Subspace Clustering via Good Neighbors

Jufeng Yang, Jie Liang, Kai Wang, Ming-Hsuan Yang

Abstract—Finding the informative clusters of a high-dimensional dataset is at the core of numerous applications in computer vision, where spectral based subspace clustering algorithm is arguably the most widely-studied methods due to its empirical performance and provable guarantees under various assumptions. Such algorithms first compute a linear representation for each sample based on a dictionary, and construct an affinity graph for spectral clustering. It is well-known that sparsity and connectivity of the affinity graph play important rules for effective subspace clustering. However, it is difficult to simultaneously optimize both factor due to their conflicting nature, and most existing methods are designed to deal with only one factor. In this paper, we propose an algorithm to optimize both sparsity and connectivity by finding good neighbors which induce key connections among samples within a subspace. First, an initial coefficient matrix is generated from the input dataset. For each sample, we find its good neighbors which not only have large coefficients but are strongly connected to each other. We reassign the coefficients of good neighbors and eliminate other entries to generate a new coefficient matrix, which can be used by spectral clustering methods. Both theoretical and empirical results show that few good neighbors obtained by the proposed algorithm can recover the subspace effectively, and the post-processing module of finding good neighbors can be complementary to most of the subspace clustering algorithms. Experiments on five benchmark datasets show that the proposed algorithm performs favorably against the state-of-the-art methods in terms of accuracy with a negligible increase in speed.

Index Terms—Spectral based subspace clustering, good neighbors, sparsity, graph connectivity.

1 Introduction

Modeling high-dimensional data has been one of the most critical issues in computer vision. As high dimensional data usually distribute in multiple low-dimensional subspaces whose structures can be extracted and exploited, numerous subspace clustering algorithms [1]–[5] based on iterative optimization, algebraic operators, statistical analysis and spectral clustering [6] have been developed in the literature to model the data points by a union of low dimensional subspaces. This paper develops a complementary module for the spectral based subspace clustering methods.

Let $\boldsymbol{x}_i$ $(1 \leqslant i \leqslant N)$ be the representation of the i-th sample in the data matrix $\boldsymbol{X} \in \mathbb{R}^{\mathcal{D}\times N}$ where $\mathcal{D}$ denotes the dimension of each sample and N is the number of samples. All samples are assumed to lie in one of K subspaces $\{\boldsymbol{\mathcal{S}}_i\}_{i=1}^K$ with the number of samples $\{n_i\}_{i=1}^K$ and the intrinsic rank $\{d_i\}_{i=1}^K$ where $n_i > d_i$. Most recent subspace clustering algorithms [1], [6] are based on the assumption that the K subspaces are independent, and thus the samples from different subspaces can be grouped into different clusters.

The key step of spectral based subspace clustering methods is to compute the coefficient matrix $\boldsymbol{Z}$ by solving an optimization problem of the form:

$$\min_{\boldsymbol{Z}} \; L(\boldsymbol{X}\boldsymbol{Z}, \boldsymbol{X}) + \lambda \|\boldsymbol{Z}\|_\xi, \tag{1}$$

where $L(\cdot,\cdot) : \mathbb{R}^{N\times N} \rightarrow \mathbb{R}^+$ denotes the loss function, λ is the trade-off parameter, and $\|\cdot\|_\xi$ denotes the regularization term where different ξ's lead to ℓ_0, ℓ_1, ℓ_2, ℓ_∞ or the nuclear norm [6]–[9]. On one hand, the coefficient matrix $\boldsymbol{Z}$ can be interpreted as a new representation of $\boldsymbol{X}$, *i.e.*, each column $\boldsymbol{z}_i$ represents the sample $\boldsymbol{x}_i$ in terms of other samples in $\boldsymbol{X}$. It induces richer correlations than the original space due to the regularization process via constraints on sparsity, subspace-preserving property and connectivity within each subspace [10] (see Section 3.2 and 3.3). On the other hand, another interpretation for $\boldsymbol{Z}$ is to consider it as an "correlation matrix", where z_{ij} reflects the similarity of samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Therefore, from the permuted $\boldsymbol{Z}$ certain block-diagonal structure [11] can be extracted when samples lie on a union of linear subspaces. Spectral clustering methods [12] can then be applied to form a low-dimension embedding of the correlation matrix $\boldsymbol{Z}$, on which groups of data can be formed using the K-means method.

It is well-known that the ℓ_2 and nuclear norm based regularizations lead to the dense coefficient matrices [13]. Although the connectivity within subspaces is guaranteed, the coefficients of the inter-subspaces are usually non-zero. Thus, the subspace-preserving property, *i.e.*, $z_{ij} = 0$ for all $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{x}_j \notin \boldsymbol{\mathcal{S}}$, is not satisfied [14] (see Section 3.2). In contrast, the ℓ_0 and ℓ_1 norms lead to a sparse $\boldsymbol{Z}$ [13]. While the sparsity regularizations enforce the subspace-preserving properties of a data matrix, the connectivity within each subspace cannot be guaranteed [15].

Spectral clustering [12] can be considered as a segmentation process on an affinity graph $\mathcal{G}$ with N vertices, where each sample in $\boldsymbol{X}$ denotes a vertex and the weights of the edges are derived from $\boldsymbol{Z}$ [16]. Hence, the clustering performance relies on not only the sparsity of the inter-subspace connections but also the connectivity among intra-subspace samples since the segmentation in subspace clustering is based on finding the connected components of $\mathcal{G}$ [15]. To optimize both properties, several recent algorithms use the mixed norm for the regularization term in (1), *e.g.*, trace Lasso $\|\boldsymbol{X}\mathrm{diag}(\boldsymbol{Z})\|_*$ [14] and elastic net [13], to interpolate between the ℓ_1 and ℓ_2 norms adaptively. Nevertheless, these schemes do not perform consistently well on different applications depending on whether good subspace structures among data points can be identified.

For clustering, it is more efficient and effective to exploit the connections in the projected space $\boldsymbol{Z}$ rather than on the data

- *J. Yang, J. Liang and K. Wang are with the College of Computer and Control Engineering, Nankai University, Tianjin, China, 300350. E-mail: yangjufeng@nankai.edu.cn; liang27jie@163.com; wangk@nankai.edu.cn*
- *M.-H. Yang is with School of Engineering, University of California, Merced, CA 95344 USA. E-mail: mhyang@ucmerced.edu*

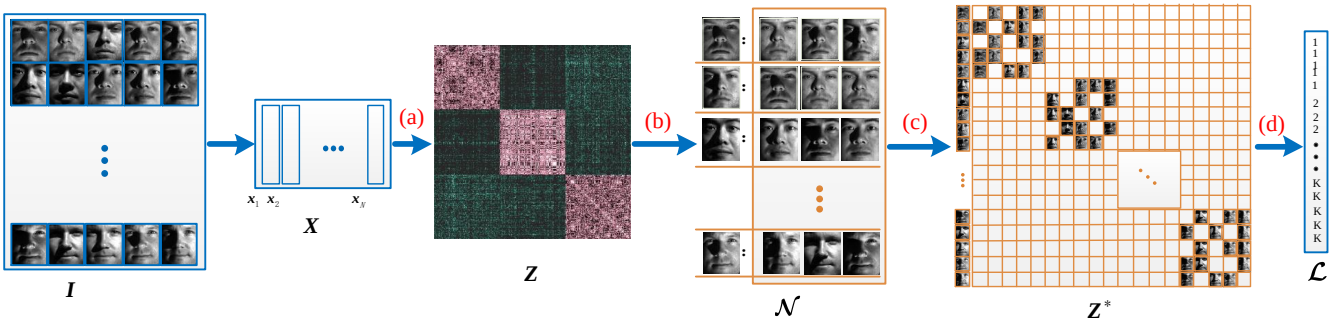


Fig. 1. Main steps of the proposed FGNSC algorithm. From the input image dataset $\boldsymbol{I}$, we generate the data matrix $\boldsymbol{X}$, each column $\boldsymbol{x}_i$ in $\boldsymbol{X}$ represents an input image. First, we compute (7) on $\boldsymbol{X}$ for the initial coefficient matrix $\boldsymbol{Z}$. Using Algorithm 1 with $\boldsymbol{Z}$, we obtain the matrix of good neighbors $\mathcal{N}$. Then, $\boldsymbol{Z}^*$ is generated by assigning new coefficients to the good neighbors and eliminating the other values. Thus $\boldsymbol{Z}^*$ maintains both sparsity and grouping effect as it has few non-zero elements yet has strong connections. Finally we implement the classic spectral clustering for the ultimate segmentation result $\mathcal{L}$. (a) Calculating (1). (b) Finding good neighbors as shown in Algorithm 1. (c) Generating $\boldsymbol{Z}^*$ by computing (15). (d) Conducting the spectral clustering.

space $\boldsymbol{X}$ [17], [18]. As shown in [19], the representations based on linear projections with different regularizations (*e.g.*, ℓ_1, ℓ_2, ℓ_∞ and nuclear norms) preserve the property of intra-subspace projection dominance (IPD), *i.e.*, the coefficients for the intra-subspace points are larger than those over inter-subspace points. However, simply preserving larger coefficients of $\boldsymbol{Z}$ does not guarantee the connectivity of each subspace [15], [20]. In this work, we preserve the key connections introduced by the proposed *good neighbors* to guarantee the latent connectivity with as few connections as possible. For each sample $\boldsymbol{x}_i$, a good neighbor $\boldsymbol{x}_j$ satisfies the following two conditions. First, the coefficient $z_{ij} \in \max_\gamma(\boldsymbol{z}_i)$ where $\max_\gamma(\cdot)$ contains the top γ values of the vector. Second, $\boldsymbol{x}_j$, together with other μ samples and $\boldsymbol{x}_i$, form a connected path where the direct connection between $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ requires $z_{pq} \in \max_\gamma(\boldsymbol{z}_p)$. We show that preserving the connections among good neighbors leads to the subspace-preserving property and the connectivity within each subspace, thereby facilitating effective subspace clustering [20].

In this paper, we propose an algorithm for finding good neighbors (FGN) which can be applied to existing subspace clustering (SC) methods. Figure 1 shows the main steps of the proposed finding good neighbors for subspace clustering (FGNSC) approach. We first generate the initial coefficient matrix $\boldsymbol{Z}$ by an off-the-shelf grouping algorithm which preserves invariability of the projection. That is, when two data points are close, the corresponding coefficients should be close, *i.e.*, $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \to 0 \Rightarrow \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 \to 0$. Next, we find good neighbors for each $\boldsymbol{x}_i$ which induce the key connections within subspace by exploiting $\boldsymbol{Z}$. Given the collection of good neighbors, we optimize the coefficient matrix $\boldsymbol{Z}$ for $\boldsymbol{Z}^*$. Concretely, if $\boldsymbol{x}_j$ is not a good neighbor of $\boldsymbol{x}_i$, we set $z^*_{ij} = 0$. Otherwise, we compute z^*_{ij} based on the original coefficients. As such, we obtain a new sparse and subspace-preserving coefficient matrix $\boldsymbol{Z}^*$, and theoretically prove that the graph constructed from $\boldsymbol{Z}^*$ has satisfactory connectivity. Similar to the existing methods [6], [21]–[23], the spectral clustering algorithm [12] is used in the last step for segmentation.

The contributions of this work are summarized as follows:

- We find good neighbors for each sample from the coefficient matrix $\boldsymbol{Z}$, and define a new metric $\mathcal{N}_e$ to evaluate the neighborhood matrix. We theoretically prove the subspace-preserving property of $\boldsymbol{Z}^*$ and connectivity of the corresponding graph $\mathcal{G}$.
- Based on the good neighbor relationship, we propose the FGNSC algorithm which can be integrated with most subspace clustering methods. In contrast to existing methods that are designed to maintain one property, the proposed method preserves both the sparsity and connectivity properties.
- We demonstrate the robustness of the FGNSC while handling the data with different assumptions and estimating the number of clusters. Experimental results on five benchmarks demonstrate that the FGNSC algorithm performs favorably against the state-of-the-art methods.

2 Related Work

Data clustering has been extensively studied in the past decades [24]–[28]. Among which, spectral based subspace clustering methods [18], [29]–[31] attract much attention due to promising performance and theoretical guarantees. The main difference among subspace clustering algorithms lies in the optimization process which learns the coefficient matrix $\boldsymbol{Z}$ with different properties, *i.e.*, sparsity and connectivity, based on different assumptions. While some schemes exploit sparsity by applying ℓ_1 or ℓ_0 minimization [32], [33] in (1), other approaches show strong connectivity with a dense coefficient matrix using ℓ_2 or nuclear regularization [10], [34], [35]. Closely related to this work are the methods that aim to bridge the gap between both properties [13], [14], [20] using mixed norms. Recently, various post-processing methods [19], [36] are proposed to further enhance the sparsity of $\boldsymbol{Z}$, but lack the guarantee of the connectivity.

2.1 Sparsity

Elhamifar and Vidal [1] propose the sparse representation based on ℓ_1 optimization as

$$\min_{\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}\|_2^2 + \lambda\|\boldsymbol{Z}\|_1 \quad \text{s.t. } \operatorname{diag}(\boldsymbol{Z}) = \boldsymbol{0}. \tag{2}$$

The obtained coefficient matrix $\boldsymbol{Z}$ recovers a sparse subspace representation but may not satisfy the graph connectivity if the dimension of the subspace is greater than three [15].

The ℓ_0 based subspace clustering methods aim to compute a sparse and subspace-preserving representation for each data

samples. Yang et al. [7] present a sparse clustering method with a regularizer based on the ℓ_0 norm by using the proximal gradient descent method. Numerous alternative methods have been proposed for ℓ_0 minimization while avoiding the non-convex problems, *e.g.*, orthogonal matching pursuit [37] and nearest subspace neighbor [33]. The scalable sparse subspace clustering by orthogonal matching pursuit (SSC-OMP) method [38] solves the following problem greedily:

$$\min_{\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}\|_2^2 \quad \text{s.t.} \ \|\boldsymbol{Z}\|_0 \leqslant \gamma. \tag{3}$$

Specifically, it compares elements in each column of $\boldsymbol{X}\boldsymbol{X}^\top$ to determine which γ positions of $\boldsymbol{z}_i$ should be non-zero and assigns the value which satisfies $\boldsymbol{x}_i = \boldsymbol{X}\boldsymbol{z}_i$. However, this general pairwise relationship does not reflect the sample correlation well especially for data pairs in the intersection of subspaces [39]. As a result, the γ positions may be incorrectly assigned and the connectivity within each subspace cannot be guaranteed.

2.2 Connectivity

An excessively sparse $\boldsymbol{Z}$ leads to unsatisfied clustering results if the non-zero elements do not contain sufficient correlations for the connectivity within each subspace [15]. As such, numerous methods have been developed for preserving more correlation information and less sparsity [11], [40]–[43].

Low-rank clustering methods [44]–[48] solve the optimization problem $\min_{\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}\|_{2,1} + \lambda\|\boldsymbol{Z}\|_*$ with the aim of generating a block diagonal solution with dense connections. However, the nuclear norm $\|\cdot\|_*$ l does not enforce subset selection well when noise exists, and $\boldsymbol{Z}$ is too dense to be an efficient representation. In [49], the least squares regression (LSR) method models the highly correlated data by minimizing the Frobenius norm of $\boldsymbol{Z}$, *i.e.*, $\|\boldsymbol{Z}\|_F = \left(\sum_{i=1}^N \sum_{j=1}^N \boldsymbol{Z}_{ij}^2\right)^{1/2}$. The smooth representation (SMR) [50] scheme emphasizes the invariability of the projection by solving the following optimization problem:

$$\min_{\boldsymbol{Z}} \alpha\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}\|_F^2 + \mathrm{tr}(\boldsymbol{Z}\tilde{\boldsymbol{L}}\boldsymbol{Z}^\top), \tag{4}$$

where $\tilde{\boldsymbol{L}}$ is the enforced Laplacian matrix and $\mathrm{tr}(\cdot)$ denotes the trace of the matrix. As LSR and SMR methods lack sparsity, the dense coefficient matrix $\boldsymbol{Z}$ retains the connections of inter-clusters which affects the clustering results.

2.3 Bridging the Gap in Optimization Model

As both sparsity and connectivity properties play important roles in spectral clustering, several methods have been developed to address two issues in one model [20], [51]–[53].

In [54], Elhamifar and Vidal introduce a tradeoff between sparsity and connectivity by optimizing

$$\min_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_1 + \lambda\|\boldsymbol{Z}\|_{r,1}. \tag{5}$$

Here, minimizing $\|\boldsymbol{Z}\|_{r,1} \triangleq \sum_{i=1}^N \|\boldsymbol{z}^i\|_2$ forces to select varying samples in sparse representations where $\boldsymbol{z}^i$ indicates the i-th row of $\boldsymbol{Z}$. Recently, You *et al.* [13] propose to balance the subspace-preserving and connectivity properties using the elastic net regularization which is defined as $\lambda\|\boldsymbol{z}\|_1 + \frac{1-\lambda}{2}\|\boldsymbol{z}\|_2^2$. Both theoretical justifications and geometric interpretation of the trade-off between subspace-preserving and the connectivity are presented [13] . Similarly, the correlation adaptive subspace segmentation method proposed by Lu *et al.* [14] takes data correlation into account by using the mixed norm of trace Lasso, *i.e.*, $\|\boldsymbol{X}\mathrm{diag}(\boldsymbol{Z})\|_*$, which adaptively interpolates between the ℓ_1 norm and the ℓ_2 norm of $\boldsymbol{Z}$. Nevertheless, the structure of the data correlations depends on the data matrix $\boldsymbol{X}$, and the norm $\|\boldsymbol{X}\mathrm{diag}(\boldsymbol{Z})\|_*$ is not effective for structure selection. Therefore, this method does not perform consistently well on different applications.

2.4 Post-Processing

For numerous vision tasks, subspaces can be dense and a sample $\boldsymbol{x}_i$ can lie on the intersection of the subspaces [**?**]. Therefore, directly optimizing on the data space may be ineffective for capturing the subspace structure of samples, especially for the optimization scheme [13] which jointly optimizes the sparsity and the connectivity properties. Recently, several post-processing modules are proposed to enhance the performance on the projected space $\boldsymbol{Z}$ [18], [55].

The L2-graph method proposed by Peng *et al.* [19] preserves the top γ values of each $\boldsymbol{z}_i$ according to the property of intra-subspace projection dominance. Although the sparsity and subspace-preserving properties can be guaranteed, some samples in $\boldsymbol{\mathcal{S}}$ may be assigned to other subspaces due to the lack of connections. Similarly, the reweighed space subspace clustering (RSSC) method [36] approximates the ℓ_0 norm minimization problem by using the ℓ_1 norm and reducing the penalty on large coefficients. It learns a weight matrix $\boldsymbol{W}$ form $\boldsymbol{Z}$ iteratively for the penalty where each element in the k-th iteration is defined as $w_{ij}^k = (|z_{ij}^k| + \epsilon)^{-1}$. While the RSSC method improves the sparsity by the close relaxation of the ℓ_0 norm, the affinity of samples from one subspace may not be a connected graph. In our work, by preserving the connections over good neighbors which not only retain high correlations but also induce key connections among samples within each subspace, the proposed FGNSC algorithm ensures the latent connectivity among each subspace while satisfying the subspace-preserving property.

3 Preliminaries and Problem Statement

For presentation clarify, we first summarize the notations used in the proposed algorithm in this section. We then introduce the spectral based subspace clustering algorithm which can be considered as an optimization or graph segmentation problem. Finally, we show how the proposed method improves the sparsity of the coefficient matrix while preserves both the connectivity and the subspace-preserving properties for clustering.

3.1 Notations

Each matrix is represented with a bold capital symbols, *e.g.*, $\boldsymbol{Z}$ denotes the correlation matrix derived from (1); the columns of a matrix are denoted as bold lowercase letter with different subscripts, *e.g.*, $\boldsymbol{z}_i$ denotes the i-th column of $\boldsymbol{Z}$; and the entries of a matrix are represented with lowercase letter with double subscripts, *e.g.*, z_{ij} denotes the (i, j)-th entry of $\boldsymbol{Z}$. We use Greek letter for representing the hyper-parameters in the algorithm, such as η, γ, and μ. Furthermore, we use calligraphy fonts to denote discrete structures or , e.g., $\boldsymbol{\mathcal{S}}$ for the subspace, $\mathcal{G}$ for the undirected graph and $\boldsymbol{\mathcal{N}}$ for the good neighbors.

We use two forms of the subscripts for vectors and matrices. The first one includes the normal lowercase letters which indicate the successive position of entries directly, *e.g.*, the subscript i of $\boldsymbol{x}_i$ denotes that $\boldsymbol{x}_i$ is the i-th column of $\boldsymbol{X}$. The second form

is by the collection of discrete positions. For example, the vector $\boldsymbol{i} \in \mathbb{R}^L$ represents a set of ordinals, and thus the columns in $\boldsymbol{X}$ which satisfy particular requirements can be denoted as $\boldsymbol{x}_{\boldsymbol{i}_l}$, where $0 \leq l \leq L$.

We adopt the convention of using the indicator function $\phi(\boldsymbol{a}, \boldsymbol{A})$ to compute the correlation between two groups as follows:

$$\phi(\boldsymbol{a}, \boldsymbol{A}) = \begin{cases} 1, & \text{if } \boldsymbol{a} \in \boldsymbol{A}; \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

There are two kinds of the combinations of the component $\boldsymbol{a}$ and the set $\boldsymbol{A}$. The first one is the combination of scalars and vectors, such as i and $\boldsymbol{i}$. The other one is the combination of column vectors and the corresponding matrix, such as $\boldsymbol{x}_i$ and $\boldsymbol{X}$.

3.2 Subspace Clustering

The goal of subspace clustering (SC) is to segment the data points into a union of linear subspaces. Let $\{\boldsymbol{\mathcal{S}}_i\}_{i=1}^K$ denote K linear subspaces of $\mathbb{R}^{\mathcal{D}}$, each of which is with the rank of d_i. Consider the data matrix $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ lying on the K subspaces, where N denotes the number of data samples and each $\boldsymbol{x}_i$ denotes a sample with $\mathcal{D}$ dimensions.

A typical subspace clustering method first constructs a sparse linear representation of each data sample using the remaining dataset as a dictionary [56]. It generates a coefficient matrix $\boldsymbol{Z} \in \mathbb{R}^{N \times N}$ by solving the following optimization problem:

$$\min \|\boldsymbol{Z}\|_\xi + \lambda \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}\|_2^2 \quad \text{s.t. } \operatorname{diag}(\boldsymbol{Z}) = \boldsymbol{0}, \tag{7}$$

where the regularization term $\|\boldsymbol{Z}\|_\xi$ denotes a specific norm of $\boldsymbol{Z}$, and different ξ leads to different $\boldsymbol{Z}$ in terms of sparsity and connectivity properties [13]. In (7), λ is the weighted parameter, and the constraint $\operatorname{diag}(\boldsymbol{Z})=\boldsymbol{0}$ eliminates the trivial solution of representing the samples as a linear combination of itself. Here, the sparsity induces the subspace-preserving property which has been extensively studied [38], [44], [54].

Definition 1. (Subspace-Preserving Property) *Given the coefficient matrix $\boldsymbol{Z}$ derived from (7), it holds the subspace-preserving property if and only if $z_{ij} = 0$ for all $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{x}_j \notin \boldsymbol{\mathcal{S}}$.*

The coefficient matrix $\boldsymbol{Z}$ can be interpreted in two ways. First, each column $\boldsymbol{z}_i$ of $\boldsymbol{Z}$ is considered as a new representation of the sample $\boldsymbol{x}_i$ regarding the remaining samples in $\boldsymbol{X}$, which is sparser than the original one yet preserving the determinative ability [10]. Second, each entry z_{ij} in $\boldsymbol{Z}$ reflects the correlation between the sample pair $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ [54]. As a result, it can be used to construct the affinity graph $\mathcal{G}$.

After generating $\boldsymbol{Z}$ by solving the optimization problem in (7), the next step of the subspace clustering is to infer the segmentation of the samples using the spectral clustering module [1], [16].

3.3 Graph Interpretation

Spectral based subspace clustering algorithms can be considered as a graph segmentation process [16]. In this section, we illustrate the subspace clustering method from the graph perspective.

Given a set of data samples $\boldsymbol{X}$, subspace clustering methods compute the optimization problem in (7) to generate the coefficient matrix $\boldsymbol{Z}$. We interpret $\boldsymbol{Z}$ as a correlation matrix which reflects the similarities between all sample pairs. Since we have no more information than the data samples in $\boldsymbol{X}$ and the similarities in $\boldsymbol{Z}$, we represent the data samples in the form of the affinity graph denoted by $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$. Specifically, each data sample $\boldsymbol{x}_i$ in $\boldsymbol{X}$ is a vertex $\boldsymbol{v}_i$ of the graph. We define a symmetric nonnegative affinity matrix $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ as $\boldsymbol{W} = |\boldsymbol{Z}| + |\boldsymbol{Z}|^\top$. A pair of vertices are connected if and only if the similarity w_{ij} between the corresponding samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is non-zero. The edge between the connected vertices $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is weighted by w_{ij}.

Definition 2. (Connectivity) *The connectivity property requires that the samples within same subspace form a connected component in the affinity graph $\mathcal{G}$ [15].*

After constructing the affinity graph $\mathcal{G}$, spectral clustering methods compute the graph Laplacian of $\mathcal{G}$ to detect the K clusters. Regarding the characteristics of the spectral clustering, an ideal $\mathcal{G}$ derived by $\boldsymbol{W}$ meets the following three requirements [16], [54]:

- If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected with the weight being $w_{ij} \neq 0$, then we have $\boldsymbol{x}_i \in \mathcal{S}$ and $\boldsymbol{x}_j \in \mathcal{S}$;
- If we have $\boldsymbol{x}_i \in \mathcal{S}$ and $\boldsymbol{x}_j \notin \mathcal{S}$, then $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are disconnected;
- The connections should be as few as possible, and yet guarantee the connectivity property within each subspace.

As a result, an ideal $\mathcal{G}$ has K connected components corresponding to K subspaces, which guarantees the performance of the spectral clustering [6].

For ease of analysis, we also construct the subgraph among the samples within a single subspace $\mathcal{S}$. Specifically, given $\boldsymbol{X}_{\boldsymbol{\mathcal{S}}}$ being the subset of $\boldsymbol{X}$ where $\boldsymbol{X}_{\boldsymbol{\mathcal{S}}}$ contains the samples $\boldsymbol{x} \in \boldsymbol{\mathcal{S}}$, and $\boldsymbol{W}_{\boldsymbol{\mathcal{S}}}$ being the corresponding block in $\boldsymbol{W}$, we construct the subgraph $\mathcal{G}_{\boldsymbol{\mathcal{S}}} = (\boldsymbol{X}_{\boldsymbol{\mathcal{S}}}, \boldsymbol{W}_{\boldsymbol{\mathcal{S}}})$. In this paper, we aim to eliminate the redundant connections among each $\mathcal{G}_{\boldsymbol{\mathcal{S}}}$ to generate $\mathcal{G}^*_{\boldsymbol{\mathcal{S}}}$, of which the connectivity property is preserved.

3.4 Problem Formulation

Given a set of noise-free data samples $\boldsymbol{X}$ lied on K independent subspaces, a sparse subspace clustering (SSC) algorithm embeds the correlation of the data samples into a sparse affinity graph since each sample $\boldsymbol{x} \in \boldsymbol{\mathcal{S}}_i$ ideally is best modeled by a linear combination of d_i samples in $\boldsymbol{\mathcal{S}}_i$. Here, d_i denotes the intrinsic rank of $\boldsymbol{\mathcal{S}}_i$. As a result, the subspace-preserving property and the connectivity within a subspace are guaranteed [56].

However, under a mild assumption on both $\boldsymbol{X}$ and $\boldsymbol{\mathcal{S}}$, these two properties are not well preserved due to the erroneous connections [15]. A robust SSC algorithm aims to generate a affinity graph which eliminates the effect of errors and preserves only the key connections between samples in the same subspace [54]. These methods model and minimize the errors in the objective functions, *e.g.*,

$$\min \|\boldsymbol{Z}\|_1 + \lambda \|\boldsymbol{E}\|_1 \quad \text{s.t. } \boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z} + \boldsymbol{E},\ z_{ii} = 0, \tag{8}$$

where $\boldsymbol{E} \in \mathbb{R}^{N \times N}$ contains the vectors of sparse error entries. Unfortunately, the structure of errors is always unknown in the original space and the complex convex optimization problem is complex to be solved efficiently [19].

In this paper, we propose to eliminate the effect of erroneous and redundant connections on the projection space (spanned by $\boldsymbol{Z}$) instead of the original data space. We exploit the intrinsic structure of the correlation matrix $\boldsymbol{Z}$ derived by (7) to find the key connections which induce both the subspace-preserving property and the connectivity within each subspace. We then retain only

the generated key connections and remove others to enhance the performance of detecting the connected components of the graph.

Problem 1. (Pruning Erroneous Connections) *Let $\boldsymbol{Z} \in \mathbb{R}^{N\times N}$ be the correlation matrix of $\boldsymbol{X}$ computed by (7), of which the intra-subspace projection dominance property [19] is guaranteed. The goal of the pruning process is to preserve the least connections in the affinity graph $\mathcal{G}$ while the following two properties of $\mathcal{G}$ are still satisfied:*

- *Subspace-preserving property in Definition 1;*
- *Connectivity property in Definition 2.*

4 Proposed Algorithm

In this section, We handle the connected pruning in Problem 1 with mild assumptions on both data samples and subspaces. We first define the good neighbors for subspace clustering based on the coefficient matrix $\boldsymbol{Z}$ (also termed as the correlation matrix) derived by (7). Next, we refine $\boldsymbol{Z}$ to generate $\boldsymbol{Z}^*$ based on the relationship of good neighbors. Finally, we use the spectral clustering method [12] to segment the data points.

4.1 Good Neighbors

For each data sample $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$, the subspace clustering algorithms utilize the self-expressive property, *i.e.*, each data sample can be reconstructed by a linear combination of other points in the dataset [54]. As a result, $\boldsymbol{x}_i$ can be written as

$$\boldsymbol{x}_i = \boldsymbol{X}\boldsymbol{z}_i, \quad \text{s.t. } z_{ii} = 0, \tag{9}$$

where the data matrix $\boldsymbol{X}$ is considered as a self-expression dictionary and $\boldsymbol{z}_i = [z_{1i}, z_{2i}, \cdots, z_{Ni}]$ contains the coefficients of the combination. With proper norm based regularization on $\boldsymbol{z}_i$, the model in (9) ensures the representation using samples in $\boldsymbol{\mathcal{S}}$.

However, given an arbitrary subspace $\boldsymbol{\mathcal{S}}$, we have $n > d$ where n is the number of samples in $\boldsymbol{\mathcal{S}}$ and d is the intrinsic rank of $\boldsymbol{\mathcal{S}}$. Thus, for $\boldsymbol{x}_i$ a model needs to select d samples for linear representation from n candidates, which is not unique in general [54]. Meanwhile, the samples can often lie near the intersection of multiple subspaces in real applications, and the combination may contain redundant or incorrect connections which affects the detection of the connected components.

The intra-subspace projection dominance property of the self-expression in (7) for $\xi = 1, 2, \infty$, *etc.*, has been shown by Peng *et al.* [19], which indicates that the coefficients over intra-subspace samples are likely to be larger than those over inter-subspace samples. Note in the linear representation system by $\boldsymbol{Z}$, $\boldsymbol{x}_j \in \boldsymbol{\mathcal{S}}$ may not necessarily choose $\boldsymbol{x}_i$ in its sparse representation even if $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{x}_i$ is represented by a linear combination of some points including $\boldsymbol{x}_j$ [54]. Namely, such asymmetric representation issues are likely to occur. Therefore, we introduce the symmetric nonnegative affinity matrix $\boldsymbol{W}$ as

$$\boldsymbol{W} = \frac{1}{2}(|\boldsymbol{Z}| + |\boldsymbol{Z}|^\top), \tag{10}$$

to guarantee that the nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected to each other. Similar to [18], [19], we first collect the samples with top η coefficients in $\boldsymbol{w}_i$ for $\boldsymbol{x}_i$, where the parameter η is empirically determined from experiments discussed in Section 7.2.

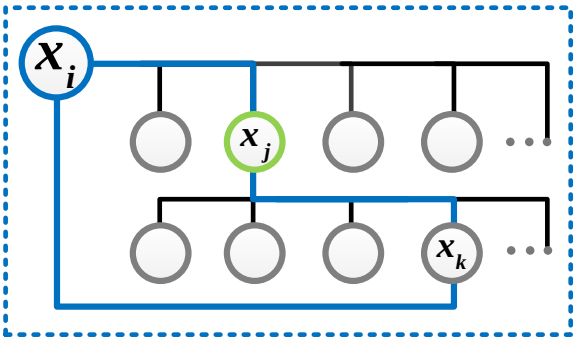


Fig. 2. Illustration of the requirement that $\boldsymbol{x}_j$ (with green circle) is the good neighbor of $\boldsymbol{x}_i$ (with blue circle) assuming $\mu = 1$. Samples with gray circle are the η-neighbors of the corresponding sample. The lines indicate the connections in the affinity graph. As shown by the blue connections, $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ are on same path as defined in Definition 5.

Definition 3. (η-neighbors) *For each sample $\boldsymbol{x}_i$, its η-neighbors (constructing the set $\boldsymbol{N}_\eta(\boldsymbol{x}_i) \in \mathbb{R}^{1\times\eta}$) are defined as follows:*

$$\boldsymbol{N}_\eta(\boldsymbol{x}_i) = \arg\max_{\boldsymbol{x}_j} \sum_{j=1}^{\eta} |w_{ij}|. \tag{11}$$

The definition of η-neighbors is in spirit similar to the work [57] which preserves the max γ entries in $\boldsymbol{w}_i$ ($\gamma < \eta < n$ where n is the number of samples in each subspace). Since larger w_{ij} usually reflects a higher similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and $d \leq \gamma < n$, the model in [57] guarantees the sparsity and subspace-preserving properties of $\boldsymbol{W}$. However, preserving only the connections with largest γ weights does not guarantee that vertices within one cluster form a connected component due to the trade-off between the sparsity and the connectivity [13]. Therefore, the spectral clustering conducted on the sparse coefficient matrix derived by [57] may potentially over-segment the samples, *i.e.*, assigning $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{x}_j \in \boldsymbol{\mathcal{S}}$ to different clusters.

To preserve both the sparsity property of $\mathcal{G}$ and connectivity property of each $\mathcal{G}_{\boldsymbol{\mathcal{S}}}$, we define the good neighbors of each sample which induce the key latent connections in a graph. For each sample $\boldsymbol{x}_i$, we preserve γ good neighbors from its η-neighbors $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ rather than γ largest entries from $\boldsymbol{w}_i$, of which the connectivity within subspace is proved in Section 5.2.

Definition 4. (Good Neighbor) *We determine whether $\boldsymbol{x}_j \in \boldsymbol{N}_\eta(\boldsymbol{x}_i)$ is a good neighbor of $\boldsymbol{x}_i$. Given the η-neighbors of $\boldsymbol{x}_i$,* i.e., *$\boldsymbol{N}_\eta(\boldsymbol{x}_i)$, if there are μ samples $\{\boldsymbol{x}_{\boldsymbol{i}_l}\}_{l=1}^{\mu} \subset \boldsymbol{N}_\eta(\boldsymbol{x}_i)$ which satisfy:*

$$\begin{aligned}&\phi(\boldsymbol{x}_j, \boldsymbol{N}_\eta(\boldsymbol{x}_{\boldsymbol{i}_1})) \times \left(\prod_{l=2}^{\mu} \phi(\boldsymbol{x}_{\boldsymbol{i}_{l-1}}, \boldsymbol{N}_\eta(\boldsymbol{x}_{\boldsymbol{i}_l}))\right) \times \\ &\phi(\boldsymbol{x}_{\boldsymbol{i}_\mu}, \boldsymbol{N}_\eta(\boldsymbol{x}_i)) \times \phi(\boldsymbol{x}_i, \boldsymbol{N}_\eta(\boldsymbol{x}_j)) = 1,\end{aligned} \tag{12}$$

where $\mu \leq \eta$, $\boldsymbol{i} \in \mathbb{R}^\mu$ is the set of ordinals and ϕ is the indicator function, then $\boldsymbol{x}_j$ is the good neighbor of $\boldsymbol{x}_i$.

For each sample $\boldsymbol{x}_i$, we consider its η-neighbors $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ as the candidates of good neighbors. We analyze the candidates sequentially to generate γ good neighbors of $\boldsymbol{x}_i$ where the parameter $\gamma < \eta$ is determined by the intrinsic rank of each subspace and will be verified via experiment in Section 7.2. If there are less than γ candidates which satisfy the condition in (12), we consider the candidates with largest coefficients as the relaxed good neighbors.

We then present the graph representation of good neighbors. For samples $\boldsymbol{X}_{\boldsymbol{\mathcal{S}}}$ in each subspace $\boldsymbol{\mathcal{S}}$, we extract only the affinities

$\boldsymbol{W}_{\boldsymbol{\mathcal{S}}}$ between the samples and its η-neighbors to construct a new subgraph $\mathcal{G}'_{\boldsymbol{\mathcal{S}}} = (\boldsymbol{X}_{\boldsymbol{\mathcal{S}}}, \boldsymbol{W}_{\boldsymbol{\mathcal{S}}})$. Each sample and one of its η-neighbors are connected via a path, which can be extended to a longer connection with multiple samples.

Definition 5. (Good Neighbor: A Graph Interpretation) *We determine whether $\boldsymbol{x}_j \in \boldsymbol{N}_\eta(\boldsymbol{x}_i)$ is a good neighbor of $\boldsymbol{x}_i$. Given the subgraph $\mathcal{G}'_{\boldsymbol{\mathcal{S}}}$ of which we have $\boldsymbol{x}_i \in \boldsymbol{V}_{\boldsymbol{\mathcal{S}}}$, and the η-neighbors $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ which are connected with $\boldsymbol{x}_i$. If there is a path between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with other μ samples $\{\boldsymbol{x}_{\boldsymbol{i}_l}\}_{l=1}^{\mu} \subset \boldsymbol{N}_\eta(\boldsymbol{x}_i)$, where $\mu \leq \eta$, $\boldsymbol{i} \in \mathbb{R}^\mu$ is the set of ordinals, then $\boldsymbol{x}_j$ is the good neighbor of $\boldsymbol{x}_i$.*

The good neighbor of $\boldsymbol{x}_i$, *e.g.*, $\boldsymbol{x}_j$, is required to be connected to not only $\boldsymbol{x}_i$ but also other μ samples selected from $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ which form a path to $\boldsymbol{x}_i$. Therefore, more latent connections between $\boldsymbol{x}_i$ and its good neighbor $\boldsymbol{x}_j$ are guaranteed while existing a typical subspace closeting method only ensures one connection for each sample pairs.

Figure 2 shows the requirement that $\boldsymbol{x}_j$ is the good neighbor of $\boldsymbol{x}_i$ when we set $\mu = 1$. In this case, $\boldsymbol{x}_j$ belongs to the η-neighbors of $\boldsymbol{x}_i$, $\boldsymbol{x}_k$ belongs to the η-neighbors of $\boldsymbol{x}_j$ and sequentially $\boldsymbol{x}_i$ belongs to the η-neighbors of $\boldsymbol{x}_k$. Namely, the good neighbor of each sample ensures a local connectivity among its η-neighbors. For $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$, we find $\gamma \leq \frac{n}{\mu}$ good neighbors, where n denotes the number of samples in $\boldsymbol{\mathcal{S}}$. Here, we set $\gamma < \eta$ while η is sufficiently large to support the selection process of good neighbors. In the remainder, we use $\boldsymbol{\mathcal{N}} \in \mathbb{R}^{\gamma \times N}$ to determine the collection of good neighbors, of which $\boldsymbol{\mathcal{N}}_i \in \mathbb{R}^\gamma$ denotes the set of good neighbors of $\boldsymbol{x}_i$. We use γ to control the sparsity and μ to control the connectivity. We discuss the details of these parameters in the next section.

4.2 Finding Good Neighbors for Subspace Clustering

In this section, we introduce a method to improve the subspace clustering algorithms by finding good neighbors. We first generate the good neighbor matrix $\boldsymbol{\mathcal{N}}$ according to the Definition 4 and 5. We then solve Problem 1 by updating $\boldsymbol{Z}$ for a new coefficient matrix $\boldsymbol{Z}^*$. Finally, we exploit the structure of the collection of multi-subspace data using a spectral clustering algorithm.

The main step for Finding Good Neighbors (FGN) are summarized in Algorithm 1. Given the collection of N data samples $\{\boldsymbol{x}_i\}_{i=1}^N$ which lie in a union of K subspaces, the FGN method first computes the coefficient matrix $\boldsymbol{Z}$ via an off-the-shelf optimization function in (7). Then the symmetric nonnegative affinity matrix $\boldsymbol{W}$ is obtained via (10).

For each sample $\boldsymbol{x}_i$, $i \in [1, 2, \cdots, N]$, we aim to find its γ good neighbors which satisfy Definition 4. We first generate its η-neighbors $\boldsymbol{N}_\eta(\boldsymbol{x}_i) \in \mathbb{R}^\eta$ by (11), where we set $\eta > \gamma$ for the pruning process. The samples in $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ are the candidate of the γ good neighbors of $\boldsymbol{x}_i$.

To determine whether $\boldsymbol{x}_j \in \boldsymbol{N}_\eta(\boldsymbol{x}_i)$ is a good neighbor of $\boldsymbol{x}_i$, we design a scoring scheme by traversing on $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ instead of using (12) which is NP-hard. We instantiate the process by setting $\mu = 1$, denoting that the path between $\boldsymbol{x}_i$ and one of its good neighbors must contain one different neighbor of $\boldsymbol{x}_i$. Given $\boldsymbol{x}_j \in \boldsymbol{N}_\eta(\boldsymbol{x}_i)$ and $\boldsymbol{N}_\eta(\boldsymbol{x}_j) = \{\boldsymbol{x}_{\boldsymbol{j}_m}\}_{m=1}^\eta$, we compute the score s_{ij} for $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ by the following equation:

$$s_{ij} = \sum_{\mathrm{m}=1}^{\eta} \phi(i, \boldsymbol{N}_\eta(\boldsymbol{x}_{\boldsymbol{j}_m})). \tag{13}$$

Algorithm 1 : Finding Good Neighbors (FGN)

Input: $\boldsymbol{Z} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N] \in \mathbb{R}^{N \times N}$, γ, η, ε.

```
compute the affinity matrix W by (10);
for i = 1 : N do
    compute the η-neighbors N_η(x_i) = {x_{j_l}}_{l=1}^η by (11);
    m = 1;
    for l = 1 : η do
        compute s_il for x_{i_l} ∈ N_η(x_i) using (13);
        if s_il > ε & m < γ then
            N_i = N_i ∪ x_{i_l};
            m = m + 1;
        end if
        if m > γ then
            N_i = N_i ∪ x̃ where x̃ is computed by (14);
        end if
    end for
end for
```

Output: Good neighbor matrix $\boldsymbol{\mathcal{N}}$.

Algorithm 2 : Finding Good Neighbors for Subspace Clustering

Input: $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{\mathcal{D} \times N}$, K, η, γ, ε.

```
Generate Z ∈ R^{N×N} via (7).
Use Algorithm 1 to obtain the good neighbor matrix N.
for i = 1 : N do
    compute z*_ij in Z* via (15).
end for
Let W* = 1/2(|Z*| + |Z*|^⊤) and compute segmentation from
    W* by spectral clustering [12].
```

Output: Labels of samples $\boldsymbol{\mathcal{L}} \in \mathbb{R}^N$.

We consider $\boldsymbol{x}_j$ as the good neighbor of $\boldsymbol{x}_i$ whenever $s_{ij} > \varepsilon$ where ε is a threshold which is discussed in Section 7.2.

We obtain each $\boldsymbol{\mathcal{N}}_i$ by picking the first γ candidates from the corresponding $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ which satisfy the aforementioned requirement. If we cannot find γ good neighbors for $\boldsymbol{x}_i$, we select the candidate $\tilde{\boldsymbol{x}}$ which holds the largest similarity in $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ for relaxation:

$$\tilde{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}_{\boldsymbol{i}_l}} |w_{i\boldsymbol{i}_l}|, \tag{14}$$

where $w_{i\boldsymbol{i}_l}$ indicates the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{\boldsymbol{i}_l}$.

We update a new coefficient matrix $\boldsymbol{Z}^*$ according to the good neighbor matrix $\boldsymbol{\mathcal{N}}$ generated from Algorithm 1. We compute each entry z^*_{ij} of $\boldsymbol{Z}^*$ by:

$$z^*_{ij} = \begin{cases} (w_{ij}) \,/\, \left(\sum_{l=1}^{\gamma} w_{i\boldsymbol{i}_l}\right), & \text{if } \boldsymbol{x}_j \in \boldsymbol{\mathcal{N}}_i; \\ 0, & \text{if } \boldsymbol{x}_j \notin \boldsymbol{\mathcal{N}}_i, \end{cases} \tag{15}$$

where $\boldsymbol{W}$ is the affinity matrix computed by (10) and $\boldsymbol{i}$ is the set of ordinals of the good neighbors of $\boldsymbol{x}_i$. Obviously, we have $0 \leq z^*_{ij} < 1$. We introduce the normalization process on the similarities to handle the problem that the followed spectral clustering puts more emphasis on keeping the stronger connections in the graph [16], [54].

After generating the updated coefficient matrix $\boldsymbol{Z}^*$, the next step is to infer the subspace structure of the data samples using

the sparse coefficients. To address this problem, we first compute the new affinity matrix $\boldsymbol{W}^*$ by

$$\boldsymbol{W}^* = \frac{1}{2}(|\boldsymbol{Z}^*| + |\boldsymbol{Z}^*|^\top). \tag{16}$$

Therefore, each node $\boldsymbol{x}_i$ connects itself to one of its good neighbors $\boldsymbol{x}_j \in \boldsymbol{\mathcal{N}}_i$ by an edge with the weight being $\frac{1}{2}(|z^*_{ij}| + |z^*_{ji}|)$. We construct a new affinity graph $\mathcal{G}^*$ according to $\boldsymbol{W}^*$. Subsequently, the normalized cut method [12] is applied to $\mathcal{G}^*$ in a way similar to [11], [42], [58] to generate the final segmentation results, *i.e.*, the labels of the samples $\boldsymbol{\mathcal{L}} \in \mathbb{R}^N$. The main procedures of FGNSC are illustrated in Figure 1 and summarized in Algorithm 2.

The proposed FGNSC algorithm is a general post-processing module that can be complementary to other spectral clustering algorithms, especially for those developed based on dense coefficient matrices. It transforms the coefficient matrix $\boldsymbol{Z}$ into a sparser $\boldsymbol{Z}^*$, and preserves the connectivity within each subspace by finding good neighbors for each sample.

5 Theoretical Analysis

In this section, we present the theoretical analysis of the subspace-preserving and the connectivity properties for the $\boldsymbol{Z}^*$ generated by the proposed algorithm.

5.1 Subspace-Preserving Property

The intrinsic requirement for the success of the spectral based subspace clustering methods is that the optimization process recovers a linear representation of each sample [54]. Specifically, the non-zero entries of the representation $\boldsymbol{z}_i$ should correspond to the subspace of the given sample $\boldsymbol{x}_i$. We prove the consistency of the proposed post-processing method by introducing the following lemma.

Lemma 1. *(**Intra-subspace projection dominance** [19]) The correlation space $\boldsymbol{Z}$ derived by the ℓ_1, ℓ_2 or nuclear norm based linear projection has the intra-subspace projection dominance property,* i.e., *for all $\boldsymbol{x}_p, \boldsymbol{x}_q \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{x}_k \notin \boldsymbol{\mathcal{S}}$, we have $z_{pq} \geq z_{pk}$ [19].*

Proposition 1. *The obtained $\boldsymbol{Z}^*$ by the proposed algorithm has the **subspace-preserving property** as defined in Definition 1,* i.e., *$z^*_{ij} = 0$ for all $x_i \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{x}_j \notin \boldsymbol{\mathcal{S}}$.*

Proof. For each sample $\boldsymbol{x}_i$, we compute its η-neighbors $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ from (11) which selects the connections with top η values of the $\boldsymbol{w}_i$. Considering the IPD in Lemma 1 and $\eta < n$ where n is the number of samples in one subspace, it follows that the candidate η-neighbors of $\boldsymbol{x}_i$ are in the same subspace with $\boldsymbol{x}_i$, *i.e.*, $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$ if and only if $\boldsymbol{N}_\eta(\boldsymbol{x}_i) \subset \boldsymbol{\mathcal{S}}$. Since we generate the good neighbors from the η-neighbors, we have $\boldsymbol{\mathcal{N}}_i \subset \boldsymbol{\mathcal{S}}$. Therefore, as we set $z^*_{ij} = 0$ if $\boldsymbol{x}_j \notin \boldsymbol{\mathcal{N}}_i$ as shown in (15), the subspace-preserving property of $\boldsymbol{Z}^*$ is thus proved. □

Note the IPD property [19] is enforced from the optimization schemes based on various norms, including the ℓ_1-, ℓ_2-, ℓ_∞ norm and the nuclear norm. Therefore, the coefficient matrix $\boldsymbol{Z}^*$ derived by the proposed FGNSC algorithm has the subspace-preserving property when combined with most existing linear representation schemes.

5.2 Connectivity

Conventional subspace clustering methods [13], [20], [59] enforces the connectivity property by preserving dense connections within each subspace. In this paper, we exploit the latent connectivity for each subspace with a sparse affinity graph of which each connection induces multiple correlations between samples. We first introduce the lemma for proving the connectivity property preserved by the proposed algorithm.

Lemma 2. *(**Connected graph** [60]) Given the graph $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$, it is connected if and only if $\mathcal{G}$ meets the following requirement:*

- $\forall \boldsymbol{v}_i, \boldsymbol{v}_j$: $e_{i\boldsymbol{i}_1} \times e_{\boldsymbol{i}_1\boldsymbol{i}_2} \times \cdots \times e_{\boldsymbol{i}_l j} \neq 0$,

where $\boldsymbol{i} \in \mathbb{R}^{l-2}$ contains the ordinals of the samples which lie on the path between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, andl is the length of the path (number of samples on the path).

Note if we have $l = 2$, then the graph $\mathcal{G}$ is fully connected. To verify the connectivity property as in Definition 2, we need to prove that each subgraph $\mathcal{G}_{\boldsymbol{\mathcal{S}}} = (\boldsymbol{V}_{\boldsymbol{\mathcal{S}}}, \boldsymbol{E}_{\boldsymbol{\mathcal{S}}})$ is connected as defined in Lemma 2. We instantiate the verification process with an arbitrary sample $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$ such that $\boldsymbol{x}_i$ is connected to the other $n-1$ points in $\boldsymbol{\mathcal{S}}$.

Proposition 2. *The subgraph $\mathcal{G}_{\boldsymbol{\mathcal{S}}} = (\boldsymbol{X}_{\boldsymbol{\mathcal{S}}}, \boldsymbol{W}^*_{\boldsymbol{\mathcal{S}}})$ is a connected graph if the parameter γ and μ satisfy:*

$$(\mu+1)\gamma^2 + \mu\gamma + 2 \geq n. \tag{17}$$

Here, $\boldsymbol{X}_{\boldsymbol{\mathcal{S}}} = \{\boldsymbol{x}_i\}_{i=1}^n$ contains the samples in $\boldsymbol{\mathcal{S}}$ and n is the number of samples in $\boldsymbol{\mathcal{S}}$. The $\boldsymbol{W}^*_{\boldsymbol{\mathcal{S}}}$ is a block of $\boldsymbol{W}^*$ in (16) derived from $\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{S}}}$ where $\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{S}}} = \{\boldsymbol{x}_{\boldsymbol{i}_l}\}_{l=1}^n$ denotes the matrix of the good neighbors for the samples in $\boldsymbol{\mathcal{S}}$.

Proof. We aim to verify the existence of $n-1$ connections for an arbitrary sample $\boldsymbol{x}_i$ either directly or latently.

Note $\boldsymbol{x}_i$ is connected to each of its good neighbors directly according to (15). Given the set of γ good neighbors for $\boldsymbol{x}_i$, *i.e.*, $\{\boldsymbol{x}_{\boldsymbol{i}_l}\}_{l=1}^\gamma$, the connectivity between $\boldsymbol{x}_i$ and these $u_{(1)}{=}\gamma$ points is guaranteed. In addition, for each of its good neighbors $\boldsymbol{x}_j$, there are μ neighbors on the path between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as shown in Definition 5. Therefore, $\boldsymbol{x}_i$ is guaranteed to be connected with another $v_{(1)}$ samples latently where $\mu \leq v_{(1)} \leq \gamma\mu$.

Next, we consider the connections derived from the good neighbors $\{\boldsymbol{x}_{\boldsymbol{i}_l}\}_{l=1}^\gamma$. For each $\boldsymbol{x}_{\boldsymbol{i}_l}$, it brings $u_{(2)_l} + v_{(2)_l}$ new connections by the samples in $\boldsymbol{\mathcal{N}}_{\boldsymbol{i}_l}$, where we have $0 \leq u_{(2)_l} + v_{(2)_l} \leq \gamma(\mu+1)$. Thus the γ good neighbors introduce $u_{(2)} + v_{(2)} \in [0, \gamma^2(\mu+1)]$ new connections for $\boldsymbol{x}_i$.

Note that all $\{\boldsymbol{x}_{\boldsymbol{i}_l}\}_{l=1}^\gamma$ have good neighbors, which can bring new latent connections for $\boldsymbol{x}_i$. For clear illustration, we take the strongest $\sum_{i=1}^2 u_{(i)} + v_{(i)}$ connections into account. According to Lemma 2, we have the following condition for a connected graph $\mathcal{G}$:

$$\sum_{m=1}^{2} \gamma^m(\mu+1) \geq n-1, \tag{18}$$

where n is the number of samples in $\boldsymbol{\mathcal{S}}$. The relationship among γ, μ and n can be described by $(\mu+1)\gamma^2 + \mu\gamma + 2 \geq n$ as in (17). For the worst case, G can also be connected by the following constraint:

$$\max(\gamma, \mu) \geq n-1. \tag{19}$$

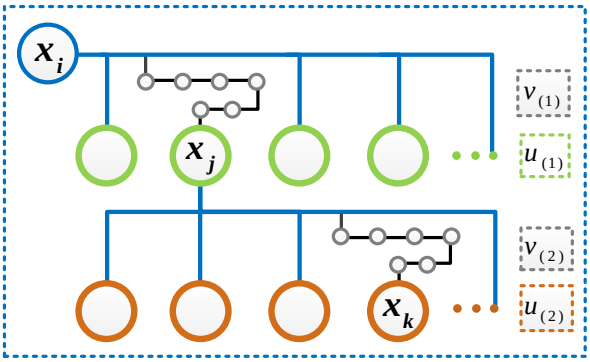


Fig. 3. Illustration of the proof on connectivity about the four types of the connections for $\boldsymbol{x}_i$ in the proof of Proposition 2, *i.e.*, $u_{(1)}$, $v_{(1)}$, $u_{(2)}$ and $v_{(2)}$. The blue lines indicate the direct connections while the black lines indicate the latent ones.

□

Figure 3 shows a diagram of four types of connections, *i.e.* $u_{(1)}$, $v_{(1)}$, $u_{(2)}$ and $v_{(2)}$. Note the worst cases that $v_{(1)}{=}\mu$ and $u_{(2)} + v_{(2)}{=}0$ correspond to a dense local geometry (μ samples are fully connected with each other and separated from the others), which should be assigned to a new subspace rather than $\boldsymbol{\mathcal{S}}$. There could be two extreme cases as follows:

- All the $N{\times}\gamma$ good neighbors cannot be found: we choose an equal number of η-neighbors derived by (11) for relaxation, of which the model is relaxed to the sparsity-based models;
- The model requires the condition in (19), *i.e.*, $\max(\gamma, \mu) \geq n - 1$: all entries of $\boldsymbol{Z}^*$ are non-zero, of which the model is equal to the approaches aiming to promote the connectivity by preserving the dense connections.

For the other cases, our model achieves a balance between the sparsity and connectivity properties. As shown in Proposition 1 and 2, the new representation $\boldsymbol{Z}^*$ is the result of the trade-off between the interclass separation and the intra-class connectivity. Since the subsequent steps of spectral clustering segments the graph by correcting the erroneous connections in the affinity graph [16], [24] which relies on the two properties, the proposed algorithm has the theoretical guarantee of clustering performance.

6 Clustering Real-World Image Data

In real world applications, subspace clustering algorithms may not perform well due to outliers or model assumptions [6]. For example, the data may not lie perfectly on a union of subspaces due to the noise or outliers introduced by the clustering operations [54]. In addition, the number of clusters K is always unknown [44] which needs to be estimated before applying the spectral clustering. In this section, we exploit the robustness of our proposed algorithm regarding both aspects.

6.1 Handling Outliers

We analyze how of the proposed FGNSC algorithm (summarized in Algorithm 2) performs on the noisy samples with outliers. The subspace-preserving property of the correlation matrix $\boldsymbol{Z}$ does not hold in the presence of outliers as inter-cluster connections [30], [61] are introduced. In contrast to existing methods [1], [31] which optimize the noisy entries in a unified framework by incorporating a term $\boldsymbol{E}$ in (8), we remove the outliers by eliminating the wrong connections according to the mapped affinity graph.

As shown in Figure 2, given the samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the good neighborhood relationship requires not only the connection between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, but also the connections from $\boldsymbol{x}_j$ to $\boldsymbol{x}_k$ and from $\boldsymbol{x}_k$ to $\boldsymbol{x}_i$. If $\boldsymbol{x}_j$ is one of the outliers which happens to be connected with $\boldsymbol{x}_i$, we have $\boldsymbol{x}_j \notin \boldsymbol{N}_\eta(\boldsymbol{x}_k)$ with a high probability. Therefore, the proposed model is robust to the outliers while a larger μ induces better robustness.

6.2 Estimating the Number of Clusters

In most real-world applications, the number of clusters, *i.e.*, K, is usually unknown when spectral clustering [2], [27], [62] is used. Theoretically, the model estimation problem in subspace clustering, *i.e.*, estimating the number of clusters, can be handled by counting the number of non-zero eigenvalues of the Laplacian matrix $\boldsymbol{L}$ [44]. Here, the Laplacian matrix $\boldsymbol{L}$ is derived by

$$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}}, \tag{20}$$

where $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{D}$ is the diagonal degree matrix where $d_{ii} = \sum_{j=1}^{N} w_{ij}$. However, the existence of the non-zero eigenvalues relies on the strict block-diagonal property of the permuted $\boldsymbol{W}$, *i.e.*, $w_{ij} \neq 0$ only if $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{x}_j \in \boldsymbol{\mathcal{S}}$. Therefore, it is generally challenging to obtain a correct estimate due to the existence of non-zero entries [19], *i.e.*, $w_{ij} \neq 0$ but $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{x}_j \notin \boldsymbol{\mathcal{S}}$. Sequentially, Liu *et al.* [44] propose a soft thresholding approach to estimate $\widehat{K}$ by:

$$\widehat{K} = N - \text{int}\left(\sum_{i=1}^{N} f_\tau(\sigma_i)\right), \tag{21}$$

where N is the number of samples in the dataset, $\{\sigma_i\}_{i=1}^N$ denotes a set of singular values of $\boldsymbol{L}$ and int$(\cdot)$ outputs the nearest integer of a real number. The soft thresholding operator $f_\tau(\cdot)$ is defined as:

$$f_\tau(\sigma_i) = \begin{cases} 1, & \text{if } \sigma > \tau; \\ \log_2(1 + \frac{\sigma^2}{\tau^2}), & \text{otherwise}, \end{cases} \tag{22}$$

where τ is the learned threshold to eliminate the errors of $\{\sigma_i\}_{i=1}^N$.

As shown in Proposition 1, the subspace-preserving property of $\boldsymbol{Z}^*$ and thus $\boldsymbol{W}^*$ is guaranteed. In addition, as described in Section 7.4 and Figure 5, $\boldsymbol{W}^*$ holds stricter block-diagonal property than $\boldsymbol{W}$ derived by other methods. Therefore, it is easier to obtain the value of τ that is robust to variation of cluster numbers in this work.

7 Experimental Results

We first demonstrate effectiveness of the proposed good neighbors for subspace clustering and then evaluate the FGNSC algorithm against the state-of-the-art methods. In addition, we analyze the efficiency of the algorithm in terms of run-time.

7.1 Experimental Settings

In the experiments except for Section 7.6, the FGNSC algorithm consists of the smooth representation (SMR, [50]) in (4), FGN in Algorithm 1 and spectral clustering [12]. We first find a proper assignment of the parameters, *e.g.*, γ, and μ. Next, we show the

TABLE 1
Evaluated datasets. The image size used in the experiments are shown in the last column.

Datasets	# of Classes	Samples per Class	Original Size	After Resizing
EYaleB	38	64	192×168	48×42
COIL-20	20	72	448×416	64×64
MNIST	10	300	28×28	28×28
USPS	10	300	16×16	16×16
AR	100	26	165×120	48×42

effectiveness of the good neighbor matrix $\boldsymbol{\mathcal{N}}$ by contrasting the error rate of $\boldsymbol{\mathcal{N}}$ using the pre-defined metric $\mathcal{N}_e$ and the accuracy of the final labels $\boldsymbol{\mathcal{L}}$. We also compare the affinity matrix derived by several methods to verify the trade-off between the subspace-preserving property and the connectivity. Finally, we evaluate the FGNSC algorithm on five benchmark datasets. In the tables on clustering results, each entry indicates the average of fifty trials with different combinations of classes (except those of the whole datasets). The source code will be made available to the public for reproducible research.

Datasets. We conduct the experiments on the extended Yale B (EYaleB [63]), COIL-20 [64], MNIST [65], USPS [66] and AR [67] datasets. The images are resized to $p \times q$ pixels to form the data vectors $\boldsymbol{x}_i \in \mathbb{R}^{pq}$, and concatenated together to form $\boldsymbol{X}$. Table 1 shows the details of the datasets.

Evaluated methods. We compare the proposed FGNSC algorithm with the state-of-the-art subspace clustering methods including SSC (with the ADMM solver) [1], [54], spatSC [41], L2-Graph [19], LSR [49], SMR [50], SSC-OMP [38], ORGEN [13], RSSC [36], NSN [33], iPursuit [18], [55], OSC [23] and LRR [44]. In the experiments, we tune the parameters for these methods to achieve the best results.

Metrics. We evaluate all the methods using two widely-used metrics in clustering: clustering accuracy (ACC) and normalized mutual information (NMI). In addition, we define the $\mathcal{N}_e$ metric to evaluate the matrix of good neighbors. Let $\boldsymbol{g}_i$ be the set of intra-class samples for $\boldsymbol{x}_i$ generated from the ground-truth, and $\boldsymbol{n}_{ij}$ be the j-th good neighbor of $\boldsymbol{x}_i$ ($j \in [1:\gamma]$ where γ denotes the number of good neighbors for each sample). We define $\mathcal{N}_e$ by

$$\mathcal{N}_e = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{\gamma}\phi(\boldsymbol{n}_{ij},\boldsymbol{g}_i)}{N\times\gamma}, \tag{23}$$

where $\phi(\cdot,\cdot)$ is the indicator function as defined in (6). A smaller value of $\mathcal{N}_e$ indicates better performance.

7.2 Parameters

For computational efficiency, we set $\varepsilon = 1$, which means that if s_{ij} from (13) satisfies $s_{ij} \geq 1$, then $\boldsymbol{x}_j$ is considered as a good neighbor of $\boldsymbol{x}_i$.

To optimize the connectivity of each subspace, we evaluate the impact of γ and μ on a subset of 8 subjects from the extended Yale B dataset. Since the connections between samples become weaker with a longer path, we fix $\mu = 1$ and optimize γ for the best assignment. Using (17) with $\mu = 1$ and $n = 64$, we have the least $\gamma \geq \frac{\sqrt{497}-1}{4} > 5$. For each experiment, we assign the value of $\gamma \in [3:12]$ and obtain the curves as shown in Figure 4. As we

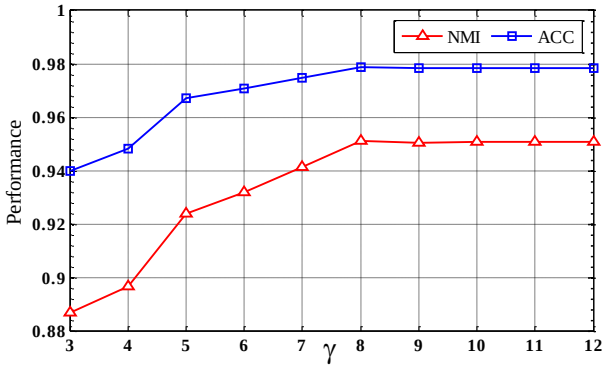


Fig. 4. Clustering performance with different γ on the extended Yale B dataset of 8 classes.

obtain the ACC= $[0.3073, 0.8281]$ and NMI= $[0.1148, 0.7528]$ with $\gamma = [1, 2]$ respectively, we do not add them to the figure for better illustration.

The curves in Figure 4 indicates that the FGNSC algorithm does not perform well with γ=$\{3, 4\}$. When γ is increased, the FGNSC algorithm achieves better performance until reaching the peak value with γ=8. Therefore, we use γ=8 and μ=1 for the rest of the experiments. Accordingly, we set $\eta = 20$ for (11) considering the balance between the computational efficiency and the effectiveness for selecting γ good neighbors.

7.3 Matrix of Good Neighbors

The coefficient matrices derived by the SSC-OMP [38] and iPursuit [18] methods is computed based on correlation of neighboring data points in the data space. In this section, we analyze the effects of $\boldsymbol{\mathcal{N}}$ by comparing the SSC-OMP, iPursuit and FGNSC methods.

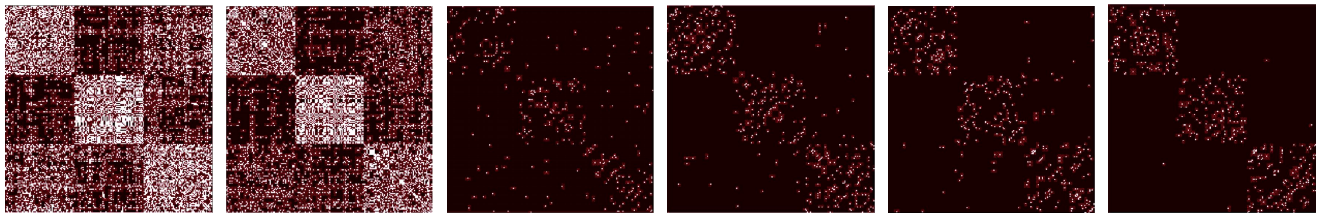
We select the images of N_c subjects from the extended Yale B dataset with $N_c \in \{3, 5, 8, 10, 15, 20, 25, 30, 35, 38\}$ and $\gamma = 8$ for all three methods. We use the $\mathcal{N}_e$ metric in (23) to evaluate the neighborhood matrix $\boldsymbol{\mathcal{N}}$ derived by these methods and ACC to evaluate the segmentation results. Table 2 shows the experimental results. Overall, the FGNSC algorithm achieves the accuracy of more than 94% on all subsets, with the error $\mathcal{N}_e$ of less than 9%. In contrast, the SSC-OMP method does not perform well. It achieves the clustering accuracy of less than 60% on the set of 38 subjects with the error rate of the neighbor matrix being 39.84%.

7.4 Affinity Matrix

We evaluate six clustering methods in terms of affinity matrix $\boldsymbol{W}$ in (10) and (16). Figure 5 shows the affinity matrices derived by the evaluated methods. We carry out the experiments on the first three digits of MNIST for clear illustration. For each digit, we choose the first 300 images to construct the subset. Figure 5(f) is the affinity matrix derived by the FGNSC algorithm. This affinity matrix is generated from the original coefficient matrix in Figure 5(b) computed by the SMR [50] method. Note the affinity matrix $\boldsymbol{W}$ in Figure 5(f) has the better sparsity and block-diagonal property, both of which are important for spectral clustering and lead to the accuracy of 98.11% and the NMI of 0.9140. In Figure 5(e), we preserve the top γ entries for each column of the $\boldsymbol{W}$ in Figure 5(b) to contrast the FGNSC algorithm with the L2-Graph in [19]. More erroneous connections are introduced in Figure 5(e) than Figure 5(f) since several digits may lie on

TABLE 2
Clustering results by the SSC-OMP, iPursuit and FGNSC methods on subsets of the extended Yale B dataset using $\mathcal{N}_e$ and ACC. Each column shows the results on subsets with different number of subspaces. The FGNSC method achieves higher accuracy with lower $\mathcal{N}_e$ on all subsets.

Metrics	Methods	3	5	8	10	15	20	25	30	35	38
$\mathcal{N}_e \downarrow$	SSC-OMP [38]	20.09	25.73	23.44	31.23	30.46	37.12	38.74	37.32	41.30	39.84
	iPursuit [18]	9.10	10.63	10.70	10.42	11.04	11.76	11.80	12.10	12.52	12.82
	FGNSC	**3.29**	**4.89**	**6.04**	**6.61**	**7.28**	**7.65**	**8.03**	**8.21**	**8.44**	**8.52**
ACC $\uparrow$	SSC-OMP [38]	94.32	85.03	78.41	72.22	71.29	59.13	54.74	61.62	51.84	57.44
	iPursuit [18]	97.19	96.19	95.98	95.03	95.47	93.41	92.77	91.55	91.62	83.68
	FGNSC	**99.34**	**98.61**	**97.86**	**97.44**	**96.60**	**96.09**	**95.72**	**95.45**	**95.41**	**94.24**



(a) (b) (c) (d) (e) (f)

Fig. 5. Affinity $\boldsymbol{W}$ derived by six methods on the first three digits of the MNIST dataset. (a) Affinity matrix by the LSR [49] method with the accuracy of 64.89%. (b) Affinity matrix by the SMR [50] method with the accuracy of 83.11%. (c) Affinity matrix by the SSC-OMP [38] method which is sparse enough but does not have the block-diagonal property. (d) Affinity matrix by the ORGEN [13] method with the accuracy of 53.22% and the NMI of 0.4912. (e) A variant of (b) which simply preserves the top γ entries for each column; it leads to the accuracy of 81.33% and the NMI of 0.4835. (f) Affinity matrix by the FGNSC algorithm with the accuracy of 98.11% and the NMI of 0.9140.

the intersection of the subspaces and wrong combinations for the representations may be generated.

7.5 Comparison to the State-of-the-Art Methods

Table 3 shows the run-time of the FGNSC algorithm and other methods using subsets from the extended Yale B dataset on a machine with a 2.93GHz CPU and 32 GB RAM. Table 4 shows the clustering results in terms of average ACC and NMI. Overall, the proposed FGNSC algorithm performs favorably against the state-of-the-art consistently on five benchmarks.

On the extended Yale B dataset, the FGNSC algorithm performs well in terms of ACC and NMI. The ACC by the FGNSC algorithm is more than 94% and the NMI of more than 0.9 on all subsets. The SMR method achieves an accuracy of 96% with $N_c = 8$ but the performance declines when the dataset becomes larger. The main reason is that the coefficient matrix $\boldsymbol{Z}$ derived by the SMR method has many redundant connections such that the spectral clustering does not perform well. The iPursuit method also achieves the NMI of more than 0.9 on all subsets. However it requires more than 400 seconds on the whole dataset. The FGNSC algorithm takes 90 seconds to process the entire extended Yale B dataset with 38 classes, but most of the run-time is on computing the initial $\boldsymbol{Z}$ by the SMR method which uses 75 seconds.

The FGNSC method performs well on the COIL-20 dataset especially for $N_c = 5$ with an average accuracy of more than 99% and an NMI of more than 0.98. The subspace structure of the COIL-20 dataset is amicable to the FGNSC algorithm since samples in the same cluster are close (as a result of dense sampling with clean backgrounds) and samples in different clusters are distinct (as a result of diverse object classes).

The FGNSC algorithm performs well on the MINST dataset especially on the subset with three digits. However, the average accuracy decreases to 75% for the whole dataset when $N_c = 10$. This can be attributed to the fact that some handwritten digits are similar (*e.g.*, 3 and 8; 1 and 7). We note that the SSC-OMP scheme performs worse, with accuracy dropping from 97.64% when $N_c = 3$ to 53.30% when $N_c = 10$.

On the USPS dataset, the FGNSC algorithm exhibits similar performance trend to that on the MNIST set. The ORGEN scheme performs better than the FGNSC method on the USPS dataset, mainly because the performance of the SMR method is poor in terms of the NMI (around 50% on all scale of subsets).

The accuracy of the ORGEN method is less than 26% on the AR dataset, while the FGNSC algorithm achieves accuracy of more than 84%. It is difficult to cluster the AR dataset as it contains real-world images of faces. While the background is simple, the face images are not aligned well. Most clustering algorithms based on sparsity, *e.g.*, SSC-OMP and SSC, do not perform well as the connections between samples are not extracted well. Overall, good neighbors can be generated by the FGNSC algorithm effectively as the subspace structure can be well reconstructed by them. This is also the main reason why the FGNSC method performs well on all the evaluated datasets.

7.6 Combination with Other Subspace Models

In this section, we combine the proposed FGN method with other subspace clustering methods, *i.e.*, LSR [49], OSC [23], spatSC [41] and LRR [44], rather than SMR [50]. For each experiment, we first compute the initialized coefficient matrix $\boldsymbol{Z}$ by each subspace scheme. The FGN method is then used to generate $\boldsymbol{Z}^*$. Finally, the spectral clustering is carried out on both $\boldsymbol{Z}$ and $\boldsymbol{Z}^*$. Table 5 shows the clustering results by all the evaluated methods.

The proposed post-processing module significantly enhances the performance of the evaluated subspace clustering algorithms with different representation terms. For instance, the low rank

TABLE 3
Run-time in seconds (s) on the subsets of extended Yale B database. On the entire extended Yale B dataset with 38 subjects, the FGNSC algorithm takes 90 seconds but most of which is on computing the initial coefficient (75 seconds).

# of subjects	SSC [1]	spatSC [41]	L2-Graph [19]	LSR [49]	SMR [50]	SSC-OMP [38]	ORGEN [13]	RSSC [36]	NSN [33]	iPursuit [18]	FGNSC
8	38.30	0.20	0.08	0.98	0.64	0.36	0.99	6.92	0.45	12.87	0.70
15	118.50	0.84	0.28	2.02	3.28	1.06	1.92	17.44	1.29	26.51	3.71
30	658.52	4.90	2.24	9.08	30.61	3.17	4.69	83.29	7.93	256.59	37.45
38	1239.90	8.66	5.76	17.93	75.24	4.88	7.17	147.39	13.78	422.35	91.73

TABLE 4
Clustering results of evaluated algorithms on five datasets: extended Yale B [63], COIL-20 [64], MNIST [65], USPS [66] and AR [67]. Each experiment is carried out fifty times and the average results are reported. The best results are shown in boldface.

Datasets	Scale	Metrics	SSC [1]	spatSC [41]	L2-Graph [19]	LSR [49]	SMR [50]	SSC-OMP [38]	ORGEN [13]	RSSC [36]	NSN [33]	iPursuit [18]	FGNSC
EYale B	8	ACC	59.61	20.84	27.53	73.46	96.43	78.41	66.78	73.42	89.43	95.98	**97.86**
		NMI	0.5518	0.1179	0.1927	0.6332	0.9270	0.5873	0.6333	0.6550	0.8013	0.9254	**0.9511**
	15	ACC	48.46	19.62	20.55	58.88	92.15	71.29	58.61	62.98	84.56	95.47	**96.60**
		NMI	0.5594	0.2590	0.2127	0.5447	0.8885	0.6372	0.6166	0.6137	0.7767	0.9202	**0.9284**
	30	ACC	36.71	18.02	16.73	58.62	89.48	61.62	54.53	58.31	78.66	91.55	**95.45**
		NMI	0.5211	0.3704	0.2603	0.5855	0.8774	0.6243	0.6291	0.6104	0.7705	0.9163	**0.9186**
	38	ACC	32.52	20.23	20.81	58.26	87.91	57.44	53.66	57.81	77.55	83.68	**94.24**
		NMI	0.5466	0.3806	0.3692	0.5917	0.8559	0.6054	0.6388	0.6152	0.7711	0.9072	**0.9084**
COIL-20	5	ACC	66.01	69.97	49.18	72.51	85.43	82.39	84.73	68.11	85.81	85.42	**99.23**
		NMI	0.6688	0.7296	0.4903	0.6639	0.7225	0.4996	0.8854	0.6323	0.8452	0.8378	**0.9814**
	10	ACC	56.46	59.44	31.93	63.70	80.83	77.54	75.07	64.55	85.32	76.21	**96.13**
		NMI	0.6233	0.7293	0.4175	0.6402	0.6993	0.6632	0.8588	0.6555	0.8905	0.7883	**0.9414**
	20	ACC	50.39	51.25	23.75	57.43	75.49	67.36	75.56	59.72	81.94	73.47	**90.07**
		NMI	0.7051	0.7082	0.3511	0.6296	0.5949	0.6552	0.8822	0.6563	0.8734	0.7788	**0.8901**
MNIST	3	ACC	71.39	79.10	73.35	77.44	89.11	97.64	78.76	38.68	38.65	37.82	**97.96**
		NMI	0.5475	0.5260	0.5190	0.5275	0.6671	0.8961	0.6622	0.0097	0.0092	0.0066	**0.9046**
	7	ACC	19.70	57.13	29.65	62.00	78.09	71.13	68.92	19.29	19.54	18.94	**85.88**
		NMI	0.5336	0.5045	0.2755	0.5326	0.5345	0.5647	0.6790	0.0125	0.0145	0.0117	**0.7551**
	10	ACC	40.16	41.83	30.50	57.80	68.87	53.30	61.50	15.30	15.50	14.50	**75.40**
		NMI	0.4872	0.4631	0.2516	0.5197	0.5195	0.4823	0.6463	0.0212	0.0168	0.0138	**0.6629**
USPS	3	ACC	65.81	84.49	48.05	81.05	85.64	92.74	89.86	57.50	58.62	57.95	**94.12**
		NMI	0.4435	0.6129	0.2401	0.6377	0.5164	0.7676	**0.8242**	0.2594	0.3253	0.2659	0.7953
	7	ACC	18.60	50.57	47.14	62.58	76.90	59.01	65.06	31.09	31.17	30.23	**80.13**
		NMI	0.5295	0.4486	0.4785	0.5219	0.5160	0.3857	0.6625	0.3067	0.3002	0.2609	**0.6927**
	10	ACC	32.34	44.73	37.37	60.40	68.83	47.33	63.57	23.15	24.40	23.30	**75.37**
		NMI	0.4814	0.4693	0.4233	0.5096	0.5136	0.3272	**0.6782**	0.2721	0.2628	0.2540	0.6390
AR	30	ACC	32.91	32.92	47.14	76.21	81.26	30.37	25.25	68.09	23.12	66.88	**87.24**
		NMI	0.5241	0.5504	0.6029	0.7913	0.6021	0.3759	0.4313	0.7598	0.3696	0.7336	**0.8566**
	60	ACC	25.58	32.70	27.35	75.19	80.63	25.21	23.09	71.63	21.78	67.44	**86.04**
		NMI	0.5162	0.5976	0.4373	0.8068	0.5602	0.4161	0.4825	0.8128	0.4321	0.7769	**0.8601**
	90	ACC	23.87	30.41	23.43	72.53	80.39	23.52	21.81	71.56	20.29	67.46	**84.22**
		NMI	0.5367	0.6173	0.4240	0.7983	0.5090	0.4404	0.5036	0.8278	0.4668	0.7891	**0.8522**
	100	ACC	25.54	29.19	25.58	74.15	79.85	23.12	20.88	76.31	20.08	65.88	**83.31**
		NMI	0.5393	0.6394	0.4920	0.8145	0.4755	0.4434	0.5054	0.8554	0.4756	0.7880	**0.8493**

representation (LRR) [44]) minimizes the nuclear norm $\|\boldsymbol{Z}\|_*$ to generate a dense coefficient matrix $\boldsymbol{Z}$, which leads to the accuracy of less than 60% and the NMI of 0.6331 on the extended Yale B dataset with 38 subjects. When combined with the FGN module, this algorithm achieves the ACC of 82.85% and the NMI of 0.8891. As the proposed FGN method preserves only the key connections and eliminates noisy ones, it performs well with different representation schemes.

8 Conclusions

In this work, we propose the FGNSC algorithm to exploit both sparsity and connectivity properties within each subspace for effective subspace clustering. We find good neighbors for each sample by utilizing the correlation information contained in the initial affinity matrix rather than the input data space. The relationship of good neighbors requires not only the direct connections derived by pairwise correlations but also the latent connections induced by other samples on the connected path. Sequentially, the good neighbors introduce high probability that the samples are within same subspace and are strongly connected. We then reassign the coefficients of the selected good neighbors and eliminate other values such that good neighbors have greater opportunity to be segmented into the same cluster. In addition, we present the theoretical guarantee of the subspace-preserving

TABLE 5
More results on combining FGN with other linear representation schemes on the extended Yale B dataset (*e.g.*, FGN-LSR is the combination of FGN in Algorithm 1 and the linear representation module in LSR [49]). The best results are shown in boldface.

Methods	Metrics	8	15	30	38
LSR [49]	ACC	73.46	58.88	58.62	58.26
	NMI	0.6332	0.5447	0.5855	0.5917
FGN-LSR	ACC	**96.11**	**90.80**	**89.85**	**90.30**
	NMI	**0.9195**	**0.8778**	**0.8779**	**0.8824**
OSC [23]	ACC	24.53	20.16	16.60	15.34
	NMI	0.1315	0.1760	0.2211	0.2424
FGN-OSC	ACC	**91.19**	**80.00**	**75.66**	**75.12**
	NMI	**0.8549**	**0.7603**	**0.7399**	**0.7412**
spatSC [41]	ACC	20.84	19.62	18.02	20.23
	NMI	0.1179	0.2590	0.3704	0.3806
FGN-spatSC	ACC	**76.02**	**73.40**	**69.80**	**70.72**
	NMI	**0.7265**	**0.7544**	**0.7411**	**0.7512**
LRR [44]	ACC	68.10	61.70	61.69	59.54
	NMI	0.6078	0.6161	0.6410	0.6331
FGN-LRR	ACC	**93.77**	**85.80**	**82.97**	**82.85**
	NMI	**0.9152**	**0.8794**	**0.8839**	**0.8891**

property and connectivity of the affinity graph. Extensive experimental results demonstrate the effectiveness and efficiency of the proposed algorithm against the state-of-the-art methods.

REFERENCES

[1] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797. 1, 2, 4, 8, 9, 11

[2] T. Beier, F.-A. Hamprecht, and J.-H. Kappes, "Fusion moves for correlation clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3507–3516. 1, 8

[3] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 586–594. 1

[4] M. Yin, Y. Guo, J. Gao, Z. He, and S. Xie, "Kernel sparse subspace clustering on symmetric positive definite manifolds," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5157–5164. 1

[5] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 430–437. 1

[6] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011. 1, 2, 4, 8

[7] Y. Yang, J. Feng, N. Jojic, J. Yang, and T.-S. Huang, "ℓ^0-sparse subspace clustering," in *European Conference on Computer Vision*, 2016, pp. 731–747. 1, 3

[8] M. Tan, S. Xiao, J. Gao, D. Xu, A. van den Hengel, and Q. Shi, "Proximal riemannian pursuit for large-scale trace-norm minimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5877–5886. 1

[9] C. Lu, J. Tang, S. Yan, and Z. Lin, "Nonconvex nonsmooth low-rank minimization via iteratively reweighted nuclear norm," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 829–839, 2016. 1

[10] M. Yin, J. Gao, and Z. Lin, "Laplacian regularized low-rank representation and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 504–517, 2016. 1, 2, 4

[11] M. Lee, J. Lee, H. Lee, and N. Kwak, "Membership representation for detecting block-diagonal structure in low-rank or sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1648–1656. 1, 3, 7

[12] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000. 1, 2, 5, 6, 7, 8

[13] C. You, C.-G. Li, D.-P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable Elastic Net subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3928–3937. 1, 2, 3, 4, 5, 7, 9, 10, 11

[14] C. Lu, J. Feng, Z. Lin, and S. Yan, "Correlation adaptive subspace segmentation by trace Lasso," in *IEEE International Conference on Computer Vision*, 2013, pp. 1345–1352. 1, 2, 3

[15] B. Nasihatkon and R. Hartley, "Graph connectivity in sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2137–2144. 1, 2, 3, 4

[16] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007. 1, 4, 6, 8

[17] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2014, pp. 977–986. 2

[18] M. Rahmani and G. Atia, "Innovation pursuit: A new approach to the subspace clustering problem," in *International Conference on Machine Learning*, 2017, pp. 2874–2882. 2, 3, 5, 9, 10, 11

[19] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-Graph for robust subspace learning and subspace clustering," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 1053–1066, 2016. 2, 3, 4, 5, 7, 8, 9, 11

[20] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When LRR meets SSC," in *Advances in Neural Information Processing Systems*, 2013, pp. 64–72. 2, 3, 7

[21] S. Wang, X. Yuan, J. Shen, T. Yao, and S. Yan, "Efficient subspace segmentation via quadratic programming," in *The Association for the Advancement of Artificial Intelligence*, 2011, pp. 519–524. 2

[22] B. Li, Y. Zhang, Z. Lin, and H. Lu, "Subspace clustering by mixture of gaussian regression," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2094–2102. 2

[23] S. Tierney, J. Gao, and Y. Guo, "Subspace clustering for sequential data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1019–1026. 2, 9, 10, 12

[24] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005. 2, 8

[25] M.-T. Law, Y. Yu, M. Cord, and E.-P. Xing, "Closed-form training of mahalanobis distance for supervised clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3909–3917. 2

[26] J. Xu, J. Han, and F. Nie, "Discriminatively embedded k-means for multi-view clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5356–5364. 2

[27] Z. Li, S. Yang, L.-F. Cheong, and K.-C. Toh, "Simultaneous clustering and model selection for tensor affinities," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5347–5355. 2, 8

[28] D. P. Hofmeyr, "Clustering by minimum cut hyperplanes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1547–1560, 2016. 2

[29] Y. Wang, Y. Wang, and A. Singh, "Differentially private subspace clustering," in *Advances in Neural Information Processing Systems*, 2015, pp. 1000–1008. 2

[30] C. You, D. P. Robinson, and R. Vidal, "Provable self-representation based outlier detection in a union of subspaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 8

[31] D. Pimentel-Alarcon and R. Nowak, "The information-theoretic requirements of subspace clustering with missing data," in *International Conference on Machine Learning*, 2016, pp. 802–810. 2, 8

[32] C.-G. Li and R. Vidal, "Structured sparse subspace clustering: A unified optimization framework," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 277–286. 2

[33] D. Park, C. Caramanis, and S. Sanghavi, "Greedy subspace clustering," in *Advances in Neural Information Processing Systems*, 2014, pp. 2753–2761. 2, 3, 9, 11

[34] M. Yin, J. Gao, Z. Lin, Q. Shi, and Y. Guo, "Dual graph regularized latent low-rank representation for subspace clustering," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4918–4933, 2015. 2

[35] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5249–5257. 2

[36] J. Xu, K. Xu, K. Chen, and J. Ruan, "Reweighted sparse subspace clustering," *Computer Vision and Image Understanding*, vol. 138, pp. 25–37, 2015. 2, 3, 9, 11

[37] E.-L. Dyer, A.-C. Sankaranarayanan, and R.-G. Baraniuk, "Greedy feature selection for subspace clustering," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2487–2517, 2013. 3

[38] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3918–3927. 3, 4, 9, 10, 11

[39] P. Purkait, T.-J. Chin, A. Sadri, and D. Suter, "Clustering with hypergraphs: the case for large hyperedges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1697–1711, 2017. 3

[40] Y. Cheng, Y. Wang, M. Sznaier, and O. Camps, "Subspace clustering with priors via sparse quadratically constrained quadratic programming," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5204–5212. 3

[41] Y. Guo, J. Gao, and F. Li, "Spatial subspace clustering for drill hole spectral data," *Journal of Applied Remote Sensing*, vol. 8, no. 1, p. 083644, 2014. 3, 9, 10, 11, 12

[42] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block-diagonal prior," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3818–3825. 3, 7

[43] B. Liu, X.-T. Yuan, Y. Yu, Q. Liu, and D.-N. Metaxas, "Decentralized robust subspace clustering," in *The Association for the Advancement of Artificial Intelligence*, 2016, pp. 3539–3545. 3

[44] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013. 3, 4, 8, 9, 10, 11, 12

[45] S. Xiao, W. Li, D. Xu, and D. Tao, "FaLRR: A fast low rank representation solver," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4612–4620. 3

[46] B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Product grassmann manifold representation and its LRR models," in *The Association for the Advancement of Artificial Intelligence*, 2016, pp. 2122–2129. 3

[47] G. Liu, H. Xu, J. Tang, Q. Liu, and S. Yan, "A deterministic analysis for LRR," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 417–430, 2016. 3

[48] C. Xu, Z. Lin, and H. Zha, "A unified convex surrogate for the Schatten-p norm." in *The Association for the Advancement of Artificial Intelligence*, 2017, pp. 926–932. 3

[49] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *European Conference on Computer Vision*, 2012, pp. 347–360. 3, 9, 10, 11, 12

[50] H. Hu, Z. Lin, J. Feng, and J. Zhou, "Smooth representation clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3834–3841. 3, 8, 9, 10, 11

[51] H. Lai, Y. Pan, C. Lu, Y. Tang, and S. Yan, "Efficient k-Support matrix pursuit," in *European Conference on Computer Vision*, 2014, pp. 617–631. 3

[52] C.-G. Li and R. Vidal, "A structured sparse plus structured low-rank framework for subspace clustering and completion," *IEEE Transactions on Signal Processing*, vol. 64, no. 24, pp. 6557–6570, 2016. 3

[53] E. Kim, M. Lee, and S. Oh, "Robust Elastic-Net subspace representation," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4245–4259, 2016. 3

[54] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013. 3, 4, 5, 6, 7, 8, 9

[55] M. Rahmani and G. Atia, "A direction search and spectral clustering based approach to subspace clustering," *arXiv preprint arXiv:1706.03860*, 2017. 3, 9

[56] Y. Wang, Y.-X. Wang, and A. Singh, "Graph connectivity in noisy sparse subspace clustering," in *Artificial Intelligence and Statistics*, 2016, pp. 538–546. 4

[57] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *The Association for the Advancement of Artificial Intelligence*, 2015, pp. 3827–3833. 5

[58] C. Peng, Z. Kang, and Q. Cheng, "Subspace clustering via variance regularized ridge regression," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 7

[59] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014. 7

[60] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM journal on computing*, vol. 1, no. 2, pp. 146–160, 1972. 7

[61] M. Soltanolkotabi and E.-J. Cands, "A geometric analysis of subspace clustering with outliers," *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2011. 8

[62] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014. 8

[63] A.-S. Georghiades, P.-N. Belhumeur, and D.-J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001. 9, 11

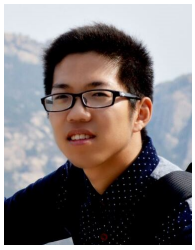
[64] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (COIL-20)," *Technical Report CUCS-005-96*, 1996. 9, 11

[65] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 9, 11

[66] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer Series in Statistics New York, 2001, vol. 1. 9, 11

[67] A.-M. Martinez, "The AR face database," *CVC Technical Report*, 1998. 9, 11

Jufeng Yang is an associate professor in the College of Computer and Control Engineering, Nankai University. He received the PhD degree from Nankai University in 2009. From 2015 to 2016, he was working at the Vision and Learning Lab, University of California, Merced. His research falls in the field of computer vision, machine learning and multimedia.



Jie Liang is currently a Master student with the College of Computer and Control Engineering, Nankai University. His current research interests include computer vision, machine learning, pattern recognition and optimization.

Kai Wang is an associate professor in the College of Computer and Control Engineering, Nankai University. He received the PhD degree in control theory and control engineering from Nankai University in 2006. His research interests include computer vision and pattern recognition. He is a member of the ACM, the IEEE Computer Society and the CCF.

Ming-Hsuan Yang is a professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in Computer Science from the University of Illinois at Urbana-Champaign in 2000. Yang has served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal of Computer Vision, Image and Vision Computing, Computer Vision and Image Understanding, and Journal of Artificial Intelligence Research. He received the NSF CAREER award in 2012 and the Google Faculty Award in 2009.