Multi-View Spectral Clustering with High-Order Optimal Neighborhood Laplacian Matrix

Weixuan Liang*, Sihang Zhou*, Jian Xiong, Xinwang Liu†, *Senior Member, IEEE,* Siwei Wang, En Zhu, Zhiping Cai, and Xin Xu

Abstract—Multi-view spectral clustering can effectively reveal the intrinsic cluster structure among data by performing clustering on the learned optimal embedding across views. Though demonstrating promising performance in various applications, most of existing methods usually linearly combine a group of pre-specified first-order Laplacian matrices to construct the optimal Laplacian matrix, which may result in limited representation capability and insufficient information exploitation. Also, storing and implementing complex operations on the $n \times n$ Laplacian matrices incurs intensive storage and computation complexity. To address these issues, this paper first proposes a multi-view spectral clustering algorithm that learns a high-order optimal neighborhood Laplacian matrix, and then extends it to the late fusion version for accurate and efficient multi-view clustering. Specifically, our proposed algorithm generates the optimal Laplacian matrix by searching the neighborhood of the linear combination of both the first-order and high-order base Laplacian matrices simultaneously. By this way, the representative capacity of the learned optimal Laplacian matrix is enhanced, which is helpful to better utilize the hidden high-order connection information among data, leading to improved clustering performance. We design an efficient algorithm with proved convergence to solve the resultant optimization problem. Extensive experimental results on nine datasets demonstrate the superiority of our algorithm against state-of-the-art methods, which verifies the effectiveness and advantages of the proposed algorithm.

Index Terms—Neighborhood kernel, High-order Laplacian matrix, Spectral clustering, Late fusion.

1 Introduction

SPECTRAL clustering is an important technique that optimally learns the low-dimensional intrinsic embedding from the noisy high-dimensional data for clustering. In recent years, in the era of big data, integrate the diverse and complementary information from multiple views to further improve the effectiveness of the algorithm is becoming an increasingly attractive hotspot in this field [1], [2], [3], [4]. As information represented by different views can be heterogeneous and biased, how to fully exploit the multi-view information and subtly fuse them to acquire a better overall vision of the whole sample set is one of the vital topics in the field of multi-view spectral clustering (MSC). According to the information fusion mechanism, the existing literature of MSC can be roughly dividing into three categories. The first category of method adopts a co-training mechanism to force the clustering results of different views to be consistent with each other [5], [6], [7]. The second category of method holds that the affinity matrix of each view is a perturbation of the optimal affinity matrix. Then, by conducting low-rank or sparse optimization, these algorithms extract an optimal consensus affinity matrix from all views [8], [9], [10], [11]. By assuming that the optimal Laplacian matrix is a linear aggregation of the base Laplacian matrices, the third category of method optimizes the combination coefficients of the base Laplacian matrices by minimizing the normalized cut of the combined matrix [12].

Although the mentioned methods have largely improved the clustering accuracy in multi-view circumstances, the storage and computational cost on the $n \times n$ Laplacian matrices limits the efficiency of these methods. To further improve the efficiency of the existing literature, a large number of methods are proposed. Zhou et al. reduce the complexity of spectral clustering by employing random Fourier features to construct the base kernels and do low-rank SVD decompositions accordingly [11]. Semertzidis et al. propose an efficient spectral clustering method for large-scale data sets in which a set of pairwise constraints are given to increase clustering accuracy and reduce clustering complexity [13]. The work in [14] adopts a novel bipartite graph, which records only the similarity between the data (n samples) and the salient point set (p samples) for spectral clustering, thus largely reduces the memory and computational complexity. Chen et al. [15] and Cai et al. [16] propose landmark-based spectral clustering and spectral dimensionality reduction, in which they adopt a representative point-based strategy to construct the similarity graph to accelerate the procedure of spectral clustering. The Nyström approach [17] samples $m(\ll n)$ columns from the affinity matrix, and then forms a low-rank approximation of the full matrix by using the correlations between the sampled columns and the remaining $n - m$ columns. As only a portion of the full matrix is computed and stored, the Nyström approach can

- $*$ *equal contribution*
- † *corresponding author*
- *Weixuan Liang, Xinwang Liu, Siwei Wang, En Zhu, and Zhiping Cai are with School of Computer, National University of Defense Technology, Changsha, Hunan, 410073, China. E-mail: {xinwangliu@nudt.edu.cn, enzhu@nudt.edu.cn}.*
- *Sihang Zhou and Xin Xu are with College of Intelligence Science and Technology, National University of Defense Technology, Changsha, Hunan, 410073, China.*
- *Jian Xiong is with School of Business Administration, Southwestern University of Finance and Economics, Chengdu, Sichuan, 611130, China.*

reduce the time and space complexity significantly. Fowlkes et al. [18] and Li et al. [19] successfully applied this to spectral clustering and propose the algorithms which can scale to very large data sets.

Existing algorithms have achieved various improvements in multi-view spectral clustering. But we observe that these algorithms bear the following drawbacks. First, the algorithms in the third category share a common assumption that the optimal Laplacian matrix lies in the linear space spanned by the base Laplacian matrices. This assumption, on the one hand, simplifies the optimization procedure. But on the other hand, as it is uncovered in recent work that it might over-reduce the feasible set of the optimal Laplacian matrix and could result in limited representation capability of the learned matrix [20], [21], [22], [23]. Second, existing algorithms do not sufficiently consider the high-order affinity information, which is important to reveal hidden neighborhood relations among samples. Both factors could adversely affect the learned Laplacian matrix, leading to unsatisfying clustering performance.

In this paper, we propose an optimal neighborhood multi-view spectral clustering algorithm (termed optimal neighborhood multi-view spectral clustering, ONMSC) to address both issues. Specifically, in our proposed algorithm, instead of restricting the optimal Laplacian matrix being a linear combination of base Laplacian matrices, we allow the optimal matrix to lie in the neighborhood of the latter. In this way, our algorithm effectively enlarges the region from which an optimal Laplacian matrix can be chosen and consequently improves its representative capacity. Moreover, we further enforce the learned optimal Laplacian matrix to be in the neighborhood of the linear combination of both the first-order and high-order base Laplacian matrices. As a consequence, the constructed optimal Laplacian matrix will be able to exploit both the first-order and high-order connection information. After that, we carefully instantiate an optimization objective function and develop an efficient algorithm with proved convergence to solve the resulting optimization problem.

Unlike the early fusion methods that merge the base affinity matrices or Laplacian matrices, late fusion multi-view clustering [24], [25], [26], [27] generates base partitions from each view independently and integrates them into a consensus one. [24] adopts the low-rank and sparse decomposition to maintain consistency and get rid of the adverse effects of noises across views for better clustering performance. [25] clusters instances from easy to difficult by a self-paced clustering ensemble method to enhance the stability of the corresponding algorithm. [26], [27] learn an optimal consensus partition by maximally aligning the consensus partition with the weighted base partitions. The mentioned multi-view algorithms reduce the computational complexity of each iteration from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ [24], [25] or $\mathcal{O}(n)$ [26], [27] while keeping comparable clustering accuracy. Inspired by the recent development of late fusion multi-view clustering [26], [27], we further extend the proposed algorithm into the late fusion fashion (denoted as ONMSC-LF) for efficient computation.

The contributions of this paper are summarized as follows:

- We discover that the current linear combination-based multi-view spectral clustering framework could: 1) limit the representation capacity of the learned Laplacian matrix; and 2) insufficiently explore the high-order neighborhood information among data.
- We propose a high-order optimal Laplacian matrix construction method to solve the above problems. In our proposed method, both first-order and high-order affinity information is fully explored and the searching space of the optimal Laplacian matrix is considerably enlarged.
- We also extend the proposed algorithm with a late fusion fashion and a Nyström sample technique to further improve the efficiency of the proposed algorithm. As a consequence, the computational complexity and the storage complexity has drastically reduced from $\mathcal{O}\left(n^3\right)$ to $\mathcal{O}\left(n\right)$ per iteration.

The notations that are used throughout the paper are summarized in Table 1. The rest of the paper is organized as follows. The related work of spectral clustering, high-order Laplacian matrix and late fusion multi-view clustering is reviewed in Section 2. The proposed optimal neighborhood multi-view spectral clustering algorithm and its late fusion version are described in Section 3. The experimental results are reported in Section 4. Finally, the paper is concluded in Section 5.

TABLE 1
Summary of notations

$\mathbf{X}$	A dataset of n samples
$\mathbf{x}_i$	The i-th sample of $\mathbf{X}$
n	The number of samples in $\mathbf{X}$
k	The number of clusters
O	The largest order number
v	The number of views
N	The neighbor number of affinity matrix
$\mathbf{A}^{(o)}$	o-th order affinity matrix
$\mathbf{D}^{(o)}$	The degree matrix of $\mathbf{A}^{(o)}$
$\mathbf{L}^{(o)}$	The Laplacian matrix of $\mathbf{A}^{(o)}$
$\mathbf{H}_p^{(o)}$	The o-th order cluster indicating matrix of p-th view
$\mathbf{W}_p^{(o)}$	The rotation matrix of $\mathbf{H}_p^{(o)}$
$\boldsymbol{\mu}$	Combination coefficients of multi-view
$\mathbf{M}$	Correlation measuring matrix of multi-view
$\mathbf{F}$	The average cluster indicating matrix
$\mathbf{L}^*$	The optimal Laplacian matrix
$\mathbf{H}^*$	The optimal cluster indicating matrix
$\mathbf{G}$	The normalized affinity matrix
λ_1	The average view balancing parameters
λ_2	The diversity balancing parameters

2 PRELIMINARIES

In this section, we first briefly introduce some important notations about spectral clustering and then revisit the 1) *linear Laplacian matrix combination-based multi-view spectral clustering* and 2) *late fusion alignment maximization based multi-view clustering*. Finally, we introduce the definition of the high-order Laplacian matrix in our paper.

2.1 Spectral Clustering

Spectral clustering is a powerful unsupervised machine learning algorithm, especially for linear inseparable data.

Denote the given data matrix as $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]^\top \in \mathbb{R}^{n\times d}$, where n is the sample number and d is the feature dimension. Given a kernel function $\kappa(\cdot,\cdot)$, the affinity matrix $\mathbf{A}$ can be constructed in a K-NN fasion. In particular, in the affinity matrix, x_i and x_j are linked if at least one of them is among the k nearest neighbors of the other in the measurement of $\kappa(\cdot,\cdot)$. The j-th element of the i-th row of $\mathbf{A}$ is:

$$\mathbf{A}_{ij} = \begin{cases} \kappa(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are linked;} \\ 0, & \text{otherwies.} \end{cases} \tag{1}$$

Denoting the i-th diagonal element in the degree matrix $\mathbf{D} \in \mathbb{R}^{n\times n}$ as

$$\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{A}_{ij}, \tag{2}$$

and the definition of the corresponding first-order normalized graph Laplacian matrix is:

$$\mathbf{L}^{(1)} \triangleq \mathbf{I}_n - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}, \tag{3}$$

where $\mathbf{I}_n$ is a $n \times n$ identity matrix. Let $\mathbf{H} \in \mathbb{R}^{n\times k}$ denotes the cluster indicating matrix, where k is the number of classes. The object function of the normalized spectral clustering [28] is:

$$\min_{\mathbf{H}^\top\mathbf{H}=\mathbf{I}} \mathrm{Tr}\left(\mathbf{H}^\top\mathbf{L}^{(1)}\mathbf{H}\right). \tag{4}$$

The optimal $\mathbf{H}$ can be easily acquired by conducting singular value decomposition (SVD) w.r.t. $\mathbf{L}^{(1)}$ and take the eigenvectors corresponding to the smallest k eigenvalues. Finally, the categorical assignment can be acquired by doing k-means on the learned optimal cluster indicating matrix $\mathbf{H}$.

2.2 Multi-view Spectral Clustering with Linear Laplacian Matrix Combination

For multi-view data, let v be be the number of views, $\mathbf{A}_1, ..., \mathbf{A}_v \in \mathbb{R}^{n\times n}$ be the affinity matrix of each view and $\mathbf{L}_1^{(1)}, ..., \mathbf{L}_v^{(1)} \in \mathbb{R}^{n\times n}$ be the corresponding first-order normalized Laplacian matrices. To exploit the complementary information from different views, [12] linearly aggregates the base Laplacian matrices from each view and learns an optimal matrix which is the most suitable for clustering. The formulation of the algorithm is:

$$\begin{aligned} &\min_{\mathbf{H}^\top\mathbf{H}=\mathbf{I}_c,\boldsymbol{\mu}} \mathrm{Tr}\left(\mathbf{H}^\top\mathbf{L}_{\boldsymbol{\mu}}^{(1)}\mathbf{H}\right), \\ &\text{s.t. } \mathbf{L}_{\boldsymbol{\mu}}^{(1)} = \sum_{p=1}^{v} \mu_p^r \mathbf{L}_p^{(1)}, \|\boldsymbol{\mu}\|_1 = 1, \boldsymbol{\mu} \geq 0, \end{aligned} \tag{5}$$

where μ_p is the combination weight of the p-th view, $\mathbf{L}_{\boldsymbol{\mu}}^{(1)}$ is the optimal Laplacian matrix for learning, and $r(\in \mathbb{N}^+)$ is a hyper-parameter to balance the contribution of each view. Although good performance has been achieved by the above method, recent literatures show that this method over reduces the feasible set of the optimal Laplacian matrix, which may lead to a less representative solution and yield even worse performance than using a single view [22].

2.3 Multi-view Clustering via Late Fusion

In multiple kernel clustering and multi-view spectral clustering, recording and doing complex operations on the $n \times n$ kernel or Laplacian matrices are storage and computational expensive. To solve the problem, Wang et al. [27] adopt a late fusion fashion and propose an efficient multi-view clustering algorithm. In their method, to reduce storage and computational cost, the authors use the light-weighted cluster indicating matrix obtained by the kernel k-means algorithm to represent the categorical information from each view. Then, by maximally aligning the linear combination of the rotated cluster indicating matrix with the optimal data partition matrix, the information of each view is efficiently and effectively fused. Let $\mathbf{H}_p (p \in [v])$ be the cluster indicating matrix of the p-th view, the formulation of the late-fusion based multi-view clustering is as follow:

$$\begin{aligned} &\max_{\mathbf{H}^*, \{\mathbf{W}_p\}_{p=1}^v, \boldsymbol{\mu}} \mathrm{Tr}(\mathbf{H}^{*\top}\mathbf{S}) + \lambda \mathrm{Tr}(\mathbf{H}^{*\top}\mathbf{F}), \\ &\text{s.t. } \mathbf{H}^{*\top}\mathbf{H}^* = \mathbf{I}_k, \mathbf{W}^\top\mathbf{W} = \mathbf{I}_k, \\ &\sum_{p=1}^{v} \mu_p^2 = 1, \mu_p \geq 0, \mathbf{S} = \sum_{p=1}^{v} \mu_p \mathbf{H}_p \mathbf{W}_p, \end{aligned} \tag{6}$$

where $\{\mathbf{W}_p\}_{p=1}^m \in \mathbb{R}^{k\times k}$ is a set of rotation matrices, and $\mathbf{S} = \sum_{p=1}^v \mu_p \mathbf{H}_p \mathbf{W}_p$ is the linear combination of the rotated cluster indicating matrices; $\mathbf{F}$ denotes the average partition matrix, which we can obtain by performing spectral clustering on the average affinity matrix $\frac{1}{v}\sum_{p=1}^v \mathbf{A}_p$; λ is a trade-off parameter to prevent $\mathbf{H}^*$ from being too far way from prior average partition.

2.4 High-Order Laplacian Matrix

First-order and second-order connections are essential concepts in graph analyzing [29]. Specifically, in graph embedding, the first-order connection refers to the local pairwise proximity between vertices in a graph. Comparatively, the second-order connection holds that vertices which have similar affinity network structure are also similar to each other. An example can be found in Fig. 1. In this figure, the circles indicate samples in the dataset and the edges indicate the first-order connection between the corresponding samples. As we can see, sample 5 and 6 are not connected, they are with a low similarity w.r.t. the definition of first-order connection. However, since both sample 5 and sample 6 are connected with sample 7, 8, 9 and 10, they share an identical neighborhood network structure. As a consequence, w.r.t. the definition of second-order connection, sample 5 and sample 6 are similar with each other.

Moreover, in recent literatures, because of the popularity of graph convolutional neural networks [30], higher-order connection information has attracted the attention of researches. In these papers, the order of connections has been explained as the receptive field of different convolutional filters. Specifically, the definition of second-order proximity in [29] is as follows:

Definition 1 (Second-order Proximity)**.** *The second-order proximity between a pair of vertices (u, v) in a network is the similarity between their neighborhood network structures.*

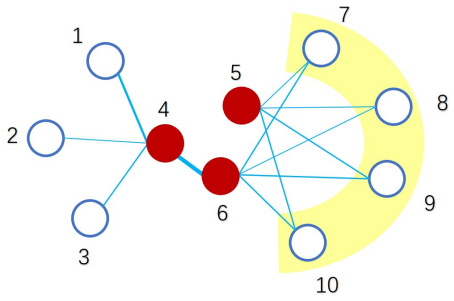


Fig. 1. A toy example of the graph of an affinity matrix. Sample 4 and 6 should be placed closely in the low-dimensional space as they are connected through a strong tie. Sample 5 and 6 should also be placed closely as they share similar neighbors.

According to the above definition, denote $\mathbf{a}_j$ as the j-th column of first-order affinity matrix $\mathbf{A}$, the mathematical definition of the second-order affinity matrix $\mathbf{A}^{(2)}$ is:

$$\mathbf{A}_{ij}^{(2)} \triangleq \mathbf{a}_i^\top \mathbf{a}_j, \forall i, j \in [n]. \tag{7}$$

Consequently, the corresponding second-order normalized Laplacian matrix can be written as:

$$\mathbf{L}^{(2)} \triangleq \mathbf{I}_n - \left(\mathbf{D}^{(2)}\right)^{-\frac{1}{2}} \mathbf{A}^{(2)} \left(\mathbf{D}^{(2)}\right)^{-\frac{1}{2}}, \tag{8}$$

where $\mathbf{D}_{ii}^{(2)} = \sum_{j=1}^{n} \mathbf{A}_{ij}^{(2)}$. According to this definition, we can readily calculate a o-order proximity via $\mathbf{A}^{(o)} = \mathbf{A}^{(o-1)}\mathbf{A}$. As shown by existing literature [29], first-order connection in the real world data is usually not sufficient to preserve the global data structure. However, existing methods in this regard do not sufficiently consider the high-order information, which is crucial to improve the learning performance, especially in unsupervised scenario.

3 The Proposed Algorithm

In this section, to explore better representation capacity and more comprehensively exploit both the first-order and high-order affinity information in data, we first propose a novel multi-view spectral clustering algorithm with optimal neighborhood Laplacian matrix. Then, to improve both the storage and computational efficiency, we extend the proposed algorithm to a late fusion version.

3.1 Multi-View Spectral Clustering with High-Order Optimal Neighborhood Laplacian Matrix

To better capture the high-order affinity information and search the optimal Laplacian matrix in a larger space, we propose the following formulation:

$$\begin{aligned} &\min_{\mathbf{H},\boldsymbol{\mu},\mathbf{L}^*} \mathrm{Tr}\left(\mathbf{H}^\top \mathbf{L}^* \mathbf{H}\right) + \sum_{o=1}^{O} \|\mathbf{L}^* - \mathbf{L}_{\boldsymbol{\mu}}^{(o)}\|_{\mathrm{F}}^2 + \alpha \boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu}, \\ &\text{s.t. } \mathbf{L}_{\boldsymbol{\mu}}^{(o)} = \sum_{p=1}^{v} \mu_p \mathbf{L}_p^{(o)} (o \in [O]), \|\boldsymbol{\mu}\|_1 = 1, \boldsymbol{\mu} \geq 0, \\ &\mathbf{L}^* \succeq 0, \mathbf{L}_{mn}^* \leq 0 (m \neq n), \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k, \end{aligned} \tag{9}$$

where $\mathbf{L}^*$ is the optimal Laplacian matrix for learning, $\mathbf{L}_{\boldsymbol{\mu}}^{(o)}$ is the linear combination of the o-order base Laplacian matrices, O is the largest order number, α is an importance balancing coefficient, and $\mathbf{M}$ is the correlation measuring matrix which records the centered kernel alignment value [31] between affinity matrices. Specifically, denote the o-order affinity matrix of the p-th view as $\mathbf{A}_p^{(o)}$, the definition of $\mathbf{M}$ is:

$$\mathbf{M}_{pq} = \sum_{o=1}^{O} \frac{\mathrm{Tr}\left(\mathbf{A}_p^{(o)} \mathbf{A}_q^{(o)}\right)}{\|\mathbf{A}_p^{(o)}\|_{\mathrm{F}} \|\mathbf{A}_q^{(o)}\|_{\mathrm{F}}}.$$

In the objective function of Eq. (9), the first term is the spectral clustering term which encourages the learned optimal Laplacian matrix to perform well in clustering. In the second term, we restrict $\mathbf{L}^*$ to be in the neighborhood of the linearly combined multi-order based Laplacian matrices by minimizing the difference between $\mathbf{L}^*$ and $\mathbf{L}_{\boldsymbol{\mu}}^{(o)}$'s at the same time. The third term is the diversity inducing term which tries to introduce more diverse information for optimal Laplacian matrix construction by minimizing the overall pair-wise correlation between the base affinity matrices [32].

In Eq. (9), the PSD and non-positive constraints are added to guarantee that the learned matrix $\mathbf{L}^*$ to be a Laplacian matrix. However, these constraints also make the corresponding optimization problem hard and inefficient to solve. To tackle the problem, we take advantage of the original definition of the Laplacian matrix, and propose the following formulation:

$$\begin{aligned} &\min_{\mathbf{H},\boldsymbol{\mu},\mathbf{P},\boldsymbol{\Lambda}} \mathrm{Tr}\left(\mathbf{H}^\top (\mathbf{I}_n - \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^\top)\mathbf{H}\right) \\ &+ \sum_{o=1}^{O} \|(\mathbf{I}_n - \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^\top) - \mathbf{L}_{\boldsymbol{\mu}}^{(o)}\|_{\mathrm{F}}^2 + \alpha \boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu}, \\ &\text{s.t. } \mathbf{L}_{\boldsymbol{\mu}}^{(o)} = \sum_{p=1}^{v} \mu_p \mathbf{L}_p^{(o)} (o \in [O]), \|\boldsymbol{\mu}\|_1 = 1, \boldsymbol{\mu} \geq 0, \\ &\mathbf{P} \in \mathbb{R}^{n \times k}, \mathbf{P}^\top \mathbf{P} = \mathbf{I}_k, 0 \leq \boldsymbol{\Lambda}_{ii} \leq 1, \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k, \end{aligned} \tag{10}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{k \times k}$ is a diagonal matrix. In the new formulation, we use $\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^\top$ to represent a low-rank normalized affinity matrix and $\mathbf{I}_n - \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^\top$ to represent the corresponding Laplacian matrix. Notably, the constraint $0 \leq \boldsymbol{\Lambda}_{ii} \leq 1$ is added to make sure that the optimization process is stable. The optimization procedure of Eq. (10) is listed in the appendix, please check Appendix A for details.

3.2 Late fusion-based Multi-View Spectral Clustering with Optimal Neighborhood Laplacian Matrix

To improve the efficiency of the proposed algorithm, we further propose the late fusion version of optimal neighborhood multi-view spectral clustering. In this version, we use more compact and light-weighted data partition matrices instead of the heavy-weighted Laplacian matrices to present the multi-level affinity information from different views to reduce the cost both in storage and computation. The

formulation of the late fusion method is as follows:

$$\begin{aligned}\max_{\mathbf{H}^*,\{\mathbf{W}_p^{(o)}\}_{p,o=1}^{v,O},\boldsymbol{\mu}} & \operatorname{Tr}(\mathbf{H}^{*\top}\mathbf{S})+\lambda_1\operatorname{Tr}(\mathbf{H}^{*\top}\mathbf{F})-\lambda_2\operatorname{Tr}\left(\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}\right)\\ \text{s.t. } & \mathbf{H}^{*\top}\mathbf{H}^*=\mathbf{I}_k,\mathbf{W}_p^{(o)\top}\mathbf{W}_p^{(o)}=\mathbf{I}_k,\\ & \mathbf{M}_{pq}=\sum_{o=1}^{O}\frac{\operatorname{Tr}\left(\mathbf{H}_p^{(o)\top}\mathbf{H}_q^{(o)}\right)}{\|\mathbf{H}_p^{(o)}\|_{\mathrm{F}}\|\mathbf{H}_q^{(o)}\|_{\mathrm{F}}},\\ & \sum_{p=1}^{v}\mu_p=1,\mu_p\geq 0,\mathbf{S}=\sum_{o=1}^{O}\sum_{p=1}^{v}\mu_p\mathbf{H}_p^{(o)}\mathbf{W}_p^{(o)},\\ & (\forall\, p,q\in[v] \text{ and } o\in[O])\end{aligned} \tag{11}$$

where $\mathbf{H}_p^{(o)}$ denotes the cluster indicating matrix of o-order affinity matrix of the p-th view, $\mathbf{M}$ is the correlation measuring matrix, and $\mathbf{F}$ is the cluster indicating matrix of average first-order affinity matrix $\frac{1}{v}\sum_{p=1}^{v}\mathbf{A}_p^{(1)}$. Correspondingly, the second term in the objective function is a generalization term which is added to avoid bad local maximal solution. λ_1 and λ_2 are hyper-parameters.

3.3 Optimization Algorithm

In this part, we design an efficient three-step alternative optimization algorithm to solve the problem in Eq. (11):

i) Update $\mathbf{H}^*$. Given $\{\mathbf{W}_p^{(o)}\}_{p,o=1}^{v,O}$ and $\boldsymbol{\mu}$, the optimization problem in Eq. (11) w.r.t. $\mathbf{H}^*$ reduces to:

$$\max_{\mathbf{H}^*}\operatorname{Tr}(\mathbf{H}^{*\top}\mathbf{C}) \text{ s.t. } \mathbf{H}^{*\top}\mathbf{H}^*=\mathbf{I}_k, \tag{12}$$

where $\mathbf{C}=\sum_{o=1}^{O}\sum_{p=1}^{v}\mu_p\mathbf{H}_p^{(o)}\mathbf{W}_p^{(o)}+\lambda_1\mathbf{F}$. The optimization of Eq. (12) could be easily solved by doing singular value decomposition(SVD) over the given matrix $\mathbf{C}$. Suppose that the matrix $\mathbf{C}$ in Eq. (12) has the economic rank-k singular value decomposition form as $\mathbf{C}_k=\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k{}^\top$, where $\mathbf{U}_k\in\mathbb{R}^{n\times k}$, $\mathbf{\Sigma}_k\in\mathbb{R}^{k\times k}$, $\mathbf{V}_k\in\mathbb{R}^{k\times k}$. Through Theorem 1, we can find that Eq. (12) has a closed-form solution, i.e $\mathbf{H}^*=\mathbf{U}_k\mathbf{V}_k{}^\top$.

Theorem 1. *Suppose that the matrix $\mathbf{C}$ in Eq. (12) has the economic rank-k singular value decomposition form as $\mathbf{C}_k=\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k{}^\top$, where $\mathbf{U}_k\in\mathbb{R}^{n\times k}$, $\mathbf{\Sigma}_k\in\mathbb{R}^{k\times k}$, $\mathbf{V}_k\in\mathbb{R}^{k\times k}$. The optimization in Eq. (12) has a closed-form solution as follows:*

$$\mathbf{H}^*=\mathbf{U}_k\mathbf{V}_k{}^\top. \tag{13}$$

Proof. By taking the the normal singular value decomposition $\mathbf{C}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, the Eq. (12) could be rewritten as,

$$\operatorname{Tr}(\mathbf{H}^{*\top}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top)=\operatorname{Tr}(\mathbf{V}^\top\mathbf{H}^{*\top}\mathbf{U}\mathbf{\Sigma}).$$

Due to the singular values of each $\mathbf{H}_p^{(o)}$ are non-negative, after rotating by $\mathbf{W}_p^{(o)}$, the singular values of $\mathbf{H}_p^{(o)}\mathbf{W}_p^{(o)}$ are the same with $\mathbf{H}_p^{(o)}$. Since $\mathbf{C}$ is the linear combination of $\mathbf{H}_p^{(o)}\mathbf{W}_p^{(o)}$'s and the singular values of $\mathbf{F}$ are also non-negative, we can obtain that the singular values of $\mathbf{C}$ are non-negative. Considering that $\mathbf{Q}=\mathbf{V}^\top\mathbf{H}^{*\top}\mathbf{U}$, we have $\mathbf{Q}\mathbf{Q}^\top=\mathbf{V}^\top\mathbf{H}^{*\top}\mathbf{U}\mathbf{U}^\top\mathbf{H}^*\mathbf{V}=\mathbf{I}_k$. Therefore we can take $\operatorname{Tr}(\mathbf{V}^\top\mathbf{H}^{*\top}\mathbf{U}\mathbf{\Sigma})=\operatorname{Tr}(\mathbf{Q}\mathbf{\Sigma})\leq\sum_{i=1}^{k}\sigma_i$. Hence, in order to maximize the value of Eq. (12), the solution should be given as Eq. (13). □

ii) Update $\{\mathbf{W}_p^{(o)}\}_{p,o=1}^{v,O}$. Given $\mathbf{H}^*$ and $\boldsymbol{\mu}$, for each single $\mathbf{W}_p^{(o)}$, the optimization problem in Eq. (11) reduces to:

$$\max_{\mathbf{W}_p^{(o)}}\operatorname{Tr}\left(\mathbf{W}_p^{(o)\top}\mathbf{A}\right) \text{ s.t. } \mathbf{W}_p^{(o)\top}\mathbf{W}_p^{(o)}=\mathbf{I}_k, \tag{14}$$

where $\mathbf{A}=\mu_p\mathbf{H}_p^{(o)\top}\mathbf{H}^*$. The solution of Eq. (14) is similar with that of Eq. (12), it also has a closed form solution as shown in Theorem 1.

iii) Update $\boldsymbol{\mu}$. Given $\mathbf{H}^*$ and $\{\mathbf{W}_p^{(o)}\}_{p,o=1}^{v,O}$, the optimization problem in Eq. (11) w.r.t $\boldsymbol{\mu}$ reduces to:

$$\begin{aligned}&\min_{\boldsymbol{\mu}}\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}-\mathbf{f}^\top\boldsymbol{\mu},\\ &\text{s.t. } \|\boldsymbol{\mu}\|_1=1,0\leq\mu_p\leq 1(p\in[v]),\end{aligned} \tag{15}$$

where $\mathbf{f}_p=\frac{\lambda_1}{\lambda_2}\operatorname{Tr}\left[\mathbf{H}^{*\top}\left(\sum_{o=1}^{O}\mathbf{H}_p^{(o)}\mathbf{W}_p^{(o)}\right)\right]$ $(p\in[v])$. Since $\mathbf{M}$ is PSD [31], the above function is a standard convex quadratic programming(QP) problem, its global optimal solution can be easily solved by the optimization toolbox of MATLAB.

In sum, our algorithm for solving Eq. (11) is outlined in Algorithm 1, where $obj_{(t)}$ denotes the objective value at the t-th iteration.

Algorithm 1 Late Fusion-based Optimal Neighborhood Multi-view Spectral Clustering

Input: Data from v views $\{\mathbf{X}_{(1)},...,\mathbf{X}_{(v)}\}$, number of cluster k, parameter λ_1, λ_2 and the neighbor number N

Output: The learned optimal cluster indicating matrix $\mathbf{H}^*$

1: Construct first-order and high-order affinity matrices and the corresponding Laplacian matrices $\mathbf{L}_p^{(o)}$ of each view. Obtain the cluster indicating matrix $\mathbf{H}_p^{(o)}$ by standard spectral clustering of $\mathbf{L}_p^{(o)}$. Initialize $\mathbf{H}^*$ as $\mathbf{0}_{n\times k}$, $\boldsymbol{\mu}$ as $\mathbf{1}_v/v$, $\mathbf{W}_p^{(o)}$ as $\mathbf{I}_{k\times k}$, and t as 1.
2: **repeat**
3: Calculate

$$\mathbf{H}_{(t-1)}^*=\sum_{p=1}^{v}\sum_{o=1}^{O}\boldsymbol{\mu}_{p_{(t-1)}}\mathbf{H}_{p_{(t-1)}}^{(o)}\mathbf{W}_{p_{(t-1)}}^{(o)}.$$

4: Calculate $\mathbf{W}_{p_{(t)}}^{(o)}$ by optimizing Eq. (14).
5: Calculate $\boldsymbol{\mu}_{(t)}$ by optimizing Eq. (15).
6: Calculate $\mathbf{H}_{(t)}^*$ by optimizing Eq. (12).
7: $t=t+1$.
8: **until** $|Obj_{(t)}-Obj_{(t-1)}|/|Obj_{(t)}|<10^{-4}$.

3.4 Algorithmic Discussion

Convergence. In each optimization iteration of Algorithm 1, two eigenvalue decomposition problems and one convex quadratic programming problem are solved. Since all these three sub-problems have optimal solution, the objective of Algorithm 1 is guaranteed to be monotonically increased when optimizing one variable with others fixed at each step. Moreover, the objective function is upper bounded by Proposition 1. As a result, since the objective value rises monotonically while is upper bounded, our proposed optimization algorithm is guaranteed to converge to a local optimum of problem Eq. (11).

Proposition 1. *The value of the objective function in Eq.* (11) *is upper bounded by* $\frac{1}{2}(1+O^2v^2+2\lambda_1)k$.

The proof of Proposition 1 can be found in Appendix B.

Computational Complexity. The computation of the proposed algorithm mainly includes four parts, i.e., an initialization procedure and three-step alternative optimization procedures. Among these steps, in the initialization procedure, calculating the corresponding cluster indicating matrices $\mathbf{H}_p^{(o)}$ requires conducting SVD on $n \times n$ Laplacian matrices, which incurs $\mathcal{O}\left(Ovn^3\right)$ complexity. In the optimization procedure, updating the optimal cluster indicating $\mathbf{H}^*$ requires solving a SVD problem on a $n \times k$ matrix, which takes $\mathcal{O}\left(nk^2\right)$ time. Updating the $\{\mathbf{W}_p^{(o)}\}_{p=1,o=1}^{v,O}$ requires solving Ov SVD problems over $k \times k$ matrices. It will incur $\mathcal{O}\left(Ovk^3\right)$ computational complexity. Finally, updating $\boldsymbol{\mu}$ requires solving a standard Quadratic Programming with Linear Constraints (QPLC) whose complexity is $\mathcal{O}\left(\epsilon^{-1}v\right)$, where ϵ is the precision of the result. Let T be the iteration number, the overall complexity of our algorithm is $\mathcal{O}\left(Ovn^3+T(Ovk^3+nk^2+\epsilon^{-1}v)\right)$. Considering that ϵ^{-1}, k, O, and v far less than n, the complexity is basically $\mathcal{O}\left(n^3+Tn\right)$.

3.5 Scale to Large-Scale Dataset

In the previous section, by analyzing the computational complexity of algorithm 1, we find the computational bottleneck mainly lies in the computation of the cluster indicating matrices of each view. This makes our algorithm hard to extend to large-scale datasets. In this section, we further adopt an efficient Nyström algorithm to make the proposed algorithm more suitable for large scale datasets.

Specifically, we follow the suggestion of Li et al. [19] to combine the Nyström algorithm with randomized SVD and propose an algorithm to efficiently learn the spectral embedding of large-scale datasets. Since calculating the eigenvectors corresponding to the smallest k eigenvalues of $\mathbf{L}$ is equivalent with calculating the eigenvectors corresponding to the largest eigenvalues of the normalized affinity matrix $\mathbf{G} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, for the application convenience of the Nyström method, we take $\mathbf{G}$ instead of $\mathbf{L}$ for calculation.

Nyström Method. The Nyström method [33] has been commonly used to constuct low-rank matrix approximation. Given a symmetric matrix $\mathbf{G} \in \mathbb{R}^{n\times n}$, this algorithm first samples $m(\ll n)$ columns from $\mathbf{G}$ (denote the columns selected as $\mathbf{E} \in \mathbb{R}^{n\times m}$). Let $\mathbf{R}$ be the $m \times m$ matrix consisting of the intersection of these m columns with the corresponding m rows of $\mathbf{G}$. The rows and columns of $\mathbf{G}$ can be rearranged such that $\mathbf{E}$ and $\mathbf{G}$ are written as:

$$\mathbf{E} = \begin{bmatrix} \mathbf{R} \\ \mathbf{R}^{'} \end{bmatrix} \text{ and } \mathbf{G} = \begin{bmatrix} \mathbf{R} & \mathbf{R}^{'\top} \\ \mathbf{R}^{'} & \mathbf{R}^{''} \end{bmatrix},$$

where $\mathbf{R}^{'} \in \mathbb{R}^{(n-m)\times m}$ and $\mathbf{R}^{''} \in \mathbb{R}^{(n-m)\times(n-m)}$. Assume that the SVD of $\mathbf{R}$ is $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{\Lambda} = \text{diag}(\sigma_1, ..., \sigma_m)$ is the diagonal matrix containing the singular values of $\mathbf{R}$ in non-increasing order. For $k \leq m$, the rank-k Nyström approximation is

$$\widetilde{\mathbf{G}} = \mathbf{E}\mathbf{R}_k^+\mathbf{E}^\top,$$

where $\mathbf{R}_k^+ = \sum_{i=1}^k \sigma_i^{-1}\mathbf{U}^{(i)}\mathbf{U}^{(i)\top}$, and $\mathbf{U}^{(i)}$ is the i-th column of $\mathbf{U}$. The time complexity of the approximation is $\mathcal{O}\left(nmk+m^3\right)$, which is much smaller than the $\mathcal{O}\left(n^3\right)$ complexity of doing standard SVD on $\mathbf{G}$ When $m \ll n$.

Randomized SVD. When the number of sampled columns m is large, the complexity $\mathcal{O}\left(nmk+m^3\right)$ of the Nyström method is still unacceptable. To tackle the problem, Halko et al. [34] propose a class of simple but efficient randomized algorithm. Our adopted algorithm includes two stages. In the first stage, we generate a $m \times (k+s)$ standard Gaussian random matrix $\mathbf{\Omega}$ (i.e., each entry of $\mathbf{\Omega}$ is an independent Gaussian random variable with zero mean and unit variance). And then, we form the matrix $\mathbf{Y} = \mathbf{R}\mathbf{\Omega}$ and construct a matrix $\mathbf{Q}$ whose columns form an orthogonal basis for the range of $\mathbf{Y}$ (by QR decomposition). As a consequence, we find an approximate basis $\mathbf{Q}$ for the range of $\mathbf{R}$, such that $\mathbf{R} \approx \mathbf{Q}\mathbf{Q}^\top\mathbf{R}$. The number of columns in $\mathbf{\Omega}$ is often set to be larger than the required rank k by an over-sampling parameter s. Typically, s is a small number such as 5 or 10. It enables $\mathbf{Y} = \mathbf{R}\mathbf{\Omega}$ to have a better chance to span the k-dimensional subspace of $\mathbf{R}$. In the second stage, $\mathbf{R}$ is restricted to the obtained subspace from $\mathbf{Y}$, leading to the reduced matrix $\mathbf{B} = \mathbf{Q}^\top\mathbf{R}\mathbf{Q}$. A standard SVD is computed on $\mathbf{B}$ to obtain $\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$. The SVD of $\mathbf{R}$ can then be approximated as

$$\mathbf{R} \approx \mathbf{Q}\mathbf{B}\mathbf{Q}^\top = (\mathbf{Q}\mathbf{V})\mathbf{\Lambda}(\mathbf{Q}\mathbf{V})^\top.$$

The operations of the randomized SVD are shown in Algorithm 2. The proposed fast spectral embedding via Nyström and randomized SVD are reported in Algorithm 3. By

Algorithm 2 Randomized SVD

Input: symmetric matrix $\mathbf{R} \in \mathbb{R}^{m\times m}$, rank k, over-sampling parameter s

Output: $\mathbf{U}$, $\mathbf{\Lambda}$

1: $\mathbf{\Omega} \leftarrow$ a $m \times (k+s)$ standard Gaussian random matrix.
2: $\mathbf{Y} \leftarrow \mathbf{R}\mathbf{\Omega}$.
3: Find an orthogonal matrix $\mathbf{Q}$ (by QR decomposition) such that $\mathbf{Y} = \mathbf{Q}\mathbf{Q}^\top\mathbf{Y}$.
4: $\mathbf{B} \leftarrow \mathbf{Q}^\top\mathbf{R}\mathbf{Q}$.
5: Perform SVD on $\mathbf{B}$ to obtain $\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$.
6: $\mathbf{U} \leftarrow \mathbf{Q}\mathbf{V}$.

Algorithm 3, matrix $\hat{\mathbf{G}} = \left(\sqrt{\frac{m}{n}}\mathbf{U}\right)\left(\frac{n}{m}\mathbf{\Lambda}\right)\left(\sqrt{\frac{m}{n}}\mathbf{U}^\top\right)$ can be regarded as an approximation of the input matrix $\mathbf{G}$, and $\mathbf{H}$ is the cluster indicating matrix. According to the conclusion in Theorem 2 of [19], the approximation error of Algorithm 3 is upper bounded.

Theorem 2 ([19]). *For the* $\mathbf{G} = (\mathbf{D})^{-\frac{1}{2}}\mathbf{A}(\mathbf{D})^{-\frac{1}{2}}$ *and* $\hat{\mathbf{G}} = \left(\sqrt{\frac{m}{n}}\mathbf{U}\right)\left(\frac{n}{m}\mathbf{\Lambda}\right)\left(\sqrt{\frac{m}{n}}\mathbf{U}^\top\right)$ *obtained by Algorithm 3,*

$$\begin{aligned}&\mathbb{E}\|\mathbf{G}-\hat{\mathbf{G}}\|_F \\ \leq&\frac{2(k+s)}{\sqrt{s-1}}\|\mathbf{G}-\mathbf{G}_k\|_F + \left(1+\frac{4(k+s)}{\sqrt{m(s-1)}}\right)n\mathbf{G}_{ii}^*,\end{aligned} \tag{16}$$

where $\mathbf{G}_k$ *is the best rank-k approximation of* $\mathbf{G}$, $\mathbf{G}_{ii}^* = \max_i \mathbf{G}_{ii}$.

Note that the approximate eigenvectors $\mathbf{H}$ obtained by algorithm 3 may not be orthogonal. According to [35], we orthogonalize $\mathbf{H}$ by the Algorithm 4.

The following proposition shows that, after performing Algorithm 4, the $\mathbf{H}$ has an orthogonalization version $\tilde{\mathbf{H}}$.

Algorithm 3 Fast Spectral Embedding via Nyström and Randomized SVD

Input: the p-th view $\mathbf{X}_p(p \in [v]) \in \mathbb{R}^{n\times d}$, number of sampled columns m, over-sampling parameter s, number of cluster k

Output: the cluster indicating matrix $\mathbf{H} \in \mathbb{R}^{n\times k}$ of the o-th order affinity matrix, the k largest approximate eigenvalues $\mathbf{\Lambda}_k$.

1: Construct o-th order affinity matrix $\mathbf{A}$ by Eq. (1) and Eq. (7).
2: Compute the degree matrix $\mathbf{D}$ by Eq. (2).
3: $\mathbf{G} \leftarrow (\mathbf{D})^{-\frac{1}{2}}\,\mathbf{A}\,(\mathbf{D})^{-\frac{1}{2}}$.
4: $\mathbf{E} \leftarrow m$ columns of $\mathbf{G}$ sampled uniformly at random without replacement.
5: $\mathbf{R} \leftarrow$ the intersection of the m columns sampled in the step 4 with the corresponding m rows of $\mathbf{G}$.
6: $[\widetilde{\mathbf{U}}, \mathbf{\Lambda}] \leftarrow$ randsvd$(\mathbf{R}, k, s)$ using Algorithm 2.
7: $\mathbf{U} \leftarrow \mathbf{E}\widetilde{\mathbf{U}}\mathbf{\Lambda}^{+}$.
8: $\mathbf{H} \leftarrow \sqrt{\frac{m}{n}}\mathbf{U}$.

Algorithm 4 Orthogonalize $\mathbf{H}$

Input: $\mathbf{H} \in \mathbb{R}^{n\times k}, \mathbf{\Lambda} \in \mathbb{R}^{k\times k}$
Output: orthogonal $\widetilde{\mathbf{H}}, \widetilde{\mathbf{\Lambda}}$

1: $\mathbf{T} \leftarrow \mathbf{H}^{\top}\mathbf{H}$.
2: eigen-decomposition: $\mathbf{T} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^{\top}$.
3: $\mathbf{K} \leftarrow \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{V}^{\top}\mathbf{\Lambda}\mathbf{V}\mathbf{\Sigma}^{\frac{1}{2}}$.
4: eigen-decomposition: $\mathbf{K} = \widetilde{\mathbf{V}}\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{V}}^{\top}$.
5: $\widetilde{\mathbf{H}} \leftarrow \mathbf{H}\mathbf{V}\mathbf{\Sigma}^{-\frac{1}{2}}\widetilde{\mathbf{V}}$.

Proposition 2. *In Algorithm 4,* $\mathbf{H}\mathbf{\Lambda}\mathbf{H}^{\top} = \widetilde{\mathbf{H}}\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{H}}^{\top}$, *and* $\widetilde{\mathbf{H}}^{\top}\widetilde{\mathbf{H}} = \mathbf{I}$.

4 Experiments

4.1 Datasets and Experimental Settings

We evaluate the clustering performance of the proposed algorithm on 9 popular datasets from various applications, including natural language processing, protein subcellular localization, and image recognition. The detailed information of these datasets is listed in Table 2. From this table, we observe that the number of samples, views, and clusters of these datasets range from 165 to 60,000, 2 to 69, and 3 to 102, respectively. For these datasets, all affinity matrices are precomputed with carefully designed similarity function and are publicly available from websites[1][2][3]. We assume that the number of clusters is pre-specified. During our experiments, we set the number of clusters to be the true number of classes.

In our experiments, the MATLAB implementation of all the compared algorithms is downloaded from the authors' websites. The hyper-parameters are set according to the suggestions of the corresponding literature. Especially, to all the compared spectral clustering algorithms,

1. http://mlg.ucd.ie/datasets/bbc.html
2. http://mkl.ucsd.edu/dataset/protein-fold-prediction
3. http://www.robots.ox.ac.uk/ vgg/data/

TABLE 2
Information of benchmark datasets. In this table, the number of samples, views, and categories of these datasets are illustrated.

Datasets	# Samples	# Views	# Clusters
YALE	165	12	15
BBCSport	554	2	5
ProteinFold	694	12	27
Flower17	1360	7	17
UCI-Digit	2000	3	10
Mfeat	2000	12	10
Nonpl	2732	69	3
Flower102	8189	4	102
MNIST	60000	3	10

the optimal neighbor numbers are carefully searched in the range of $[0.1s, 0.2s, \ldots, s]$, where $s = n/k$ is the average sample number in each category. As to our proposed method, the parameter λ_1 and λ_2 are chosen in the range of $[2^{-15}, 2^{-12}, \ldots, 2^{15}]$. K-means clustering is adopted on the final representation to assign an appropriate label for each sample. In the experiment, to reduce the effect of randomness caused by k-means, we repeat the clustering process for 50 times with random initialization and report the result with the smallest k-means distortion. The clustering performance is evaluated in terms of three widely used criteria, including clustering accuracy (ACC), normalized mutual information (NMI), and purity. All our experiments are conducted on a desktop computer with 3.6GHz Intel Core i7 CPU, 64GB RAM, and MATLAB 2018a (64bit).

4.2 Ablation Study

In our first experiment, we study the effectiveness of each proposed component, i.e., the neighborhood learning mechanism (NLM) and the high-level connection information (HCI) by careful ablation study. Also, the optimal order number of the high-order Laplacian matrix is exploited. Specifically, six algorithms are designed and tested. The average clustering performance on all eight datasets are listed in Table 3.

Effectiveness of the designed algorithm. Among the compared algorithms, the baseline method (BL) indicates a classic linear Laplacian matrix combination with matrix-induced regularization [32]. For high-level connection information extraction, the order number of the Laplacian matrix is fixed as 2. As we can see from Table 3, both the neighborhood learning mechanism and the high-level connection information is capable of improving the spectral clustering performance of the corresponding algorithm. Specifically, HCI and NLM improve the ACC of the baseline algorithm for 2.04% and 3.63% on average, respectively. Moreover, by combining these two designs, the resultant algorithm can improve 4.64% over the baseline algorithm in terms of ACC.

The optimal order-number. We also test the effect of different order-number of the high-order Laplacian matrices. In this part, the second-, third-, forth- and fifth-order algorithms are compared. As one can see in Table 4, the second- and the third-order algorithms provide comparably good performance. However, as the orders of the Laplacian matrices keep get higher, the range of neighborhood also gets larger and the discriminative capacity of the corresponding algorithms start to decrease a little bit. As a consequence, for the sake of the clustering performance and the com-

TABLE 3
Ablation study. Average clustering performance on eight datasets of four algorithms. In the compared algorithms, BL indicates the baseline method, NLM indicates neighborhood learning mechanism, HCI indicates high-level connection information.

Methods	BL	BL+HCI	BL+NLM	BL+NLM+HCI
ACC (%)	64.00	66.04	67.63	**68.64**
NMI (%)	63.58	64.53	66.42	**67.65**
Purity (%)	68.32	70.01	71.78	**72.29**

TABLE 4
Average clustering performance comparison with different Laplacian matrix order number.

Methods	2nd-order	3rd-order	4th-order	5th-order
ACC (%)	68.64	**68.94**	67.58	65.76
NMI (%)	**67.65**	67.04	66.23	64.94
Purity (%)	**72.29**	72.21	71.87	69.88

putational efficiency, the order number of our proposed algorithm is fixed as two in all our following experiments.

4.3 Comparison with state-of-the-art algorithms

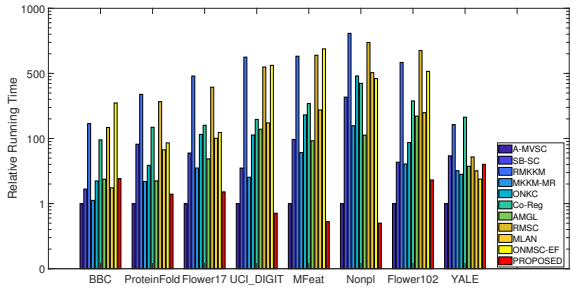
To verify the effectiveness of the proposed algorithm, we further compare it with six state-of-the-art multi-view spectral clustering algorithms and three multiple kernel clustering algorithms. Among these methods, (1) average multi-view spectral clustering (**A-MVSC**) uniformly weights Laplacian matrices from each view to generate a new Laplacian matrix for clustering (2) Single best spectral clustering (**SB-SC**) performs spectral clustering on every single view separately and reports the best performance. (3) Co-regularized Spectral Clustering (**Co-reg**) [5] is a representative of the co-training methods. (4) Auto-weighted Multiple Graph Learning (**AMGL**) [36] is a linear combination-based method. (5) Multi-view Learning with Adaptive Neighbors (**MLAN**) [37], and (6) Robust Multi-view Spectral Clustering **RMSC** [38] are consensus Laplacian construction methods. Also, since the affinity matrices in each view can be treated as kernels, three multiple kernel clustering algorithms, i.e., (7) **ONKC** [22], (8) **MKKM-MR** [32], and (9) **RMKKM** [39], are also included for a more comprehensive comparison. The early fusion and late fusion version of our proposed algorithm refers to ONMSC-EF and ONMSC-LF in Table 5, respectively.

The ACC, NMI, and purity, of the algorithms mentioned above are reported in Table 5. As seen, in all of the eight datasets, both the proposed ONMSC-LF and ONMSC-EF show superior performance gains over the state-of-the-art algorithms w.r.t. all the three metrics. Also, the proposed algorithm significantly outperforms existing linear combination based algorithms, including RMKKM, MKKM-RM, and AMGL with comparable computational consumption. This validates the effectiveness of optimal neighborhood spectral clustering and the high-order information again.

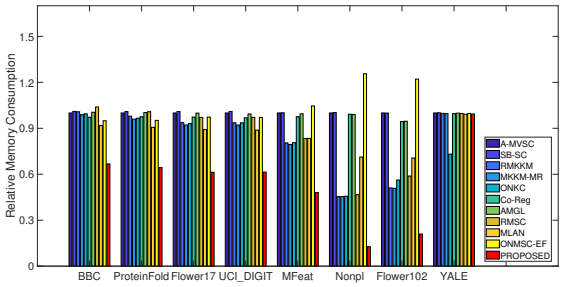
4.4 Running Time and Memory Consumption

To compare the computational efficiency of the proposed algorithms, we record the running time and maximum memory consumption of various algorithms on the benchmark datasets and report them in Fig. 2. The result in figure 2 is obtained by dividing by the result of A-MVSC. As we can see in Fig. 2-(a), without the acceleration of the late-fusion mechanism and the Nyström algorithm, the computational time of the early fusion version of our algorithm is comparable with most of the compared state-of-the-art algorithms. Also, with the enhancement of the mentioned techniques, the late fusion version of our proposed algorithm achieves $\mathcal{O}(n)$ time complexity.

Moreover, from Fig. 2-(b), we can see that the proposed ONMSC-LF has the smallest memory consumption compared to other algorithms. The results in Fig. 2 well demonstrate the efficiency of the proposed algorithm both in computation and storage.



(a) Illustration of the relative running time of the compared algorithms.



(b) Illustration of the relative maximum memory consumption of the compared algorithms.

Fig. 2. The running time and maximum memory consumption comparison of different algorithms on eight benchmark datasets. Both the running time and the memory cost of the compared algorithms are divided by the corresponding results of A-MVSC.

4.5 Visualization of Algorithm Performance

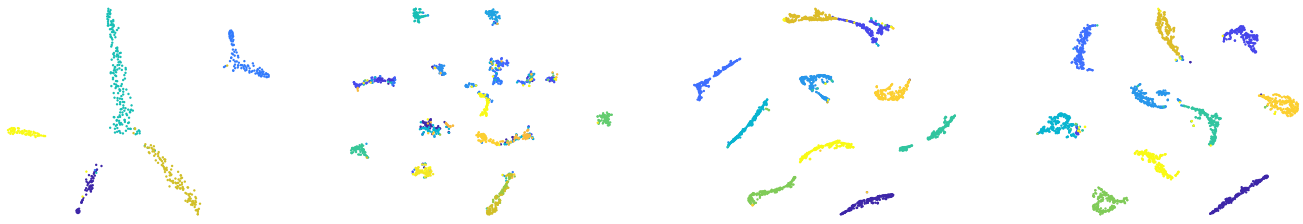
To illustrate the performance of the compared algorithms more intuitively, we further adopt t-distributed stochastic neighbor embedding(t-SNE) [40] to visualized the acquired cluster structure of six representative datasets. Specifically, only the results of our proposed algorithm and the second best results are illustrated. As we can see from Fig. 3, the proposed algorithm better reveals the underlying local geometric structure of in all the six datasets, validating its superior performance.

4.6 Parameter Sensitivity and Convergence

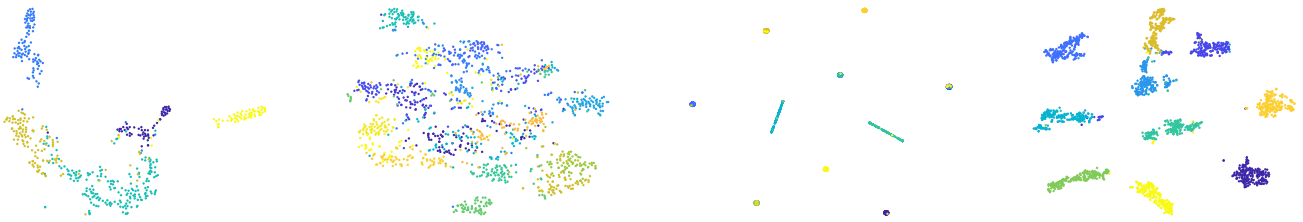
Parameter Sensitivity. The proposed ONMSC-LF introduces three hyper-parameters, i.e., the average view balancing coefficient λ_1, the diversity balancing coefficient λ_2, and the neighbor number N for affinity matrix construction. To

TABLE 5
ACC, NMI, purity comparison of different clustering algorithms on eight benchmark datasets. In this table, the boldface indicates the best performance among all the compared algorithms.

Datasets	A-MVSC	SB-SC	RMKKM [39]	MKKM-MR [32]	ONKC [22]	Co-reg [5]	AMGL [36]	RMSC [38]	MLAN [37]	ONMSC-EF	ONMSC-LF
ACC(%)											
BBCSports	66.18	76.65	63.79	66.18	68.20	85.66	86.39	86.03	70.58	95.77	**97.61**
ProteinFold	30.69	34.58	30.98	36.46	37.90	34.87	36.88	33.00	28.38	**41.21**	40.48
Flower17	51.02	42.05	48.38	60.00	60.88	52.72	56.32	53.90	53.38	66.39	**67.5**
UCI-Digit	88.75	75.40	40.45	90.40	91.05	84.80	92.85	90.40	97.15	97.6	**97.85**
MFeat	95.20	86.00	65.30	83.20	97.05	84.30	84.35	84.15	96.55	**98.1**	97.00
Nonpl	49.37	57.50	62.77	56.59	59.57	55.27	56.91	60.65	44.98	65.84	**68.41**
Flower102	27.29	33.12	28.17	39.91	41.56	37.26	33.34	32.97	24.19	43.31	**44.47**
YALE	46.06	44.24	58.79	60.00	61.21	51.52	60.0	56.36	57.58	64.85	**66.06**
NMI(%)											
BBCSport	53.92	59.38	39.62	53.93	54.64	71.27	73.7	73.89	65.34	87.19	**92.00**
ProteinFold	40.95	42.33	38.78	45.32	46.93	43.34	44.18	43.91	27.86	49.33	**49.5**
Flower17	50.18	45.14	50.73	57.11	58.58	52.13	56.97	53.89	55.38	65.54	**66.35**
UCI-Digit	80.59	68.38	46.87	83.22	83.96	73.51	86.65	81.8	93.4	94.39	**94.86**
MFeat	89.83	75.78	62.67	78.12	93.07	80.99	81.57	81.69	92.89	**95.51**	93.43
Nonpl	16.55	15.26	17.34	15.51	24.04	12.55	15.19	20.35	6.14	**25.35**	24.44
Flower102	46.32	48.99	48.17	57.27	59.13	54.18	51.63	53.36	34.94	60.12	**60.74**
YALE	49.04	50.42	59.70	61.29	62.27	57.01	61.2	59.11	57.07	64.40	**64.46**
Purity(%)											
BBCSport	77.2	79.59	67.83	77.21	77.76	85.66	86.39	86.03	74.44	95.77	**97.61**
ProteinFold	37.17	41.21	36.6	42.65	45.24	40.78	42.07	42.36	31.84	47.98	**49.56**
Flower17	51.98	44.63	51.54	61.03	61.69	56.47	58.16	53.24	55.07	68.52	**69.7**
UCI-Digit	88.75	76.1	44.2	90.4	91.05	77.75	92.85	82.9	97.15	97.6	**97.85**
MFeat	95.2	86	66.25	83.2	97.05	84.3	84.35	84.1	96.55	**98.1**	97
Nonpl	72.18	71.12	71.71	63.91	75.34	66.07	69.94	70.5	60.35	**76.13**	75.66
Flower102	32.27	38.78	27.61	33.86	47.64	44.08	39.71	40.24	31.15	50.78	**51.75**
YALE	48.48	47.88	59.39	60.21	61.82	54.55	60.61	57.58	58.18	65.45	**66.67**



(a) BBCsport (proposed) (b) Flower17 (proposed) (c) UCI-Digit (proposed) (d) Mfeat (proposed)



(e) BBCsport (RMSC) (f) Flower17 (ONKC) (g) UCI-Digit (MLAN) (h) Mfeat (ONKC)

Fig. 3. Clustering structure visualization through the t-SNE algorithm [40]. In this figure, the results on four datasets, i.e., BBCSport, Flower17, UCIDight, Mfeat are illustrated. Among the sub-figures, the first row represents the results of our proposed algorithm, while the second row represents the best cluster structure other than our proposed algorithm.

test the sensitivity of the proposed algorithm against these three parameters, we fix one parameter and tune the other in a large range. Parameter sensitivity w.r.t. λ_1 and λ_2 are displayed in Fig. 4. The performance variation against N is illustrated in Fig. 5. In this figure, the performance variation w.r.t. different N is compared with the second-best state-of-the-art algorithms on eight datasets. From these figures, we observe that i) all the parameters are effective in improving the algorithm performance; ii) the proposed algorithm is stable against the these parameters that it achieves good performance in a wide range of parameter settings. iii) the proposed algorithm is relatively sensitive to the neighbor numbers.

Algorithm Convergence. Six examples of the objective values of our algorithm at each iteration are shown in Fig. 6. As observed from a) to l), the objective function monotonically increases and reaches convergence quickly. In a), g) and k), we can find that the clustering performance reaches the best results, as the objective function value reaches convergence. Because the best value of the objective function may not lead to the best clustering performance, the results in c), e), and i) have a slight variation but gradually increases with the increasing of iterations. In general, all the figures above have clearly demonstrated the effectiveness of the learned consensus matrix $\mathbf{H}^*$. Furthermore, as shown in other figures, the objective function does monotonically increase at each iteration, and it usually converges within 30 iterations.

4.7 Scale the Algorithm to Large Datasets

Being able to work appropriately on large scale datasets is an essential criterion to the practicality of a MVSC algorithm. To show the effectiveness of our proposed method, we further conduct an experiment on the MNIST dataset[4]. To construct the dataset, we first adopt three deep neural networks, i.e., VGG19 [41], DenseNet121 [42], and ResNet101 [43], which are pre-trained on the ImageNet[5] dataset as feature extractors in three different views. Then, we conduct Algorithm 3 and Algorithm 4 to obtain the cluster indicating matrix of each order and of each view. In this step, to test the influence of the anchor number of the Nyström algorithm we tune the sampled anchor points number from [10 50 100 500 1000 6000], and report the corresponding results in Fig. 7. We also visualize the clustering structure of four representative digits, i.e., 1, 3, 5, 9. From Fig. 7-(a) we observe that as the number of sampled columns increases, the performance becomes better and quickly reaches a plateau at the number of 50 anchor points. This result indicates that our algorithm can achieve favourable performance with efficient approximation of the cluster indicating matrices. The best ACC, NMI, and purity are **94.09**, **89.9**, and **94.09**, respectively. From Fig. 7-(b) we can obviously see that the cluster structure represented by our algorithm well reveal the underlying manifold of the corresponding dataset.

In addition, we compare the proposed algorithm with several state-of-the-art large-scale multi-view clustering algorithms, including: (1) single best k-means **(SB-KM)**: which performs standard k-means on every single view separately and reports the best performance, (2) robust multi-view k-means clustering **(RMKMC)** [44]: which is a robust large-scale multi-view k-means clustering algorithm, (3) large-scale multi-view subspace clustering **(LMVSC)** [45]: which replaces the full reconstruction matrix with an anchor-based reconstruction matrix for efficient subspace clustering. The ACC, NMI, purity and runtime of the above algorithms are reported in Table 6. According to the results, our method shows the best performance and competitive running time when compared with the state-of-the-art large-scale multi-view clustering methods.

TABLE 6
ACC, NMI, purity comparison of different large-scale clustering algorithms on MNIST dataset.

	SB-KM	RMKMC	LMVSC	Proposed
ACC	72.84	72.72	88.56	**94.09**
NMI	61.79	64.91	77.14	**89.9**
Purity	72.85	75.30	88.58	**94.09**
Time(s)	318.8	2935.4	69.4	27.4

5 Conclusion

This paper proposes an optimal neighborhood multi-view spectral clustering (ONMSC) algorithm and its late fusion extension. In the proposed algorithms, the early fusion version (ONMSC-EF) enlarges the searching space of optimal Laplacian matrix from the linear combination of the first-order base Laplacian matrices to the neighborhood of both the first-order and high-order Laplacian combinations. The late fusion version (ONMSC-EF) introduces a late fusion learning mechanism and drastically reduces both the computational and storage complexity of the proposed algorithm. As a consequence, the scalability of the proposed algorithm is largely improved. A three-step algorithm with proved convergence is designed to solve the resulting optimization problem. Comprehensive experimental results demonstrate the effectiveness and the superior performance of our proposed algorithm. In the future, we plan to develop new acceleration algorithms including anchor points [**?**], landmarks [**?**], to further reduce the computational complexity of constructing the base clustering indicator matrices. We also intend to extend our algorithm to a more general framework and use it as a platform to revisit existing multi-view spectral clustering algorithms.

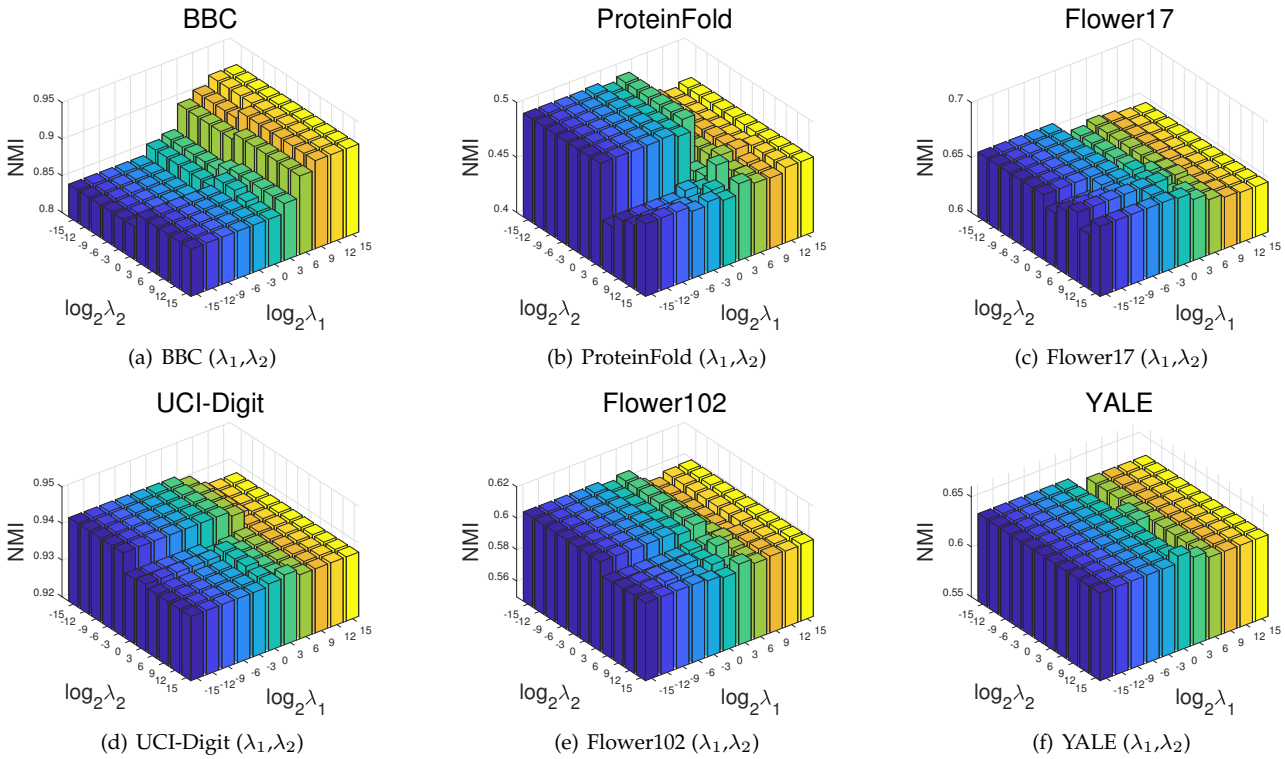
Acknowledgment

This work was supported by the Natural Science Foundation of China (project no. 61773392 and 61672528).

References

[1] V. R. De Sa, "Spectral clustering with two views," in *ICML workshop on learning with multiple views*, 2005, pp. 20–27.
[2] Y. Yang and H. Wang, "Multi-view clustering: a survey," *Big Data Mining and Analytics*, pp. 83–107, 2018.
[3] X. Yu, G. Yu, J. Wang, and C. Domeniconi, "Co-clustering ensembles based on multiple relevance measures," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2019.
[4] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE Transactions on Neural Networks and learning systems (TNNLS)*, pp. 4833–4843, 2018.

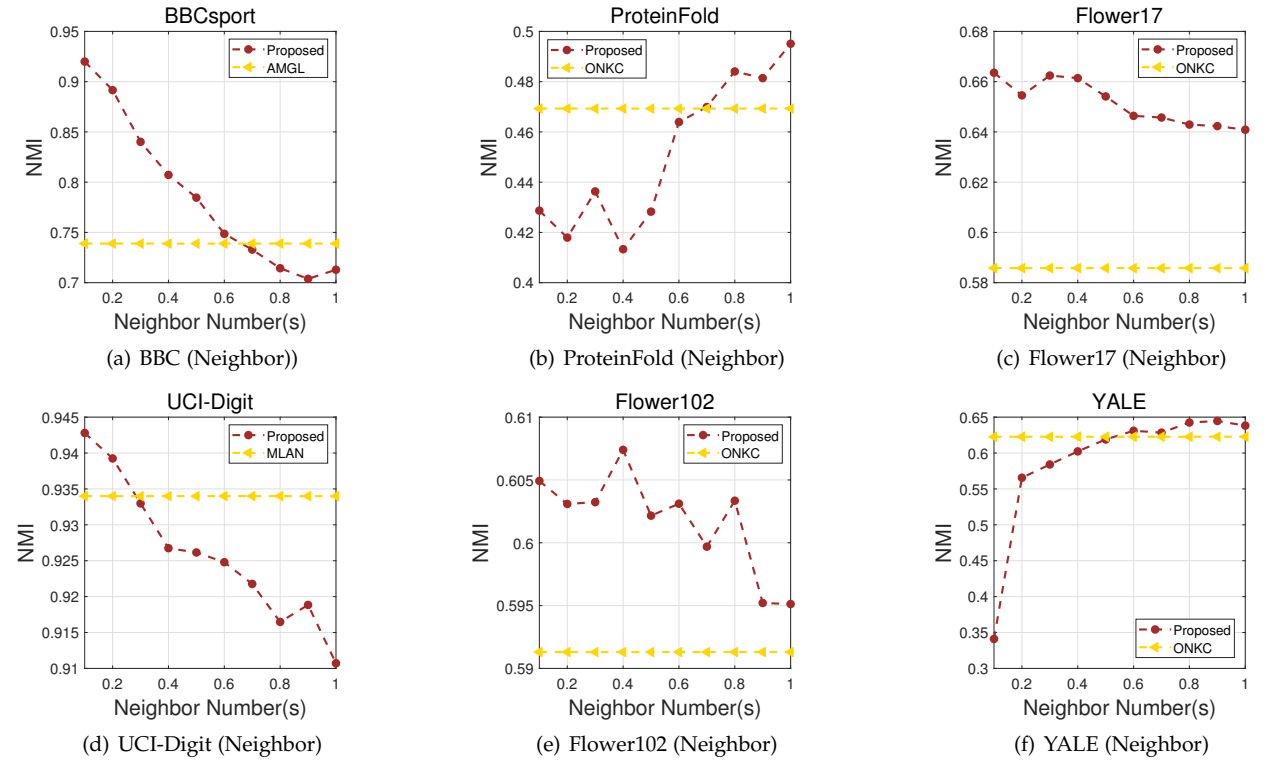
4. http://yann.lecun.com/exdb/mnist/
5. http://www.image-net.org/



(a) BBC (λ_1,λ_2) (b) ProteinFold (λ_1,λ_2) (c) Flower17 (λ_1,λ_2)

(d) UCI-Digit (λ_1,λ_2) (e) Flower102 (λ_1,λ_2) (f) YALE (λ_1,λ_2)

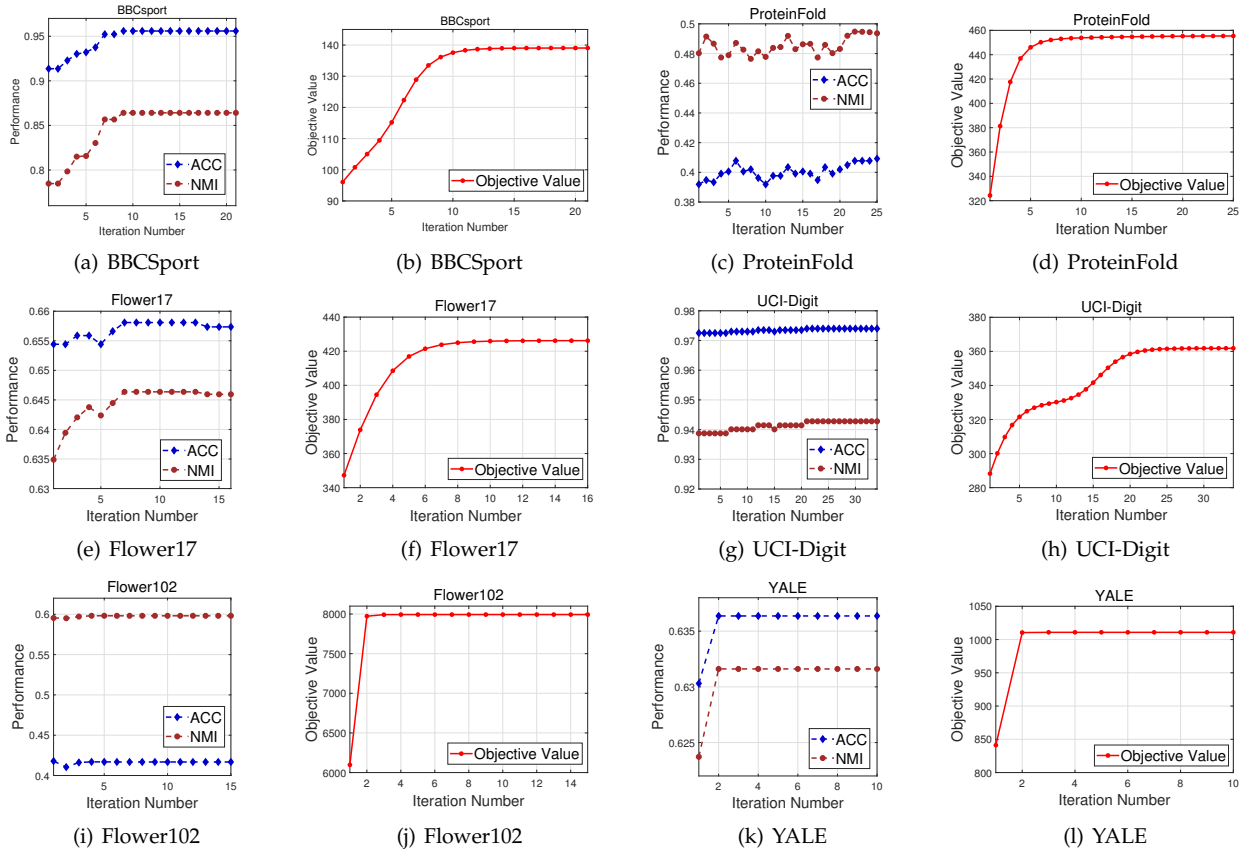
Fig. 4. Illustration of parameter sensitivity against the two hyper-parameters λ_1 and λ_2. In the experiment, both λ_1 and λ_2 are tested in the range $[2^{-15}, 2^{-12}, \cdots, 2^{15}]$. Among the figures, sub-figure a) - f) are corresponding to the performance variation of NMI on BBCSport, ProteinFold, Flower17, UCI-Digit, Mfeat, Nonpl, Flower102, and YALE respectively.



(a) BBC (Neighbor)) (b) ProteinFold (Neighbor) (c) Flower17 (Neighbor)

(d) UCI-Digit (Neighbor) (e) Flower102 (Neighbor) (f) YALE (Neighbor)

Fig. 5. Illustration of parameter sensitivity against neighbor number N. Sub-figure a) - f) are corresponding to the performance variation of NMI on BBCSport, ProteinFold, Flower17, UCI-Digit, Flower102, and YALE, respectively. The brown curves record the results of the proposed algorithm, while the yellow lines are corresponding to the second-best performance on the corresponding dataset.

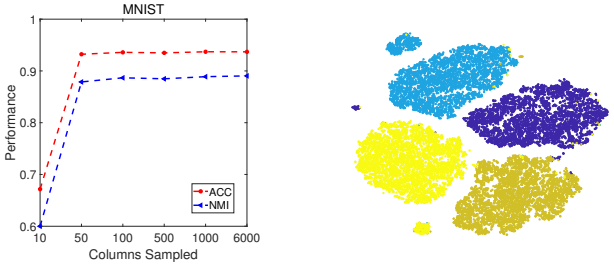


(a) BBCSport (b) BBCSport (c) ProteinFold (d) ProteinFold

(e) Flower17 (f) Flower17 (g) UCI-Digit (h) UCI-Digit

(i) Flower102 (j) Flower102 (k) YALE (l) YALE

Fig. 6. Illustration of performance variation and algorithm convergence. In these figures, the first and the third columns are corresponding to the performance evolution as the iteration increases. The blue cures are corresponding to the clustering accuracy and the brown curves are corresponding to the NMI. The second and the fourth columns represent the variation of the objective values. Results on BBCSport, ProteinFold, Flower17, UCI-Digit, Flower102 and YALE datasets are reported.



(a) Performance variation. (b) Clustering results visualization.

Fig. 7. Results illustration on the MNIST dataset. (a) represents the performance variation against the selected anchor point number of the Nyström algorithm. (b) illustrates the intrinsic cluster structure of the proposed algorithm.

[5] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *In Neural Information Processing Systems (NIPS)*, 2011, pp. 1413–1421.

[6] S. Huang, H. Wang, D. Li, Y. Yang, and T. Li, "Spectral co-clustering ensemble," *Knowledge-Based Systems (KBS)*, pp. 46–55, 2015.

[7] C. Wang, J. Lai, and P. S. Yu, "Multi-view clustering based on belief propagation," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, pp. 1007–1021, 2016.

[8] F. Nie, J. Li, X. Li *et al.*, "Self-weighted multiview clustering with multiple graphs." in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 2564–2570.

[9] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE Transactions on Image Processing (TIP)*, pp. 1261–1270, 2019.

[10] C. Tang, X. Zhu, X. Liu, M. Li, P. Wang, C. Zhang, and L. Wang, "Learning joint affinity graph for multi-view subspace clustering," *IEEE Transactions on Multimedia (TMM)*, 2018.

[11] P. Zhou, Y.-D. Shen, L. Du, F. Ye, and X. Li, "Incremental multi-view spectral clustering," *Knowledge-Based Systems (KBS)*, pp. 73–86, 2019.

[12] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (SMC)*, pp. 1438–1446, 2010.

[13] T. Semertzidis, D. Rafailidis, M. G. Strintzis, and P. Daras, "Large-scale spectral clustering based on pairwise constraints," *Information Processing and Management (IPM)*, pp. 616–624, 2015.

[14] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[15] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," 2011.

[16] D. Cai and D. Cai, "Large scale spectral clustering via landmark-based sparse representation," 2015, pp. 1669–1680.

[17] A. E. Alaoui and M. W. Mahoney, "Fast randomized kernel methods with statistical guarantees," in *In Neural Information Processing Systems (NIPS)*, 2014.

[18] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 214–225, 2004.

[19] M. Li, J. T. Kwok, and B.-L. Lu, "Making large-scale nyström approximation possible," in *International Conference on Machine Learning (ICML)*, 2010, pp. 631–638.

[20] F. R. Bach, "Exploring large feature spaces with hierarchical multiple kernel learning," in *In Neural Information Processing Systems (NIPS)*, 2009, pp. 105–112.

[21] C. Cortes, M. Mohri, and A. Rostamizadeh, "Learning non-linear combinations of kernels," in *In Neural Information Processing Systems (NIPS)*, 2009, pp. 396–404.

[22] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, and J. Yin, "Optimal neighborhood kernel clustering with multiple kernels," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[23] J.-H. Li, C.-D. Wang, P.-Z. Li, and J.-H. Lai, "Discriminative metric learning for multi-view graph partitioning," *Pattern Recognition*, pp. 199–213, 2018.

[24] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 2843–2849.
[25] P. Zhou, L. Du, X. Liu, Y. Shen, M. Fan, and X. Li, "Self-paced clustering ensemble," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, pp. 1–15, 2020.
[26] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, "Late fusion incomplete multi-view clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.
[27] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, and J. Yin, "Multi-view clustering via late fusion alignment maximization," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 3778–3784.
[28] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *In Neural Information Processing Systems (NIPS)*, 2002, pp. 849–856.
[29] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *International World Wide Web Conferences (WWW)*, 2015, pp. 1067–1077.
[30] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *In Neural Information Processing Systems (NIPS)*, 2016, pp. 3844–3852.
[31] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research (JMLR)*, pp. 795–828, 2012.
[32] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.
[33] C. K. I. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *In Neural Information Processing Systems (NIPS)*, 2001, pp. 682–688.
[34] N. Halko, P. Martinsson, and J. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, pp. 217–288, 2011.
[35] M. Li, X.-C. Lian, J. T. Kwok, and B.-L. Lu, "Time and space efficient spectral clustering via column sampling," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 2297–2304.
[36] F. Nie, J. Li, X. Li *et al.*, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, pp. 1881–1887.
[37] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.
[38] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2014.
[39] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y.-D. Shen, "Robust multiple kernel k-means using l21-norm," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.
[40] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research (JMLR)*, pp. 2579–2605, 2008.
[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.
[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[44] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.
[45] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Weixuan Liang is a graduate student in National University of Defense Technology (NUDT), China. His current research interests include kernel learning, unsupervised multiple-view learning, scalable clustering and deep unsupervised learning.

Sihang Zhou received his PhD degree in computer science from National University of Defense Technology (NUDT), China in 2019. He received his M.S. degree in computer science from the same school in 2014 and his bachelor's degree in information and computing science from the University of Electronic Science and Technology of China (UESTC) in 2012. He is now a lecturer of College of Intelligence Science and Technology, NUDT. His current research interests include machine learning, pattern recognition, and medical image analysis.

Jian Xiong received the B.S. degree in engineering, and the M.S. and Ph.D. degrees in management from National University of Defense Technology, Changsha, China, in 2005, 2007, and 2012, respectively. He is an Associate Professor with the School of Business Administration, Southwestern University of Finance and Economics. His research interests include data mining, multiobjective evolutionary optimization, machine learning, multiobjective decision making, project planning, and scheduling. Dr. Xiong has published 20+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-EC, etc.

Xinwang Liu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Assistant Researcher of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 60+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, NeurIPS, ICCV, CVPR, AAAI, IJCAI, etc. He serves as the associated editor of Information Fusion Journal. More information can be found at https://xinwangliu.github.io/.

En Zhu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor at School of Computer Science, NUDT, China. His main research interests are pattern recognition, image processing, machine vision and machine learning. Dr. Zhu has published 60+ peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation.

Siwei Wang is a graduate student in National University of Defense Technology (NUDT), China. His current research interests include kernel learning, unsupervised multiple-view learning, scalable clustering and deep unsupervised learning. He has published several peer-reviewed papers such as AAAI, IJCAI, etc. He served as reviewers of AAAI20, IJCAI20 and IEEE TCYB, IEEE TNNLS, etc.

Zhiping Cai received the B.Eng., M.A.Sc., and Ph.D degrees in computer science and technology from the National University of Defense Technology (NUDT), China, in 1996, 2002, and 2005, respectively. He is a full professor in the College of Computer, NUDT. His current research interests include network security and big data. He is a senior member of the CCF and a member of the IEEE. His doctoral dissertation has been rewarded with the Outstanding Dissertation Award of the Chinese PLA.

Xin Xu (M'07-SM'12) received the B.S. degree in electrical engineering from the Department of Automatic Control, National University of Defense Technology (NUDT), Changsha, China, in 1996, and the Ph.D. degree in control science and engineering from the College of Mechatronics and Automation, NUDT, in 2002. He has been a Visiting Professor with The Hong Kong Polytechnic University, the University of Alberta, the University of Guelph, and the University of Strathclyde, U.K. He is currently a Full Professor with the Institute of Unmanned Systems, College of Intelligence Science and Technology, NUDT. He has co-authored more than 160 papers in international journals and conferences and four books. His research interests include intelligent control, reinforcement learning, approximate dynamic programming, machine learning, robotics, and autonomous vehicles. He is a member of the IEEE CIS Technical Committee on Approximate Dynamic Programming and Reinforcement Learning and the IEEE RAS Technical Committee on Robot Learning. He received the National Science Fund for Outstanding Youth in China and the second-class National Natural Science Award of China. He has served as an Associate Editor or Guest Editor for Information Sciences, International Journal of Robotics and Automation, IEEE Transactions on Systems, Man, and Cybernetics: Systems, Intelligent Automation and Soft Computing, the International Journal of Adaptive Control and Signal Processing and Acta Automatica Sinica.