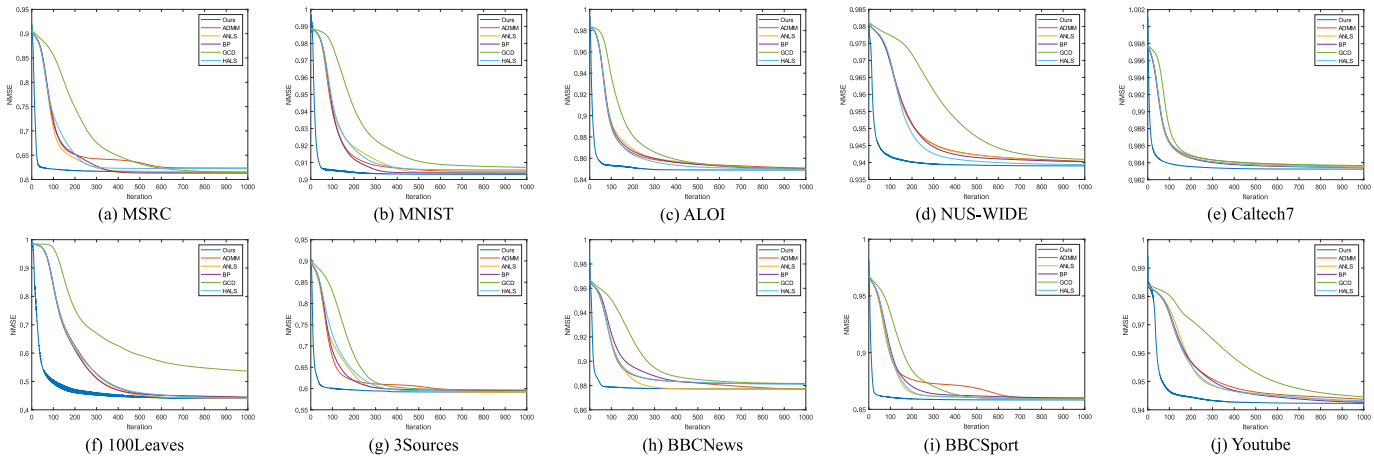

Fig. 11. Convergence curves of various algorithms for solving SymNMF problem.

our method not only has shorter running time than deep learning methods, but also faster than traditional methods on multiple datasets.

Convergence. The convergence curves of our method are depicted in Fig. 5. It can be seen that as the number of iterations increases, not only does our loss gradually decrease, but the accuracy also increases steadily. This shows that our algorithm is convergent and has excellent stability during training.

Parameter Sensitivity. Fig. 6 shows that our algorithm is insensitive to parameters, which means that it is stable and does not depend on precise parameter tuning to achieve a good performance.

4.3 Multi-View Classification

In multi-view classification, the experimental results are also elaborated in four parts: performance, runtime, convergence and parameter sensitivity.

Performance. Our method is evaluated against 3 classic and 5 state-of-the-art algorithms, by Accuracy (ACC). As shown in Table 3, compared to other algorithms, our method always achieves better performance, or even a significant boost. Further, take 5%, 10%,..., 60% of the labeled data to observe the accuracy change trend of all algorithms, and the results are shown in Fig. 7. It can be found that when the labeled data is only 10% or even 5%, our method not only achieves better performance than other algorithms, but also achieves performance close to 50%-60% of the labeled data. This shows that our dependence on the number of labels is lower, and it is more in line with the situation of a small number of labels in reality.

Runtime. The running time of all multi-view classification algorithms is shown in Fig. 8. Similar to the case of multi-view clustering, our model is not only faster than deep learning ones, but also competitive with traditional ones on multiple datasets.

Convergence. The convergence curves of our model are depicted in Fig. 9. It can be seen that as the number of iterations increases, not only does our loss gradually decrease, but the accuracy also increases steadily. Compared with the case of multi-view clustering, our model also converges faster and more stably in multi-view classification.

Parameter Sensitivity. It can be seen from Fig. 10 that the parameter sensitivity in multi-view classification is roughly the same as in multi-view clustering. Obviously, our algorithm is parameter-insensitive, implying its stability and low dependence on precise parameter tuning.

4.4 Ablation Study

In the ablation study, our model is modified variously to verify the effectiveness of each component. There are different variants as follows.

TABLE 4
Accuracy of Our Model Variants on Multiple Datasets

Dataset	S1	S2	S3	S4	S5	S6	S7
MSRC	**92.0**	32.9	49.5	84.8	89.5	81.9	88.1
MNIST	**91.7**	16.9	58.5	66.4	91.0	84.6	86.8
BBCNews	**93.1**	31.4	35.2	89.5	90.1	83.2	84.8
Youtube	**47.8**	13.2	18.2	20.6	39.4	37.9	36.8
NUS-WIDE	**39.4**	16.4	17.5	29.5	34.8	35.2	35.8
ALOI	**97.9**	16.5	41.9	67.6	91.7	78.0	84.8
Caltech7	**84.5**	54.3	26.0	53.3	36.5	56.2	57.0
3Sources	**80.0**	32.5	50.3	76.9	76.9	66.3	76.9
BBCSport	**98.6**	26.7	53.5	84.2	98.0	83.3	98.2
100Leaves	**92.1**	18.9	14.9	81.3	86.1	85.5	86.6

- S1: full model without modification.
- S2: keep $\sum_{i=1}^{n_v}(\mathbf{W}^{(i)}+\lambda_i\mathbf{I})\mathbf{G}^{(i)}$ but parameterize $[\sum_{i=1}^{n_v}(\mathbf{G}^{(i)T}\mathbf{G}^{(i)}+\lambda_i\mathbf{I})]^{-1}$.
- S3: remove L_2 loss.
- S4: remove L_1 loss.
- S5: construct $\mathbf{W}$ using $\mathbf{W}=\sum_{i=1}^{n_v}\mathbf{W}^{(i)}/n_v$.
- S6: perform spectral clustering directly on $\mathbf{W}$.
- S7: solve by traditional optimization.

As can be seen from Table 4, the absence of any component leads to a drop in performance, indicating that each component is necessary. It is worth mentioning that the sharply declining performance in variant S2 also reflects the correctness of our previous parameterization of the first half.

When degenerating to a single view, our model becomes a symmetric non-negative matrix factorization problem, which can be solved by a variety of traditional algorithms such as Alternating Direction Method of Multipliers (ADMM) [50], Alternating Nonnegative Least Squares (ANLS), Block Principal Pivoting (BPP) [51], [52], Greedy Coordinate Descent (GCD) [53], and Hierarchical Alternating Least Squares (HALS) [54]. Further, our model is compared with these traditional optimization algorithms. As can be seen from Fig. 11, our model not only requires fewer iterations, but also generally has lower Normalized Mean Square Error (NMSE).

5 Conclusion

In this paper, inspired by traditional optimization and deep learning, we propose a differentiable multi-view spectral clustering with flexible extension. The core idea is to partially parameterize the traditional optimization process and then guide the learning through various loss functions. Clearly, our model absorbs the strengths of both, such as flexibility and stability. Theoretical and experimental analyses demonstrate that our method not only achieves competitive performance, but also enjoys stable training with fewer iterations. Furthermore, in semi-supervised classification tasks, our model has lower dependence on the number of labels. However, interpretability and flexibility are our greatest strengths. This allows the output reliability to be assessed and the model to be easily extended for different problems. In the future, we plan to further explore improvements and applications of this model.

References

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

[2] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[3] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[4] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.

[5] S. Wang, A. Gittens, and M. W. Mahoney, "Scalable kernel k-means clustering with nyström approximation: Relative-error bounds," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 431–479, 2019.

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

[7] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 23–32.

[8] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 8547–8555.

[9] H. Zhong et al., "Graph contrastive clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9224–9233.

[10] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 11, no. 9, pp. 1074–1085, Sep. 1992.

[11] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[12] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *Proc. IEEE Int. Conf. Data Mining*, 2001, pp. 107–114.

[13] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 977–986.

[14] F. Nie, X. Wang, M. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1969–1976.

[15] S. Kumar, J. Ying, J. V. de Miranda Cardoso, and D. P. Palomar, "A unified framework for structured graph learning via spectral constraints," *J. Mach. Learn. Res.*, vol. 21, no. 22, pp. 1–60, 2020.

[16] J. Yang, J. Liang, K. Wang, P. L. Rosin, and M.-H. Yang, "Subspace clustering via good neighbors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1537–1544, Jun. 2020.

[17] X. Liu, "SimpleMKKM: Simple multiple kernel k-means," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 16, 2022, doi: 10.1109/TPAMI.2022.3198638.

[18] X. Liu et al., "Localized simple multiple kernel k-means," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9293–9301.

[19] X. Liu, "Incomplete multiple kernel alignment maximization for clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Oct. 01, 2021, doi: 10.1109/TPAMI.2021.3116948.

[20] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2564–2570.

[21] X. Li, M. Chen, F. Nie, and Q. Wang, "A multiview-based parameter free framework for group detection," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4147–4153.

[22] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for image clustering and semi-supervised classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1501–1511, Mar. 2018.

[23] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, Jun. 2020.

[24] 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.

[25] S. Huang, I. Tsang, Z. Xu, and J. C. Lv, "Measuring diversity in graph learning: A unified framework for structured multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5869–5883, Dec. 2022.

[26] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multiview clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

[27] X. Liu et al., "One pass late fusion multi-view clustering," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 6850–6859.
[28] 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.*, vol. 44, no. 9, pp. 5042–5055, Sep. 2022.
[29] S. Liu et al., "Efficient one-pass multi-view subspace clustering with consensus anchors," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 7576–7584.
[30] F. Nie, J. Li, and X. Li, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1881–1887.
[31] H. Tao, C. Hou, F. Nie, J. Zhu, and D. Yi, "Scalable multi-view semi-supervised classification via adaptive regression," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4283–4296, Sep. 2017.
[32] Z. Han, C. Zhang, H. Fu, and J. T. Zhou, "Trusted multi-view classification with dynamic evidential fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 03, 2022, doi: 10.1109/TPAMI.2022.3171983.
[33] D. Matthew Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[34] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4829–4837.
[35] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6541–6549.
[36] B. Kim, R. Khanna, and O. O. Koyejo, "Examples are not enough, learn to criticize! criticism for interpretability," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2288–2296.
[37] W. Pang Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1885–1894.
[38] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019.
[39] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.
[40] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Learning efficient sparse and low rank models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1821–1833, Sep. 2015.
[41] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9061–9071.
[42] J. Liu and X. Chen, "ALISTA: Analytic weights are as good as learned weights in LISTA," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–33.
[43] J. Sun et al., "Deep ADMM-Net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 10–18.
[44] X. Xie, J. Wu, G. Liu, Z. Zhong, and Z. Lin, "Differentiable linearized ADMM," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6902–6911.
[45] J. T. Zhou et al., "SC2Net: Sparse LSTMs for sparse coding," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4588–4595.
[46] Z. Wang, S. Chang, J. Zhou, M. Wang, and T. S. Huang, "Learning a task-specific deep architecture for clustering," in *Proc. Int. Conf. Data Mining*, 2016, pp. 369–377.
[47] X. Peng, Y. Li, I. W. Tsang, H. Zhu, J. Lv, and J. T. Zhou, "XAI beyond classification: Interpretable neural clustering," *J. Mach. Learn. Res.*, vol. 23, pp. 1–28, 2022.
[48] D. Chris, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proc. SIAM Int. Conf. Data Mining*, 2005, pp. 606–610.
[49] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation. III," *SIAM J. Numer. Anal.*, vol. 7, no. 1, pp. 1–46, 1970.
[50] S. Lu, M. Hong, and Z. Wang, "A nonconvex splitting method for symmetric nonnegative matrix factorization: Convergence analysis and optimality," *IEEE Trans. Signal Process.*, vol. 65, no. 12, pp. 3120–3135, Jun. 2017.
[51] J. Kim and H. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *Proc. IEEE 8th Int. Conf. Data Mining*, 2008, pp. 353–362.
[52] D. Kuang, S. Yun, and H. Park, "SymNMF: Nonnegative low-rank approximation of a similarity matrix for graph clustering," *J. Glob. Optim.*, vol. 62, no. 3, pp. 545–574, 2015.
[53] C.-J. Hsieh and I. S. Dhillon, "Fast coordinate descent methods with variable selection for non-negative matrix factorization," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2011, pp. 1064–1072.
[54] Z. Zhu, X. Li, K. Liu, and Q. Li, "Dropping symmetry for fast symmetric nonnegative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5160–5170.

Zhoumin Lu received the MS degree in computer technology from Fuzhou University, China, in 2021. He is currently working toward the PhD degree with the School of Computer Science, Northwestern Polytechnical University. His research interests include machine learning, deep learning and their applications, such as pattern recognition and data mining.

Feiping Nie (Senior Member, IEEE) received the PhD degree in computer science from Tsinghua University, China, in 2009, and is currently a full professor with Northwestern Polytechnical University, China. His research interests are machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing and information retrieval. He has published more than 100 papers in the following journals and conferences: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *International Journal of Computer Vision*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Knowledge and Data Engineering*, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, ACM MM. His papers have been cited more than 20000 times and the H-index is 84. He is now serving as associate editor or PC member for several prestigious journals and conferences in the related fields.

Rong Wang received the BS degree in information engineering, the MS degree in signal and information processing, and the PhD degree in computer science from Xi'an Research Institute of Hi-Tech, Xi'an, China, in 2004, 2007 and 2013, respectively. During 2007 and 2013, he also received the PhD degree from the Department of Automation, Tsinghua University, Beijing, China. He is currently an associate professor with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests focus on machine learning and its applications.

Xuelong Li (Fellow, IEEE) is a full professor with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China.