Interpretable Subspace Clustering

Zheng Zhang, Peng Zhou, *Senior Member, IEEE,* Aiting Yao, Liang Du, Xinwang Liu, *Senior Member, IEEE*

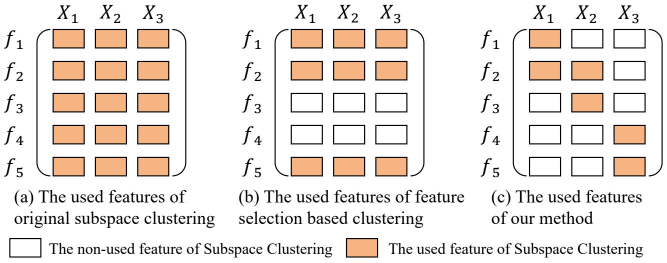
***Abstract*—Subspace clustering is one of the most popular clustering methods due to its effectiveness. Although subspace clustering methods have been demonstrated to achieve promising performance, they still lack interpretability, especially when handling high-dimensional complicated data. To bridge this gap, this paper focuses on the interpretability of subspace clustering and proposes a novel interpretable subspace clustering method. Our goal is to answer two key questions about the interpretability in subspace clustering: (1) when handling an individual sample, which features should work for this sample? (2) Which cluster or subspace will the features that work put this sample into? To answer these two questions, we design two new interpretability regularized terms and plug them into the subspace clustering. In this way, we show that interpretability can be used to improve the clustering performance in turn. Extensive experiments on benchmark data sets demonstrate the effectiveness of our method in terms of clustering performance and interpretability.**

***Index Terms*—Machine learning, interpretability, subspace clustering**

I. Introduction

CLUSTERING is a fundamental unsupervised task in machine learning. It aims to partition the data samples into several clusters, where the data in the same cluster are close to each other, and the data in different clusters are far apart from each other. Since clustering does not need human annotations, it has been widely used in many machine learning applications, such as recommendation systems [1], [2], data mining [3], [4], and image segmentation [5], [6].

In recent decades, many clustering methods have been proposed, such as k-means based methods [7], [8], [9], [10], graph based methods [11], [12], [13], [14], [15], density based methods [16], [17], and subspace based methods [18], [19]. Among them, subspace clustering can discover the intrinsic subspace structure of non-linear complicated data, and thus has attracted much attention [18], [19], [20], [4], [21], [22]. For example, Sparse Subspace Clustering (SSC) [23] pointed out that each sample can be represented by a few number of samples from the same subspace, thereby proposing a sparse subspace clustering methodology. LRR [19] achieved robust subspace clustering by seeking the low-rank representation that can recover underlying subspace structures and detect outliers under various data corruption scenarios. LPP [24] proposed a projection-based subspace clustering method that learned a mapping matrix to transform the original input space into a lower-dimensional subspace. These methods learn the relationships among data samples and partition the data into several subspaces based on these relations, obtaining the final clustering results from the subspaces. Although the subspace clustering methods have demonstrated promising performance, they may lack interpretability, especially when handling high-dimensional complicated data. In high-dimensional data, the relations between samples are complicated and difficult to recognize. The reason why two samples of this complicated high-dimensional data belong to the same subspace is not obvious, and existing subspace clustering methods fail to provide an explanation.


X_1 X_2 X_3
f_1
f_2
f_3
f_4
f_5
The non-used feature of Subspace Clustering
The used feature of Subspace Clustering

(a) The used features of original subspace clustering

(b) The used features of feature selection based clustering

(c) The used features of our method

Fig. 1: Explanation of differences among subspace clustering, feature selection-based clustering, and our method. Each column (i.e., X_1, X_2, X_3) denotes a sample, and each row ($f_1, \cdots, f_5$) represents a feature. (a) shows the features used in conventional subspace clustering methods. Conventional subspace clustering uses all features for clustering. (b) shows the features used in feature selection based methods. These methods use the same features for all samples for clustering. (c) shows the features used in our method. For different samples, we use different features for clustering, which can explain why we make such a prediction on an individual.

To explain the subspace clustering or to tell the reason why two samples belong to the same subspace, in this paper, we dive into the interpretable subspace clustering task. Given a data set, especially the complicated high-dimensional data, our interpretable method should answer the following two key questions: *1) when handling an individual sample, which features should work for this sample? 2) Which cluster or subspace will the features that work put this sample into?* If we can answer these two questions, we can tell why the two samples are put into the same cluster.

Therefore, the key point of our interpretable subspace clustering is the features. One natural idea is to apply the feature selection methods [25], [26], [27], [28], [29], [30]. However, our task is different from feature selection. Conventional feature selection methods try to select several informative

Zheng Zhang and Peng Zhou are with Anhui Provincial International Joint Research Center for Advanced Technology in Medical Imaging, School of Computer Science and Technology, Anhui University, Hefei, 230601, China. Email: $zheng-zhang@stu.ahu.edu.cn$, $zhoupeng@ahu.edu.cn$. Peng Zhou is the corresponding author.

Aiting Yao is with the Department of New Networks, Peng Cheng Laboratory, Shenzhen, 518000, China. Email: $yaoat@pcl.ac.cn$

Liang Du is with School of Computer and Information Technology, Shanxi University, Taiyuan, 030006, China. Email: $duliang@sxu.edu.cn$

Xinwang Liu is with the College of Computer, National University of Defense Technology, Changsha, 410073, China. Email: $xinwangliu@nudt.edu.cn$.

features for *all* data, which means no matter which sample is handled, the features that work are exactly the same [31], [32]. Therefore, it still fails to answer Question 1, because if the features that work for all data are the same, we cannot tell why we put some samples into one cluster but put other samples into another cluster. Fig. 1 shows an example of feature selection and our task. In Fig. 1, a column denotes a sample and a row denotes a feature. The yellow block means the feature of the data that is used in clustering. Fig. 1(a) shows the features used in traditional subspace clustering, i.e., all features of all data are used. It is uninterpretable because it cannot tell why a sample belongs to a cluster. Fig. 1(b) shows the features used in clustering with feature selection. We can see that, for all data, it uses the same features. It can tell which features are important for clustering, but it still cannot tell why we put a sample into a cluster but put another sample into another cluster. Fig. 1(c) shows the features used in our method. It can be seen that, for different samples, the features that work are also different, which can answer Question 1 that for each individual, which features should work. Notice that, to answer Question 1, one of the most recent techniques is the local feature selection, which tries to select features for each individual instead of for all data [33], [34]. However, local feature selection can only answer Question 1 but fails to answer Question 2. To answer Question 2, we should dive into the subspaces and try to capture the relations between the feature and subspace for an individual sample.

To this end, in this paper, we propose a novel Interpretable Subspace Clustering (ISC) method. The basic idea is that during the subspace clustering, we also learn an indicator matrix to show which features should be used for each sample. Notice that for each individual, only a few features work. Therefore, the indicator matrix in our method should be sparse for each sample. To this end, we use a sparse regularized term to control the sparsity for each sample. With this regularized term, we can answer Question 1. To answer Question 2, we carefully design another novel regularized term to characterize the relations between features and subspaces. Thanks to this regularized term, we can easily tell, given an individual, which subspace the selected features will put the data into. Intuitively, if we know an algorithm better, we can improve it more easily. To show whether the interpretability can improve or guide the subspace clustering, we plug these two regularized terms into the subspace clustering framework, leading to the objective function of our interpretable subspace clustering method. Then, we provide an iterative optimization method to solve the introduced objective function. At last, extensive experiments are conducted to show the effectiveness and interpretability of the proposed method.

The main contributions of this paper are summarized as follows:

- We propose a novel interpretable subspace clustering method that can answer the two key questions about the interpretability. To the best of our knowledge, this is the first work to achieve interpretability in subspace clustering.
- We design a novel regularized term to capture the relations between features and subspaces, which makes the decisions interpretable. Although the motivation of the regularized term is interpretability, we show that it can also guide the subspace clustering in turn.
- Extensive experiments demonstrate that our method has comparable or even better clustering performance compared to state-of-the-art subspace clustering methods, but has better interpretability.

II. Related Work

In this section, we introduce related work on subspace clustering and interpretable clustering. In the following, we use bold uppercase and lowercase characters to denote a matrix and a vector, respectively. For an arbitrary matrix $\mathbf{A}$, we denote its (i,j)-th element as A_{ij}, the k-th column vector and k-th row vector of $\mathbf{A}$ is $\mathbf{A}_{:,k}$ and $\mathbf{A}_{k,:}$, respectively.

A. Subspace Clustering

Over the past few decades, subspace clustering has witnessed significant advancements [23], [19], [20], [35], [18], [22], [4]. Sparse Subspace Clustering (SSC) [23] stands as one of the most important self-representation methods, which aims to learn a coefficient matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ by selecting the fewest possible samples to represent each data point. Specifically, given a data matrix $\mathbf{X} \in \mathbb{R}^{d\times N}$ with N samples and d features, the SSC's objective function is formulated as follows:

$$\begin{aligned} \min_{\mathbf{S}} \quad & \|\mathbf{S}\|_1 + \lambda\|\mathbf{X} - \mathbf{XS}\|_F^2 \\ \text{s.t.} \quad & \mathbf{S}^T\mathbf{1} = \mathbf{1}, \quad \operatorname{diag}(\mathbf{S}) = \mathbf{0}. \end{aligned} \tag{1}$$

where $\operatorname{diag}(\mathbf{S})$ indicates the diagonal vector of the matrix $\mathbf{S}$, and $\mathbf{1}$ denotes the vector whose elements are all 1's. The first term is to control the sparsity of $\mathbf{S}$, which means we only need a few samples (not all samples) to reconstruct each data. The second term is the reconstruction term, which reconstructs each data point with the coefficients in $\mathbf{S}$. The first constraint makes sure that the reconstruction coefficients of each sample sum to 1. The second constraint is to avoid the trivial solution that each sample is reconstructed by itself.

After obtaining the coefficient matrix $\mathbf{S}$ by optimizing Eq.(1), the subspaces are partitioned from $\mathbf{S}$. Specifically, they construct the sparse similarity matrix $\tilde{\mathbf{S}} = (\mathbf{S} + \mathbf{S}^T)/2$, and then run spectral clustering on $\tilde{\mathbf{S}}$ to obtain several clusters where each cluster represents a subspace.

Following this framework, many variants of subspace clustering have been proposed. For example, low-rank-based approaches were developed to more effectively uncover the intrinsic structure of subspaces [19], [20], [36]. Greedy Subspace Clustering (GSC) [22] employed a greedy search strategy to identify nearest subspace neighbors, thereby recovering underlying subspace structures. Orthogonal Matching Pursuit (OMP) [35] addressed the computational overhead of traditional SSC through iterative orthogonal projection. Zhou et al. [4] proposed a projected subspace clustering method that jointly optimizes local subspace structures and global structures. Most recently, some works tried to extend the subspace clustering to deep learning, leading to the deep subspace clustering. For example, Cai et al. [37] proposed a novel deep subspace clustering method out of the self-expressive framework, which applied an iterative refining method to learn the subspace, and the refined subspace bases helped the representation learning in turn. Furthermore, Cai et al. [38] designed a deep clustering method in a generative way. The method learned a Wasserstein embedding for clustering by minimizing the Wasserstein distance between the prior and marginal distribution in the embedding space instead of the original data space, which conventional methods often do. In [39], Yu et al. proposed a new deep subspace clustering method with contrastive learning and applied it to multi-view clustering. It applied the Cauchy-Schwarz divergence to characterize the interaction of the representation and clustering, which improved the encoder's feature extraction capacity.

Although these existing methods have achieved promising performance, they lack interpretability as introduced before. This is the main motivation of our interpretable subspace clustering.

B. Interpretable Clustering and Local Feature Selection

Existing interpretable clustering techniques can be roughly categorized into four types: decision tree based, rule based, prototype based, and feature selection based interpretable clustering methods.

The core idea of decision tree based methods is to construct a decision tree using pseudo-labels (generated by clustering algorithms like k-means or spectral clustering) and the original data points, aiming to minimize clustering error and reduce tree complexity simultaneously [40], [41], [42], [43], [44], [45]. Moshkovitz et al. [40], [41] were the first to propose a top-down decision tree algorithm specifically designed for k-means clustering. Then, Fleissner et al. [44] argued that the aforementioned methods failed to address nonlinear data problems. Then, they proposed a novel decision tree based clustering approach operating in kernel space. Rule based clustering methods aim to discover a set of if-$then$ rules from data for clustering purposes. The interpretability of a clustering algorithm's decision process can be achieved by identifying and representing the rule set that characterizes each cluster [46], [47], [48], [49]. For example, Saisubramanian et al. [46] proposed a method that predefined a set of feature points of interest as rule conditions, requiring samples within each cluster to maximally satisfy these prescribed rules. The proportion of rule-satisfying samples within each cluster served as an interpretability metric. Wang et al. [47] designed a rapid fuzzy rule clustering method based on granular computing. The core objective of prototype based interpretable clustering is to discover representative exemplars that characterize each cluster's structure [50], [51], [52], [53]. For example, Davidson et al. [50] developed an approximation algorithm that balanced the number of selected exemplars against clustering performance by simultaneously maximizing cluster diameters while minimizing the number of exemplars. Pan et al. [51] proposed a deep prototype-based interpretable clustering framework by embedding the Centroid-Integration Unit (CIUnit) within the autoencoder network.

Feature selection based approaches are another popular interpretable clustering method. Notice that, conventional feature selection approaches determine a global feature subset shared across all samples by focusing only on the most relevant subset for all data, such as [25], [26], [4], [28], [30]. However, since the above methods select the same features for all samples to cluster, it still lacks interpretability [33], [34]. To tackle this issue, Armanfard et al. [33] first proposed a local feature selection method for the classification task that allows each sample to select its own features. After that, several local feature selection methods have been proposed. For example, INTERPRET3C [54] introduced sample-wise adaptive feature selection via a Gumbel-Softmax-based gating network with sparsity-inducing regularization, enabling interpretable clustering on the filtered features. The IDC [34] framework introduced a dual-gating mechanism consisting of local feature selection, optimized with sparsity-promoting and orthogonality constraints.

In this paper, we also focus on the feature selection based method. We design new interpretability regularized terms based on the local feature selection for subspace clustering, leading to interpretable subspace clustering.

III. Method

In this section, we introduce our ISC method in detail. Given a data set $\mathbf{X} \in \mathbb{R}^{d\times N}$ containing N samples and d features, we aim to partition the samples into C clusters.

Following the framework of subspace clustering Eq.(1), we also apply a self-reconstructive approach, which reconstructs $\mathbf{X}$ with a coefficient matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$. However, different from conventional subspace clustering methods, our interpretable method tries to find out which features work for each sample. To this end, we introduce a learnable sample-feature indicator matrix $\mathbf{W} \in \{0,1\}^{d\times N}$. If the i-th feature of the j-th sample works, $W_{ij} = 1$ and $W_{ij} = 0$ otherwise.

Equipped with $\mathbf{W}$, in the self-reconstruction, we wish to use samples with their worked features to reconstruct the original data. Therefore, we adjust Eq.(1) as follows:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{W}} \quad & ||\mathbf{S}||_1 + \lambda||\mathbf{X} - (\mathbf{X} \odot \mathbf{W})\mathbf{S}||_F^2 \\ \text{s.t.} \quad & \mathbf{W} \in \{0,1\}^{d\times N}, \mathbf{S}^T\mathbf{1} = \mathbf{1}, \text{diag}(\mathbf{S}) = 0, \end{aligned} \tag{2}$$

where $\odot$ denotes the Hadamard product, i.e., the element-wise product.

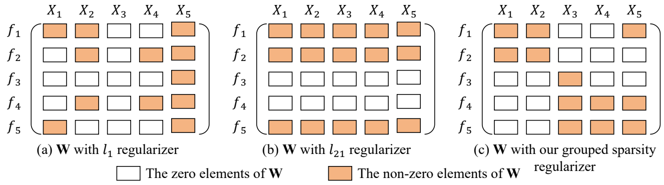
$\mathbf{W}$ is the key to our interpretability. As introduced before, to realize the interpretability, we should learn $\mathbf{W}$ to answer the following two questions:

- Q1. When handling an individual sample, which features should work for this sample?
- Q2. Which cluster or subspace will the features that work put this sample into?

In the following, we will introduce our formulation to answer these two questions.

A. Q1: Which Features should Work for a Sample?

To answer Question 1, we need to enforce sparsity on $\mathbf{W}$. Notice that $\mathbf{W} \in \{0,1\}^{d\times N}$ is discrete and thus difficult to optimize. To this end, we relax $\mathbf{W}$ to $\mathbf{W} \in [0,1]^{d\times N}$.


X_1 X_2 X_3 X_4 X_5
f_1
f_2
f_3
f_4
f_5
□ The zero elements of **W**
▨ The non-zero elements of **W**

(a) **W** with l_1 regularizer (b) **W** with l_{21} regularizer (c) **W** with our grouped sparsity regularizer

Fig. 2: An illustration of the comparison of the ℓ_1 regularizer, ℓ_{21} regularizer, and our grouped sparsity regularizer.

However, this relaxation cannot guarantee the sparsity of **W**. To achieve sparsity, one of the simplest ways is to use the ℓ_1 norm on **W**, i.e., $\|\mathbf{W}\|_1$. However, ℓ_1 norm can only enforce **W** to be sparse but cannot control where the zero-elements appear. In our task, given any sample, we wish a few features work. It means that, given any column in **W**, there should be a few non-zero elements (indicating the worked features) and a lot of zero elements (indicating the useless features). Therefore, we do not want to create a column with all-zeros, which means the given sample does not have any worked features. To this end, we wish that in each column, there should be some non-zero elements. The conventional ℓ_1 norm cannot realize it.

In our task, we can regard each column of **W** as a group. The group should have two properties: 1) There are no all-zero groups. 2) Each group is sparse. To this end, inspired by [55], we impose the following group sparsity regularized term on **W**:

$$\|\mathbf{W}\|_G = \sum_{j=1}^{N}\left(\sum_{i=1}^{d} W_{ij}\right)^2. \tag{3}$$

The outer summation is an ℓ_2 norm on $\sum_{i=1}^{d} W_{ij}$. As we know, ℓ_2 norm does not lead to sparse results, and thus each $\sum_{i=1}^{d} W_{ij}$ will be non-zero. It means that, given any sample (e.g., the j-th sample), $\sum_{i=1}^{d} W_{ij} > 0$ indicates that there should be some non-zero W_{ij} and thus for each sample, there should be some features that work.

The inner summation in Eq.(3) is an ℓ_1 norm on W_{ij} (notice that $W_{ij} > 0$ and thus we do not need the absolute value operation on W_{ij}). The ℓ_1 norm can make sure that in the j-th column, some W_{ij} should be zeros, and thus the non-zero elements indicate the features that work for the i-th sample. It is worthy to clarify that this term is different from some widely-used sparse regularized terms, such as the ℓ_1 regularizer $\|\mathbf{W}\|_1 = \sum_{i=1}^{d}\sum_{j=1}^{N}|W_{ij}|$ and the ℓ_{21} regularizer $\|\mathbf{W}\|_{21} = \sum_{i=1}^{d}\sqrt{\sum_{j=1}^{N} W_{ij}^2}$. Fig. 2 shows a simple comparison of these three regularizers. Fig. 2(a) shows the result of the ℓ_1 regularizer. It can impose sparsity on the whole **W** matrix, but it cannot control where the zero elements are. For example, for some instances, e.g. X_3, there are no selected features and for some instances, e.g. X_5, all features are selected. Fig. 2(b) shows the result of ℓ_{21} regularizer. It imposes the ℓ_1 norm on the row, which will lead to the ℓ_2 norm of some rows being zero, and thus it can cause the row sparsity. The row sparsity means that, for all instances, the selected features are the same. Fig. 2(c) shows the sparsity of $\|\cdot\|_G$, where for each instance, there are several selected features and non-selected features.

Therefore, the group sparsity regularized term $\|\mathbf{W}\|_G$ can satisfy the above-mentioned two properties. Then, we plug it into Eq.(2), leading to

$$\begin{aligned} \min_{\mathbf{S},\mathbf{W}} \quad & ||\mathbf{S}||_1 + \lambda||\mathbf{X} - (\mathbf{X}\odot\mathbf{W})\mathbf{S}||_F^2 + \gamma||\mathbf{W}||_G \\ \text{s.t.} \quad & \mathbf{W}\in[0,1]^{d\times N},\ \mathbf{S}^T\mathbf{1} = \mathbf{1},\ \mathrm{diag}(\mathbf{S}) = 0. \end{aligned} \tag{4}$$

B. Q2: Which Subspace will the Worked Features Put a Sample Into?

To answer Question 2, we need to construct the relation between **W** and the subspaces. Therefore, we should first construct the subspaces.

In conventional subspace clustering, they obtain subspaces from **S**, i.e., they run spectral clustering on the similarity matrix $\tilde{\mathbf{S}} = (\mathbf{S}+\mathbf{S}^T)/2$. In more detail, they first construct the Laplacian $\mathbf{L} = \mathbf{D} - \tilde{\mathbf{S}}$, where **D** is a diagonal matrix whose diagonal element $D_{ii} = \sum_{j=1}^{N}\tilde{S}_{ij}$. Then, they obtain the C eigenvectors of **L** corresponding to the smallest C eigenvalues and run k-means on the eigenvectors to obtain the discrete subspace indicator matrix $\mathbf{Y}\in\{0,1\}^{N\times C}$, where if the i-th sample belongs to the j-th subspace, $Y_{ij} = 1$ and $Y_{ij} = 0$ otherwise. Obviously, **Y** can also be regarded as the cluster indicator, and they use **Y** as the final clustering result.

Therefore, in our framework, we need to construct the relation between **W** and **Y**. However, in the above-mentioned spectral clustering procedure, they obtain **Y** in two separate steps: eigenvalue decomposition on **L** and k-means on the eigenvectors. Since we need to plug **Y** into our formulation, we wish to obtain **Y** from **S** by a single term, i.e., we need to obtain **Y** in one step.

To this end, we adjust the spectral clustering to the discrete spectral clustering inspired by the ratio-cut [56]. In the ratio-cut, we first regard $\tilde{\mathbf{S}}$ as an adjacency matrix of a graph, and try to partition the graph into C subgraphs $\{\mathcal{X}^1, \mathcal{X}^2, ..., \mathcal{X}^C\}$ by maximizing the intra-subgraph similarity and minimizing the inter-subgraph similarity:

$$\mathrm{RatioCut}(\mathcal{X}^1, \mathcal{X}^2, ..., \mathcal{X}^C) = \sum_{k=1}^{C}\frac{\mathrm{cut}(\mathcal{X}^k, \bar{\mathcal{X}}^k)}{|\mathcal{X}^k|}, \tag{5}$$

where $\bar{\mathcal{X}}^k$ is the complement of $\mathcal{X}_k$. Here, $\mathrm{cut}(\mathcal{X}^k, \bar{\mathcal{X}}^k) = \sum_{V_i\in\mathcal{X}_k}\sum_{V_j\in\bar{\mathcal{X}}_k}\tilde{S}_{ij}$ is a cut of the graph, i.e., the summation of the edge weights between $\mathcal{X}_k$ and $\bar{\mathcal{X}}_k$. Introducing the diagonal matrix **D** where $D_{ii} = \sum_{j=1}^{N}\tilde{S}_{ij}$ and a one-hot indicator vector $\mathbf{y}_k\in\mathbb{R}^N$ where the i-th element in $\mathbf{y}_k$ is 1 only if the i-th vertice V_i belongs to $\mathcal{X}^k$, we can obtain:

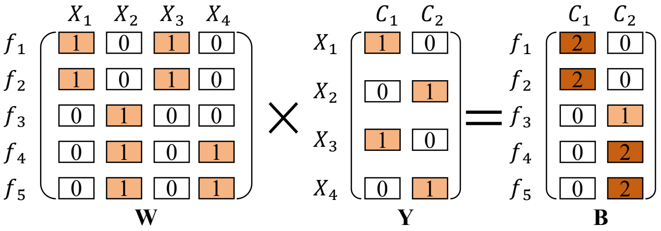
$$\begin{aligned} \mathrm{cut}(\mathcal{X}^k, \bar{\mathcal{X}}^k) &= \sum_{V_i\in\mathcal{X}^k} D_{ii} - \mathrm{cut}(\mathcal{X}^k, \mathcal{X}^k) \\ &= \mathbf{y}_k^T\mathbf{D}\mathbf{y}_k - \mathbf{y}_k^T\tilde{\mathbf{S}}\mathbf{y}_k \\ &= \mathbf{y}_k^T\mathbf{L}\mathbf{y}_k, \end{aligned} \tag{6}$$

where $\mathbf{L} = \mathbf{D} - \tilde{\mathbf{S}}$ is the Laplacian matrix. Take it into Eq.(5), we have:

$$\begin{aligned}\mathrm{RatioCut}(\mathcal{X}^1, \mathcal{X}^2, ..., \mathcal{X}^C) &= \sum_{k=1}^{C} \frac{\mathbf{y}_k^T \mathbf{L} \mathbf{y}_k}{\mathbf{y}_k^T \mathbf{y}_k} \\ &= \mathrm{tr}(\mathbf{Y}^T \mathbf{L} \mathbf{Y} (\mathbf{Y}^T \mathbf{Y})^{-1})\end{aligned} \tag{7}$$

Therefore, we can obtain $\mathbf{Y}$ from $\mathbf{S}$ by minimizing the term $\mathrm{tr}(\mathbf{Y}^T \mathbf{L} \mathbf{Y} (\mathbf{Y}^T \mathbf{Y})^{-1})$.

Now, we have obtained the subspace $\mathbf{Y}$ and will construct the relation between $\mathbf{Y}$ and $\mathbf{W}$. To answer Question 2, we should find out which features work for a given subspace.



$$\begin{array}{c|cccc} & X_1 & X_2 & X_3 & X_4 \\ \hline f_1 & 1 & 0 & 1 & 0 \\ f_2 & 1 & 0 & 1 & 0 \\ f_3 & 0 & 1 & 0 & 0 \\ f_4 & 0 & 1 & 0 & 1 \\ f_5 & 0 & 1 & 0 & 1 \end{array} \times \begin{array}{c|cc} & C_1 & C_2 \\ \hline X_1 & 1 & 0 \\ X_2 & 0 & 1 \\ X_3 & 1 & 0 \\ X_4 & 0 & 1 \end{array} = \begin{array}{c|cc} & C_1 & C_2 \\ \hline f_1 & 2 & 0 \\ f_2 & 2 & 0 \\ f_3 & 0 & 1 \\ f_4 & 0 & 2 \\ f_5 & 0 & 2 \end{array}$$

Fig. 3: A toy example of the $\mathbf{WY} = \mathbf{B}$ with 4 samples (X_1, X_2, X_3, X_4) and 5 features $(f_1, f_2, f_3, f_4, f_5)$ in 2 subspace (C_1, C_2). Matrix $\mathbf{B}$ shows the relevance between the features and the subspace. For example, X_1 and X_3 belong to the first subspace, and in X_1 and X_3, the worked features are both f_1 and f_2. Therefore, features f_1 and f_2 are highly related to subspace C_1. Correspondingly, B_{11} and B_{21}, which shows the relevance of the features f_1 and f_2 to subspace C_1 respectively, have a relatively large value.

To this end, we compute a matrix $\mathbf{B} = \mathbf{WY} \in \mathbb{R}^{d \times C}$. B_{ij} is the correlation of the i-th feature and j-th subspace. Fig. 3 shows a toy example of 4 samples with 5 features in 2 subspaces. The larger B_{ij} is, the more relative the i-th feature and j-th subspace is. $B_{ij} = 0$ means that the i-th feature does not work for the j-th subspace. In this example, we can see that the first two features relate to the first subspace, and the last two features relate to the second subspace. The third feature may also relate to the second subspace, but the relation is not as strong as the last two features.

Therefore, we wish $\mathbf{B}$ has the following properties: 1) There are no all-zero columns in $\mathbf{B}$ because for each subspace, there must be some features that work. 2) Each column of $\mathbf{B}$ is sparse because for each subspace, only a few features work. Now we show that the sparsity of each column of $\mathbf{B}$ is beneficial to the consistency of features and subspaces, i.e., the samples in the same subspace will select the same or similar features. Considering the i-th column of $\mathbf{B}$, denoted as $\mathbf{B}_{.i}$, we have $\mathbf{B}_{.i} = \mathbf{WY}_{.i}$, where $\mathbf{Y}_{i.}$ denotes the i-th column of matrix $\mathbf{Y}$. Notice that $\mathbf{Y}_{.i}$ is a 0-1 vector, where if the p-th sample belongs to the i-th subspace, then the p-th element in $\mathbf{Y}_{.i}$ is 1 and otherwise is 0. Supposing that the i-th subspace contains the samples $X_{i_1}, \cdots, X_{i_m}$, we have

$$\mathbf{B}_{.i} = \mathbf{WY}_{.i} = \begin{bmatrix} W_{1,i_1} + W_{1,i_2} + \cdots + W_{1,i_m} \\ W_{2,i_1} + W_{2,i_2} + \cdots + W_{2,i_m} \\ \cdots \\ W_{d,i_1} + W_{d,i_2} + \cdots + W_{d,i_m} \end{bmatrix}. \tag{8}$$

The sparsity of $\mathbf{B}_{.i}$ makes that there are as many zeros in $W_{1,i_1} + W_{1,i_2} + \cdots + W_{1,i_m}, \cdots, W_{d,i_1} + W_{d,i_2} + \cdots + W_{d,i_m}$ as possible. Without loss of generality, considering a simple example, we suppose that sample X_{i_1} selects the 1-st, 2-nd, and 3-rd features, which means $W_{1,i_1}, W_{2,i_1}, W_{3,i_1} > 0$, and other $W_{j,i_1} = 0$ for $j \neq 1, 2, 3$. Therefore, the 1-st, 2-nd, and 3-rd elements in $\mathbf{B}_{.i}$ are greater than 0, and other elements are zeros. Now, it is X_{i_2}'s turn to select features. If X_{i_2} selects a new feature, e.g. the 4-th feature, then $W_{4,i_2} > 0$, which makes the 4-th element of $\mathbf{B}_{.i}$ be non-zero, and thus decreases the number of zeros in $\mathbf{B}_{.i}$. If X_{i_2} selects the same features as X_{i_1}, i.e., the 1-st, 2-nd, or 3-rd features, since these elements in $\mathbf{B}_{.i}$ are already non-zeros, it will not decrease the number of zeros in $\mathbf{B}_{.i}$. Therefore, to make $\mathbf{B}_{.i}$ be as sparse as possible, X_{i_2} should select the same features as X_{i_1}. This analysis applies to any other samples X_{i_j}'s in the i-th subspace. Therefore, the samples in the same subspace are prone to selecting similar features, which shows the consistency of features and subspaces.

We can see that these two properties of $\mathbf{B}$ are very similar to the two properties of $\mathbf{W}$ introduced in Section III-A. Therefore, we can also use the group sparsity regularized term on $\mathbf{B}$ to constrain the relation between features and subspaces, leading to our novel regularized term:

$$\|\mathbf{B}\|_G = \|\mathbf{WY}\|_G. \tag{9}$$

Taking Eqs.(7) and (9) into Eq.(4), we obtain our final objective function:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{W},\mathbf{Y}} \quad & ||\mathbf{S}||_1 + \lambda \|\mathbf{X} - (\mathbf{X} \odot \mathbf{W})\mathbf{S}\|_F \\ & + \beta\, \mathrm{tr}(\mathbf{Y}^T \mathbf{L} \mathbf{Y} (\mathbf{Y}^T \mathbf{Y})^{-1}) + \gamma(\|\mathbf{W}\|_G + \|\mathbf{WY}\|_G) \\ \text{s.t.} \quad & \mathbf{W} \in [0,1]^{d \times N},\ \mathbf{S}^T \mathbf{1} = \mathbf{1},\ \mathrm{diag}(\mathbf{S}) = \mathbf{0}, \\ & \mathbf{Y} \in \{0,1\}^{N \times C},\ \sum_{j=1}^{C} Y_{ij} = 1. \end{aligned} \tag{10}$$

To achieve the interpretability, in Eq.(10), we use $\|\mathbf{W}\|_G$ to answer Question 1 and use $\|\mathbf{WY}\|_G$ to answer Question 2.

IV. Optimization

In this section, we introduce how to optimize Eq.(10). We first handle the group sparsity regularized term $\|\mathbf{W}\|_G$:

$$\|\mathbf{W}\|_G = \sum_{j=1}^{N} (\sum_{i=1}^{d} W_{ij})^2 = \mathrm{tr}(\mathbf{W}^T \mathbf{1} \mathbf{1}^T \mathbf{W}). \tag{11}$$

Similarly, for the term $\|\mathbf{WY}\|_G$, we have

$$\|\mathbf{WY}\|_G = \mathrm{tr}(\mathbf{Y}^T \mathbf{W}^T \mathbf{1} \mathbf{1}^T \mathbf{WY}) = \mathrm{tr}(\mathbf{W}^T \mathbf{1} \mathbf{1}^{\mathbf{T}} \mathbf{WYY}^T),$$

and thus

$$\|\mathbf{W}\|_G + \|\mathbf{WY}\|_G = \mathrm{tr}(\mathbf{W}^T \mathbf{1} \mathbf{1}^{\mathbf{T}} \mathbf{W}(\mathbf{YY}^T + \mathbf{I})). \tag{12}$$

We take it into Eq.(10) and apply the Alternating Direction Method of Multipliers (ADMM) method to optimize it. In more detail, we introduce an auxiliary variable $\mathbf{A} =$

$\mathbf{S} - \mathrm{diag}(\mathbf{S})$ and the Lagrange multipliers $\boldsymbol{\delta} \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{\Delta} \in \mathbb{R}^{N\times N}$, and obtain the following Lagrange formula:

$$\begin{aligned}\mathcal{L} = \min_{\mathbf{A},\mathbf{S},\mathbf{Y},\mathbf{W}} \quad & \|\mathbf{S}\|_1 + \lambda\|\mathbf{X} - (\mathbf{X}\odot\mathbf{W})\mathbf{A}\|_F^2 + \\ & \beta\,\mathrm{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}) + \gamma\,\mathrm{tr}(\mathbf{W}^T\mathbf{1}\mathbf{1}^T\mathbf{W}(\mathbf{Y}\mathbf{Y}^T + \mathbf{I})) \\ & \frac{\rho}{2}\|\mathbf{A}^T\mathbf{1} - \mathbf{1}\|_F^2 + \frac{\rho}{2}\|\mathbf{A} - \mathbf{S} + \mathrm{diag}(\mathbf{S})\|_F^2 + \\ & \boldsymbol{\delta}^T(\mathbf{A}^T\mathbf{1} - \mathbf{1}) + \mathrm{tr}(\boldsymbol{\Delta}^T(\mathbf{A} - \mathbf{S} + \mathrm{diag}(\mathbf{S}))), \\ \text{s.t. } \mathbf{Y} \in \{0,1\}^{N\times C}, \quad & \sum_{j=1}^{C} Y_{ij} = 1, \quad \mathbf{W} \in [0,1]^{d\times N},\end{aligned} \tag{13}$$

where $\rho > 0$ is an adaptive penalty parameter. Next, we update the variables $\mathbf{A}$, $\mathbf{S}$, $\mathbf{W}$, and $\mathbf{Y}$ iteratively.

A. *Updating* A

By removing terms unrelated to $\mathbf{A}$, the objective function can be simplified to an unconstrained optimization problem:

$$\begin{aligned}\mathcal{L} = \min_{\mathbf{A}} \quad & \lambda\|\mathbf{X} - (\mathbf{X}\odot\mathbf{W})\mathbf{A}\|_F^2 + \beta\,\mathrm{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}) \\ & \frac{\rho}{2}\|\mathbf{A}^T\mathbf{1} - \mathbf{1}\|_F^2 + \frac{\rho}{2}\|\mathbf{A} - \mathbf{S} + \mathrm{diag}(\mathbf{S})\|_F^2 + \\ & \boldsymbol{\delta}^T(\mathbf{A}^T\mathbf{1} - \mathbf{1}) + \mathrm{tr}(\boldsymbol{\Delta}^T(\mathbf{A} - \mathbf{S} + \mathrm{diag}(\mathbf{S}))).\end{aligned} \tag{14}$$

Notice that $\mathbf{L}$ is related to $\mathbf{A}$, and the following equivalence holds:

$$\mathrm{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}) = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij}\|\mathbf{F}_{i,:} - \mathbf{F}_{j,:}\|_2^2, \tag{15}$$

where $\mathbf{F} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1/2}$. By introducing matrix $\mathbf{Q}$ whose (i,j)-th element is $Q_{ij} = \|\mathbf{F}_{i,:} - \mathbf{F}_{j,:}\|_2^2$, we can derive the following formula:

$$\mathrm{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}) = \frac{1}{2}\,\mathrm{tr}(\mathbf{A}^T\mathbf{Q}) \tag{16}$$

Taking it into Eq.(14), and setting the derivative of $\mathcal{L}$ with respect to $\mathbf{A}$ to zero, we obtain:

$$\begin{aligned}&(2\lambda\mathbf{E}^T\mathbf{E} + \rho\mathbf{1}\mathbf{1}^T + \rho\mathbf{I})\mathbf{A} \\ &= 2\lambda\mathbf{E}^T\mathbf{X} + \rho(\mathbf{1}\mathbf{1}^T + \mathbf{S}) - \beta\mathbf{Q} - \mathbf{1}\boldsymbol{\delta}^T - \boldsymbol{\Delta},\end{aligned} \tag{17}$$

where $\mathbf{E} = \mathbf{X}\odot\mathbf{W}$. If $N < d$, we can directly obtain the solution of $\mathbf{A}$ as:

$$\begin{aligned}\mathbf{A} =&(2\lambda\mathbf{E}^T\mathbf{E} + \rho\mathbf{1}\mathbf{1}^T + \rho\mathbf{I})^{-1}\cdot \\ &(2\lambda\mathbf{E}^T\mathbf{X} + \rho(\mathbf{1}\mathbf{1}^T + \mathbf{S}) - \beta\mathbf{Q} - \mathbf{1}\boldsymbol{\delta}^T - \boldsymbol{\Delta}).\end{aligned} \tag{18}$$

The time complexity of the inverse $(2\lambda\mathbf{E}^T\mathbf{E} + \rho\mathbf{1}\mathbf{1}^T + \rho\mathbf{I})^{-1}$ is $O(N^3)$ which is acceptable when N is small. However, when N is large, especially when $N > d$, the time complexity is too high. To tackle this problem, we derive an accelerated method to compute Eq.(18). We denote $\mathbf{J} = \mathbf{I} + \epsilon\mathbf{E}^T\mathbf{E}$ where $\epsilon = \frac{2\lambda}{\rho}$. It can be seen that $(2\lambda\mathbf{E}^T\mathbf{E} + \rho\mathbf{1}\mathbf{1}^T + \rho\mathbf{I})^{-1} = \frac{1}{\rho}(\mathbf{J} + \mathbf{1}\mathbf{1}^T)^{-1}$. Therefore, we compute $(2\lambda\mathbf{E}^T\mathbf{E} + \rho\mathbf{1}\mathbf{1}^T + \rho\mathbf{I})^{-1}$ incrementally.

In more detail, we first compute the inverse of $\mathbf{J}$. According to the Woodbury Identity, we have:

$$\mathbf{J}^{-1} = (\mathbf{I} + \epsilon\mathbf{E}^T\mathbf{E})^{-1} = \mathbf{I} - \mathbf{E}^T(\epsilon^{-1}\mathbf{I} + \mathbf{E}\mathbf{E}^T)^{-1}\mathbf{E}. \tag{19}$$

Notice that $\epsilon^{-1}\mathbf{I} + \mathbf{E}\mathbf{E}^T$ is a d-by-d matrix whose inverse can be computed in $O(d^3)$ which is smaller than $O(N^3)$ if $d < N$. Then, according to Sherman-Morrison Identity, we have:

$$(\mathbf{J} + \mathbf{1}\mathbf{1}^T)^{-1} = \mathbf{J}^{-1} - \frac{\mathbf{J}^{-1}\mathbf{1}\mathbf{1}^T\mathbf{J}^{-1}}{1 + \mathbf{1}^T\mathbf{J}^{-1}\mathbf{1}}, \tag{20}$$

Taking it into Eq.(18), we can obtain the solution of $\mathbf{A}$ when $N > d$.

To sum up, if $N > d$, the time complexity of computing $\mathbf{A}$ is $O(dN^2 + d^2N + d^3) = O(dN^2)$. If $N < d$, the time complexity of computing $\mathbf{A}$ is $O(dN^2 + N^3) = O(dN^2)$.

B. *Updating* S

By fixing other variables, we can obtain the closed-form solution of $\mathbf{S}$ as:

$$S_{ij} = \begin{cases} R_{ij}\,\mathrm{sgn}(P_{ij}), & \text{if } |P_{ij}| > \frac{1}{\rho} \text{ and } i \neq j \\ 0, & \text{otherwise,} \end{cases} \tag{21}$$

where $R_{ij} = |A_{ij} + \frac{\Delta_{ij}}{\rho}| - \frac{1}{\rho}$ and $P_{ij} = A_{ij} + \frac{\Delta_{ij}}{\rho}$. The $\mathrm{sgn}(\cdot)$ represents the sign function.

C. *Updating* W

We update $\mathbf{W}$ using a Quasi-Newton method. By taking the derivative of the objective function Eq.(13) w.r.t $\mathbf{W}$, we have:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{W}} =& 2\mathbf{X}\odot\left((\mathbf{X}\odot\mathbf{W})\,\mathbf{S}\mathbf{S}^T\right) - \\ & 2\mathbf{X}\odot\left(\mathbf{X}\mathbf{S}^T\right) + 2\gamma\left(\mathbf{1}\mathbf{1}^T\mathbf{W}\left(\mathbf{I} + \mathbf{Y}\mathbf{Y}^T\right)\right).\end{aligned} \tag{22}$$

Equipped with the partial derivative Eq.(22), we employ the constrained L-BFGS-B to optimize $\mathbf{W}$ [57].

D. *Updating* Y

When updating $\mathbf{Y}$, the objective function Eq.(13) can be simplified to the following formulation:

$$\begin{aligned}\min_{\mathbf{Y}} \quad & \sum_{k=1}^{C}\frac{\mathbf{Y}_{:,k}^T\mathbf{L}\mathbf{Y}_{:,k}}{\mathbf{Y}_{:,k}^T\mathbf{Y}_{:,k}} + \gamma\mathbf{Y}_{:,k}^T\hat{\mathbf{L}}\mathbf{Y}_{:,k} \\ \text{s.t.} \quad & \mathbf{Y} \in \{0,1\}^{N\times C}, \sum_{k=1}^{C}\mathbf{Y}_{:,k} = \mathbf{1},\end{aligned} \tag{23}$$

where $\hat{\mathbf{L}} = \mathbf{W}^T\mathbf{1}\mathbf{1}^T\mathbf{W}$. Notice that each row of $\mathbf{Y}$ contains only one 1 and other elements are all zeros. Therefore, we can solve $\mathbf{Y}$ row-by-row. In more detail, considering the i-th row of $\mathbf{Y}$, we try C possible result $[1,0,\cdots,0]$, $[0,1,0,\cdots,0]$, $\cdots$, $[0,\cdots,0,1]$ and replace the i-th row of $\mathbf{Y}$ with the one which leads to the smallest objective function. To accelerate this process, we adopt the method proposed in [58], [59] for efficient computation.

E. *Updating Lagrange Multipliers*

At last, we update the Lagrange multipliers $\boldsymbol{\delta}$, $\boldsymbol{\Delta}$ and step size ρ as follows:

$$\begin{cases} \boldsymbol{\delta} = \boldsymbol{\delta} + \rho(\mathbf{A}\mathbf{1} - \mathbf{1}), \\ \boldsymbol{\Delta} = \boldsymbol{\Delta} + \rho(\mathbf{A} - \mathbf{S}), \\ \rho = 2\rho. \end{cases} \tag{24}$$

F. Algorithm and Discussion

Algorithm 1 summarizes the process of our ISC. We initialize the local feature selection matrix $\mathbf{W}$ with an all-ones matrix. The k-nn method is used to construct an initial graph $\mathbf{S}$, and then the spectral clustering method is applied on $\mathbf{S}$ to get the initial pseudo-labels $\mathbf{Y}$.

Now we analyze the time complexity. When solving matrix $\mathbf{A}$ with Eq.(17), the time complexity is $O(dN^2)$. The time complexity of computing matrix $\mathbf{S}$ is $O(N^2)$. For updating matrix $\mathbf{W}$ using the gradient descent method, the time complexity of computing the gradient of $\mathbf{W}$ is $O(dN^2)$. The time complexity of computing $\mathbf{Y}$ is $O(CN^2)$ [58]. The overall time complexity of the entire algorithm is $O(dN^2 + CN^2)$. According to [60], the mainstream subspace clustering methods, especially the methods that use the ADMM algorithm, often have similar time complexity. The complexity of our method is comparable to the mainstream subspace clustering methods [19], [20], [4]. Notice that the operation that costs $O(dN^2)$ in our method is the matrix multiplication, which is often fast in practice. The matrix multiplication can be easily parallelized for further speedup. In the future, we will try to tackle this problem to speed up the method.

Algorithm 1 Interpretable Subspace Clusterting

Input: Data matrix $\mathbf{X}$, number of clusters C.
Output: Clustering results $\mathbf{Y}$ and feature selection matrix $\mathbf{W}$.

```
Construct the initial local feature selection matrix $\mathbf{W}$ and
similarity matrix $\mathbf{S}$ with $k$-nn. Run spectral clustering on
$\mathbf{X}$ to obtain the initial label matrix $\mathbf{Y}$.
while not converge do
    Update $\mathbf{A}$ by Eq.(18).
    Update $\mathbf{S}$ by Eq.(21).
    Update $\mathbf{W}$ with L-BFGS-B method.
    Update $\mathbf{Y}$ by solving Eq.(23).
    Update the lagrange mltipliers $\delta$ and $\mathbf{\Delta}$ by Eq.(24).
end while
Obtain the final clustering result $\mathbf{Y}$ and feature selection
matrix $\mathbf{W}$.
```

V. Experiments

In this section, we compare our method with several state-of-the-art clustering methods on benchmark data sets.

A. Data Sets

We conduct experiments on 12 benchmark data sets, including JAFFE [61], SONAR[1], HEART[1], ORL [62], PROSTATE[2], BASEHOOK[2], LUNG[2], NCI9[2], COIL20[3], TOX171[62], MAGIC[1] and AMAZON[4]. The basic information of these data sets are summarized in Table I .

TABLE I: Description of the data sets.

	# Instance	# Features	# Clusters
JAFFE	213	676	10
HEART	303	13	2
ORL	400	1024	40
PROSTATE	102	5966	2
BASEHOOK	1993	4862	2
LUNG	203	3312	4
COIL20	1440	1024	20
NCI9	60	9712	9
SONAR	208	60	2
TOX171	171	5748	4
MAGIC	19020	10	2
AMAZON	1000	607	2

B. Experimental Setup

To evaluate the clustering performance comprehensively, we compare our ISC with the following three kinds of methods:

(1) Subspace Clustering Methods

- KSS [18], which is a K-subspaces method for subspace clustering.
- TSC [36], which is a thresholding-based method for subspace clustering.
- GSC [22], which is a greedy subspace clustering method that recovers subspaces by a greedy algorithm.
- LRR [19], which is a low-rank representation-based method for subspace clustering.
- LRR-SSC [20], which integrates low-rank and sparse representation for subspace clustering.
- SSC-OMP [35], which applies an orthogonal matching pursuit method to sparse subspace clustering.
- RWSC [4], which is a reweighted clustering approach that jointly considers local and global structural information.

(2) Interpretable Clustering Methods

- IDC [34], which is a local feature selection based sample-specific interpretable deep clustering method.
- EXKMEANS [40], which is an expanding explainable k-means clustering method using a decision tree.
- SHALLOW [42], which constructs a shallow decision tree to improve the interpretability of k-means.

(3) Feature Selection based Clustering Methods

Since our method is based on feature selection, we also compare with some state-of-the-art feature selection based clustering methods:

- BSFS [25], which is a balanced spectral feature selection method by selecting the discriminative features and revealing the balanced structure of data.
- BLSFS [26], which is a bi-level balanced spectral feature selection method that combines classification level and feature level, using pseudo-labels.
- FSDK [27], which is a fast sparse discriminative k-means for feature selection by combining least-squares regression and discriminative k-means.
- GOD-cPSO [31], which is a synchronous feature selection method that combines graph embedding with collaborative particle swarm optimization to select the most discriminative features.

[1] https://archive.ics.uci.edu/dataset/
[2] https://jundongl.github.io/scikit-feature/datasets.html
[3] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html/
[4] https://archive.ics.uci.edu/dataset/331/sentiment+labelled+sentences

- SPDFS [32], which proposes an unsupervised feature selection method with $l_{2,0}$-norm constrained sparse projection.

Notice that, although the above-mentioned feature selection methods [25], [26], [27], [31], [32] are applied to select features in their original literatures, since they are embedded methods, which also learn the pseudo-labels of clustering to guide the feature selection, for a fair comparison, we regard their pseudo-labels $\mathbf{Y}$ as their clustering results, which is similar to our method. Their $\mathbf{Y}$'s are initialized using the same spectral clustering method as ours. We compare the $\mathbf{Y}$ obtained from our method to the $\mathbf{Y}$'s obtained from the feature selection methods BSFS, BLSFS, FSDK, GOD-cPSO, and SPDFS.

To evaluate the clustering performance of the proposed ISC, we report both Accuracy (ACC) and Normalized Mutual Information (NMI) metrics, comparing against all baseline methods. For subspace clustering methods that output similarity matrices [18], [36], [22], [19], [20], [35], [4], we consistently construct a k-nn graph ($k = 5$) followed by spectral clustering for fair comparison.

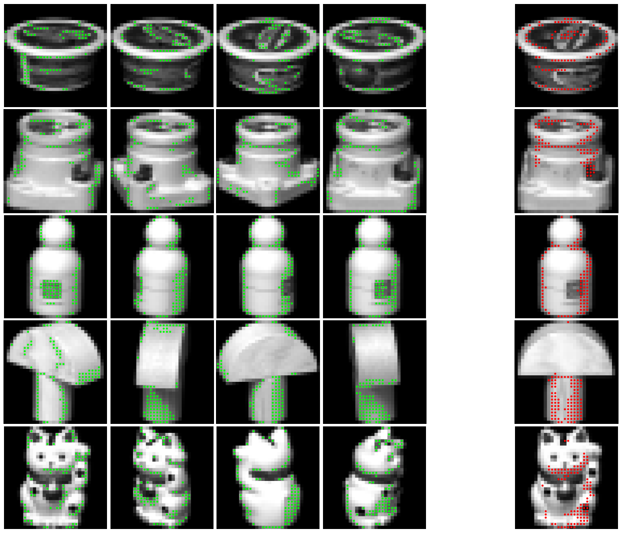
In our method, we fix the hyperparameter $\beta = 10^{-3}$ and tune λ within the range $[1, 10^8]$ and γ within $[10^{-3}, 10^2]$ by grid search. For other compared methods, we tune the parameters as suggested in their papers. For all methods on all data sets, the number of clusters is set to the true number of classes.

All experiments are conducted using MATLAB 2022b on a PC computer with Windows 11, 3.61-GHz CPU, and 64 GB of memory.

C. Experimental Results on Clustering Performance

We run experiments 10 times and report the average clustering results and standard deviation w.r.t. ACC and NMI in Tables II and III, respectively. From Tables II and III, we find that although our method focuses on enhancing the interpretability of subspace clustering, it can also outperform other subspace clustering methods, interpretable methods, and feature selection-based clustering methods on most data sets. When compared with interpretable clustering methods and feature selection methods, our method outperformed all the competing methods on all data sets in terms of ACC and NMI. The main reasons may be threefold as follows.

- In our interpretable clustering method, we explain the clustering process during the clustering. In this way, the explanation (i.e., $\mathbf{W}$) can guide the clustering in turn. However, SHALLOW [42] and EXKMEANS [41] are post-hoc interpretable clustering methods, which can hardly directly improve clustering performance with interpretability. Although IDC [34] is also a local feature selection based interpretable clustering model, it ignores the relations between the features and clusters, i.e., it does not constrain the consistency of the selected features in the same cluster. The inconsistency of features in the same cluster may misguide the clustering.
- Our method introduces a learnable local feature selection matrix $\mathbf{W}$ based on conventional subspace clustering. This weighting matrix amplifies the influence of significant feature values while suppressing redundant or irrelevant features in the clustering process, which is helpful for clustering.
- We impose a group sparsity regularized term $\|\mathbf{WY}\|_G$ on the selected features for each cluster. Although this term is used for interpretability, since this term also involves the clustering result $\mathbf{Y}$, which can affect the clustering result $\mathbf{Y}$ in turn. When minimizing $\|\mathbf{WY}\|_G$, it enforces feature consistency within clusters, i.e., the features selected within each cluster remain highly similar, which can effectively reduce intra-cluster sample distances and reveal clearer clustering structure. Therefore, even compared with other subspace clustering methods, our methods can also often achieve better performance.



(a) The selected features of each sample in each cluster.

(b) The selected features of each cluster.

Fig. 4: Interpretability results on the COIL20 data set. Each row in the figure represents a cluster, and each image represents one sample in a cluster. (a) shows the selected features for each sample by green dots. (b) shows the selected features for each cluster by red dots.

D. Experimental Results on Interpretability

To demonstrate the interpretability of our proposed method, we design experiments to answer the two questions raised in the Introduction section: 1) when handling an individual sample, which features should work for this sample? 2) Which subspace or cluster will the features that work put this sample into?

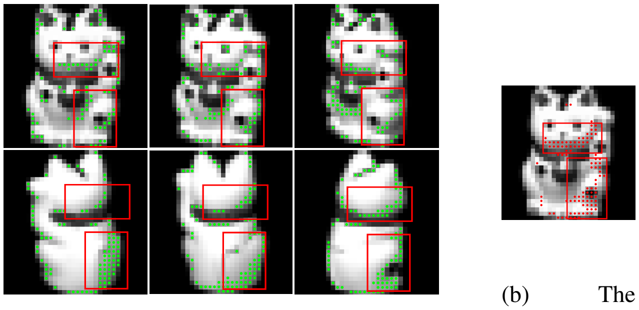
To show the answer to the two questions, we present visualization results on the COIL20 data set. First, we show the answer to Question 1. In Fig. 4(a), we show the selected features for each sample. Here, we show samples from 5 classes in 5 rows, where each row shows 4 different samples in such a cluster. For each sample, the selected 100 features (i.e., the selected pixels) are shown in green color. We can see that, for different objects, we choose quite different features, which

TABLE II: The average ACC results and standard deviation of our method and other methods. Red, green, and blue texts indicate the best, the second-best results, and the third-best results, respectively.

Methods	JAFFE	HEART	ORL	PROSTATE	BASEHOOK	LUNG	COIL20	NCI9	SONAR	TOX171	MAGIC	AMAZON
KSS [18]	0.7793 ± 0.0000	0.6040 ± 0.0099	0.4767 ± 0.0000	0.6078 ± 0.0000	0.7195 ± 0.0000	0.5320 ± 0.0000	**0.9215** ± 0.0000	0.5283 ± 0.0137	0.5481 ± 0.0000	0.4269 ± 0.0000	0.6032 ± 0.0000	0.5740 ± 0.0000
TSC [36]	0.9812 ± 0.0000	0.7393 ± 0.0000	0.6900 ± 0.0138	0.6078 ± 0.0000	0.9463 ± 0.0000	0.8768 ± 0.0000	0.8444 ± 0.0000	0.5350 ± 0.0266	0.5625 ± 0.0000	0.5390 ± 0.0000	0.5869 ± 0.0000	0.6420 ± 0.0000
GSC [22]	0.9953 ± 0.0000	0.7508 ± 0.0052	0.6777 ± 0.0199	0.5647 ± 0.0186	0.5265 ± 0.0011	0.7537 ± 0.0000	0.8591 ± 0.0117	0.3900 ± 0.0394	0.5231 ± 0.0426	0.4246 ± 0.0276	0.6402 ± 0.0000	0.6167 ± 0.0326
LRR [19]	0.9465 ± 0.0000	0.7525 ± 0.0000	**0.7545** ± 0.0119	0.5784 ± 0.0000	0.6232 ± 0.0000	0.7877 ± 0.0016	0.7149 ± 0.0108	0.4217 ± 0.0324	0.5529 ± 0.0000	0.4807 ± 0.0452	0.5815 ± 0.0000	0.5520 ± 0.0000
LRR-SSC [20]	0.9906 ± 0.0000	0.7393 ± 0.0000	0.7312 ± 0.0165	0.6275 ± 0.0000	0.9508 ± 0.0000	0.8227 ± 0.0000	0.7939 ± 0.0015	0.5167 ± 0.0484	0.6442 ± 0.0000	0.5263 ± 0.0000	0.6254 ± 0.0000	0.6000 ± 0.0000
SSC-OMP [35]	0.8423 ± 0.0392	0.6139 ± 0.0000	0.6570 ± 0.0301	0.5882 ± 0.0000	0.9649 ± 0.0000	0.7409 ± 0.0025	0.5119 ± 0.0284	0.4650 ± 0.0372	0.5192 ± 0.0000	0.4152 ± 0.0000	0.6502 ± 0.0000	0.5040 ± 0.0000
RWSC [4]	0.9374 ± 0.0621	0.8251 ± 0.0000	0.5942 ± 0.0290	0.6196 ± 0.0165	0.5227 ± 0.0265	0.4921 ± 0.0378	0.5185 ± 0.0366	0.3783 ± 0.0478	0.6875 ± 0.0000	0.3632 ± 0.0171	**0.7096** ± 0.0000	0.5170 ± 0.0000
IDC [34]	0.7229 ±0.1408	0.5413 ± 0.0000	0.4383 ± 0.0346	0.6176 ± 0.0084	0.5937 ± 0.0546	0.7286 ± 0.0553	0.6308 ± 0.0529	0.3617 ± 0.0289	0.5336 ± 0.0000	0.5087 ± 0.0033	0.6484 ± 0.0001	0.5310 ± 0.0218
EXKMEANS [41]	0.8845 ± 0.0785	0.6284 ± 0.0478	0.5468 ± 0.0230	0.5784 ± 0.0000	0.5882 ± 0.0185	0.6675 ± 0.1143	0.5715 ± 0.0563	0.3883 ± 0.0509	0.5736 ± 0.0255	0.4398 ± 0.0526	0.6406 ± 0.0108	0.5541 ± 0.0376
SHALLOW [42]	0.9634 ± 0.0079	0.6139 ± 0.0000	0.5055 ± 0.0216	0.5784 ± 0.0000	0.5795 ± 0.0000	0.7975 ± 0.068	0.5665 ± 0.0243	0.4100 ± 0.0251	0.5385 ± 0.0000	0.4094 ± 0.0000	0.6245 ± 0.0021	0.5768 ± 0.0186
FSDK [27]	0.9812 ± 0.0000	0.6172 ± 0.0000	0.6825 ± 0.0000	0.7157 ± 0.0000	0.5048 ± 0.0000	0.8670 ± 0.0000	0.8486 ± 0.0000	0.4833 ± 0.0000	0.5337 ± 0.0000	0.4795 ± 0.0000	0.6039 ± 0.0000	0.5120 ± 0.0000
BSFS [25]	0.9765 ± 0.0000	0.7393 ± 0.0000	0.6425 ± 0.0000	0.6176 ± 0.0000	0.5876 ± 0.0000	0.7488 ± 0.0000	0.8132 ± 0.0000	0.5333 ± 0.0000	0.7163 ± 0.0000	0.5497 ± 0.0000	0.5830 ± 0.0000	0.6310 ± 0.0000
BSLFS [26]	0.9812 ± 0.0000	0.5941 ± 0.0000	0.1637 ± 0.0300	0.6471 ± 0.0000	0.5735 ± 0.0000	0.8621 ± 0.0000	0.8590 ± 0.0000	0.2167 ± 0.0000	0.8269 ± 0.0000	0.2632 ± 0.0000	0.6063 ± 0.0000	0.5570 ± 0.0000
GOD-cPSO [31]	**1.0000** ± 0.0000	0.6587 ± 0.0469	0.6520 ± 0.0076	0.5627 ± 0.0095	0.5048 ± 0.0002	0.8670 ± 0.0286	0.8230 ± 0.0057	0.4800 ± 0.0233	0.5519 ± 0.0527	0.4485 ± 0.0347	0.5363 ± 0.0015	0.5083 ± 0.0007
SPDFS [32]	**1.0000** ± 0.0000	0.8083 ± 0.0328	0.6167 ± 0.0191	0.5794 ± 0.0117	0.5681 ± 0.0126	0.7438 ± 0.0040	0.7588 ± 0.0169	0.5150 ± 0.0053	0.5438 ± 0.0235	0.4959 ± 0.0281	0.6188 ± 0.0087	0.5496 ± 0.0140
ISC (OURS)	**1.0000** ± 0.0000	**0.8911** ± 0.0000	0.7235 ± 0.0236	**0.7255** ± 0.0000	**0.9905** ± 0.0000	**0.8966** ± 0.0000	0.9174 ± 0.0000	**0.6000** ± 0.0000	**0.9663** ± 0.0000	**0.6082** ± 0.0000	0.6545 ± 0.0000	**0.6530** ± 0.0000

TABLE III: The average NMI results and standard deviation of our method and other methods. Red, green, and blue texts indicate the best, the second-best results, and the third-best results, respectively.

Methods	JAFFE	HEART	ORL	PROSTATE	BASEHOOK	LUNG	COIL20	NCI9	SONAR	TOX171	MAGIC	AMAZON
KSS [18]	0.8399 ± 0.0000	0.0296 ± 0.0000	0.6910 ± 0.0066	0.0453 ± 0.0000	0.1452 ± 0.0000	0.3454 ± 0.0000	**0.9665** ± 0.0000	0.5305 ± 0.0086	0.0076 ± 0.0000	0.1740 ± 0.0000	0.0048 ± 0.0000	0.0195 ± 0.0000
TSC [36]	0.9731 ± 0.0000	0.1739 ± 0.0000	0.8109 ± 0.0057	0.0380 ± 0.0000	0.7042 ± 0.0000	0.7000 ± 0.0000	0.8981 ± 0.0000	0.5241 ± 0.0198	0.0098 ± 0.0000	0.2559 ± 0.0000	0.0099 ± 0.0000	0.0655 ± 0.0000
GSC [22]	0.9917 ± 0.0000	0.1875 ± 0.0093	0.8130 ± 0.0058	0.0156 ± 0.0147	0.0239 ± 0.0011	0.5240 ± 0.0000	0.9084 ± 0.0111	0.3897 ± 0.0411	0.0061 ± 0.0193	0.1764 ± 0.0018	0.0048 ± 0.0000	0.0608 ± 0.0183
LRR [19]	0.9651 ± 0.0000	0.1890 ± 0.0000	**0.8650** ± 0.0057	0.0191 ± 0.0000	0.4440 ± 0.0000	0.5643 ± 0.0036	0.7836 ± 0.0037	0.4349 ± 0.0295	0.0086 ± 0.0000	0.2105 ± 0.0000	0.0085 ± 0.0000	0.5520 ± 0.0000
LRR-SSC [20]	0.9832 ± 0.0000	0.1696 ± 0.0000	0.8485 ± 0.0086	0.0473 ± 0.0000	0.7175 ± 0.0000	0.5909 ± 0.0000	0.8671 ± 0.0000	0.5094 ± 0.0353	0.0691 ± 0.0000	0.2342 ± 0.0000	0.0214 ± 0.0000	0.0484 ± 0.0000
SSC-OMP [35]	0.8383 ± 0.0122	0.0358 ± 0.0000	0.8150 ± 0.0120	0.0226 ± 0.0000	0.7812 ± 0.0000	0.5046 ± 0.0016	0.5877 ± 0.0118	0.5040 ± 0.0235	0.0132 ± 0.0000	0.0991 ± 0.0000	0.0029 ± 0.0000	0.0008 ± 0.0000
RWSC [4]	0.9582 ± 0.0351	0.3386 ± 0.0000	0.7632 ± 0.0153	0.0441 ± 0.0109	0.0080 ± 0.0102	0.2647 ± 0.0138	0.6832 ± 0.0277	0.3096 ± 0.0453	0.1011 ± 0.0000	0.0607 ± 0.0248	**0.0795** ± 0.0000	0.0172 ± 0.0000
IDC [34]	0.8349 ± 0.0945	0.0000 ± 0.0000	0.6827 ± 0.0229	0.0699 ± 0.0068	0.0388 ± 0.0468	0.2584 ± 0.2703	0.7446 ± 0.0245	0.3119 ± 0.0426	0.0056 ± 0.0177	0.2777 ± 0.0726	0.0100 ± 0.0000	0.0044 ± 0.0059
EXKMEANS [40]	0.88445 ± 0.0785	0.0695 ± 0.0434	0.7357 ± 0.0188	0.0183 ± 0.0000	0.0252 ± 0.0080	0.5576 ± 0.0799	0.7344 ± 0.0189	0.3912 ± 0.0467	0.0179 ± 0.0113	0.1551 ± 0.0390	0.0279 ± 0.0042	0.0446 ± 0.0419
SHALLOW [42]	0.9500 ± 0.0080	0.0603 ± 0.0000	0.7014 ± 0.0140	0.0185 ± 0.0000	0.0196 ± 0.0000	0.6328 ± 0.0380	0.6854 ± 0.0095	0.3886 ± 0.0269	0.0068 ± 0.0000	0.1262 ± 0.0000	0.0167 ± 0.0000	**0.0887** ± 0.0288
FSDK [27]	0.9731 ± 0.0000	0.0458 ± 0.0000	0.8251 ± 0.0000	0.1430 ± 0.0000	0.0035 ± 0.0000	0.6994 ± 0.0000	0.9414 ± 0.0000	0.5489 ± 0.0000	0.0268 ± 0.0000	0.1393 ± 0.0000	0.0076 ± 0.0000	0.0005 ± 0.0000
BSFS [25]	0.9615 ± 0.0000	0.1739 ± 0.0000	0.7732 ± 0.0000	0.0403 ± 0.0000	0.0223 ± 0.0000	0.5007 ± 0.0000	0.8754 ± 0.0035	0.5094 ± 0.0000	0.1831 ± 0.0000	**0.3206** ± 0.0000	0.0219 ± 0.0000	0.0501 ± 0.0000
BSLFS [26]	0.9699 ± 0.0000	0.0256 ± 0.0000	0.2040 ± 0.0441	0.0688 ± 0.0000	0.0158 ± 0.0000	0.5211 ± 0.0000	0.8987 ± 0.0000	0.0491 ± 0.0000	0.3534 ± 0.0000	0.0000 ± 0.0000	0.0060 ± 0.0000	0.0095 ± 0.0000
GOD-cPSO [31]	**1.0000** ± 0.0000	0.1264 ± 0.0549	0.8010 ± 0.0068	0.0116 ± 0.0041	0.0036 ± 0.0002	0.7036 ± 0.0636	0.9064 ± 0.0074	0.5061 ± 0.0201	0.0285 ± 0.0151	0.2645 ± 0.0606	0.0009 ± 0.0004	0.0002 ± 0.0000
SPDFS [32]	**1.0000** ± 0.0000	0.2996 ± 0.0684	0.7866 ± 0.0156	0.0197 ± 0.0058	0.0212 ± 0.0059	0.3122 ± 0.0033	0.8296 ± 0.0120	0.5368 ± 0.0022	0.0160 ± 0.0131	0.3314 ± 0.0308	0.0133 ± 0.0024	0.0082 ± 0.0043
ISC (OURS)	**1.0000** ± 0.0000	**0.5123** ± 0.0000	0.8507 ± 0.0128	**0.1538** ± 0.0000	**0.9225** ± 0.0000	**0.7145** ± 0.0000	0.9509 ± 0.0000	**0.6047** ± 0.0000	**0.8188** ± 0.0000	0.3189 ± 0.0000	0.0270 ± 0.0000	0.0687 ± 0.0000



(a) Common feature regions selected across different samples are shown by red rectangles.

(b) The selected features of this cluster.

Fig. 5: An example of selected features on the COIL20 dataset. (a) Common feature regions selected across different samples are shown by red rectangles. (b) The feature regions selected for a cluster are shown in red rectangles.

means for different samples, the features that work should also be different. Moreover, for each sample, our method tends to select contour points in the images as discriminative features. This is in line with human vision, because humans also often recognize different objects by their contours.

To answer Question 2, from Fig. 4(a), we can see that, for different samples in the same cluster (i.e., the images in the same row), we often select similar features. These common features are the features related to the cluster. To show it more clearly, we show the selected features for a cluster in Fig. 4(b). In more detail, we select features according to the matrix $\mathbf{WY}$. Intuitively, considering the j-th cluster, i.e., the j-th column of $\mathbf{WY}$, the larger the i-th element is, the more important the i-th feature is to the j-th cluster. Based on this observation, we plot the most important 100 features for each cluster, and show the results in Fig. 4(b). The selected features for each cluster are shown in red. We can see that, for images whose contour information remains consistent among different samples in one cluster (e.g., the first, second, and third rows in Fig. 4(b)), the features selected for their corresponding subspace are consistent with those of the individual samples within that subspace. For images that have large variations of contours among different samples in one cluster (e.g., the fourth and fifth rows in Fig. 4(b)), our method tries to select the common discriminative features. For example, we show the cluster of the lucky cat in COIL 20 as a case study in Fig. 5. Fig. 5(a) shows the features selected for each sample in the cluster of lucky cat, and Fig. 5 (b) shows the selected features for this cluster. We highlight the commonly selected features among different samples with the red blocks in Fig. 5(a). These features are also selected for this cluster as shown in Fig. 5(b).

Furthermore, Fig. 6 provides a comparative analysis against current state-of-the-art interpretable clustering methods and feature selection-based clustering methods. The number of selected features is gradually increased from 20 to 200. We find that our approach identifies more interpretable features and focuses more effectively on discriminative regions of the samples. Notice that the EXKMEANS [40], SHALLOW [42], and IDC [34] methods select fewer than 200 features due to inherent limitations of the algorithm itself. Unlike feature selection methods that identify identical features for all samples, our proposed method adaptively selects sample-specific features, which is more helpful for interoperability.

Besides, we show another example on the AMAZON data set. AMAZON is a text data set, containing the reviews on the AMAZON website. The data set has two classes: the positive reviews and the negative reviews. The features of the data set are the words. In Table IV, we show 5 reviews of each cluster. In each review, we highlight the top one selected word according to our feature indicator matrix $\mathbf{W}$ with red color. From Table IV, we can see that the selected words, e.g. ”great”, ”waste”, and ”fails”, can well represent the attitude of the reviews. Therefore, these selected words or features can well explain why we put some reviews into a cluster.

E. Ablation Study

In this section, we conduct some ablation studies to show the effects of our new proposed regularized terms $\|\mathbf{W}\|_G$ and $\|\mathbf{WY}\|_G$. In more detail, we compare our method with three degenerated versions of ISC. The first is the one without both the $\|\mathbf{W}\|_G$ and $\|\mathbf{WY}\|_G$. The second one only removes $\|\mathbf{W}\|_G$ and the third one only removes $\|\mathbf{WY}\|_G$. The results are shown in Table V.

From Table V, we find that $\|\mathbf{WY}\|_G$ is more important than $\|\mathbf{W}\|_G$. This is because both $\|\mathbf{WY}\|_G$ and $\|\mathbf{W}\|_G$ can achieve row-wise sparsity in $\mathbf{W}$, but $\|\mathbf{WY}\|_G$ additionally enforces feature consistency within cluster, resulting in superior performance compared to $\|\mathbf{W}\|_G$. It is the term $\|\mathbf{WY}\|_G$ that constructs the relations between the selected features and clusters and applies the interpretability (i.e., $\mathbf{W}$) to guide the clustering. Although the motivation of the term is interpretability, it can also improve the clustering performance, demonstrating that improving the clustering performance with interpretability is feasible. $\|\mathbf{W}\|_G$ is to impose the group sparsity on $\mathbf{W}$, which is used to answer the first question of interpretability. This term does not involve either the data matrix $\mathbf{X}$ or the clustering result $\mathbf{Y}$. It means that this term is just for interpretability, but not for clustering. $\mathbf{W}$ does not control the cluster structure directly, and if we remove it, the clustering performance will be affected slightly as shown in Table V.

F. Efficiency Results

Fig. 7 shows the convergence curves of ISC on COIL20, JAFFE, NCI9, and ORL data sets. The results on other data sets are similar. From Fig. 7, we find that ISC converges fast. It often converges within ten iterations.

Fig. 8 shows the running time of ISC compared with other clustering methods on all data sets. Since some methods are very time-consuming, we report the logarithm of the time in seconds for better comparison. From Fig. 8, we can see that our method is comparable to the mainstream methods, and even faster than some methods, such as IDC.

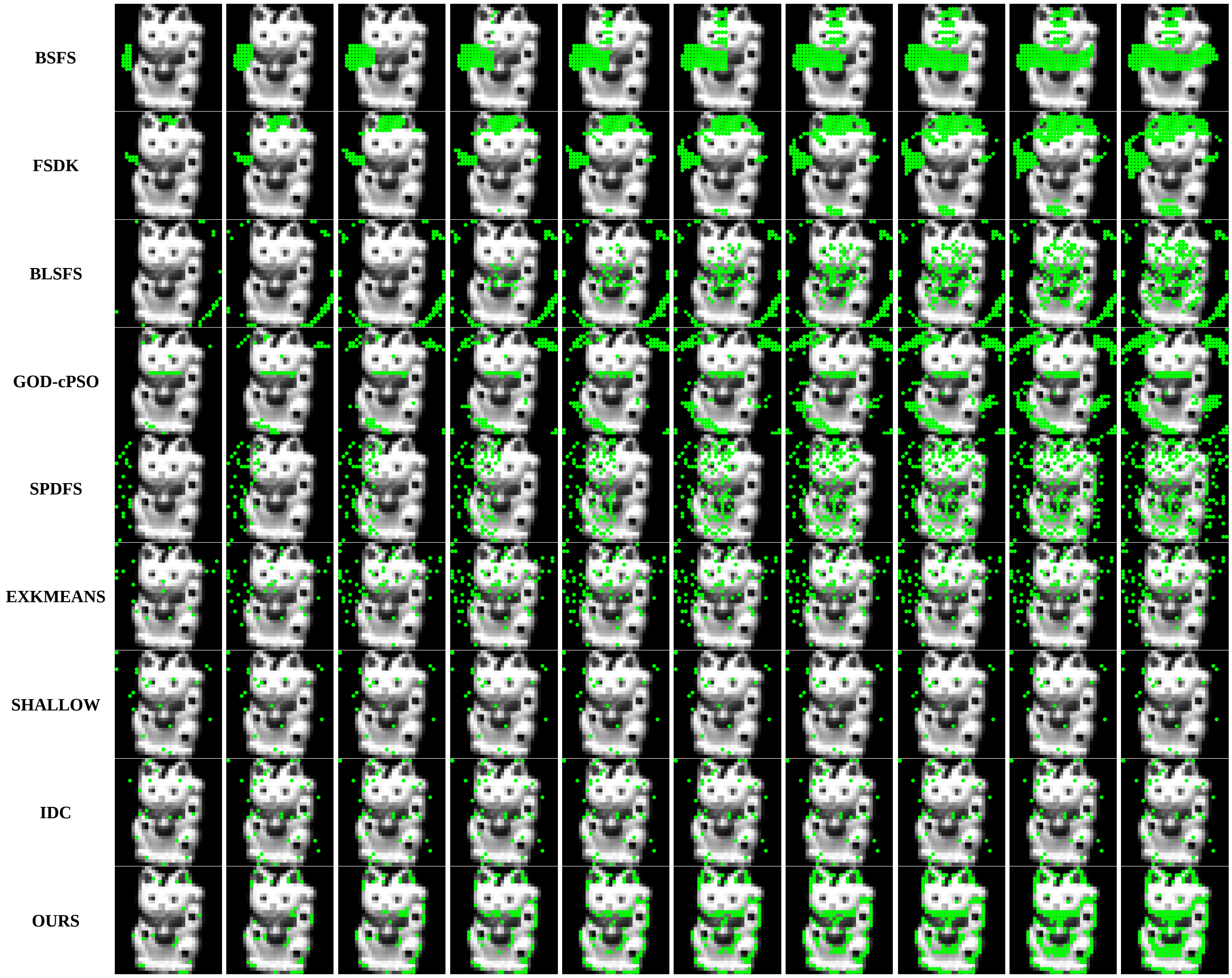
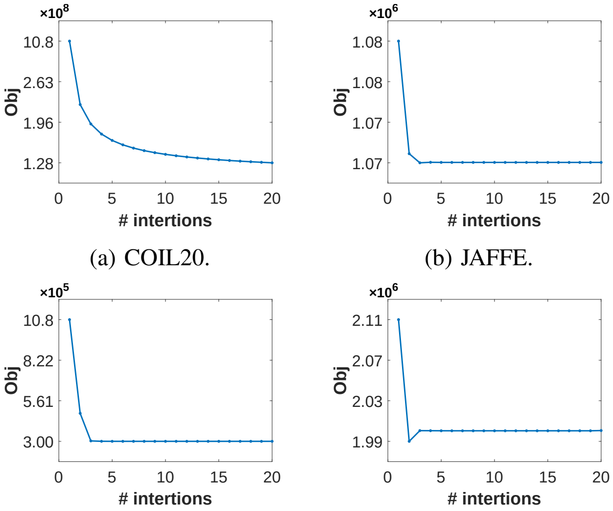


Fig. 6: Comparison of Interpretability with Different Methods. The green dots represent the features selected by different methods for clustering.



(a) COIL20.

(b) JAFFE.

(c) NCI9.

(d) ORL.

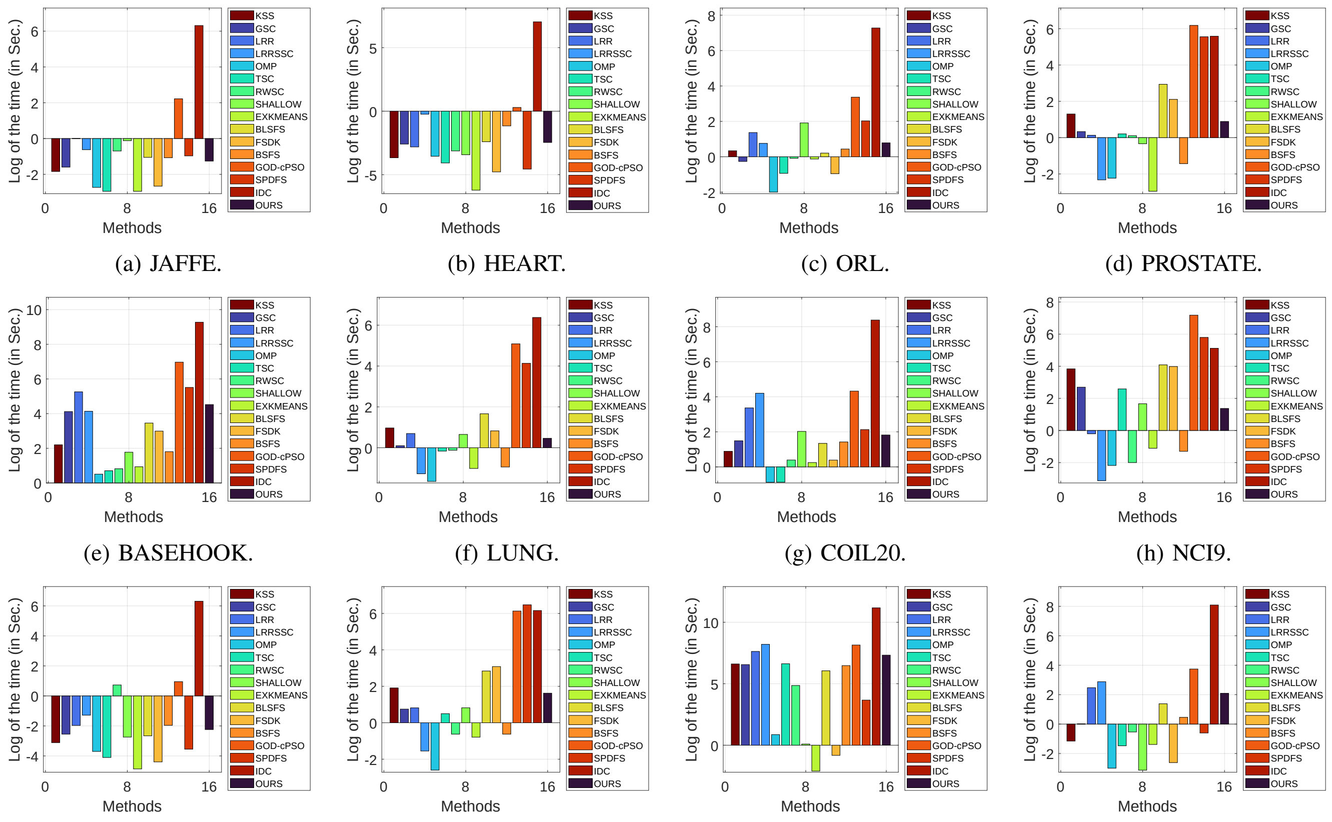
Fig. 7: Convergence curves on (a) COIL20, (b) JAFFE, (c) NCI9, (d) ORL.

G. *Parameter Study*

In this section, we show the effects of the hyperparameters λ and γ. Fig. 9 presents the ACC and NMI results on JAFFE, NCI9, PROSTATE, and COIL20 data sets. The results on other data sets are similar. The results demonstrate that both λ and γ can achieve stable performance across wide parameter ranges, indicating the method's low sensitivity to parameter selection. From the parameter sensitivity, we recommend selecting the parameter γ from the range $\{1, 10, 100\}$ and λ from the range $\{10^6, 10^7, 10^8\}$ for a new data set, which can often achieve a relatively good performance.

VI. Conclusion

In this paper, we proposed a novel Interpretable Subspace Clustering method. Compared to existing methods, ISC differs in two key aspects. On one hand, unlike other subspace clustering methods, which focus on improving clustering performance by learning structural information from data, our approach primarily focuses on enhancing the interpretability of clustering and improving the clustering performance with interpretability. On the other hand, in contrast to conventional

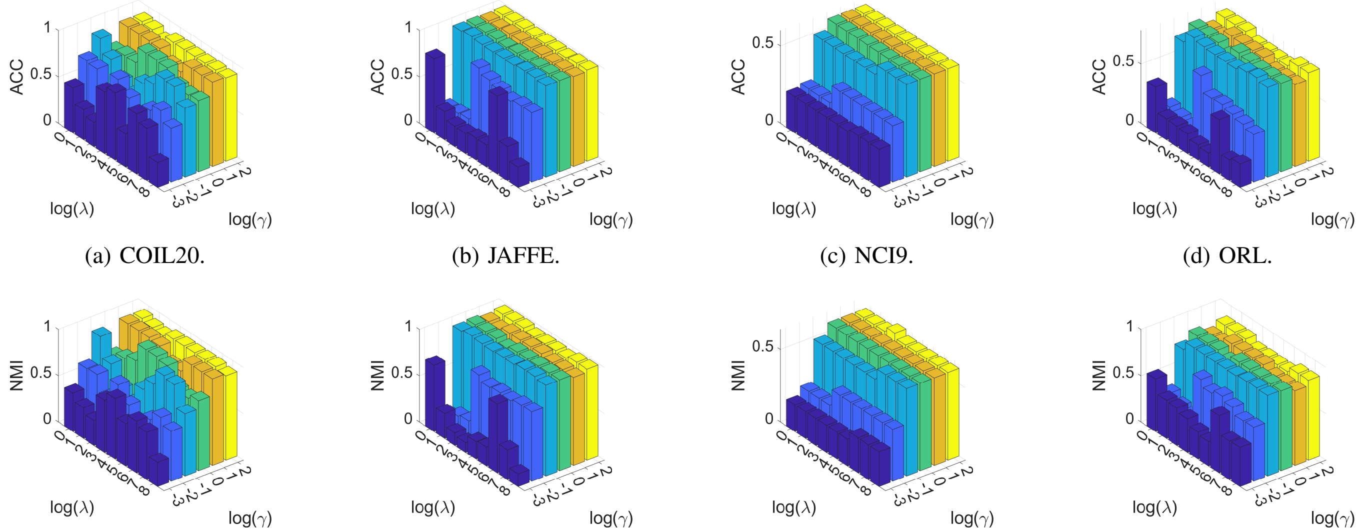


(a) JAFFE. (b) HEART. (c) ORL. (d) PROSTATE.

(e) BASEHOOK. (f) LUNG. (g) COIL20. (h) NCI9.

(i) SONAR. (j) TOX171. (k) MAGIC. (l) AMAZON.

Fig. 8: Running time of all methods on all data sets.



(a) COIL20. (b) JAFFE. (c) NCI9. (d) ORL.

(e) COIL20. (f) JAFFE. (g) NCI9. (h) ORL.

Fig. 9: Clustering results on COIL20, JAFFE, NCI9 and ORL with respect to different values of λ and γ. (a) ACC on COIL20. (b) ACC on JAFFE. (c) ACC on NCI9. (d) ACC on ORL. (e) NMI on COIL20. (f) NMI on JAFFE. (g) NMI on NCI9. (h) NMI on ORL.

TABLE IV: An example of selected features on AMAZON data set. AMAZON data set is a text data set of reviews on AMAZON website. The features of AMAZON are words. The data set has two classes: the positive reviews and the negative reviews. We show 5 reviews in each cluster. In each review, we select the top one feature (i.e., the top one word) according to our $\mathbf{W}$, which is shown in red color.

POSITIVE	NEGATIVE
This phone works **great**.	It was a **waste** of my money.
Has been working **great**.	don't **waste** your money.
I would definitely recommend the Jabra BT250v for those who are looking for **comfort**, clarity and a great price!	**Unfortunately** the ability to actually know you are receiving a call is a rather important feature and this phone is pitiful in that respect.
The only good thing was that it fits **comfortably** on small ears.	**Unfortunately** it's easy to accidentally activate them with the gentle-touch buttons if you accidentally touch the phone to your face while listening.
I was very **impressed** with the price of the cases.	This phone tries very hard to do everything but **fails** at it's very ability to be a phone.

TABLE V: Ablation Study.

Methods	Metric	JAFFE	HEART	ORL	PROSTATE	BASEHOOK	LUNG	COIL20	NCI9	SONAR	TOX171	MAGIC	AMAZON
W.O.	ACC	**1.0000**	0.7162	0.7075	0.5686	0.6016	0.8670	**0.9201**	0.5500	0.5712	0.444	0.5873	0.5200
$\|\|\mathbf{W}\|\|_G + \|\mathbf{WY}\|_G$	NMI	**1.0000**	0.1398	0.8517	0.0137	0.0304	0.6662	**0.9531**	0.5584	0.0121	0.2185	0.0255	0.0048
W.O.	ACC	**1.0000**	**0.8911**	0.7040	0.7157	0.9754	**0.8966**	0.9174	**0.6000**	**0.9663**	0.5965	**0.6736**	0.6380
$\|\|\mathbf{W}\|\|_G$	NMI	**1.0000**	**0.5123**	0.8416	0.1411	0.8471	**0.7145**	0.9509	**0.6047**	**0.8188**	0.3474	**0.0395**	0.0557
W.O.	ACC	**1.0000**	0.7162	0.6983	0.5098	0.6011	0.8818	**0.9201**	**0.6000**	0.6346	0.4845	0.6615	0.5280
$\|\|\mathbf{WY}\|\|_G$	NMI	**1.0000**	0.1398	0.8386	0.0001	0.0327	0.6639	**0.9531**	**0.6047**	0.0517	0.2778	0.0210	0.0092
ISC	ACC	**1.0000**	**0.8911**	**0.7235**	**0.7255**	**0.9905**	**0.8966**	0.9174	**0.6000**	**0.9663**	**0.6082**	0.6545	**0.6530**
	NMI	**1.0000**	**0.5123**	**0.8507**	**0.1538**	**0.9225**	**0.7145**	0.9509	**0.6047**	**0.8188**	**0.3189**	0.0270	**0.0687**

feature selection methods that uniformly select the same discriminative features for all samples, our method adopts a sample-specific feature selection strategy. By identifying and presenting the most important features for each sample and each subspace with our new proposed regularized terms, our model achieved interpretable results. Furthermore, extensive experiments on benchmark data sets by comparing with state-of-the-art clustering methods shew our superiority in interpretability and clustering performance.

Although our ISC can achieve better clustering performance and interpretability, it has a relatively high computational complexity. In the future, we will study the speedup strategy to tackle the scalability issue.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China grants 62576002, 62176001, and 62306002. This work is supported by the National Science Fund for Distinguished Young Scholars of China grants 62325604. This work is also supported by the Natural Science Project of Anhui Provincial Education Department grants 2023AH030004.

REFERENCES

[1] R. Zhang, Y. Chen, C. Wu, and F. Wang, "Cluster-driven gnn-based federated recommendation with biased message dropout," in *Proc. IEEE Int. Conf. Multimedia Expo*. IEEE, 2023, pp. 594–599.

[2] J. Nie, Z. Zhao, L. Huang, W. Nie, and Z. Wei, "Cross-domain recommendation via user-clustering and multidimensional information fusion," *IEEE Trans. Multimedia.*, vol. 25, pp. 868–880, 2021.

[3] J. Yang and C.-T. Lin, "Autonomous clustering by fast find of mass and distance peaks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 7, pp. 5336–5349, 2025.

[4] J. Zhou, C. Huang, C. Gao, Y. Wang, W. Pedrycz, and G. Yuan, "Reweighted subspace clustering guided by local and global structure preservation," *IEEE Trans. Cybern.*, vol. 55, no. 3, pp. 1436–1449, 2025.

[5] Y. Chen, Z. Wang, and X. Bai, "Fuzzy sparse subspace clustering for infrared image segmentation," *IEEE Trans. Image Process.*, vol. 32, pp. 2132–2146, 2023.

[6] T. Naous, S. Sarkar, A. Abid, and J. Zou, "Clustering plotted data by image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 21 499–21 504.

[7] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y.-D. Shen, "Robust multiple kernel k-means using l21-norm." in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 15, 2015, pp. 3476–3482.

[8] P. Zhou, R. Li, and L. Du, "Fair kernel k-means: from single kernel to multiple kernel," *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 37, pp. 99 686–99 714, 2024.

[9] X. Liu, "Simplemkkm: Simple multiple kernel k-means," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5174–5186, 2022.

[10] X. Liu, Y. Zhang, L. Liu, C. Tang, L. Peng, L. Lan, and D. Hu, "Sample adaptive localized simple multiple kernel k-means and its application in parcellation of human cerebral cortex," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–14, 2025.

[11] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining.*, 2014, pp. 977–986.

[12] R. Wang, H. Chen, Y. Lu, Q. Zhang, F. Nie, and X. Li, "Discrete and balanced spectral clustering with scalability," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14 321–14 336, 2023.

[13] F. Nie, J. Xue, W. Yu, and X. Li, "Fast clustering with anchor guidance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 4, pp. 1898–1912, 2023.

[14] J. Yang and C. Lin, "Enhanced adjacency-constrained hierarchical clustering using fine-grained pseudo labels," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 8, no. 3, pp. 2481–2492, 2024.

[15] P. Zhou, R. Li, Z. Ling, L. Du, and X. Liu, "Fair clustering ensemble

with equal cluster capacity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 3, pp. 1729–1746, 2025.

[16] H. Yan, M. Wang, and J. Xie, "Ann-dpc: Density peak clustering by finding the adaptive nearest neighbors," *Knowledge-Based Syst.*, vol. 294, pp. 111 748–111 767, 2024.

[17] H. Lin, J. Hou, and H. Yuan, "Adaptive nearest neighbor density peak clustering based on fuzzy logic," in *Int. Conf. Pattern Recognition*. Springer, 2024, pp. 125–139.

[18] P. Wang, H. Liu, A. M.-C. So, and L. Balzano, "Convergence and recovery guarantees of the k-subspaces method for subspace clustering," in *Proc. 39th Int. Conf. Mach. Learn.* PMLR, 2022, pp. 22 884–22 918.

[19] 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, 2012.

[20] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When lrr meets ssc," in *IEEE Trans. Inf. Theory.*, vol. 65, no. 9, 2013, pp. 5406–5432.

[21] J. Bian, D. Zhao, F. Nie, R. Wang, and X. Li, "Robust and sparse principal component analysis with adaptive loss minimization for feature selection." *Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 3, pp. 3601–3614, 2024.

[22] D. Park, C. Caramanis, and S. Sanghavi, "Greedy subspace clustering," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 27. Curran Associates, Inc., 2014.

[23] 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, 2013.

[24] X. He and P. Niyogi, "Locality preserving projections," *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 16, 2003.

[25] P. Zhou, J. Chen, L. Du, and X. Li, "Balanced spectral feature selection," *IEEE Trans. Cybern.*, vol. 53, no. 7, pp. 4232–4244, 2022.

[26] Z. Hu, J. Wang, K. Zhang, W. Pedrycz, and N. R. Pal, "Bi-level spectral feature selection," *Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 4, pp. 6597–6611, 2024.

[27] F. Nie, Z. Ma, J. Wang, and X. Li, "Fast sparse discriminative k-means for unsupervised feature selection," *Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 9943–9957, 2024.

[28] P. Zhou, L. Du, X. Li, Y.-D. Shen, and Y. Qian, "Unsupervised feature selection with adaptive multiple graph learning," *Pattern Recognition*, vol. 105, pp. 107 375–107 389, 2020.

[29] X. Zhang, M. Fan, D. Wang, P. Zhou, and D. Tao, "Top-k feature selection framework using robust 0-1 integer programming," *Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 3005–3019, 2021.

[30] S. Wang, F. Nie, Z. Wang, R. Wang, and X. Li, "Outliers robust unsupervised feature selection for structured sparse subspace," *Trans. Knowl. Data Eng.*, vol. 36, no. 03, pp. 1234–1248, 2024.

[31] J. Zhong, R. Shang, S. Xu, and Y. Li, "Graph embedding orthogonal decomposition: A synchronous feature selection technique based on collaborative particle swarm optimization," *Pattern Recognition*, vol. 152, p. 110453, 2024.

[32] X. Dong, F. Nie, L. Tian, R. Wang, and X. Li, "Unsupervised discriminative feature selection with l2,0-norm constrained sparse projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[33] N. Armanfard, J. P. Reilly, and M. Komeili, "Local feature selection for data classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1217–1227, 2016.

[34] J. Svirsky and O. Lindenbaum, "Interpretable deep clustering for tabular data," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 47 314–47 330.

[35] M. Tschannen and H. Bölcskei, "Noisy subspace clustering via matching pursuits," *IEEE Trans. Inf. Theory.*, vol. 64, no. 6, pp. 4081–4104, 2018.

[36] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Trans. Inf. Theory.*, vol. 61, no. 11, pp. 6320–6342, 2015.

[37] J. Cai, J. Fan, W. Guo, S. Wang, Y. Zhang, and Z. Zhang, "Efficient deep embedded subspace clustering," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 21–30.

[38] J. Cai, Y. Zhang, S. Wang, J. Fan, and W. Guo, "Wasserstein embedding learning for deep clustering: A generative approach," *IEEE Trans. Multim.*, vol. 26, pp. 7567–7580, 2024.

[39] X. Yu, Y. Jiang, G. Chao, and D. Chu, "Deep contrastive multi-view subspace clustering with representation and cluster interactive learning," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 1, pp. 188–199, 2025.

[40] M. Moshkovitz, S. Dasgupta, C. Rashtchian, and N. Frost, "Explainable k-means and k-medians clustering," in *Proc. 37th Int. Conf. Mach. Learn.* PMLR, 2020, pp. 7055–7065.

[41] N. Frost, M. Moshkovitz, and C. Rashtchian, "Exkmc: Expanding explainable k-means clustering," *arXiv:2006.02399*, 2020.

[42] E. Laber, L. Murtinho, and F. Oliveira, "Shallow decision trees for explainable k-means clustering," *Pattern Recognition*, vol. 137, pp. 109 239–109 250, 2023.

[43] H. Hwang and S. E. Whang, "Xclusters: explainability-first clustering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 7, 2023, pp. 7962–7970.

[44] M. Fleissner, L. C. Vankadara, and D. Ghoshdastidar, "Explaining kernel clustering via decision trees," in *Proc. 20th Inte. Conf. Learn. Represent.*, 2024.

[45] M. Jiang, L. Hu, Z. He, and Z. Chen, "Interpretable multi-view clustering," *Pattern Recognition*, pp. 111 418–111 428, 2025.

[46] S. Saisubramanian, S. Galhotra, and S. Zilberstein, "Balancing the tradeoff between clustering value and interpretability," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 351–357.

[47] X. Wang, X. Liu, and L. Zhang, "A rapid fuzzy rule clustering method based on granular computing," *Appl. Soft Comput.*, vol. 24, pp. 534–542, 2014.

[48] S. Saisubramanian, S. Galhotra, and S. Zilberstein, "Balancing the tradeoff between clustering value and interpretability," in *Proc. AAAI/ACM Conf. Artif. Intell.*, 2020, pp. 351–357.

[49] E. Carrizosa, K. Kurishchenko, A. Marín, and D. R. Morales, "On clustering and interpreting with rules by means of mathematical optimization," *Comput. Oper. Res.*, vol. 154, pp. 106 180–106 199, 2023.

[50] I. Davidson, M. Livanos, A. Gourru, P. Walker, and J. V. S. Ravi, "An exemplars-based approach for explainable clustering: Complexity and efficient approximation algorithms," in *Proc. SIAM Int. Conf. Data Min.* SIAM, 2024, pp. 46–54.

[51] Y. Pan, Y. Yao, and I. Tsang, "Pc-x: Profound clustering via slow exemplars," in *Conf. Parsim. Learn.* PMLR, 2024, pp. 1–19.

[52] C. You, C. Li, D. P. Robinson, and R. Vidal, "Scalable exemplar-based subspace clustering on class-imbalanced data," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 67–83.

[53] J. Cui and L. Chu, "Interpretable deep graph-level clustering: A prototype-based approach," in *International Conference on Pattern Recognition*. Springer, 2024, pp. 114–129.

[54] I. Salles, P. Mejia-Domenzain, V. Swamy, J. Blackwell, and T. Käser, "Interpret3c: Interpretable student clustering through individualized feature selection," in *Int. Conf. Artif. Intell. Education*. Springer, 2024, pp. 382–390.

[55] D. Kong, R. Fujimaki, J. Liu, F. Nie, and C. Ding, "Exclusive feature learning on arbitrary structures via l1,2-norm," *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 27, 2014.

[56] U. Von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, pp. 395–416, 2007.

[57] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Trans. Math. Software*, vol. 23, no. 4, pp. 550–560, 1997.

[58] R. Li, H. Hu, L. Du, J. Chen, B. Jiang, and P. Zhou, "One-stage fair multi-view spectral clustering," in *Proc. 32nd ACM Int. Conf. Multimedia*, 2024, pp. 1407–1416.

[59] R. Wang, H. Chen, Y. Lu, Q. Zhang, F. Nie, and X. Li, "Discrete and balanced spectral clustering with scalability," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14 321–14 336, 2023.

[60] F. Pourkamali-Anaraki, J. Folberth, and S. Becker, "Efficient solvers for sparse subspace clustering," *Signal Processing*, vol. 172, p. 107548, 2020.

[61] M. J. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 12, pp. 1357–1362, 2002.

[62] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–45, 2017.

Zheng Zhang received the Master's degree in computer science and technology from the School of Computer Science and Technology, Anhui University, Hefei, Anhui, China in 2021-2024, respectively. He is currently pursuing the Ph.D degree with the School of Computer Sciences and Technology, Anhui University, Hefei, Anhui, China, He is current research interests include cluster, interpretable machine learning, subspace learning and data mining.

Peng Zhou received the B.E. degree in computer science and technology from University of Science and Technology of China in 2011, and Ph.D. degree in Computer Science from Institute of Software, Chinese Academy of Sciences in 2017. He is currently a Professor with the School of Computer Science and Technology, Anhui University. He has published more than 80 papers in highly regarded conferences and journals, including IEEE TPAMI, IEEE TKDE, IEEE TNNLS, IEEE TCYB, ACM TKDD, NeurIPS, IJCAI, AAAI, MM, etc. His research interests include machine learning and data mining.

Aiting Yao received her Ph.D. degree in Computer Science and Technology from Anhui University, China, in 2024. She is currently a Postdoctoral Research Fellow at Peng Cheng Laboratory. Her research interests include edge computing, privacy protection, and federated learning. During her Ph.D. studies, she was a visiting scholar and joint Ph.D. student at the School of Information Technology, Deakin University, Australia, supported by the China Scholarship Council. She has participated in several national research projects and served as the principal investigator of the Anhui Provincial Outstanding Ph.D. Dissertation Project. Her research has been published in leading journals and conferences such as FGCS, Ad Hoc Networks, Complex and Intelligent Systems, and IEEE TrustCom.

Liang Du received the B.E. degree in software engineering from Wuhan University, in 2007, and the Ph.D. degree in computer science from the Institute of Software, University of Chinese Academy of Sciences, in 2013. From July 2013 to July 2014, he was a Software Engineer with Alibaba Group. He was also an Assistant Researcher with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. He is currently an Associate Professor with Shanxi University. He has published more than 40 papers in top conferences and journals, including KDD, IJCAI, AAAI, ICDM, TKDE, SDM, and CIKM. His research interests include clustering with noise and heterogeneous data, ranking for feature selection, active learning, and document summarization.

Xinwang Liu received his PhD degree from National University of Defense Technology (NUDT), China, in 2013. He is now Professor at School of Computer, NUDT. His current research interests include kernel learning, multi-view clustering and unsupervised feature learning. Dr. Liu has published 80+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, ICML, NeurIPS, CVPR, ICCV, AAAI, IJCAI, etc. He is an Associate Editor of IEEE T-NNLS and Information Fusion Journal. More information can be found at https://xinwangliu.github.io/.