# Multi-View Spectral Clustering with Optimal Neighborhood Laplacian Matrix

**Sihang Zhou,[1] Xinwang Liu,[1*] Jiyuan Liu,[1] Xifeng Guo,[1] Yawei Zhao,[1]**
**En Zhu,[1] Yongping Zhai,[2] Jianping Yin,[3] Wen Gao[4]**
[1]College of Computer, National University of Defense Technology, Changsha 410073, China
[2]College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha 410073, China
[3]School of Cyberspace Science, Dongguan University of Technology, Guangdong 523808, China
[4]School of Electronics Engineering and Computer Science, Peking University, Beijing, China, 100871
sihangjoe@gmail.com, xinwangliu@nudt.edu.cn

## Abstract

Multi-view spectral clustering aims to group data into different categories by optimally exploring complementary information from multiple Laplacian matrices. However, existing methods usually linearly combine a group of pre-specified first-order Laplacian matrices to construct an optimal Laplacian matrix, which may result in limited representation capability and insufficient information exploitation. In this paper, we propose a novel optimal neighborhood multi-view spectral clustering (ONMSC) algorithm to address these issues. Specifically, the proposed algorithm generates an optimal Laplacian matrix by searching the neighborhood of both the linear combination of the first-order and high-order base Laplacian matrices simultaneously. This design enhances the representative capacity of the optimal Laplacian and better utilizes the hidden high-order connection information, leading to improved clustering performance. An efficient algorithm with proved convergence is designed to solve the resultant optimization problem. Extensive experimental results on 9 datasets demonstrate the superiority of our algorithm against state-of-the-art methods, which verifies the effectiveness and advantages of the proposed ONMSC.

## Introduction

Multi-view spectral clustering (MVSC), which makes use of the information of multi-view data to improve clustering accuracy, has become an important research topic in the past decades (De Sa 2005; Yang and Wang 2018). According to the complementary information extraction mechanism, current algorithms can be roughly grouped into three categories. The **first** category adopts a co-training mechanism and forces the clustering results of different views to be consistent with each other (Kumar and Daumé 2011; Huang et al. 2015). The **second** category assumes that the predefined affinity matrices are perturbation of an optimal affinity matrix. Then, by utilizing low-rank or sparse optimization, these algorithms construct an optimal consensus affinity matrix from all views (Nie et al. 2017; Zhan et al. 2019; Tang et al. 2018; Zhou et al. 2019). By assuming that the optimal Laplacian matrix is a linear aggregation of the base Laplacian matrices from each view, the **third** category of methods optimize the combination coefficients of the base Laplacian matrices by minimizing the normalized cut of the combined matrix (Xia et al. 2010). Methods in the third category have received intensive attention during the past few years, and progress continues being made along this line of research (Li et al. 2015; Nie et al. 2016; Zhao, Ding, and Fu 2017; Zong et al. 2018; Huang, Kang, and Xu 2018; Kang et al. 2018; Zhou et al. 2020). The work in (Li et al. 2015) approximates the similarity graphs with bipartite graphs and makes the proposed algorithm applicable to large-scale problems. In (Nie et al. 2016), researchers introduce a parameter-free framework which could serve for both unsupervised and semi-supervised learning circumstances. To propose a more appropriate view-weighting scheme, (Zong et al. 2018) adopts the largest canonical angle to measure the difference between spectral clustering results of different views. Our work also falls into the third category.

Although various improvements have been achieved by existing algorithms, we observe that algorithms from the third category bear the following drawbacks. **First**, these algorithms share a common assumption that the optimal Laplacian matrix lies in the linear space spanned by the base Laplacian matrices. This assumption, on one hand, simplifies the optimization procedure. On the other hand, 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 capacity of the learned matrix (Bach 2009; Cortes, Mohri, and Rostamizadeh 2009; Liu et al. 2017; Li et al. 2018). **Second**, existing algorithms do not sufficiently consider the high-order affinity information, which is important to reveal hidden neighborhood relation 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 to address both issues. Specifically, instead of restricting the optimal Laplacian matrix exactly being a linear combination of base matrices, our algorithm allows the optimal matrix to lie in the neighbor-

*corresponding author

hood of the latter. By this way, our algorithm effectively enlarges the region from which an optimal Laplacian matrix can be chosen for clustering. Then, 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. This contributes to exploit both first-order and high-order connection information. After that, we carefully instantiate an optimization objective and develop an efficient algorithm with proved convergence to solve the resulting optimization problem. The contributions of this paper are summarized as follows:

i) We, for the first time, discover that 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*.

ii) We provide a flexible optimal Laplacian matrix construction mechanism to solve the aforementioned issues. Also, an efficient algorithm with proved convergence is proposed to solve the resulting optimization problem.

iii) Comprehensive experimental study on eight benchmark datasets and the large scale MNIST dataset indicates superior performance of the proposed algorithm.

## Background and Notations

In this section, we first briefly introduce some important notations and then revisit the basic form of linear Laplacian matrix combination-based multi-view spectral clustering. Finally, we introduce the definition of the high-order Laplacian matrix in our paper.

### Notations

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n\times d}$ denote the data matrix, where $n$ is the sample number and $d$ is the feature dimension. Given the dataset $\mathbf{X}$ and a kernel function $\kappa(\cdot,\cdot)$, the adjacent matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ can then be constructed in a k-NN fashion. In particular, in the affinity matrix, $\mathbf{x}_i$ and $\mathbf{x}_j$ is 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:

$$A_{ij} = \begin{cases} \kappa(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are linked} \\ 0, & \text{otherwise} \end{cases}$$

Denoting the $i$-th diagonal element in the degree matrix $\mathbf{D} \in \mathbb{R}^{n\times n}$ as $D_{ii} = \sum_{j=1}^{n} A_{ij}$, the definition of the corresponding first-order normalized graph Laplacian matrix is:

$$\mathbf{L}^{(1)} \triangleq \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}.$$

Let $\mathbf{H} \in \mathbb{R}^{n\times c}$ denote the cluster indicator matrix, where $c$ is the number of classes, the object function of the normalized spectral clustering (Ng, Jordan, and Weiss 2002) is:

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

### Multi-view Spectral Clustering with Linear Laplacian Matrix Combination

For multi-view data, let $v$ be the number of views, $\mathbf{A}_1, \ldots, \mathbf{A}_v \in \mathbb{R}^{n\times n}$ be the affinity matrix of each view and $\mathbf{L}_1^{(1)}, \ldots, \mathbf{L}_v^{(1)} \in \mathbb{R}^{n\times n}$ be the corresponding first-order normalized Laplacian matrices. To exploit the complementary information from different views, (Xia et al. 2010) linearly aggregates the base Laplacian matrices from each view and learns an optimal matrix which can best suit the need of clustering. The formulation of the algorithm is:

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

where $\mu_p$ is the combination weight of the $p$-th view, $r$ is a hyper-parameter to balance the contribution of each view, and $\mathbf{L}_{\boldsymbol{\mu}}^{(1)}$ is the optimal Laplacian matrix for learning. Although good performance has been achieved by the above method, recent literature shows 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 (Liu et al. 2017).

### High-Order Laplacian Matrix

First-order and second-order connections are essential concepts in graph analyzing (Tang et al. 2015). Specifically, in graph embedding, the first-order connection refers to the local pairwise proximity between vertices in a graph. Comparatively, the second-order connection assumes that vertices sharing many connections to other vertices are also similar to each other. Moreover, in recent literature, because of the popularity of graph convolutional neural networks (Defferrard, Bresson, and Vandergheynst 2016), 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 (Tang et al. 2015) is as follow.

**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.*

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:

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

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

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

where $D_{ii}^{(2)} = \sum_{j=1}^{n} A_{ij}^{(2)}$. 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 (Tang et al. 2015), 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.

# Optimal Neighborhood Multi-view Spectral Clustering

## The Proposed Formulation

In this section, to explore better representation capacity and more comprehensively exploit both the first-order and high-order affinity information in data, we propose a novel multi-view spectral clustering algorithm with optimal neighborhood Laplacian matrix. Generally, our algorithm simultaneously seeks an optimal Laplacian matrix $\mathbf{L}^*$ in the neighborhood of both the linear combination of the first-order and high-order base Laplacian matrices. This idea can be intuitively implemented as follows

$$\begin{aligned}\min_{\mathbf{H}^\top\mathbf{H}=\mathbf{I}_c,\boldsymbol{\mu},\mathbf{L}^*} & \operatorname{Tr}\left(\mathbf{H}^\top\mathbf{L}^*\mathbf{H}\right)+\sum_{o=1}^{O}\|\mathbf{L}^*-\mathbf{L}_{\boldsymbol{\mu}}^{(o)}\|_{\mathrm{F}}^2+\alpha\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu},\\ \text{s.t. } & \mathbf{L}_{\boldsymbol{\mu}}^{(o)}=\sum_{p=1}^{v}\mu_p\mathbf{L}_p^{(o)}(o\in[O]),\|\boldsymbol{\mu}\|_1=1,\boldsymbol{\mu}\geq 0,\\ & \mathbf{L}^*\succeq 0,\ L^*_{mn}\leq 0(m\neq n),\end{aligned} \tag{4}$$

where $\mathbf{L}^*$ is the optimal Laplacian matrix for learning, $\mathbf{L}_{\mu}^{(o)}$ is the linear combination of the $o$-order base Laplacian matrices, $O$ is the largest order number and $[O]$ is equivalent with $\{1,\ldots,O\}$. $\alpha$ is an importance balancing coefficient, $\mathbf{M}$ is the correlation measuring matrix which records the centered kernel alignment value (Cortes, Mohri, and Rostamizadeh 2012) between affinity matrices. Specifically, denote the $o$-order affinity matrix of the $p$-th view as $\mathbf{A}_p^{(o)}$, the matrix $\mathbf{M}$ can be defined as:

$$M_{pq}=\sum_{o=1}^{O}\frac{\operatorname{Tr}(\mathbf{A}_p^{(o)}\mathbf{A}_q^{(o)})}{\|\mathbf{A}_p^{(o)}\|_{\mathrm{F}}\|\mathbf{A}_q^{(o)}\|_{\mathrm{F}}},\quad (p,q\in\{1,\ldots,v\}).$$

In the objective function of Eq. (4), 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 (Liu et al. 2016).

In the formulation, 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:

$$\begin{aligned}\min_{\mathbf{H}^\top\mathbf{H}=\mathbf{I}_c,\boldsymbol{\mu},\mathbf{W},\boldsymbol{\Lambda}} & \operatorname{Tr}\left(\mathbf{H}^\top(\mathbf{I}_n-\mathbf{W}\boldsymbol{\Lambda}\mathbf{W}^\top)\mathbf{H}\right)\\ &+\sum_{o=1}^{O}\|(\mathbf{I}_n-\mathbf{W}\boldsymbol{\Lambda}\mathbf{W}^\top)-\mathbf{L}_{\boldsymbol{\mu}}^{(o)}\|_{\mathrm{F}}^2+\alpha\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu},\\ \text{s.t. } & \mathbf{L}_{\boldsymbol{\mu}}^{(o)}=\sum_{p=1}^{v}\mu_p\mathbf{L}_p^{(o)}(o\in[O]),\|\boldsymbol{\mu}\|_1=1,\boldsymbol{\mu}\geq 0,\\ & \mathbf{W}\in\mathbb{R}^{n\times c},\mathbf{W}^\top\mathbf{W}=\mathbf{I}_c,0\leq\Lambda_{ii}\leq 1,\end{aligned} \tag{5}$$

where $\boldsymbol{\Lambda}\in\mathbb{R}^{c\times c}$ is a diagonal matrix. In the new formulation, we use $\mathbf{W}\boldsymbol{\Lambda}\mathbf{W}^\top$ to represent a low rank normalized affinity matrix and $\mathbf{I}_n-\mathbf{W}\boldsymbol{\Lambda}\mathbf{W}^\top$ to represent the corresponding Laplacian matrix. Notably, the constraint $0\leq\Lambda_{ii}\leq 1$ is added to make sure that the optimization process is stable.

## Alternate Optimization Framework

In the following, we design an efficient alternative four-step optimization algorithm to solve the problem in Eq. (5):

**i) Optimizing $\boldsymbol{\Lambda}$**. Given $\mathbf{H}$, $\boldsymbol{\mu}$ and $\mathbf{W}$, the optimization problem in Eq. (5) w.r.t. $\boldsymbol{\Lambda}$ reduces to:

$$\begin{aligned}&\min_{\boldsymbol{\Lambda}}\operatorname{Tr}\left(O\boldsymbol{\Lambda}^2+2\boldsymbol{\Lambda}\mathbf{C}\right),\\ &s.t.\ 0\leq\Lambda_{ii}\leq 1,\ \Lambda_{ij}=0(i\neq j),\end{aligned} \tag{6}$$

where $\mathbf{C}=\mathbf{W}^\top(\sum_{o=1}^{O}\mathbf{L}_{\boldsymbol{\mu}}^{(o)}-\frac{1}{2}\mathbf{H}\mathbf{H}^\top)\mathbf{W}-O\mathbf{I}$. Denoting $\hat{\boldsymbol{\lambda}}=\mathbf{diag}(\boldsymbol{\Lambda})$ as the diagonal vector of matrix $\boldsymbol{\Lambda}$, $\hat{\mathbf{c}}=\mathbf{diag}(\mathbf{C})$ as the diagonal vector of matrix $\mathbf{C}$, problem (6) is equivalent with the following formulation:

$$\min_{\hat{\boldsymbol{\lambda}}}\frac{1}{2}\hat{\boldsymbol{\lambda}}^\top\hat{\boldsymbol{\lambda}}+\frac{1}{O}\hat{\boldsymbol{\lambda}}^\top\hat{\mathbf{c}},\ s.t.\ \mathbf{0}\leq\hat{\boldsymbol{\lambda}}\leq\mathbf{1}. \tag{7}$$

Since the objective function is separable, the optimization problem is equivalent to, for any $i\in[n]$,

$$\min_{\hat{\boldsymbol{\lambda}}_i}\left(\hat{\boldsymbol{\lambda}}_i+\frac{\hat{\mathbf{c}}_i}{O}\right)^2,\ s.t.\ 0\leq\hat{\boldsymbol{\lambda}}_i\leq 1. \tag{8}$$

Denote the optimal solution of this problem as $\hat{\boldsymbol{\lambda}}^*$, it has a closed form

$$\hat{\boldsymbol{\lambda}}_i^*=\mathbf{Proj}_{[0,1]}\left(-\frac{\hat{\mathbf{c}}_i}{O}\right),\quad \text{for any } i\in[n],$$

where, $\mathbf{Proj}_{[0,1]}(\cdot)$ represents to project a real number to $[0,1]$.

**ii) Optimizing $\mathbf{H}$**. Given $\mathbf{W}$, $\boldsymbol{\mu}$, $\boldsymbol{\Lambda}$, the optimization problem in Eq. (5) w.r.t. $\mathbf{H}$ reduces to a standard spectral clustering problem as follow:

$$\min_{\mathbf{H}^\top\mathbf{H}=\mathbf{I}}\operatorname{Tr}\left(\mathbf{H}^\top(\mathbf{I}_n-\mathbf{W}\boldsymbol{\Lambda}\mathbf{W}^\top)\mathbf{H}\right). \tag{9}$$

It is not difficult to know that the optimal solution of $\mathbf{H}$ in Eq. (9) is the $\mathbf{c}$ smallest eigenvectors of $\mathbf{I}-\mathbf{W}\boldsymbol{\Lambda}\mathbf{W}^\top$.

**iii) Optimizing $\boldsymbol{\mu}$**. Given $\mathbf{H}$, $\mathbf{W}$ and $\boldsymbol{\Lambda}$, the optimization problem in Eq. (5) w.r.t. $\boldsymbol{\mu}$ reduces to the following formulation:

$$\begin{aligned}&\min_{\boldsymbol{\mu}}\ \boldsymbol{\mu}^\top(\alpha\mathbf{M}+\hat{\mathbf{M}})\boldsymbol{\mu}-2\boldsymbol{\mu}^\top\boldsymbol{\tau},\\ &s.t.\ \ \|\boldsymbol{\mu}\|_1=1,\ \boldsymbol{\mu}\geq 0,\end{aligned} \tag{10}$$

where $\mathbf{M}$ is the affinity correlation matrix and $\boldsymbol{\tau}\in\mathbb{R}^v$, $\tau_p=\operatorname{Tr}\left((\mathbf{I}-\mathbf{W}\boldsymbol{\Lambda}\mathbf{W}^\top)(\sum_{o=1}^{O}\mathbf{L}_p^{(o)})\right)$. The definition of $\hat{\mathbf{M}}$ is:

$$\hat{\mathbf{M}}\triangleq\left\{\sum_{o=1}^{O}\operatorname{Tr}\left(\mathbf{L}_p^{(o)}\mathbf{L}_q^{(o)}\right)\right\}\in\mathbb{R}^{v\times v},\forall p,q\in[v].$$

Since both $\mathbf{M}$ and $\hat{\mathbf{M}}$ are PSD (Cortes, Mohri, and Rostamizadeh 2012), and the constraints of Eq. (10) is convex,

the corresponding QP problem is convex. Its global optimal solution can be easily solved by the optimization toolbox of MATLAB.

**iv) Optimizing $\mathbf{W}$.** Given $\mathbf{H}$, $\boldsymbol{\mu}$ and $\mathbf{\Lambda}$, through simple deduction, the optimization problem in Eq. (5) w.r.t. $\mathbf{W}$ reduces to the following formulation:

$$\min_{\mathbf{W}^\top \mathbf{W}=\mathbf{I}_c} \mathrm{Tr}\left(\mathbf{\Lambda}\mathbf{W}^\top \mathbf{B}\mathbf{W}\right), \tag{11}$$

where $\mathbf{B} = \sum_{o=1}^{O} \mathbf{L}_{\boldsymbol{\mu}}^{(o)} - \frac{1}{2}\mathbf{H}\mathbf{H}^\top$. Without matrix $\mathbf{\Lambda}$, problem (11) can be simply solved by calculating the eigenvectors corresponding to the smallest $c$ eigenvalues of $\mathbf{B}$.

Generally, the existence of $\mathbf{\Lambda}$ makes the problem hard to solve. Nevertheless, through careful theoretical justification, we find that (11) *does not become more difficult due to* $\mathbf{\Lambda}$. The theoretical analysis of this conclusion is presented as follows.

**Theorem 1.** *When* $\mathbf{\Lambda}$ *is a non-negative diagonal matrix and* $\mathbf{B}$ *is a symmetric matrix, Eq.* (11) *has the same solution as Eq.* (12)*,*

$$\min_{\mathbf{W}^\top \mathbf{W}=\mathbf{I}_c} \mathrm{Tr}\left(\mathbf{W}^\top \mathbf{B}\mathbf{W}\right). \tag{12}$$

*Proof.* Denote $\mathbf{W}_j$ as the $j$-th column of matrix $\mathbf{W}$.

$$\mathrm{Tr}\left(\mathbf{\Lambda}\mathbf{W}^\top \mathbf{B}\mathbf{W}\right) = \sum\nolimits_{j=1}^{c} \Lambda_{jj}\mathbf{W}_j^\top \mathbf{B}\mathbf{W}_j. \tag{13}$$

As a consequence, the optimization problem in Eq. (11) can be transferred into the following form:

$$\min_{\mathbf{W}_1,\ldots,\mathbf{W}_c} \sum\nolimits_{j=1}^{c} \Lambda_{jj}\mathbf{W}_j^\top \mathbf{B}\mathbf{W}_j,$$
$$s.t.\, \mathbf{W}_i \in \mathbb{R}^n,\, \mathbf{W}_i^\top \mathbf{W}_i = 1, \mathbf{W}_i^\top \mathbf{W}_j = 0, (\forall i \neq j \in [c]). \tag{14}$$

Since there is no upper and lower bound on $\mathbf{W}$ in the formulation, according to the method of Lagrange multipliers, the optimal points of (14) should be the critical points of the following formulation:

$$\mathcal{L}(\mathbf{W}_1,\ldots,\mathbf{W}_c) = \sum_{j=1}^{c} \Lambda_{jj}\mathbf{W}_j^\top \mathbf{B}\mathbf{W}_j - \sum_{j=1}^{c} \rho_{jj}(\mathbf{W}_j^\top \mathbf{W}_j - 1) - \sum_{i=1}^{c}\sum_{j=1,j\neq i}^{c} \rho_{ij}(\mathbf{W}_i^\top \mathbf{W}_j),$$

where $\rho_{ij}$s are the Lagrange multipliers. Take the derivative of $\mathcal{L}(\mathbf{W}_1,\ldots,\mathbf{W}_c)$ with respect to $\mathbf{W}_i$ and set it to 0. With the help of Karush–Kuhn–Tucker (KKT) conditions (Bertsekas 1997), through simple deduction, for $\Lambda_{ii} > 0$, we have

$$\mathbf{B}\mathbf{W}_i = \frac{\rho_{ii}}{\Lambda_{ii}}\mathbf{W}_i$$

As a consequence, the optimal points of Eq. (14) are the eigenvectors of matrix $\mathbf{B}$. Denote $\beta_i$ and $\mathbf{u}_i$ as the $i$-th eigenvalue and the corresponding eigenvector of matrix $\mathbf{B}$. Since $\mathbf{B}$ is a symmetric matrix, the $\mathbf{u}_i$s are orthogonal with each other and $\mathbf{B}$ can be rewritten in the following form:

$$\mathbf{B} = \sum\nolimits_{i=1}^{n} \beta_i \mathbf{u}_i \mathbf{u}_i^\top. \tag{15}$$

Substitute Eq. (15) into the objective of Eq. (14) and set $\mathbf{W}_i$s as the eigenvectors of $\mathbf{B}$, we have

$$\mathrm{Tr}\left(\mathbf{\Lambda}\mathbf{W}^\top \mathbf{B}\mathbf{W}\right) = \sum\nolimits_{i=1}^{c} \Lambda_{ii}\hat{\beta}_i,$$

where $\{\hat{\beta}_i\}_{i=1}^{c}$ are $c$ arbitrary eigenvalues of $\mathbf{B}$. Obviously, the minimum value can be achieved when the smallest $c$ eigenvalues are selected. Correspondingly, the optimal solution can be achieved by calculating the $c$ eigenvectors with the smallest eigenvalues, which is the same as the solution of Eq. (12). □

**Remark 1.** *Theorem 1 implies that* $\mathbf{\Lambda}$ *in Eq.* (11) *does not make the optimization of* $\mathbf{W}$ *more difficult.*

In sum, our algorithm for solving Eq. (5) is outlined in Algorithm 1, where $obj_{(t)}$ denotes the objective value at the $t$-th iteration.

**Algorithm 1** Optimal Neighborhood Multi-View Spectral Clustering

**Input:** Data from $v$ views $\{\mathbf{X}_{(1)},\ldots,\mathbf{X}_{(v)}\}$, number of clusters $c$, parameter $\alpha$ and the neighbor number $N$.
**Output:** The learned optimal neighborhood Laplacian matrix $\mathbf{L}^*$

```
Construct first-order and high-order affinity matrices and the corresponding Laplacian matrices. Initialize H_(0) as 0_{n×c}, μ_(0) as 1_v/v, Λ as I_c. Initialize t as 1.
repeat
    Calculate L^(o)_{μ_(t)} = Σ_{p=1}^{v} μ_{p_(t-1)} L_p^(o);
    Calculate W_(t) by optimizing Eq. (11);
    Calculate Λ_(t) by optimizing Eq. (7);
    Calculate H_(t) by optimizing Eq. (9);
    Calculate μ_(t) by optimizing Eq. (10);
    t = t + 1.
until |Obj_(t) − Obj_(t−1)| < 10^{−4} × |Obj_(t)|.
```

## Algorithmic Discussion

**Convergence.** In each of the optimization iteration of our proposed algorithm, two convex quadratic programming problems and two eigenvalue decomposition problems are being solved. As a consequence, the objective of Algorithm 1 is guaranteed to be monotonically decreased when optimizing one variable with others fixed at each iteration. At the same time, the objective is lower-bounded by zero. As a result, our algorithm is guaranteed to converge to a local optimum of problem Eq. (5).

**Computational Complexity.** Our optimization algorithm is composed of four sub-problems. The overall procedure is reported in Algorithm 1. Specifically, as analyzed in the previous sections, the optimization of $\mathbf{W}$ and $\mathbf{H}$ requires solving a SVD decomposition problem on a $n^2$ matrix, which leads to $\mathcal{O}\left(n^3\right)$ complexity. Additionally, updating $\boldsymbol{\mu}$ requires solving a standard Quadratic Programming with Linear Constraints (QPLC) whose complexity is $\mathcal{O}\left(\epsilon^{-1}v\right)$. Here, $v$ is the number of views and $\epsilon$ is the precision of the result.

The update of $\mathbf{\Lambda}$ has a closed form solution, its complexity is $\mathcal{O}(n)$. Overall, the total complexity of our algorithm is $\mathcal{O}\left(T(n^3+\epsilon^{-1}v+n)\right)$, where $T$ is the number of iterations. Under the condition $\epsilon^{-1} \ll n^2$, the total complexity is basically $\mathcal{O}\left(Tn^3\right)$.

Note that the existing popular linear combination-based multi-view spectral clustering algorithms like AMGL (Nie et al. 2016) also require to conduct SVD decomposition on a $n^2$ matrix iteratively. Comparing with it, Algorithm 1 does not lead to much extra computational cost since QPLC can be solved efficiently. This is verified by empirical studies, please refer to Table 4.

# Experiments

## 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 sub-cellular localization, and image recognition. The detailed information of these datasets is listed in Table 1. From this table, we observe that the number of samples, views and the number of categories of these datasets range from 554 to 60,000, 2 to 69, and 3 to 102, respectively. For these datasets, all affinity matrices are pre-computed with carefully designed similarity function and are publicly available from websites[1][2][3].

Table 1: Benchmark datasets

| Datasets | # Samples | # Views | # Clusters |
|---|---|---|---|
| BBCSport | 554 | 2 | 5 |
| ProteinFold | 694 | 12 | 27 |
| Flower17 | 1360 | 7 | 17 |
| Caltech101mit | 1530 | 25 | 102 |
| UCI-Digit | 2000 | 3 | 10 |
| Mfeat | 2000 | 12 | 10 |
| Nonpl | 2732 | 69 | 3 |
| Flower102 | 8189 | 4 | 102 |
| MNIST | 60000 | 3 | 10 |

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. Specially, to all the compared spectral clustering algorithms, the optimal neighbor numbers are carefully searched in the range of $[0.1s, 0.2s, \ldots, s]$, where $s = n/c$ is the average sample number in each category. As to our proposed method, the regularization parameter is chosen in the range of $[2^0, 2^3, \ldots, 2^{15}]$. 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 a 3.6GHz Intel Core i7 CPU and 64GB RAM, MATLAB 2017a (64bit).

[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: 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 3: Performance comparison when different orders of affinity information is preserved.

| 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 |

## 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.

**Effectiveness of the designed algorithm.** Among the compared algorithms, the baseline method (BL) indicates a classic linear Laplacian matrix combination with matrix-induced regularization (Liu et al. 2016). For high-level connection information extraction, the order number of the Laplacian matrix is fixed as 2. As we can see from Table 2, 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 preserving affinity information of different orders in Eq. (5). In this part, the second-, third-, fourth- and fifth-order algorithms are compared. As one can see in Table 3, the second- and third-order algorithms provide comparably good performance. However, as the orders of the Laplacian matrices keep getting higher, the range of neighborhood also gets larger, and the discriminative capacity of the corresponding algorithms start to decrease a little bit. Consequently, for the sake of the clustering performance and the computational efficiency, the order number of our proposed algorithm is fixed as 2 in all the following experiments.

Table 4: ACC, NMI, purity, and time consumption (in seconds) 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 (Du et al. 2015) | MKKM-MR (Liu et al. 2016) | ONKC (Liu et al. 2017) | Co-reg (Kumar and Daumé 2011) | AMGL (Nie et al. 2016) | RMSC (Xia et al. 2014) | MLAN (Nie, Cai, and Li 2017) | Proposed |
|---|---|---|---|---|---|---|---|---|---|---|
| ACC (%) | | | | | | | | | | |
| BBCSport | 66.18 | 76.65 | 63.79 | 66.18 | 68.20 | 85.66 | 86.39 | 86.03 | 70.58 | **95.77** |
| ProteinFold | 30.69 | 34.58 | 30.98 | 36.46 | 37.90 | 34.87 | 36.88 | 33.00 | 28.38 | **41.21** |
| Flower17 | 51.02 | 42.05 | 48.38 | 60.00 | 60.88 | 52.72 | 56.32 | 53.90 | 53.38 | **66.39** |
| Caltech101mit | 35.55 | 33.13 | 29.67 | 35.82 | 37.32 | 33.33 | 37.64 | 31.50 | 26.33 | **40.84** |
| UCI-Digit | 88.75 | 75.40 | 40.45 | 90.40 | 91.05 | 84.80 | 92.85 | 90.40 | 97.15 | **97.60** |
| MFea | 95.20 | 86.00 | 65.30 | 83.20 | 97.05 | 84.30 | 84.35 | 84.15 | 96.55 | **98.10** |
| Nonpl | 49.37 | 57.50 | 62.77 | 56.59 | 59.57 | 55.27 | 56.91 | 60.65 | 44.98 | **65.84** |
| Flower102 | 27.29 | 33.12 | 28.17 | 39.91 | 41.56 | 37.26 | 33.34 | 32.97 | 24.19 | **43.31** |
| NMI (%) | | | | | | | | | | |
| BBCSport | 53.92 | 59.38 | 39.62 | 53.93 | 54.64 | 71.27 | 73.70 | 73.89 | 65.34 | **87.19** |
| ProteinFold | 40.95 | 42.33 | 38.78 | 45.32 | 46.93 | 43.34 | 44.18 | 43.91 | 27.86 | **49.33** |
| Flower17 | 50.18 | 45.14 | 50.73 | 57.11 | 58.58 | 52.13 | 56.97 | 53.89 | 55.38 | **65.54** |
| Caltech101mit | 59.90 | 59.06 | 55.86 | 60.38 | 61.41 | 58.20 | 61.79 | 58.40 | 43.25 | **63.77** |
| UCI-Digit | 80.59 | 68.38 | 46.87 | 83.22 | 83.96 | 73.51 | 86.65 | 81.80 | 93.40 | **94.39** |
| MFea | 89.83 | 75.78 | 62.67 | 78.12 | 93.07 | 80.99 | 81.57 | 81.69 | 92.89 | **95.51** |
| Nonpl | 16.55 | 15.26 | 17.34 | 15.51 | 24.04 | 12.55 | 15.19 | 20.35 | 6.14 | **25.35** |
| Flower102 | 46.32 | 48.99 | 48.17 | 57.27 | 59.13 | 54.18 | 51.63 | 53.36 | 34.94 | **60.12** |
| Purity (%) | | | | | | | | | | |
| BBCSport | 77.20 | 79.59 | 67.83 | 77.21 | 77.76 | 85.66 | 86.39 | 86.03 | 74.44 | **95.77** |
| ProteinFold | 37.17 | 41.21 | 36.60 | 42.65 | 45.24 | 40.78 | 42.07 | 42.36 | 31.84 | **47.98** |
| Flower17 | 51.98 | 44.63 | 51.54 | 61.03 | 61.69 | 56.47 | 58.16 | 53.24 | 55.07 | **68.52** |
| Caltech101mit | 37.12 | 35.09 | 31.70 | 37.65 | 39.08 | 35.75 | 39.28 | 33.27 | 28.56 | **43.39** |
| UCI-Digit | 88.75 | 76.10 | 44.20 | 90.40 | 91.05 | 77.75 | 92.85 | 82.90 | 97.15 | **97.60** |
| MFea | 95.20 | 86.00 | 66.25 | 83.20 | 97.05 | 84.30 | 84.35 | 84.10 | 96.55 | **98.10** |
| Nonpl | 72.18 | 71.12 | 71.71 | 63.91 | 75.34 | 66.07 | 69.94 | 70.50 | 60.35 | **76.13** |
| Flower102 | 32.27 | 38.78 | 27.61 | 33.86 | 47.64 | 44.08 | 39.71 | 40.24 | 31.15 | **50.78** |
| Computational time ($s$) | | | | | | | | | | |
| Average | 2.67 | 4.79 | 61.51 | 3.44 | 38.04 | 15.27 | 8.95 | 7.76 | 5.86 | 7.57 |

## 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**) (Kumar and Daumé 2011) is a representative of the co-training methods. (4) Auto-weighted Multiple Graph Learning (**AMGL**) (Nie et al. 2016) is a linear combination-based method. (5) Multi-view Learning with Adaptive Neighbors (**MLAN**) (Nie, Cai, and Li 2017), and (6) Robust Multi-view Spectral Clustering **RMSC** (Xia et al. 2014) 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** (Liu et al. 2017), (8) **MKKM-MR** (Liu et al. 2016), and (9) **RMKKM** (Du et al. 2015), are also included for more comprehensive comparison.

The ACC, NMI, purity, and the computational time of the above-mentioned algorithms are reported in Table 4. As can be seen, in all of the eight datasets, the proposed ONMSC shows superior performance gains over the state-of-the-art algorithms w.r.t. all the three metrics. Specifically, comparing with the second best baseline algorithms, take the ACC as an example, the proposed algorithm achieves an improvement of $9.38\%$, $3.31\%$, $5.51\%$, $3.20\%$, $0.45\%$, $1.05\%$, $3.07\%$ and $1.75\%$ on BBC Sport, Protein Fold, Flower17, Caltech-MIT, Digit, Multiple Feature, Non-plant and Flower102 datasets, respectively. Also, the proposed algorithm significantly outperforms existing linear combination based algorithms, including RMKKM, MKKM-RM, and AMGL with comparable computational consumption. This again validates the effectiveness of optimal neighborhood spectral clustering and the high-level information.

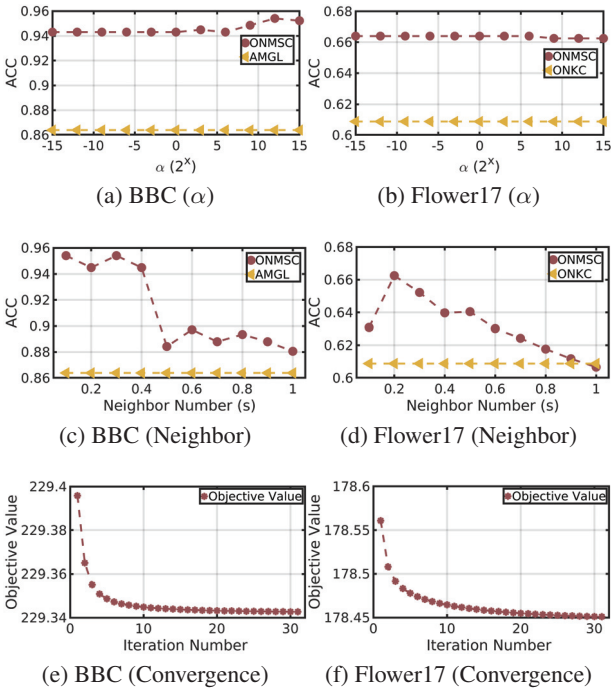
## Parameter Sensitivity and Convergence

**Parameter Sensitivity**. The proposed algorithm introduces two hyper-parameters, i.e., the diversity balancing coefficient $\alpha$ and the neighbor number $N$ for affinity matrix construction. To test the sensitivity of the proposed algorithm against these two parameters, we fix one parameter and tune the other in a large range. The comparison between the proposed algorithm with the second best baseline algorithm on two datasets (BBC Sport and Flower17) are illustrated in Fig. 1 (a-d). From these figures, we observe: i) both $\alpha$ and $N$ are effective in improving the algorithm performance; ii) the algorithm is stable against $\alpha$; iii) the proposed algorithm is relatively sensitive to the neighbor numbers. However, setting a small neighbor around $0.2s$ helps achieve preferable performance; iv) thanks to the robustness of the design, the performance of our algorithm significantly surpasses the second best algorithm in most of the times.

**Algorithm Convergence**. Two examples of the objective values of our algorithm at each iteration are shown in Fig. 1 (e-f). As observed from these figures, the objective value is monotonically decreased and the algorithm quickly converges in less than thirty iterations.

## Scale the Algorithm to Large Datasets

Being able to work appropriately on large scale datasets is an important criterion to the practicality of a MVSC algorithm. To show the effectiveness of our proposed method,




(a) BBC ($\alpha$) (b) Flower17 ($\alpha$)

(c) BBC (Neighbor) (d) Flower17 (Neighbor)

(e) BBC (Convergence) (f) Flower17 (Convergence)

Figure 1: Illustration of parameter sensitivity and algorithm convergence. The algorithm sensitivity against the variation of two hyper-parameters are shown in the upper four figures. The bottom two figures show the convergence curve of the algorithm. Results on BBCSport and Flower17 datasets are reported.

Table 5: Experimental results on the MNIST dataset with 60,000 samples.

| Methods | A-MVSC | RMSC | AMGL | Proposed |
|---|---|---|---|---|
| ACC (%) | 63.75 | 70.36 | 73.36 | **80.79** |
| NMI (%) | 55.33 | 64.10 | 63.09 | **71.24** |
| Purity (%) | 64.79 | 72.24 | 73.46 | **80.80** |
| Time (s) | 34.21 | 301.5 | 196.8 | 227.6 |

we further conduct an experiment on the MNIST dataset[4]. To construct the dataset, we first adopt three deep neural networks, i.e., VGG19 (Simonyan and Zisserman 2014), DenseNet121 (Huang et al. 2017), and ResNet101 (He et al. 2016), which are pre-trained on the ImageNet[5] dataset as feature extractors in three difference views.

Since our proposed algorithm includes two SVD operations for the calculation of $\mathbf{W}$ and $\mathbf{H}$, the $\mathcal{O}\left(n^3\right)$ computational complexity and $\mathcal{O}\left(n^2\right)$ memory consumption hinders the algorithm from scaling to large datasets. To solve the problem, we adopt the Nyström algorithm and a sampling strategy (Li et al. 2011). Specifically, in the experiment, we first select $s$ landmark samples with a k-means algorithm and construct affinity matrices with only the sub-sample set. Then, with these affinity matrices, we conduct a multi-view spectral clustering algorithm and learn the combination weight for each view. Finally, with these weights we integrate the first-order or high-order affinity matrices on the whole dataset and conduct an orthogonal Nyström algorithm (Li et al. 2011) to acquire the final label for the dataset. In this setting, the total computational consumption of the proposed algorithm becomes $\mathcal{O}\left(ns^2 + Ts^3\right)$ and the largest memory consumption becomes $\mathcal{O}\left(ns\right)$, which is much more affordable for the large scale clustering tasks.

Four algorithms, i.e., average multi-view spectral clustering, RMSC (Xia et al. 2014), AMGL (Nie et al. 2016), and the proposed algorithm, are compared in the experiment. The testing procedure is similar with that in the previous sub-section and the same Nyström algorithm is applied to all the compared algorithms. The number of landmark samples $(s)$ is fixed as 3000 for all comparing algorithms. As shown in Table 5, our proposed algorithm surpasses the compared algorithms with comparable computational consumption. Specifically, it outperforms the second best algorithm for about 7% in all three criteria.

[4]http://yann.lecun.com/exdb/mnist/

[5]http://www.image-net.org/

## Conclusion

This paper proposes an optimal neighborhood multi-view spectral clustering (ONMSC) algorithm, which 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. In this way, the representative capacity of the learned Laplacian matrix is effectively improved, and more comprehensive sample affinity information is extracted. A four-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 extend our algorithm to a more general framework and use it as a platform to revisit existing multi-view spectral clustering algorithms.

## Acknowledgement

This work was supported by the National Key R&D Program of China 2018YFB1003203 and the National Science Foundation of China under Grant No. 61672528, 61976064, 61773392, 61872377, and 61922088.

## References

Bach, F. R. 2009. Exploring large feature spaces with hierarchical multiple kernel learning. In *NIPS*, 105–112.

Bertsekas, D. P. 1997. Nonlinear programming. *Journal of the Operational Research Society* 48(3):334–334.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2009. Learning non-linear combinations of kernels. In *NIPS*, 396–404.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2012. Algorithms for learning kernels based on centered alignment. *JMLR* 13(Mar):795–828.

De Sa, V. R. 2005. Spectral clustering with two views. In *ICML workshop on learning with multiple views*, 20–27.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 3844–3852.

Du, L.; Zhou, P.; Shi, L.; Wang, H.; Fan, M.; Wang, W.; and Shen, Y.-D. 2015. Robust multiple kernel k-means using l21-norm. In *IJCAI*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Huang, S.; Wang, H.; Li, D.; Yang, Y.; and Li, T. 2015. Spectral co-clustering ensemble. *Knowledge-Based Systems* 84:46–55.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *CVPR*, 4700–4708.

Huang, S.; Kang, Z.; and Xu, Z. 2018. Self-weighted multi-view clustering with soft capped norm. *Knowledge-Based Systems* 158:1–8.

Kang, Z.; Lu, X.; Yi, J.; and Xu, Z. 2018. Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification. *arXiv preprint arXiv:1806.07697*.

Kumar, A., and Daumé, H. 2011. A co-training approach for multi-view spectral clustering. In *ICML*, 393–400.

Li, M.; Lian, X.-C.; Kwok, J. T.; and Lu, B.-L. 2011. Time and space efficient spectral clustering via column sampling. In *CVPR 2011*, 2297–2304. IEEE.

Li, Y.; Nie, F.; Huang, H.; and Huang, J. 2015. Large-scale multi-view spectral clustering via bipartite graph. In *AAAI*.

Li, J.-H.; Wang, C.-D.; Li, P.-Z.; and Lai, J.-H. 2018. Discriminative metric learning for multi-view graph partitioning. *Pattern Recognition* 75:199–213.

Liu, X.; Dou, Y.; Yin, J.; Wang, L.; and Zhu, E. 2016. Multiple kernel k-means clustering with matrix-induced regularization. In *AAAI*.

Liu, X.; Zhou, S.; Wang, Y.; Li, M.; Dou, Y.; Zhu, E.; and Yin, J. 2017. Optimal neighborhood kernel clustering with multiple kernels. In *AAAI*.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *NIPS*, 849–856.

Nie, F.; Li, J.; Li, X.; et al. 2016. Parameter-free auto-weighted multiple graph learning: A framework for multi-view clustering and semi-supervised classification. In *IJCAI*, 1881–1887.

Nie, F.; Li, J.; Li, X.; et al. 2017. Self-weighted multiview clustering with multiple graphs. In *IJCAI*, 2564–2570.

Nie, F.; Cai, G.; and Li, X. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *WWW*, 1067–1077.

Tang, C.; Zhu, X.; Liu, X.; Li, M.; Wang, P.; Zhang, C.; and Wang, L. 2018. Learning joint affinity graph for multi-view subspace clustering. *IEEE TMM*.

Xia, T.; Tao, D.; Mei, T.; and Zhang, Y. 2010. Multiview spectral embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 40(6):1438–1446.

Xia, R.; Pan, Y.; Du, L.; and Yin, J. 2014. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*.

Yang, Y., and Wang, H. 2018. Multi-view clustering: a survey. *Big Data Mining and Analytics* 1(2):83–107.

Zhan, K.; Nie, F.; Wang, J.; and Yang, Y. 2019. Multiview consensus graph clustering. *IEEE TIP* 28(3):1261–1270.

Zhao, H.; Ding, Z.; and Fu, Y. 2017. Multi-view clustering via deep matrix factorization. In *AAAI*.

Zhou, P.; Shen, Y.-D.; Du, L.; Ye, F.; and Li, X. 2019. Incremental multi-view spectral clustering. *Knowledge-Based Systems* 174:73–86.

Zhou, S.; Zhu, E.; Liu, X.; Zheng, T.; Liu, Q.; Xia, J.; and Yin, J. 2020. Subspace segmentation-based robust multiple kernel clustering. *Information Fusion* 53:145–154.

Zong, L.; Zhang, X.; Liu, X.; and Yu, H. 2018. Weighted multi-view spectral clustering based on spectral perturbation. In *AAAI*.