Consistent Affinity Representation Learning with Dual Low-rank Constraints for Multi-view Subspace Clustering

Lele Fu[a], Jieling Li[b], Chuan Chen[c,*]

[a]*School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou, China*
[b]*Department of Information and Communication Engineering, Xiamen University, Xiamen, China*
[c]*School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China*

Abstract

Multi-view clustering aims to achieve better accuracy of data clustering by leveraging complementary information embedded in multi-view data. How to learn a consistent clustering-friendly affinity representation matrix is a crucial issue. In this paper, we propose a consistent affinity representation learning method with dual low-rank constraints for multi-view subspace clustering. To be specific, for capturing the high-order correlations and global consensus among views, we collect the subspace representations of all views into a 3-order tensor, which is imposed with the tensor singular value decomposition (t-SVD) based tensor nuclear norm for achieving the low-rank recovery. Thus, we learn a consistent affinity matrix by fusing multiple subspace representations on the Grassmann manifold rather than handling them in the Euclidean space. In order to enhance the global cluster structure in the uniform subspace, the low-rank constraint is imposed on the consistent affinity matrix. Furthermore, the local geometric structure of the uniform subspace is encoded via graph regularization. The established model can be solved via the alternating direction method of multipliers algorithm (ADMM). Ultimately, the proposed method is experimentally validated to be superior to other state-of-the-art clustering algorithms.

Keywords: Multi-view clustering, consistent representation, subspace learning, Grassmann manifold.

*Corresponding author
Email addresses: fulle@mail2.sysu.edu.cn (Lele Fu), chenchuan@mail.sysu.edu.cn (Chuan Chen)

1. Introduction

Nowadays, heterogeneous features of data are increasingly frequent and readily available. Compared to single-view data with monotonous and inadequate information, multi-view data possess richer and more complete semantic profiles. In face of this particular data format, traditional single-view representation learning generally deal with them via feature concatenation. Nevertheless, this approach cannot utilize the potential correlations between multiple views and leads to inferior performance. To this end, multi-view representation learning [1, 2, 3] has gained widespread interests.

Although different views of data are available with different statistical information, a unified data structure information is shared among them. In a nutshell, multi-view representation learning aims to incorporate the differences and complementarities from multiple views to learn a consistent representation, which is used to perform the downstream tasks such as multi-view unsupervised clustering [4, 5, 6] or multi-view semi-supervised classification [7, 8, 9]. However, in real scenarios, data are often presented without any label information. Therefore, it is more interesting and challenging to study how to achieve multi-view representation learning under unsupervised paradigm. In this paper, we concentrate on learning a consistent affinity representation matrix from multi-view data lacking the help of label information, then which is used to execute the clustering task.

Differing from some distance metrics such as Gaussian kernel function and Cosine similarity function that compute the similarity between samples via predefined calculations, self-representation based subspace learning explores the latent relationships between data points by dictionary mapping using the data matrix itself as the basis vector. Due to its encouraging performance, extensive multi-view self-representation based subspace clustering approaches have been proposed. For instance, Cao et al. [10] adopted the Hilbert Schmidt Independence Criterion (HSIC) to enhance the diversity of different views for exploring the complementary information. Zhang et al. [11] aimed at finding the shared latent representation and exploring the complementary information among multiple views. Niu et al. [12] leveraged the low-rank matrix factorization to handle the incomplete multi-view subspace clustering. Zhang et al. [13] employed the weighted Schatten p-norm to learn the robust multiple subspace representations. Wang et al. [14] proposed a deep multi-view subspace clustering method with preserving the global and local structures. These works are all based on matrix level, which are lim-

ited in mining the high-order correlations between views. Towards this end, some tensor-oriented methods are developed. The works [15, 16] extended the low-rank representation (LRR) [17] to tensor form and recovered the low-rank property of subspace tensor representation. Cheng et al. [18] learned the low-dimensional latent tensor space from the heterogeneous and high-dimensional multi-view data. Chen et al. [19] proposed a generalized nonconvex low-rank tensor norm to distinguish the meanings of diverse singular values. Wang et al. [20] learned the task-driving affinity matrix in the tensor space for accurate multi-view clustering.

When dealing with multiple learned representations, all of the above methods face a common problem, i.e., how to align these representations. Most methods learn a latent space via minimizing their distances in the Euclidean space or directly adopt the averaging measure. Nevertheless, these linear approaches perturb the local structure of data and are vulnerable to noises, which may lead to undesirable clustering results. Fortunately, some previous works [21, 22, 23] have demonstrated that the fusion of varying subspaces on the Grassmann manifold is more beneficial to protect the vertex connectivity, thus obtaining the unified informative representation matrix. In this paper, we propose a consistent affinity representation learning method on the Grassmann manifold for multi-view subspace clustering, which uses dual low-rank constraints to achieve double enhancement. Specifically, we stack all self-representation matrices as a 3-order tensor, which is imposed with the tensor singular value decomposition (t-SVD) based tensor nuclear norm (TNN). Thus, the consistency across views is enhanced and the high-order correlations between views are captured. For aligning the self-representation matrices in the low-rank tensor space, we fuse them on the Grassmann manifold to obtain a consistent affinity representation matrix. Then, we apply the low-rank constraint on the consistent affinity matrix to further enhance its cluster structure. Moreover, the graph regularization is leveraged to encode the local geometric structure inside the uniformity subspace. For the convenience of the following description, the proposed method is referred to as CARLDLC. Fig. 1 figuratively illustrates the framework of the proposed CARLDLC. Generally, the major contributions of this paper are as follows:

- To explore the high-order correlations and reinforce the consistency between different views, we stack multiple self-representation matrices as a 3-order tensor, which is constrained with the t-SVD based TNN to exploit the low-rank property.
- For learning a well-structured consistent affinity representation matrix, we align diverse subspace representations in the optimized low-rank tensor s-

pace on the Grassmann manifold. Furthermore, the low-rank constraint is imposed on the learned consistent affinity matrix to enhance its cluster structure. Meanwhile, we utilize the graph regularization to encode the local structure of the uniformity subspace.

- An optimization algorithm for the proposed CARLDLC based on the alternating direction method of multipliers algorithm (ADMM) is developed. We also conduct comprehensive experiments to verify the superiority of CARLDLC compared to other state-of-the-art clustering algorithms.

In summary, we conclude the motivations for proposing CARLDLC herein. 1) Compared to matrix-based multi-view clustering methods, we want to explore the high-order correlations between views from a global perspective, namely we combine multiple subspace representations into a 3-order tensor with low-rank constraint, thus more effectively excavating the complementary information among them. 2) Compared to the multi-view clustering methods using linear fusion scheme in the Euclidean space, we want to more robustly integrate various subspace representations. In light of the effectiveness of robust fusion on the Grassmann manifold, we merge multiple subspace representations in the low-rank tensor space into a uniformity affinity representation on the Grassmann manifold.

We organize the rest of this paper as follows. Section 2 briefly reviews several popular multi-view clustering models and the Grassmann manifold. In Section 3, some preliminaries are introduced. Section 4 elaborates the proposed CARLDLC. Experimental results are presented in Section 5. In Section 6, we summarize the paper.

2. Related Works

2.1. Multi-view Clustering

In recent years, a great deal of multi-view clustering methods have been proposed. The works [24, 25] were designed to learn a consistent spectral embedding matrix, on which the K-means algorithm was adopted to obtain the clustering results. However, the works [26, 27] regarded the existence of capacity difference in diverse views, they learned a view-specific spectral embedding from each view and instead learned a consensus non-negative embedding, then integrated them with certain learning manner. Graph learning has received widespread attentions in multi-view clustering due to its powerful ability to depict relationships between

samples. Some works [28, 29, 30] desired to learn a unified graph matrix by integrating multiple views, which was used as the input of spectral clustering. Considering that graph-based methods often suffer from high computational complexity, the works [31, 32] constructed an anchor graph for each view using a small number of samples, then fused them to match the global consistent graph matrix. For the same purpose, bipartite graph technology was introduced to approximate the similarity graphs by [33, 34, 35], which can improve the efficiency of models. Subspace clustering is an effective technology for discovering clusters in different subspaces. To explore a discriminate subspace where the data are easily distinguishable, researchers have attempted to develop a large variety of proposals. For instance, the works [10, 11, 36] aimed to learn a latent shared subspace representation via different regularization terms. Robust integration of various subspace representations is an important concern for multi-view subspace clustering. Recently, some fusion methods based on Grassmann manifold have been proposed. For addressing high-dimensional multi-view data, Guo et al. [37] proposed an improved low-rank model using the matrix tri-factorization on Grassmann manifold. Wang et al. [38] explored the self-representation tensor from multi-view data tensor, then whose frontal slices were merged on the Grassmann manifold into a consensus affinity matrix. Rong et al. [23] learned a consistent clustering-friendly affinity matrix via aligning all subspace representations on the Grassmann manifold. Jing et al. [39] fused multiple hypergraph Laplacian matrices on the Grassmann manifold to obtain a robust affinity graph. To handle the non-linear structure of multi-view data, the kernel trick was leveraged to tackle this issue in [13, 40]. Deep learning has been demonstrated to deliver excellent capability of representation learning. Zheng et al. [41] proposed a deep multi-view subspace learning network with adversarial setting and co-attention mechanism. Lu et al. [42] merged different views by considering the dynamic contribution of each view captured by attention mechanism.

2.2. Grassmann manifold

According to [43], the set consisting of all r-dimensional linear subspaces embedded in an n-dimensional Euclidean space is termed as Grassmann manifold. An orthonormal matrix $\mathbf{M} \in \mathbb{R}^{n\times r}$ ($\mathbf{M}^T\mathbf{M} = \mathbf{I}$) is used to represent an item on the Grassmann manifold. Moreover, we utilize $span(\mathbf{M})$ to denote the r-dimensional subspace that is spanned by the column vectors of $\mathbf{M}$. Suppose there are two r-dimensional subspaces $span(\mathbf{M}^{(1)})$ and $span(\mathbf{M}^{(2)})$, their geodesic distance on the Grassmann manifold can be defined based on a group of principal angles $\{\theta_i\}_{i=1}^r$. The value range of principal angle θ_i is from 0^o to

360° (or greater values), for the sake of brevity, the smallest angle is adopted. The distance between two subspaces is formulated as

$$\begin{aligned} \mathrm{D}_{proj}^{2}\left(\mathbf{M}^{(1)}, \mathbf{M}^{(2)}\right) &= \sum_{i=1}^{r} \sin^{2} \boldsymbol{\theta}_{i} \\ &= r - \sum_{i=1}^{r} \cos^{2} \boldsymbol{\theta}_{i} \\ &= r - Tr\left(\mathbf{M}^{(1)}\mathbf{M}^{(1)^{T}}\mathbf{M}^{(2)}\mathbf{M}^{(2)^{T}}\right), \end{aligned} \tag{1}$$

Specifically, [43] leverages the one-to-one mapping manner to match the $span(\cdot)$ operation, i.e., $span(\mathbf{M}) = \mathbf{M}\mathbf{M}^{T}$, then $\sum_{i=1}^{r} \cos^{2} \boldsymbol{\theta}_{i}$ can be viewed as the calculation of cosine similarity between $span(\mathbf{M}^{(1)})$ and $span(\mathbf{M}^{(2)})$. Moreover, it can be seen that the distance between subspace $\mathbf{M}^{(1)}$ and subspace $\mathbf{M}^{(2)}$ reaches a minimum when they are same.

When the above formula is extended to the fusion of multiple subspaces $\{\mathbf{M}^{(v)}\}_{v=1}^{m}$ for learning the integrative subspace $\mathbf{M}$, the following Eq. (2) can be obtained:

$$\begin{aligned} D_{proj}^{2}\left(\{\mathbf{M}^{(v)}\}_{v=1}^{m}, \mathbf{M}\right) &= \sum_{v=1}^{m} D_{proj}^{2}\left(\mathbf{M}^{(v)}, \mathbf{M}\right) \\ &= \sum_{v=1}^{m} \left(r - Tr(\mathbf{M}^{(v)}\mathbf{M}^{(v)^{T}}\mathbf{M}\mathbf{M}^{T})\right) \\ &= rm - \sum_{v=1}^{m} Tr\left(\mathbf{M}^{(v)}\mathbf{M}^{(v)^{T}}\mathbf{M}\mathbf{M}^{T}\right). \end{aligned} \tag{2}$$

It can be seen that each $\mathbf{M}^{(v)}$ is forced to be close to the integrative $\mathbf{M}$, which maximizes the agreement between varying views.

3. PRELIMINARIES

The meanings of some common notations are presented in Table 1. For a transparent comprehension of the equations in this paper, we introduce some related definitions herein. Given tensors $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, their **t-product** $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$ is computed by

$$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}} = bvfold(bcirc(\boldsymbol{\mathcal{A}}) \cdot bvec(\boldsymbol{\mathcal{B}})). \tag{3}$$

Table 1: Descriptions of various notations.

Notations	Descriptions
$\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$	A matrix
$\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$	A tensor
$\mathbf{X}^{(v)} \in \mathbb{R}^{d^{(v)} \times n}$	The feature matrix of the v-th view. $d^{(v)}$ and n denote the dimension and the number of samples of the feature matrix, respectively.
$\|\|\mathbf{M}\|\|_{2,1}$	The $l_{2,1}$-norm, i.e., $\|\|\mathbf{M}\|\|_{2,1} = \sum_j \|\|\mathbf{M}(:,j)\|\|_2$.
$\|\|\mathbf{M}\|\|_{\infty}$	The ∞-norm, i.e., $\|\|\mathbf{M}\|\|_{\infty} = max_i \sum_j \|\mathbf{M}_{ij}\|$.
$Tr(\cdot)$	The trace operator
$\boldsymbol{\mathcal{T}}_{ijk}$	The (i, j, k)-th item of $\boldsymbol{\mathcal{T}}$
$\boldsymbol{\mathcal{T}}^{(i)}$	The i-th frontal slice of $\boldsymbol{\mathcal{T}}$
$\boldsymbol{\mathcal{T}}_f$	The result of performing fast Fourier transformation (FFT) on $\boldsymbol{\mathcal{T}}$ along the third dimension, i.e., $\boldsymbol{\mathcal{T}}_f = fft(\boldsymbol{\mathcal{T}}, [\,], 3)$. Moreover, $\boldsymbol{\mathcal{T}}$ is rederived via the inverse FFT, i.e., $\boldsymbol{\mathcal{T}}_f = ifft(\boldsymbol{\mathcal{T}}, [\,], 3)$.
$bcirc(\boldsymbol{\mathcal{T}})$	$bcirc(\boldsymbol{\mathcal{T}}) = \begin{bmatrix} \boldsymbol{\mathcal{T}}^{(1)} & \boldsymbol{\mathcal{T}}^{(n_3)} & \cdots & \boldsymbol{\mathcal{T}}^{(2)} \\ \boldsymbol{\mathcal{T}}^{(2)} & \boldsymbol{\mathcal{T}}^{(1)} & \cdots & \boldsymbol{\mathcal{T}}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{T}}^{(n_3)} & \boldsymbol{\mathcal{T}}^{(n_3-1)} & \cdots & \boldsymbol{\mathcal{T}}^{(1)} \end{bmatrix}$
$bvec(\boldsymbol{\mathcal{T}})$	$bvec(\boldsymbol{\mathcal{T}}) = \left[\boldsymbol{\mathcal{T}}^{(1)}; \boldsymbol{\mathcal{T}}^{(2)}; \cdots; \boldsymbol{\mathcal{T}}^{(n_3)}\right]$
$bdiag(\boldsymbol{\mathcal{T}})$	$bdiag(\boldsymbol{\mathcal{T}}) = \begin{bmatrix} \boldsymbol{\mathcal{T}}^{(1)} & & & \\ & \boldsymbol{\mathcal{T}}^{(2)} & & \\ & & \ddots & \\ & & & \boldsymbol{\mathcal{T}}^{(n_3)} \end{bmatrix}$

The operator $bvfold(\cdot)$ can be viewed as the inverse operation of $bvec(\cdot)$, i.e., $bvfold(bvec(\boldsymbol{\mathcal{A}})) = \boldsymbol{\mathcal{A}}$. For tensor $\boldsymbol{\mathcal{A}}$, it is orthogonal if it satisfies

$$\boldsymbol{\mathcal{A}}^T * \boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{A}}^T = \boldsymbol{\mathcal{I}}, \tag{4}$$

where $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{n_1 \times n_1 \times n_2}$ is an identity tensor, whose first frontal slice is a $n_1 \times n_1$ identity matrix and the other frontal slices are all zeros.

Definition 1. ***(t-SVD)*** *For tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its t-SVD is defined as*

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{D}} * \boldsymbol{\mathcal{V}}^T, \tag{5}$$

where both $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal, $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor, whose each frontal slice is a diagonal matrix.

Definition 2. ***(t-SVD based TNN)*** *According to [44], the t-SVD based TNN of $\boldsymbol{\mathcal{X}}$ is defined as*

$$\|\boldsymbol{\mathcal{X}}\|_{\circledast} = \sum_{i=1}^{\min\{n_1,n_2\}} \sum_{j=1}^{n_3} |\boldsymbol{\mathcal{D}}_f^{(j)}(i,i)|, \tag{6}$$

where $\boldsymbol{\mathcal{D}}_f^{(j)}$ is gained by $\boldsymbol{\mathcal{X}}_f^{(j)} = \boldsymbol{\mathcal{U}}_f^{(j)} * \boldsymbol{\mathcal{D}}_f^{(j)} * \boldsymbol{\mathcal{V}}_f^{(j)^T}$.

4. The Proposed Method

4.1. Problem Formulation

For a data feature matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, it is well known that the self-representation based subspace learning is formulated as $\mathbf{X} = \mathbf{XZ} + \mathbf{E}$, where $\mathbf{Z} \in \mathbb{R}^{n \times n}$ and $\mathbf{E} \in \mathbb{R}^{d \times n}$ denote the self-representation matrix and error matrix, respectively. Specifically, $\mathbf{Z}$ also depicts the underlying relationships of data points. In multi-view scenarios, we want to explore the self-representation matrix $\mathbf{Z}^{(v)}$ for the v-th view on the basis of the above formula. Thus, inspired by the works [15, 16], we reorganize multiple subspace matrices $\{\mathbf{Z}^{(v)}\}_{v=1}^m$ into a 3-order tensor $\boldsymbol{\mathcal{Z}}$, which is imposed with the t-SVD based TNN to capture the high-order correlation and global consensus between views.

After optimization, a low-rank representation tensor $\widetilde{\boldsymbol{\mathcal{Z}}}$ is obtained, which cannot be directly employed for implementing clustering task. Therefore, how to acquire a unified affinity representation matrix $\mathbf{A}$ is posed ahead of us. With respect to existing works, there are two common ways to get the consistent affinity

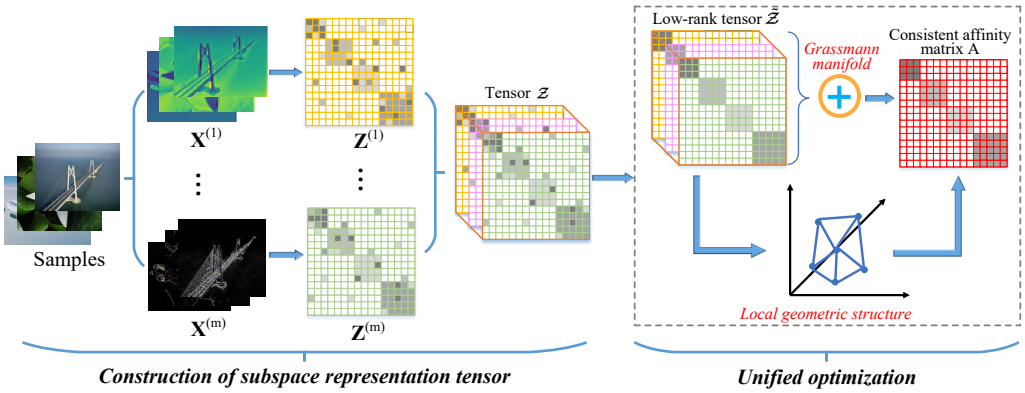


Figure 1: The flowchart of the proposed CARLDLC. $\{\mathbf{Z}^{(v)}\}_{v=1}^{m}$ denote the subspace representations derived by the original multi-view data $\{\mathbf{X}^{(v)}\}_{v=1}^{m}$, which are assembled into a 3-order tensor. The t-SVD based TNN is adopted to recover the low-rank component. Thus, we integrate multiple subspace representations to obtain the consistent affinity matrix $\mathbf{A}$ on the Grassmann manifold. While optimizing the low-rank tensor $\tilde{\mathcal{Z}}$ and the consistent affinity matrix $\mathbf{A}$, the local geometric structure of the uniformity subspace is also encoded.

matrix. The first way is to average all subspace representations $\{\mathbf{Z}^{(v)}\}_{v=1}^{m}$, i.e., $\mathbf{A} = \frac{1}{m}\sum_{v=1}^{m}(\mathbf{Z}^{(v)} + \mathbf{Z}^{(v)^T})/2$. The second way is to fuse $\{\mathbf{Z}^{(v)}\}_{v=1}^{m}$ in a weighted manner to obtain a consistent affinity matrix, i.e., solving the minimization problem $\min_{\mathbf{A},\mathbf{w}} \sum_{v=1}^{m} w^{(v)}||\mathbf{Z}^{(v)} - \mathbf{A}||_F^2 + \lambda||\mathbf{w}||_2^2$, where $\mathbf{w}$ is weight vector composed of the weight coefficients $\{w^{(v)}\}_{v=1}^{m}$. Both of these approaches are linear solutions in Euclidean space, which may destroy the local structure of data and be susceptible to noises. Thus, we attempt to fuse multiple subspace representations $\{\mathbf{Z}^{(v)}\}_{v=1}^{m}$ into a consistent subspace representation on the Grassmann manifold. From Eq. (2), we can realize that the alignment and fusion of multiple subspaces on the Grassmann manifold are achieved by minimizing the geometric distance between the consistent basis matrix and the view-specific basis matrices, which has been illustrated to be an effective and robust fusion method in the existing works [21, 22, 23].

Assuming that the consistent subspace is spanned by the basis matrix $\mathbf{M}$, then the affinity between samples in the consistent subspace is given by $\mathbf{A} = \mathbf{M}\mathbf{M}^T$. Likewise, supposing that the v-th subspace is spanned based on the basis matrix $\mathbf{M}^{(v)}$, the affinity between samples in the v-th subspace is given by $\mathbf{Z}^{(v)} + \mathbf{Z}^{(v)^T} = \mathbf{M}^{(v)}\mathbf{M}^{(v)^T}$. According to Eq. (2), the integration of multiple subspaces is trans-

formed into the following problem:

$$\min_{\mathbf{A}} rm - \sum_{v=1}^{m} Tr\left((\mathbf{Z}^{(v)} + \mathbf{Z}^{(v)^T})\mathbf{A}\right). \tag{7}$$

For the consensus affinity matrix $\mathbf{A}$, we want to further enhance its cluster structure, so the matrix nuclear norm is employed to constrain it. Meanwhile, rm is a constant and can be discarded, then the optimization problem with respect to $\mathbf{A}$ is rewritten as

$$\min_{\mathbf{A}} - \sum_{v=1}^{m} Tr\left((\mathbf{Z}^{(v)} + \mathbf{Z}^{(v)^T})\mathbf{A}\right) + ||\mathbf{A}||_*. \tag{8}$$

Furthermore, we take inspiration from graph learning theory [45, 46] and are concerned with a basic fact that no matter in any view, two closed data points have high similarity, i.e., the data locality. In other words, it is expected that the subspace representations $\mathbf{z}_i^{(v)}$ and $\mathbf{z}_j^{(v)}$ that are located close together should have a large similarity value $\mathbf{A}_{ij}$. Then, we have

$$\sum_i \sum_j ||\mathbf{z}_i^{(v)} - \mathbf{z}_j^{(v)}||_2^2 \mathbf{A}_{ij} = \frac{1}{2} Tr(\mathbf{Z}^{(v)} \mathbf{L} \mathbf{Z}^{(v)^T}). \tag{9}$$

The Laplacian matrix $\mathbf{L}$ is calculated by $\mathbf{L} = \mathbf{D} - (\mathbf{A} + \mathbf{A}^T)/2$, $\mathbf{D}$ denotes a diagonal matrix and is defined as $\mathbf{D}_{ii} = \sum_j (\frac{\mathbf{A}+\mathbf{A}^T}{2})_{ij}$.

For the sake of learning a consistent clustering-friendly affinity matrix, we present the final objective function of the proposed CARLDLC as follows

$$\begin{aligned} \min_{\boldsymbol{\mathcal{Z}},\mathbf{E},\mathbf{A}} & \|\boldsymbol{\mathcal{Z}}\|_{\circledast} + \sum_{v=1}^{m} \left(\alpha Tr(\mathbf{Z}^{(v)} \mathbf{L} \mathbf{Z}^{(v)^T}) - \beta Tr((\mathbf{Z}^{(v)} + \mathbf{Z}^{(v)^T})\mathbf{A})\right) \\ & + \gamma ||\mathbf{A}||_* + \lambda \|\mathbf{E}\|_{2,1} \\ \text{s.t. } & \mathbf{X}^{(v)} = \mathbf{X}^{(v)} \mathbf{Z}^{(v)} + \mathbf{E}^{(v)}, \\ & \boldsymbol{\mathcal{Z}} = \Phi\left(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \ldots, \mathbf{Z}^{(m)}\right), \\ & \mathbf{E} = \left[\mathbf{E}^{(1)}; \mathbf{E}^{(2)}; \ldots; \mathbf{E}^{(m)}\right], \end{aligned} \tag{10}$$

where $\Phi(\cdot)$ denotes the operation of stacking multiple matrices into a 3-order tensor. $\mathbf{E}$ is constrained by $l_{2,1}$-norm, which aims to resist the impact of sample-specific noises. α, β, γ, and λ are four tradeoff parameters used to balance the five terms. It is worth emphasizing that the proposed CARLDLC enhances the

global consistency, exploits the high-order correlations between views via low-rank tensor optimization. Then, multiple subspace representations are merged to obtain the consistent affinity matrix on the Grassmann manifold. Meanwhile, we attach importance to preserving the local structure within the uniformity subspace. These three considerations ensure that the proposed CARLDLC achieves excellent clustering results.

4.2. Optimization Process

We update each variable of Eq. (10) based on ADMM. For tackling the inseparability of variable $\boldsymbol{\mathcal{Z}}$ and $\mathbf{A}$, the auxiliary variables $\boldsymbol{\mathcal{H}}$ and $\mathbf{G}$ are introduced. Thus, the augmented Lagrangian function of Eq. (10) is written as

$$\begin{aligned}
&\boldsymbol{\mathcal{L}}\left(\{\mathbf{Z}^{(v)}\}_{v=1}^{m};\mathbf{E};\boldsymbol{\mathcal{H}};\mathbf{A};\mathbf{G}\right)\\
&=\|\boldsymbol{\mathcal{H}}\|_{\circledast}+\alpha\sum_{v=1}^{m}Tr(\mathbf{Z}^{(v)}\mathbf{L}\mathbf{Z}^{(v)^T})-\beta\sum_{v=1}^{m}Tr\left((\mathbf{Z}^{(v)}+\mathbf{Z}^{(v)^T})\mathbf{A}\right)\\
&+\sum_{v=1}^{m}\left(\langle\mathbf{J}^{(v)},\mathbf{X}^{(v)}-\mathbf{X}^{(v)}\mathbf{Z}^{(v)}-\mathbf{E}^{(v)}\rangle+\frac{\mu}{2}||\mathbf{X}^{(v)}-\mathbf{X}^{(v)}\mathbf{Z}^{(v)}-\mathbf{E}^{(v)}\|_F^2\right)\\
&+\langle\boldsymbol{\mathcal{K}},\boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{H}}\rangle+\frac{\mu}{2}\|\boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{H}}\|_F^2+\gamma||\mathbf{G}||_*+\lambda\|\mathbf{E}\|_{2,1}+\langle\mathbf{M},\mathbf{A}-\mathbf{G}\rangle\\
&+\frac{\mu}{2}||\mathbf{A}-\mathbf{G}||_F^2
\end{aligned}\tag{11}$$

where $\{\mathbf{J}^{(v)}\}_{v=1}^{m}$, $\boldsymbol{\mathcal{K}}$, and $\mathbf{M}$ are Lagrange multipliers, μ is the penalty parameter. $\langle\cdot,\cdot\rangle$ denotes the inner product operation. The ways to update variables are equivalent to solving several subproblems as follows.

$\mathbf{Z}^{(v)}$-Subproblem: When keeping the related items, the subproblem of $\mathbf{Z}^{(v)}$ becomes the following form:

$$\begin{aligned}
\mathbf{Z}^{(v)^*}&=\arg\min_{\mathbf{Z}^{(v)}}\alpha Tr(\mathbf{Z}^{(v)}\mathbf{L}\mathbf{Z}^{(v)^T})-\beta Tr((\mathbf{Z}^{(v)}+\mathbf{Z}^{(v)^T})\mathbf{A})\\
&+\langle\mathbf{J}^{(v)},\mathbf{X}^{(v)}-\mathbf{X}^{(v)}\mathbf{Z}^{(v)}-\mathbf{E}^{(v)}\rangle+\frac{\mu}{2}||\mathbf{X}^{(v)}-\mathbf{X}^{(v)}\mathbf{Z}^{(v)}-\mathbf{E}^{(v)}||_F^2\\
&+\langle\mathbf{K}^{(v)},\mathbf{Z}^{(v)}-\mathbf{H}^{(v)}\rangle+\frac{\mu}{2}||\mathbf{Z}^{(v)}-\mathbf{H}^{(v)}||_F^2.
\end{aligned}\tag{12}$$

Setting the partial derivative with respect to $\mathbf{Z}^{(v)}$ to zero, we get

$$\begin{aligned}
&\mathbf{Z}^{(v)}(2\alpha\mathbf{L})+(\mu\mathbf{I}+\mu\mathbf{X}^{(v)^T}\mathbf{X}^{(v)})\mathbf{Z}^{(v)}=\beta(\mathbf{A}+\mathbf{A}^T)\\
&-\mathbf{K}^{(v)}+\mu\mathbf{X}^{(v)^T}\mathbf{X}^{(v)}-\mu\mathbf{X}^{(v)^T}\mathbf{E}^{(v)}+\mathbf{X}^{(v)^T}\mathbf{J}^{(v)}+\mu\mathbf{H}^{(v)}.
\end{aligned}\tag{13}$$

It can been observed that Eq. (13) conforms to the form of Sylvester equation [47], i.e., $\mathbf{A}\mathbf{Z}^{(v)} + \mathbf{Z}^{(v)}\mathbf{B} = \mathbf{C}$. Then the function $lyap(\cdot)$ in MATLAB can be used to obtain the solution of $\mathbf{Z}^{(v)}$.

E-Subproblem: Discarding the items irrelevant to $\mathbf{E}$, the $\mathbf{E}$ subproblem is expressed as

$$\begin{aligned}\mathbf{E}^* &= \arg\min_{\mathbf{E}} \lambda\|\mathbf{E}\|_{2,1} + \sum_{v=1}^{m}\big(\langle \mathbf{J}^{(v)}, \mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}\rangle \\ &\quad + \frac{\mu}{2}||\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}\|_F^2\big) \\ &= \arg\min_{\mathbf{E}} \lambda||\mathbf{E}||_{2,1} + \frac{\mu}{2}\sum_{v=1}^{m}||\mathbf{E}^{(v)} - (\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \frac{1}{\mu}\mathbf{J}^{(v)})\|_F^2 \\ &= \arg\min_{\mathbf{E}} \lambda||\mathbf{E}||_{2,1} + \frac{\mu}{2}||\mathbf{E} - \mathbf{F}||_F^2.\end{aligned} \tag{14}$$

For convenience, we denote $\mathbf{F}^{(v)} = \mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \frac{1}{\mu}\mathbf{J}^{(v)}$, and $\mathbf{F} = [\mathbf{F}^{(1)}; \cdots; \mathbf{F}^{(m)}]$. According to [17], we update $\mathbf{E}$ by

$$\mathbf{E}^*_{:,j} = \begin{cases} \frac{||\mathbf{F}_{:,j}||_2 - \frac{\lambda}{\mu}}{||\mathbf{F}_{:,j}||_2}\mathbf{F}_{:,j}, & \|\mathbf{F}_{:,j}\|_2 > \frac{\lambda}{\mu} \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{15}$$

A-Subproblem: Fixing $\{\mathbf{Z}^{(v)}\}_{v=1}^m$, $\mathbf{E}$, $\boldsymbol{\mathcal{H}}$, and $\mathbf{G}$, we obtain the subproblem with respect to $\mathbf{A}$.

$$\begin{aligned}\mathbf{A}^* = \arg\min_{\mathbf{A}} &\sum_{v=1}^{m}\Big(\alpha Tr(\mathbf{Z}^{(v)}\mathbf{L}\mathbf{Z}^{(v)^T}) - \beta Tr((\mathbf{Z}^{(v)} + \mathbf{Z}^{(v)^T})\mathbf{A})\Big) \\ &+ \langle \mathbf{M}, \mathbf{A} - \mathbf{G}\rangle + \frac{\mu}{2}||\mathbf{A} - \mathbf{G}||_F^2.\end{aligned} \tag{16}$$

Taking the derivative of $\mathbf{A}$ and setting its value to zero, the updating approach is as follows:

$$\mathbf{A}^* = \left(\sum_{v=1}^{m}\Big(\alpha\mathbf{Z}^{(v)^T}\mathbf{Z}^{(v)} + \beta(\mathbf{Z}^{(v)} + \mathbf{Z}^{(v)^T})\Big) + \mu\mathbf{G} - \mathbf{M}\right)/\mu. \tag{17}$$

$\boldsymbol{\mathcal{H}}$-Subproblem: Ignoring the terms unrelated to $\boldsymbol{\mathcal{H}}$, the $\boldsymbol{\mathcal{H}}$ subproblem be-

comes

$$\begin{aligned}\boldsymbol{\mathcal{H}}^* &= \arg\min_{\boldsymbol{\mathcal{H}}} ||\boldsymbol{\mathcal{H}}||_{\circledast} + \langle \boldsymbol{\mathcal{K}}, \boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{H}} \rangle + \frac{\mu}{2}||\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{H}}||_F^2 \\ &= \arg\min_{\boldsymbol{\mathcal{H}}} ||\boldsymbol{\mathcal{H}}||_{\circledast} + \frac{\mu}{2}||\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{H}} + \frac{1}{\mu}\boldsymbol{\mathcal{K}}||_F^2 \\ &= \arg\min_{\boldsymbol{\mathcal{H}}} ||\boldsymbol{\mathcal{H}}||_{\circledast} + \frac{\mu}{2}||\boldsymbol{\mathcal{H}} - (\boldsymbol{\mathcal{Z}} + \frac{1}{\mu}\boldsymbol{\mathcal{K}})||_F^2.\end{aligned} \tag{18}$$

Before optimizing $\boldsymbol{\mathcal{H}}$, we take a cue from the work [16] and rotate $\boldsymbol{\mathcal{H}}$ from $n \times n \times m$ to $n \times m \times n$. Fig. 2 illustrates the approach of tensor rotation. The rotation is necessary. According the definition of t-SVD based tensor nuclear norm, after rotation, the low-rank property across views can be captured, that is, the elements at the same position of varying subspace representations tend to coincide, thus exploring the high-order correlations.

Inspired by [48], we update $\boldsymbol{\mathcal{H}}$ by the tensor tubal-shrinkage operator

$$\boldsymbol{\mathcal{H}}^* = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{T}}_{m/\mu}(\boldsymbol{\mathcal{D}}) * \boldsymbol{\mathcal{V}}^T, \tag{19}$$

where $\boldsymbol{\mathcal{T}}_{m/\mu}(\boldsymbol{\mathcal{D}}) = \boldsymbol{\mathcal{D}} * \boldsymbol{\mathcal{F}}$. $\boldsymbol{\mathcal{F}}$ denotes an f-diagonal tensor, the calculation method of its diagonal element of the j-th frontal slice is

$$\boldsymbol{\mathcal{F}}_f(i,i,j) = max\{1 - \frac{m/\mu}{\boldsymbol{\mathcal{D}}(i,i,j)}, 0\}. \tag{20}$$

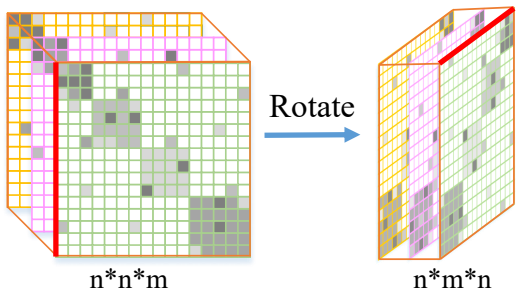


Figure 2: Illustration of tensor rotation. The dimensions of the tensor before and after rotation are $n \times n \times m$ and $n \times m \times n$, respectively. The position of the red line before and after rotation graphically demonstrates how the tensor rotates.

G-Subproblem: When the terms associated with G are kept, the following

subproblem is gained:

$$\begin{aligned}\mathbf{G}^* &= \arg\min_{\mathbf{G}} \gamma||\mathbf{G}||_* + \langle \mathbf{M}, \mathbf{A}-\mathbf{G}\rangle + \frac{\mu}{2}||\mathbf{A}-\mathbf{G}||_F^2 \\ &= \arg\min_{\mathbf{G}} \gamma||\mathbf{G}||_* + \frac{\mu}{2}||\mathbf{A}-\mathbf{G}+\frac{1}{\mu}\mathbf{M}||_F^2 \\ &= \arg\min_{\mathbf{G}} \gamma||\mathbf{G}||_* + \frac{\mu}{2}||\mathbf{G}-(\mathbf{A}+\frac{1}{\mu}\mathbf{M})||_F^2.\end{aligned} \tag{21}$$

According to [49], Eq. (21) can be updated via the singular value threshold algorithm, i.e.,

$$\mathbf{G}^* = \mathbf{U} * \mathcal{S}_{\gamma/\mu}(\mathbf{\Sigma}) * \mathbf{V}^T, \tag{22}$$

where $\mathbf{U}$, $\mathbf{\Sigma}$, and $\mathbf{V}$ are obtained by implementing SVD on the matrix $(\mathbf{A}+\frac{1}{\mu}\mathbf{M})$. The shrinkage operator $\mathcal{S}_{\gamma/\mu}(\mathbf{\Sigma})$ is defined as $\mathcal{S}_{\gamma/\mu}(\mathbf{\Sigma}) = max(\mathbf{\Sigma}-\gamma/\mu, 0) + min(\mathbf{\Sigma}+\gamma/\mu, 0)$.

Update the Lagrangian multipliers $\mathbf{J}^{(v)}$, $\boldsymbol{\mathcal{K}}$, $\mathbf{M}$, and penalty parameter μ:

$$\begin{aligned}&\mathbf{J}^{(v)*} = \mathbf{J}^{(v)} + \mu(\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}); \\ &\boldsymbol{\mathcal{K}}^* = \boldsymbol{\mathcal{K}} + \mu(\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{K}}); \\ &\mathbf{M}^* = \mathbf{M} + \mu(\mathbf{A}-\mathbf{G}); \\ &\mu^* = min(\omega * \mu, \mu_{max}).\end{aligned} \tag{23}$$

In Algorithm 1, we summarize the outline of the proposed CARLDLC. Among these subproblems, the updating approaches of $\mathbf{Z}^{(v)}$, $\mathbf{E}$, $\boldsymbol{\mathcal{H}}$, and $\mathbf{G}$ result in high computational complexity. To be specific, updating $\mathbf{Z}^{(v)}$ by solving Sylvester equation takes $\mathcal{O}(n^3)$. Hence, updating the set $\{\mathbf{Z}^{(v)}\}_{v=1}^m$ needs $\mathcal{O}(mn^3)$. For the subproblem of $\mathbf{E}$, it costs $\mathcal{O}(mn^2)$. In the optimization process of $\boldsymbol{\mathcal{H}}$, FFT, inverse FFT, and SVD operations are used, which take $\mathcal{O}(m^2n^2 + mn^2log(n))$ in total. For solving $\mathbf{G}$, the SVD is performed, so $\mathcal{O}(n^3)$ is costed. Let t be the number of iterations, the computational complexity of the proposed CARLDLC is $\mathcal{O}(t(mn^2log(n) + (m+1)n^3 + (m^2+m)n^2))$.

4.3. Discussion

Compared with multi-view clustering methods MVLDAM [38] and LRTG [50] related to the proposed CARLDLC, there are several significant differences between them.

1) MVLDAM *vs* CARLDLC:

Table 2: Statistics of six datasets.

Dataset	Samples	Views	Clusters	Features
BBCsports	544	2	5	3,183/3,203
Citeseer	3,312	2	6	3,703/3,312
MSRC	210	6	7	1,302/48/512/100/256/210
UCI	2,000	3	10	240/76/6
WikipediaArticles	693	2	10	128/10
Youtube	2,000	6	10	2,000/1,024/64/512/64/647

Algorithm 1 Consistent Affinity Representation Learning with Dual Low-rank Constraints for Multi-view Subspace Clustering

Input: $\boldsymbol{\mathcal{X}} = \left\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(m)}\right\}$, $\mathbf{X}^{(v)} \in \mathbb{R}^{d^{(v)} \times n}$, α, β, γ, λ.
Output: Consistent affinity representation matrix $\mathbf{A}$.

Initialize $\boldsymbol{\mathcal{Z}}_0 = \boldsymbol{\mathcal{H}}_0 = \boldsymbol{\mathcal{K}}_0 = \mathbf{0}$, $\mathbf{E}_0 = \mathbf{0}$, $\mathbf{J}_0^{(v)} = \mathbf{0}$, $\mathbf{M}_0 = \mathbf{0}$, $\omega = 2$, $\varepsilon = 10^{-6}$, μ_0, $\mu_{max} = 10^{10}$, $k = 0$.
while not convergent **do**
  for $v = 1 : m$ **do**
    Update $\mathbf{Z}_{k+1}^{(v)}$ by Eq. (13);
  end for
  Update $\mathbf{E}_{k+1}$ by Eq. (15);
  Update $\mathbf{A}_{k+1}$ by Eq. (17);
  Update $\boldsymbol{\mathcal{H}}_{k+1}$ by Eq. (19);
  Update $\mathbf{G}_{k+1}$ by Eq. (22);
  Update $\mathbf{J}_{k+1}^{(v)}$, $\boldsymbol{\mathcal{K}}_{k+1}$, $\mathbf{M}_{k+1}$, and μ_{k+1} by Eq. (23);
  Check the convergence conditions:
  $||\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}_{k+1}^{(v)} - \mathbf{E}_{k+1}^{(v)}||_\infty \leq \varepsilon$,
  $||\mathbf{Z}_{k+1}^{(v)} - \mathbf{H}_{k+1}^{(v)}||_\infty \leq \varepsilon$,
  $||\mathbf{A}_{k+1} - \mathbf{G}_{k+1}||_\infty \leq \varepsilon$.
  $k = k + 1$;
end while
return Matrix $\mathbf{A}$;

- Different construction ways of subspace representation tensor. Given a multi-view dataset $\{\mathbf{X}^{(v)}\}_{v=1}^{m}$, MVLDAM reorganizes it to a multi-view data tensor $\mathcal{X} \in \mathbb{R}^{d\times n\times m}$, where $d = \sum_{v=1}^{m} d^{(v)}$. Then MVLDAM explores the subspace representation tensor $\mathcal{C} \in \mathbb{R}^{n\times n\times m}$ via $\min_{\mathcal{C}} ||\mathcal{X} - \mathcal{X} * \mathcal{C}||_F^2$. However, the proposed CARLDLC first learns the view-specific subspace representation $\mathbf{Z}^{(v)}$ by $\mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \mathbf{E}^{(v)}$, then assembles $\{\mathbf{Z}^{(v)}\}_{v=1}^{m}$ into a third-order tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{n\times n\times m}$.

- Different selections of tensor nuclear norm. In order to preserve the low-rank property of subspace representation tensor, MVLDAM adopts the definition of $||\boldsymbol{\mathcal{X}}||_{TNN} = (1/n_3)\sum_{i=1}^{n_3} ||\boldsymbol{\mathcal{X}}_f^{(i)}||_*$, where $||\cdot||_*$ denotes the nuclear norm of a matrix. Nevertheless, CARLDLC utilizes t-SVD based tensor nuclear norm, i.e., $\|\boldsymbol{\mathcal{X}}\|_{\circledast} = \sum_{i=1}^{\min\{n_1,n_2\}}\sum_{j=1}^{n_3} |\boldsymbol{\mathcal{D}}_f^{(j)}(i,i)|$, where $\boldsymbol{\mathcal{D}}_f$ is obtained via performing the t-SVD on $\boldsymbol{\mathcal{X}}_f$.

- Different constraints on the consistent affinity matrix. MVLDAM imposes l_1-norm on the consistent affinity matrix $\mathbf{W}$ to enforce its information capability, while CARLDLC constrains the consensus affinity matrix $\mathbf{A}$ with matrix nuclear norm. The features of the same clusters are similar, while the features of diverse clusters are variable. Therefore, it is beneficial to use the low-rank constraint to obtain a clustering-driven affinity matrix.

- Different dispositions of the error matrix. Learning subspace representations in either tensor or matrix forms results in an inherent bias tensor/matrix that portrays noise and outliers. Imposing a certain regularization constraint on this bias tensor/matrix is conducive to enhancing the robustness of models. MVLDAM does not take into account the effect of bias tensor, while CARLDLC uses the $l_{2,1}$-norm constraint on the bias matrix $\mathbf{E}$, which strengthens its robustness.

- Different treatments of local structure. Discovering the hidden local structure in data has positive significance for boosting clustering performance, so CARLDLC leverages the graph regularization $Tr(\mathbf{Z}^{(v)}\mathbf{L}\mathbf{Z}^{(v)^T})$ to preseve the locality of data features. However, MVLDAM ignores it.

2) LRTG vs CARLDLC:

- Different learning methods for the consistent affinity matrix A. LRTG directly explores the graph embedding $\mathbf{A}$ from multiple subspace representations $\{\mathbf{Z}^{(v)}\}_{v=1}^{m}$, which constructs the consistent affinity matrix $\mathbf{A}$ mainly

from a perspective of locality. While CARLDLC first considers robustly integrating multiple subspace representations $\{\mathbf{Z}^{(v)}\}_{v=1}^{m}$ to initially obtain a consensus affinity representation $\mathbf{A}$, thus imposes the low-rank constraint on it to enhance the global cluster structures. Furthermore, we also attach importance to the data locality, and use the second term in Eq. (10) to encode the local structures within the uniform subspace. Alternatively, another perspective on the role of the second term is that it can smooth the elements in $\mathbf{A}$ and boost its information characterization capability [10].

- Different ways to capture the low-rank property. LRTG recovers a low-rank tensor $\widetilde{\mathcal{Z}}$ via Tucker decomposition. While CARLDLC explores the low-rank property of tensor $\mathcal{Z}$ using the t-SVD based tensor nuclear norm.

5. Experiments

5.1. Datasets

We choose six publicly real-world datasets to investigate the performance of the proposed CARLDLC, their detailed statistics are introduced as follows. Table 2 also shows the details of these datasets.

BBCsports [1] is comprised of 544 documents with 5 themes about sports, which are excerpted from BBC website, each news is represented by 2 features.

Citeseer contains 3,312 documents derived from 6 categories. There are two kinds of feature representations in this dataset: content feature and citation feature.

MSRC [2] consists of 210 images from 7 categories such as cars, trees, and cows, etc. Each image has 6 feature representations: SIFT feature, LBP feature, CM feature, HOG feature, GIST feature, and CENTRIST feature.

UCI [51] includes 2,000 handwritten digit images from 0 to 9. Three features are extracted from each image: PIX feature, FOU feature, and MOR feature.

WikipediaArticles [3] is an article dataset collected and compiled by the Wikipedia editors. In the experiments, each sample is represented by 2 views.

Youtube [4] is a video dataset containing 2,000 samples with 10 categories, each sample has six types of features, including HOG feature, CH feature, HME feature, VS feature, MFCC feature, and SS feature.

[1] http://mlg.ucd.ie/datasets/segment.html
[2] http://research.microsoft.com/en-us/projects/objectclassrecognition/
[3] http://lig-membres.imag.fr/grimal/data.html
[4] http://archive.ics.uci.edu/ml/datasets

5.2. *Baselines and Evaluation Metrics*

We select a conventional single-view clustering algorithm and ten multi-view clustering methods as the baselines, the details of them are provided as follows.

K-means stitches together multiple views to form the available feature matrix, on which it is performed to get the clustering results.

GMC [52] fused the graph matrices of all views to construct the unified graph matrix with a certain number of connected components.

$\mathrm{SM^2SC}$ [53] explored the essential components from view-specific representations that are aligned with the consistent subspace representation.

LMVSC [31] learned multiple anchor graphs for each view, then an efficient fusion method was developed to integrate them.

MCDCF [54] proposed a multi-layer concept factorization to capture the hierarchical information of multi-view data.

LTMSC [15] aimed at recovering a low-rank tensor space using the sum of nuclear norms (SNN) to constrain the self-representation tensor.

t-SVD-MSC [16] leveraged t-SVD based TNN to learn a low-rank subspace representation tensor.

HLR-$\mathbf{M^2VS}$ [7] highlighted the protection of the local structure of data while learning a low-rank tensor space.

MVLDAM [38] directly explored the subspace representation tensor from multi-view data tensor, then fused multiple subspace representations into a common affinity matrix on the Grassmann manifold.

CGL [55] respectively learned affinity matrices using the spectral embedding matrices of multiple features, which are stacked into a 3-order tensor with low-rank constraint.

CoMSC [56] learned the robust representations of multiple views via the eigendecomposition, then a consensus subspace representation was calculated based on them.

We employ six evaluation metrics to evaluate the clustering results, including Accuaracy (ACC), Normalized Mutual Information (NMI), Purity, Adjusted Rand Index (ARI), F-score, Precision. For ARI, the range of value is [-1, 1], as for the other metrics, their values range in [0, 1]. Furthermore, higher values of the six evaluation metrics mean better performance. Considering the impact of stability, each round of comparison experiment is run ten times, and we document the mean and variance.

5.3. Experimental Results

Tables 3, 4, 5, 6, 7, 8 report the clustering results for all compared algorithms on six challenging datasets, and the numbers in bold indicate the highest values on each evaluation metric. We discuss the observations revealed by these statistics as follows:

- K-means performs poorly on the concatenation features in most cases, mainly because stitching together multi-view data both destroys the semantic information of individual view and renders the synthetic data more complicated, making it impossible for K-means to clearly partition the data. On the contrary, the majority of multi-view approaches achieve good performance owing to their ability to leverage the complementary information of multi-view data. However, there are exceptions occurring. For example, on the WikipediaArticles dataset, K-means gains favorable clustering results and outperforms some multi-view methods including GMC, MCDCF, and LMVSC. For investigating the reason, we run K-means on the first view with ACC and NMI of 19.9% and 6.3%, respectively, and run it on the second view with ACC and NMI of 56.4% and 54.1%, respectively. It can be seen that the semantic information hidden in the two views in the dataset differs significantly, which may seriously interfere with the view fusion effect of some multi-view methods.

Table 3: Comparison of clustering results (%) on the BBCsports dataset.

Methods	ACC	NMI	Purity	ARI	F-score	Precision
K-means	37.5±4.5	5.4±4.9	39.2±5.2	2.3±3.2	39.0±2.1	24.9±2.3
GMC	81.0±0.0	72.3±0.0	84.3±0.0	72.2±0.0	79.4±0.0	72.7±0.0
SM^{2}SC	90.8±0.0	82.2±0.0	90.8±0.0	85.5±0.0	89.0±0.0	88.7±0.0
LMVSC	89.3±0.0	81.4±0.0	89.3±0.0	84.7±0.0	88.4±0.0	86.8±0.0
MCDCF	94.5±4.1	88.3±2.3	95.3±2.1	90.7±2.5	92.9±1.4	92.1±1.9
LTMSC	68.2±0.0	50.4±0.0	72.8±0.0	47.0±0.0	61.2±0.0	54.4±0.0
t-SVD-MSC	94.9±0.0	85.3±0.0	94.9±0.0	85.4±0.0	88.9±0.0	87.1±0.0
HLR-M^{2}VS	94.9±0.0	85.4±0.0	94.9±0.0	86.4±0.0	89.6±0.0	89.3±0.0
MVLDAM	90.3±0.0	80.5±0.0	90.3±0.0	74.9±0.0	81.3±0.0	75.9±0.0
CGL	93.6±0.0	86.5±0.0	93.6±0.0	87.8±0.0	90.7±0.0	93.0±0.0
CoMSC	90.8±0.0	81.8±0.0	90.8±0.0	85.4±0.0	88.9±0.0	89.0±0.0
CARLDLC	**97.1±0.0**	**90.1±0.0**	**97.1±0.0**	**91.8±0.0**	**93.8±0.0**	**92.2±0.0**

Table 4: Comparison of clustering results (%) on the Citeseer dataset.

Methods	ACC	NMI	Purity	ARI	F-score	Precision
K-means	36.2±3.0	14.1±3.3	37.2±3.8	11.5±3.3	31.9±2.8	24.9±2.2
GMC	21.7±0.0	0.8±0.0	21.8±0.0	0.1±0.0	30.3±0.0	17.9±0.0
SM^{2}SC	52.5±0.0	28.7±0.0	57.8±0.0	25.1±0.0	39.4±0.0	37.0±0.0
LMVSC	45.9±0.0	21.5±0.0	48.1±0.0	17.3±0.0	32.8±0.0	31.2±0.0
MCDCF	58.9±2.1	31.1±1.9	59.2±2.0	30.6±1.5	45.5±1.8	37.9±2.5
LTMSC	62.3±0.0	35.1±0.0	64.8±0.0	35.0±0.0	46.4±0.0	47.1±0.0
t-SVD-MSC	41.0±0.0	18.5±0.0	43.3±0.0	13.9±0.0	29.7±0.0	30.7±0.0
HLR-M^{2}VS	58.3±0.0	35.1±0.0	61.8±0.0	32.7±0.0	44.3±0.0	45.8±0.0
MVLDAM	22.0±0.0	1.0±0.0	22.2±0.0	0.2±0.0	30.3±0.0	17.9±0.0
CGL	37.0±0.0	13.8±0.0	40.5±0.0	11.9±0.0	28.5±0.0	27.0±0.0
CoMSC	44.7±0.0	19.8±0.0	46.9±0.0	17.8±0.0	32.3±0.0	32.6±0.0
CARLDLC	**67.8±0.0**	**41.4±0.0**	**70.3±0.0**	**42.8±0.0**	**53.0±0.0**	**53.3±0.0**

Table 5: Comparison of clustering results (%) on the MSRC dataset.

Methods	ACC	NMI	Purity	ARI	F-score	Precision
K-means	45.0±3.2	38.1±2.4	47.6±2.8	24.3±2.6	37.4±2.2	30.1±2.3
GMC	89.5±0.0	81.6±0.0	89.5±0.0	76.7±0.0	80.0±0.0	78.6±0.0
SM^{2}SC	91.4±1.4	82.3±1.1	91.4±1.2	81.1±1.5	83.7±1.0	83.7±1.8
LMVSC	82.4±0.0	71.3±0.0	82.4±0.0	65.1±0.0	70.1±0.0	68.1±0.0
MCDCF	85.6±0.8	77.7±0.8	85.6±0.7	70.8±0.6	75.0±0.5	72.3±0.8
LTMSC	83.4±0.1	74.9±0.0	83.4±0.1	68.8±0.1	73.2±0.1	71.9±0.1
t-SVD-MSC	98.1±0.0	96.0±0.0	98.1±0.0	95.5±0.0	96.2±0.0	96.3±0.0
HLR-M^{2}VS	99.1±0.0	97.8±0.0	99.1±0.0	97.8±0.0	98.1±0.0	98.0±0.0
MVLDAM	87.1±0.0	77.5±0.0	87.1±0.0	73.1±0.0	76.8±0.0	76.0±0.0
CGL	76.1±0.2	68.5±0.2	76.1±0.2	59.3±0.2	65.1±0.2	63.6±0.3
CoMSC	76.7±0.0	63.4±0.0	76.7±0.0	54.1±0.0	60.6±0.0	59.8±0.0
CARLDLC	**1.0±0.0**	**1.0±0.0**	**1.0±0.0**	**1.0±0.0**	**1.0±0.0**	**1.0±0.0**

Table 6: Comparison of clustering results (%) on the UCI dataset.

Methods	ACC	NMI	Purity	ARI	F-score	Precision
K-means	38.8±1.0	46.6±0.3	44.2±0.8	31.4±0.1	38.9±0.2	35.4±0.4
GMC	74.8±0.0	80.9±0.0	77.4±0.0	71.0±0.0	74.1±0.0	68.0±0.0
SM^{2}SC	84.2±0.0	79.9±0.0	84.2±0.0	74.9±0.0	77.4±0.0	77.2±0.0
LMVSC	82.5±0.0	73.1±0.0	82.5±0.0	66.2±0.0	69.6±0.0	69.0±0.0
MCDCF	73.7±1.2	75.8±1.1	73.7±1.5	64.3±1.4	68.0±1.2	64.8±1.3
LTMSC	79.8±0.1	76.7±1.0	81.5±0.1	71.8±1.4	74.6±1.3	73.3±1.2
t-SVD-MSC	95.5±0.0	93.2±0.0	95.5±0.0	92.4±0.0	93.2±0.0	93.0±0.0
HLR-M^{2}VS	85.6±0.1	87.0±0.1	86.2±0.1	81.9±0.1	83.7±0.1	82.0±0.4
MVLDAM	95.9±0.0	91.3±0.0	95.9±0.0	91.0±0.0	91.9±0.0	91.8±0.0
CGL	84.3±1.7	90.5±1.6	88.6±1.8	83.2±2.8	85.0±2.5	78.9±2.8
CoMSC	77.8±0.0	78.4±0.0	81.6±0.0	69.3±0.0	72.6±0.0	66.9±0.0
CARLDLC	**97.9±0.0**	**95.0±0.0**	**97.9±0.0**	**95.3±0.0**	**95.8±0.0**	**95.7±0.0**

Table 7: Comparison of clustering results (%) on the WikipediaArticles dataset.

Methods	ACC	NMI	Purity	ARI	F-score	Precision
K-means	54.7±0.0	51.5±0.0	58.6±0.0	39.0±0.0	45.8±0.0	45.0±0.0
GMC	44.9±0.0	36.1±0.0	48.2±0.0	14.5±0.0	28.4±0.0	19.1±0.0
SM^{2}SC	55.1±0.0	50.8±0.0	59.3±0.0	40.7±0.0	47.0±0.0	48.5±0.0
LMVSC	55.6±0.0	47.5±0.0	57.0±0.0	33.1±0.0	41.0±0.0	38.0±0.0
MCDCF	47.3±2.7	35.1±1.9	50.0±1.4	24.5±1.3	32.6±1.1	32.9±1.2
LTMSC	53.1±0.3	49.5±0.5	57.5±0.0	40.7±0.0	47.1±0.0	48.1±0.0
t-SVD-MSC	55.6±0.1	48.4±0.2	58.0±0.2	40.8±0.1	47.1±0.1	48.0±0.1
HLR-M^{2}VS	54.6±0.0	48.2±0.0	57.4±0.0	40.8±0.0	47.1±0.0	47.8±0.0
MVLDAM	54.7±0.0	47.1±0.0	57.7±0.0	37.7±0.0	44.2±0.0	46.2±0.0
CGL	54.2±0.1	49.8±0.0	59.4±0.1	37.1±0.1	44.1±0.1	43.2±0.1
CoMSC	21.1±0.0	7.2±0.0	23.2±0.0	2.9±0.0	13.1±0.0	13.6±0.0
CARLDLC	**57.3±0.2**	**53.3±0.2**	**61.3±0.1**	**43.3±0.2**	**49.5±0.2**	**49.7±0.2**

Table 8: Comparison of clustering results (%) on the Youtube dataset.

Methods	ACC	NMI	Purity	ARI	F-score	Precision
K-means	24.6±1.2	15.2±0.4	27.9±1.0	8.2±0.6	19.5±0.5	15.8±0.5
GMC	11.7±0.0	2.0±0.0	12.1±0.0	0.0±0.0	18.1±0.0	10.0±0.0
SM^{2}SC	30.6±0.1	18.1±0.1	34.1±0.1	11.1±0.1	20.1±0.1	19.8±0.1
LMVSC	27.2±0.0	14.8±0.0	29.2±0.0	7.9±0.0	17.3±0.0	16.9±0.0
MCDCF	17.4±1.2	8.2±0.9	21.1±1.0	3.2±1.3	13.7±1.3	12.6±1.0
LTMSC	30.0±0.1	18.4±0.1	32.1±0.0	11.2±0.1	20.3±0.1	19.8±0.1
t-SVD-MSC	26.3±1.7	16.2±0.9	33.4±1.1	9.8±0.5	19.3±0.5	20.4±0.5
HLR-M^{2}VS	30.6±0.0	17.9±0.0	32.3±0.0	11.2±0.0	20.2±0.0	19.8±0.0
MVLDAM	28.1±0.0	14.6±0.0	29.4±0.0	8.6±0.0	19.2±0.0	16.5±0.0
CGL	32.9±0.1	19.7±0.1	35.4±0.1	12.5±0.1	21.5±0.1	20.9±0.0
CoMSC	24.6±0.0	10.4±0.0	26.0±0.0	5.9±0.0	15.4±0.0	15.1±0.0
CARLDLC	**33.0±0.1**	**21.4±0.1**	**36.2±0.2**	**14.2±0.1**	**23.2±0.1**	**22.1±0.1**

- The proposed CARLDLC leads in performance on almost all datasets. Furthermore, the improvements are significant on some datasets. For example, on the Citeseer dataset, CARLDLC improves over the next best method LTMSC by 5.5%, 6.3%, 5.5%, 7.8%, 6.6%, and 6.2% on the six evaluation metrics. Even more, perfect clustering results are achieved on the MSRC dataset. There are a number of reasons for the leading situation. Compared to the matrix-based methods GMC, SM^{2}SC, LMVSC, and MCDCF, which are dedicated to learning a shared affinity matrix or a low-dimensional representation matrix from varying views in the Euclidean space, CARLDLC fetches the high-order correlations through low-rank tensor optimization, and merges different views on the Grassmann manifold to obtain the consistent affinity matrix, which is further imposed with the low-rank constraint for strengthening the cluster structure. Thus, CARLDLC achieves better performance. In comparison to the tensor-based methods LTMSC, t-SVD-MSC, HLR-M^{2}VS, and CGL which gain the final affinity matrix via averaging all slices in the optimized low-rank tensor, CARLDLC fuses multiple slices of the learned low-rank tensor on the Grassmann manifold, and pays attention to encoding the local structure of the uniformity subspace. Given these measures, CARLDLC performs better. As for MVLDAM, which is also based on the low-rank tensor factorization and Grassmann manifold, we have explained their differences in Subsection 4.3. From the clustering

results, we can see that the proposed CARLDLC outperforms MVLDAM.

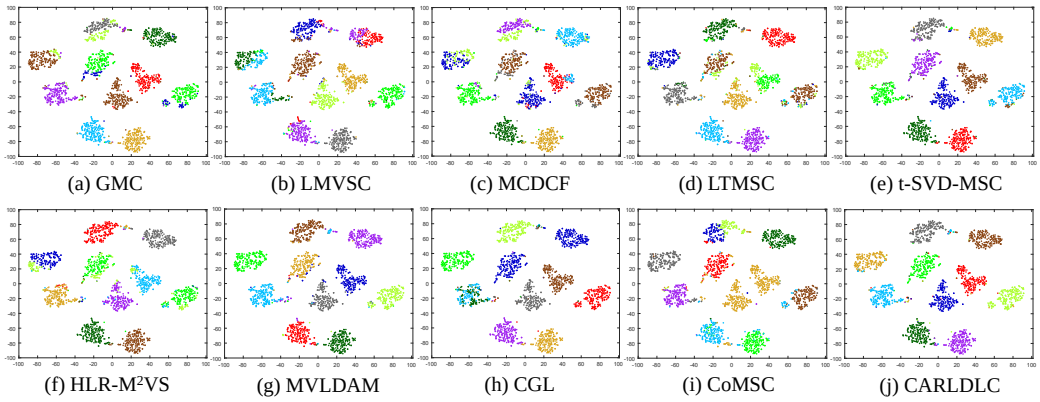
- With the help of t-SNE technology, we provide visualizations of clustering results of ten multi-view clustering methods on the UCI dataset, which are presented in Fig. 3. Data points of different clusters are rendered in different colors. It can be seen that the clusters divided by GMC, LMVSC, MVDCF, LTMSC, HLR-M^{2}VS, and CoMSC are heavily intermingled with instances from other clusters, CGL does not divide enough ten clusters, which all demonstrate that the clustering results are unsatisfactory. Nonetheless, the clusters segmented by t-SVD-MSC, MVLDAM, and CARLDLC are relatively purity, illustrating the clustering results are superior. The situations can also be confirmed by the numerical results in Table 6.

- We visualize varying affinity matrices learned by nine multi-view clustering algorithms on the BBCsports dataset in Fig. 4. A good affinity matrix can well profile the cluster structure of samples, i.e., the diagonal-block structures in the visualization. It can be observed that the methods GMC, LTMSC, t-SVD-MSC, HLR-M^{2}VS, MVLDAM, and CGL do not portray the diagonal-block structures clearly enough. On the contrary, SM^{2}SC and CoMSC can learn the diagonal-block structures well, but do not sufficiently suppress the elements at non-diagonal-block positions, indicating the lack of protection of the local structures. Fortunately, the proposed CARLDLC can perceive the diagonal-block structures well while suppressing elements at other locations.

- In addition, the visualizations of each self-representation matrix $\mathbf{Z}^{(v)}$ and the consistent affinity matrix $\mathbf{A}$ are presented in Fig. 5, which are obtained after CARLDLC converging on UCI dataset. It can be seen that the image corresponding to the consistent affinity matrix $\mathbf{A}$ most clearly depict the diagonal-block structures of data. Accordingly, it can be inferred that $\mathbf{A}$ as the input of spectral clustering algorithm is able to obtain the optimal clustering results.

5.4. Ablation Experiments

For examining the roles played by encoding the local structure of the uniformity subspace and integrating varying views on the Grassmann manifold, we perform some ablation experiments to verify. Table 9 illustrates the performance of the proposed CARLDLC and its three variants for ACC and NMI on six datasets.



(a) GMC (b) LMVSC (c) MCDCF (d) LTMSC (e) t-SVD-MSC

(f) HLR-M^{2}VS (g) MVLDAM (h) CGL (i) CoMSC (j) CARLDLC

Figure 3: A visualization comparison of ten clustering methods via t-SNE on the UCI dataset.

CARLDLC-OL represents the variant that does not incorporate the local structure, i.e., discarding the second item of the objective function Eq. (10). In the case that we do not align various views on the Grassmann manifold, but directly learn the consistent affinity matrix from multiple self-representation matrices, i.e., abandoning the third term in Eq. (10), the proposed CARLDLC devolves to CARLDLC-OG. Moreover, we further undertake an interesting comparison experiment, namely the linear fusion manner of multiple subspace representations in the Euclidean space is used to substitute the fusion on the Grassmann manifold. The objective function of new model is rewritten as

$$\begin{aligned}
&\min_{\boldsymbol{\mathcal{Z}},\mathbf{E},\mathbf{A},\mathbf{w}} \|\boldsymbol{\mathcal{Z}}\|_{\circledast} + \sum_{v=1}^{m}\left(\alpha Tr(\mathbf{Z}^{(v)}\mathbf{L}\mathbf{Z}^{(v)^T})\right) + \sum_{v=1}^{m}\omega^{(v)}||\mathbf{Z}^{(v)} - \mathbf{A}||_F^2 \\
&\qquad +\gamma||\mathbf{A}||_* + \lambda\|\mathbf{E}\|_{2,1} + \beta||\mathbf{w}||_2^2 \\
&\text{s.t. } \mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \mathbf{E}^{(v)}, \mathbf{w} = \{\omega^{(v)}\}_{v=1}^{m}, \\
&\quad \boldsymbol{\mathcal{Z}} = \Phi\left(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \ldots, \mathbf{Z}^{(m)}\right), \\
&\quad \mathbf{E} = \left[\mathbf{E}^{(1)}; \mathbf{E}^{(2)}; \ldots; \mathbf{E}^{(m)}\right],
\end{aligned} \tag{24}$$

which is abbreviated as CARLDLC-LF. From the Eq. (2) and Eq. (24), we can observe that the fusion on the Grassmann manifold aligns basis vectors in the uniform subspace with the basis vectors in varying subspaces, while the fusion in the Euclidean space narrows the distances between all elements in the consistent affinity representation and all subspace representations, the former fusion method is more robust to noise than the latter [21]. It can be observed that CARLDLC

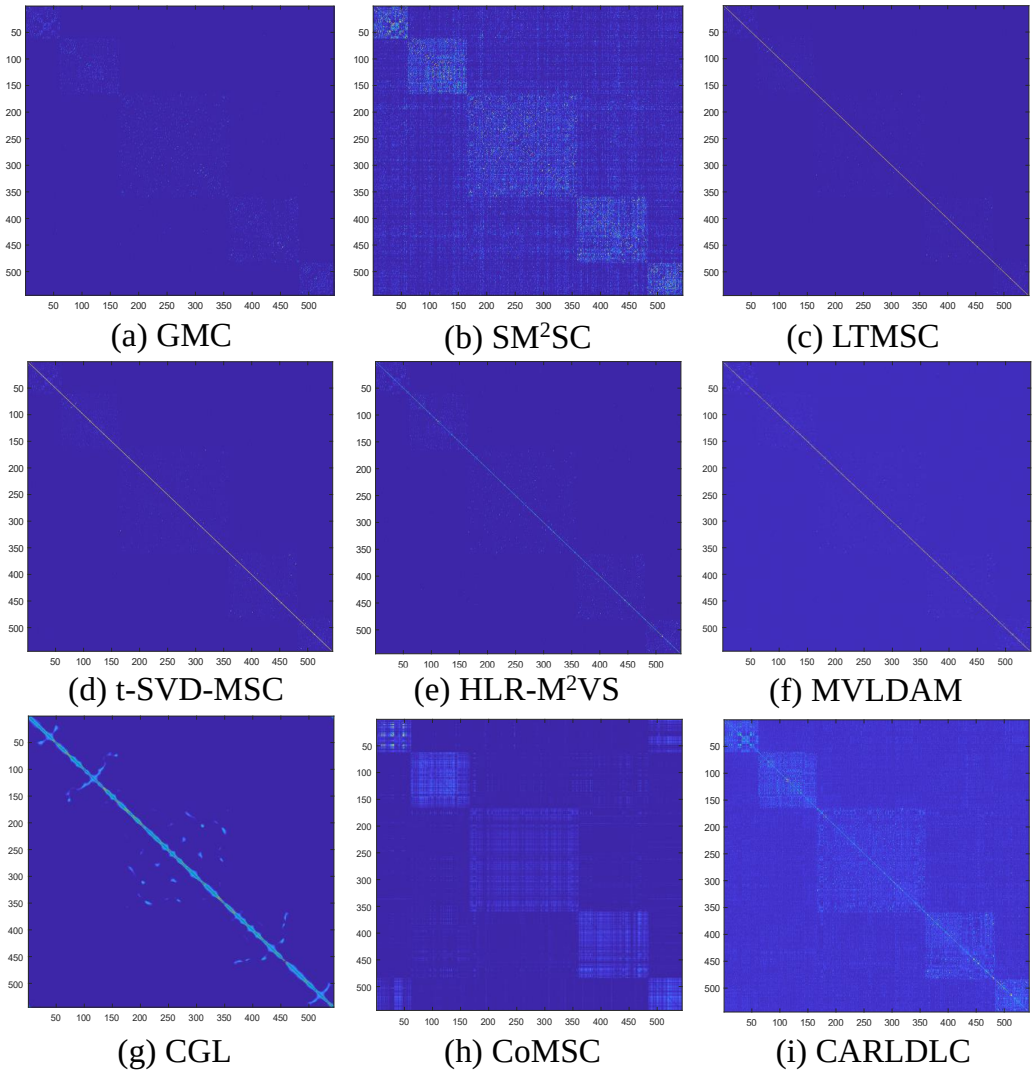
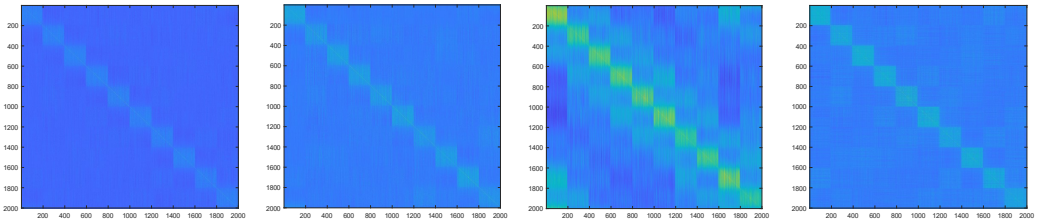


Figure 4: Comparison of varying affinity matrices learned by nine multi-view clustering algorithms on BBCsports dataset.



(a) UCI: $Z^{(1)}$ (b) UCI: $Z^{(2)}$ (c) UCI: $Z^{(3)}$ (d) UCI: A

Figure 5: Visualizations of affinity matrices of different views and the consistent subspace on UCI dataset.

realizes the best performance, which confirms that the preservation of local structure and the fusion on Grassmann manifold are necessary and valid. Notably, If we perform the averaging fusion for multiple subspace representations $\{\mathbf{Z}^{(v)}\}_{v=1}^{m}$ to obtain the consistent affinity representation $\mathbf{A}$, i.e., removing the second, the third, and the fourth terms in Eq. (10), the proposed CARLDLC degenerates to t-SVD-MSC model. Thus, we can see that CARLDLC outperforms t-SVD-MSC on six datasets from Tables 3, 4, 5, 6, 7, 8.

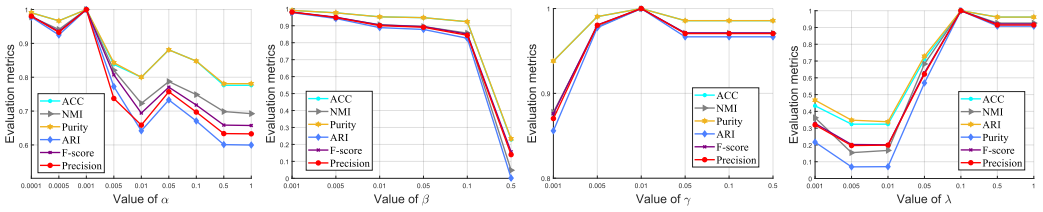
Table 9: Comparison of clustering results (ACC/NMI, %) of CARLDLC and its three variants.

Datasets	BBCsports	Citeseer	MSRC	UCI	WikipediaArticles	Youtube
CARLDLC-OL	95.6/86.3	67.5/40.1	98.6/97.1	94.3/90.0	34.6/18.6	32.3/18.2
CARLDLC-OG	95.8/86.6	42.0/18.2	96.7/92.4	93.7/93.9	52.8/44.6	30.7/19.0
CARLDLC-LF	91.9/80.9	53.8/27.1	54.3/41.9	95.0/90.1	55.3/49.0	31.0/17.1
CARLDLC	**97.1/90.1**	**67.8/41.4**	**1.0/1.0**	**97.9/95.0**	**57.3/53.3**	**33.0/21.4**

5.5. Model Analysis

(1) Parameter selection: In the objective function Eq. (10), there are four hyperparameters used to balance the five terms, including α, β, γ, and λ. To investigate the effect of each parameter on the proposed CARLDLC, we record the clustering results with different values of each parameter when the other parameters are fixed. Due to space limitations, Fig. 6 shows the performance of CARLDLC with varying parameter settings on the MSRC dataset. Specifically, α ranges in {1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 5e-2, 1e-1, 5e-1, 1} with fixed $\beta = 0.002$, $\gamma = 0.01$, and $\lambda = 0.1$. β ranges in {1e-3, 5e-3, 1e-2, 5e-2, 1e-1, 5e-1} with fixed $\alpha = 0.001$, $\gamma = 0.01$, and $\lambda = 0.1$. γ is tuned in {1e-3, 5e-3, 1e-2, 5e-2, 1e-1, 5e-1} while keeping $\alpha = 0.001$, $\beta = 0.002$, and $\lambda = 0.1$. λ varies in {1e-3, 5e-3,

1e-2, 5e-2, 1e-1, 5e-1, 1} with setting $\alpha = 0.001$, $\beta = 0.002$, and $\gamma = 0.01$. It can

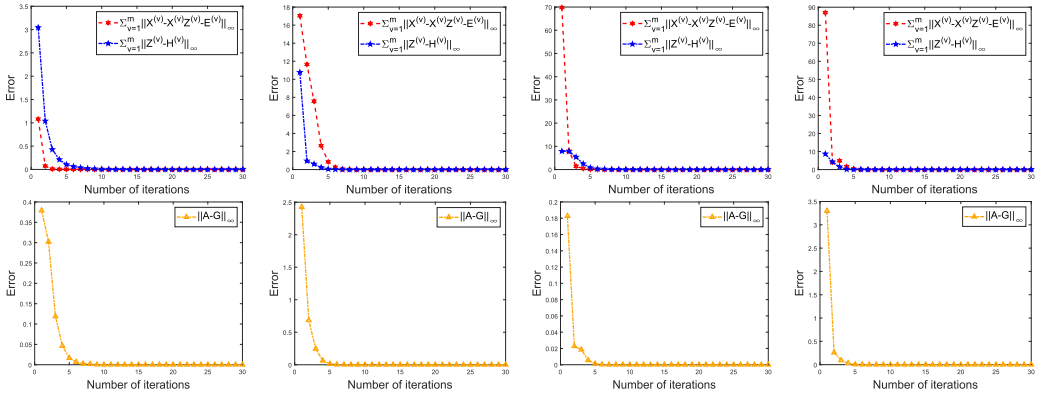


(a) α (b) β (c) γ (d) λ

Figure 6: Clustering performance with different values of parameters α, β, γ, and λ on the MSRC dataset.

be seen that the values of these hyperparameters have impressive effects on the performance of CARLDLC, which means that the settings of these hyperparameters are necessary. Overall, the proposed CARLDLC performs well when α, β, γ, and λ range in [1e-4, 1e-3], [1e-3, 1e-1], [5e-3, 5e-1], and [1e-1, 1], respectively.

(2) Convergence analysis: Fig. 7 displays the convergence curves on the datasets BBCsports, Citeseer, MSRC, and WikipediaArticles, where the blue line indicates the error defined as $\sum_{v=1}^{m} ||\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}||_{\infty}$, the red line represents the error calculated by $\sum_{v=1}^{m} ||\mathbf{Z}^{(v)} - \mathbf{H}^{(v)}||_{\infty}$, and the yellow line denotes the error defined as $||\mathbf{A} - \mathbf{G}||_{\infty}$. The values of these three error terms drop very quickly and tend to be stable in 15 iterations, which suggests that the proposed CARLDLC exhibits good convergence properties.



(a) BBCsports (b) Citeseer (c) MSRC (d) WikipediaArticles

Figure 7: Convergence curves on datasets BBCsports, Citeseer, MSRC, and WikipediaArticles.

(3) Running time comparison: The average running time of various multi-view clustering algorithms on six datasets is presented in Fig. 8. For the sake of fairness, all experiments are run in Matlab 2018b equipped on a personal computer with 3.00GHz Intel Core i5-9500 CPU and 24GB RAM. It can be found that the models GMC, LMVSC, and CoMSC are most efficient in terms of operation, especially for LMVSC, the introduction of anchor graph technology turns out to a linear computational complexity and can dramatically decreases the running time. It is worth noting that MCDCF and LTMSC require more time to implement on the larger datasets such as Citeseer, UCI, and Youtube, probably because MCDCF performs a multi-layer concept factorization, and LTMSC uses the SNN based TNN, thus making them less efficient. Furthermore, the operational efficiency of our method is moderate, but it is acceptable on the basis of achieving excellent clustering results.

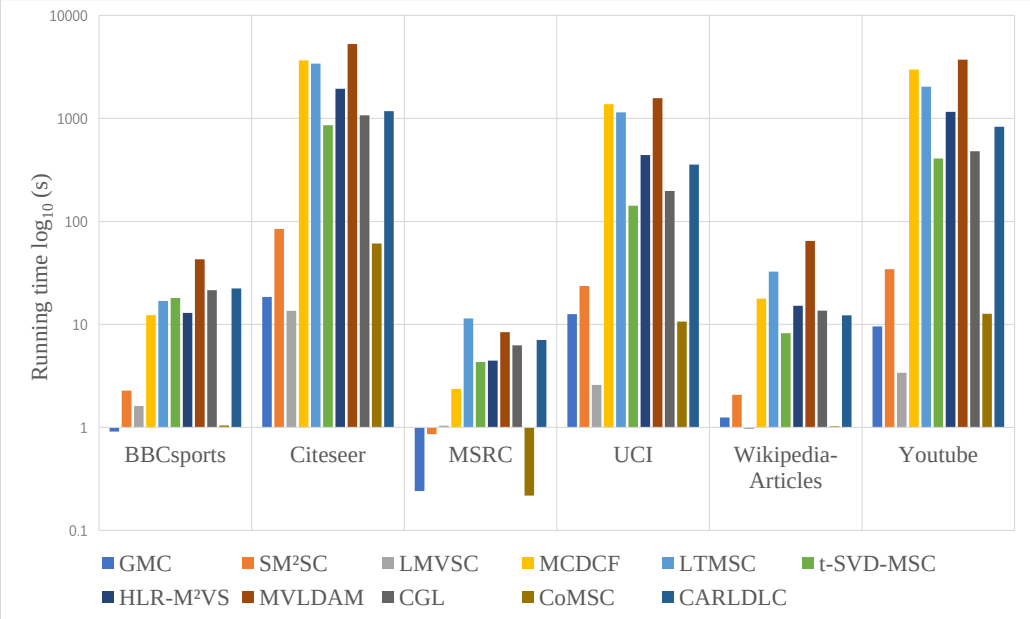


Figure 8: Comparison of running time of various multi-view clustering methods on six datasets.

6. Conclusion

In this paper, we propose a consistent affinity representation matrix learning method with dual low-rank constraints, which can be applied for multi-view sub-

space clustering and is termed as CARLDLC. To be specific, we assemble all self-representation matrices into a 3-order tensor. In pursuit of the global consensus, t-SVD based TNN is used to constrain the target tensor representation. Different from most multi-view clustering methods that fuse varying views in the Euclidean space, we integrate different views to obtain the consistent affinity matrix on the Grassmann manifold, which is further imposed with the low-rank constraint to enhance its cluster structure. We also pay attentions to preserving the local structure of the consistent subspace via graph regularization. For solving the proposed CARLDLC, a recursive optimization algorithm based on ADMM is developed. Moreover, a great deal of experiments are conducted to demonstrate the superiority of CARLDLC compared to other clustering methods. For the future work, considering the problem of possible missing data in multi-view data, we want to fill in the missing parts using low-rank tensor learning, so as to be suitable for downstream task.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (11801595), the Guang dong Basic and Applied Basic Research Foundation (2019A1515011043), the Natural Science Foundation of Guangdong (2018A030310076).

[1] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[2] Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43–54, 2017.

[3] Yu Wang, Chuan Chen, Jinrong Lai, Lele Fu, Yuren Zhou, and Zibin Zheng. A self-representation method with local similarity preserving for fast multi-view outlier detection. *ACM Transactions on Knowledge Discovery from Data*, 2022. url = https://doi.org/10.1145/3532191.

[4] Lele Fu, Pengfei Lin, Athanasios V Vasilakos, and Shiping Wang. An overview of recent multi-view clustering. *Neurocomputing*, 402:148–161, 2020.

[5] Ziheng Li, Zhanxuan Hu, Feiping Nie, Rong Wang, and Xuelong Li. Multi-view clustering based on generalized low rank approximation. *Neurocomputing*, 471:251–259, 2022.

[6] Sheng Huang, Yunhe Zhang, Lele Fu, and Shiping Wang. Learnable multi-view matrix factorization with graph embedding and flexible loss. *IEEE Transactions on Multimedia*, 2022. doi=10.1109/TMM.2022.3157997.

[7] Yuan Xie, Wensheng Zhang, Yanyun Qu, Longquan Dai, and Dacheng Tao. Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning. *IEEE Transactions Cybernetics*, 50(2):572–586, 2020.

[8] Lele Fu, Zhaoliang Chen, Sujia Huang, Sheng Huang, and Shiping Wang. Multi-view learning via low-rank tensor optimization. In *2021 IEEE International Conference on Multimedia and Expo*, pages 1–6, 2021.

[9] Shiping Wang, Lele Fu, Zhewen Wang, Haiping Xu, and William Zhu. Multigraph random walk for joint learning of multiview clustering and semisupervised classification. *IEEE Transactions on Computational Social Systems*, 9(3):926–939, 2022.

[10] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–594, 2015.

[11] Changqing Zhang, Huazhu Fu, Qinghua Hu, Xiaochun Cao, Yuan Xie, Dacheng Tao, and Dong Xu. Generalized latent multi-view subspace clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1):86–99, 2020.

[12] Guoli Niu, Youlong Yang, and Liqin Sun. One-step multi-view subspace clustering with incomplete views. *Neurocomputing*, 438:290–301, 2021.

[13] Xiaoqian Zhang, Zhenwen Ren, Huaijiang Sun, Keqiang Bai, Xinghua Feng, and Zhigui Liu. Multiple kernel low-rank representation-based robust multi-view subspace clustering. *Information Sciences*, 551:324–340, 2021.

[14] Qianqian Wang, Jiafeng Cheng, Quanxue Gao, Guoshuai Zhao, and Licheng Jiao. Deep multi-view subspace clustering with unified and discriminative learning. *IEEE Transactions on Multimedia*, 23:3483–3493, 2021.

[15] Changqing Zhang, Huazhu Fu, Si Liu, Guangcan Liu, and Xiaochun Cao. Low-rank tensor constrained multiview subspace clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1582–1590, 2015.

[16] Yuan Xie, Dacheng Tao, Wensheng Zhang, Yan Liu, Lei Zhang, and Yanyun Qu. On unifying multi-view self-representations for clustering by tensor multi-rank minimization. *International Journal of Computer Vision*, 126(11):1157–1179, 2018.

[17] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[18] Miaomiao Cheng, Liping Jing, and Michael K. Ng. Tensor-based low-dimensional representation learning for multi-view clustering. *IEEE Transactions on Image Processing*, 28(5):2399–2414, 2019.

[19] Yongyong Chen, Shuqin Wang, Chong Peng, Zhongyun Hua, and Yicong Zhou. Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering. *IEEE Transactions on Image Processing*, 30:4022–4035, 2021.

[20] Haiyan Wang, Guoqiang Han, Junyu Li, Bin Zhang, Jiazhou Chen, Yu Hu, Chu Han, and Hongmin Cai. Learning task-driving affinity matrix for accurate multi-view clustering through tensor subspace learning. *Information Sciences*, 563:290–308, 2021.

[21] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. Clustering on multi-layer graphs via subspace analysis on grassmann manifolds. *IEEE Transactions on Signal Processing*, 62(4):905–918, 2014.

[22] Hao Ding, Michael Sharpnack, Chao Wang, Kun Huang, and Raghu Machiraju. Integrative cancer patient stratification via subspace merging. *Bioinform.*, 35(10):1653–1659, 2019.

[23] Wentao Rong, Enhong Zhuo, Hong Peng, Jiazhou Chen, Haiyan Wang, Chu Han, and Hongmin Cai. Learning a consensus affinity matrix for multi-view clustering via subspaces merging on grassmann manifold. *Information Sciences*, 547:68–87, 2021.

[24] Tian Xia, Dacheng Tao, Tao Mei, and Yongdong Zhang. Multiview spectral embedding. *IEEE Transactions on Systems Man Cybernetics Part B*, 40(6):1438–1446, 2010.

[25] Feiping Nie, Jing Li, and Xuelong Li. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *Proceedings of the International Joint Conferences on Artificial Intelligence Organization*, pages 1881–1887, 2016.

[26] Zhanxuan Hu, Feiping Nie, Rong Wang, and Xuelong Li. Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding. *Information Fusion*, 55:251–259, 2020.

[27] S. El Hajjar, Fadi Dornaika, and Fahed Abdallah. Multi-view spectral clustering via constrained nonnegative embedding. *Information Fusion*, 78:209–217, 2022.

[28] Feiping Nie, Jing Li, and Xuelong Li. Self-weighted multiview clustering with multiple graphs. In *Proceedings of the International Joint Conferences on Artificial Intelligence Organization*, pages 2564–2570, 2017.

[29] Feiping Nie, Guohao Cai, and Xuelong Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2408–2414, 2017.

[30] Guangqi Jiang, Huibing Wang, Jinjia Peng, Dongyan Chen, and Xianping Fu. Graph-based multi-view binary learning for image clustering. *Neurocomputing*, 427:225–237, 2021.

[31] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. Large-scale multi-view subspace clustering in linear time. In *Proceedings of the International Joint Conferences on Artificial Intelligence Organization*, pages 4412–4419, 2020.

[32] Qianyao Qiang, Bin Zhang, Fei Wang, and Feiping Nie. Fast multi-view discrete clustering with anchor graphs. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 9360–9367, 2021.

[33] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. Large-scale multi-view spectral clustering via bipartite graph. In Blai Bonet and Sven Koenig, editors, *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2750–2756, 2015.

[34] Shudong Huang, Zenglin Xu, Ivor W. Tsang, and Zhao Kang. Auto-weighted multi-view co-clustering with bipartite graphs. *Information Sciences*, 512:18–30, 2020.

[35] Xuelong Li, Han Zhang, Rong Wang, and Feiping Nie. Multi-view clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. doi:10.1109/TPAMI.2020.3011148.

[36] Xiaoqian Zhang, Jing Wang, Xuqian Xue, Huaijiang Sun, and Jiangmei Zhang. Confidence level auto-weighting robust multi-view subspace clustering. *Neurocomputing*, 475:38–52, 2022.

[37] Jipeng Guo, Yanfeng Sun, Junbin Gao, Yongli Hu, and Baocai Yin. Low rank representation on product grassmann manifolds for multi-view subspace clustering. In *Proceedings of the International Conference on Pattern Recognition*, pages 907–914, 2021.

[38] Haiyan Wang, Guoqiang Han, Bin Zhang, Guihua Tao, and Hongmin Cai. Multi-view learning a decomposable affinity matrix via tensor self-representation on grassmann manifold. *IEEE Transactions on Image Processing*, 30:8396–8409, 2021.

[39] Peiguang Jing, Yuting Su, Zhengnan Li, and Liqiang Nie. Learning robust affinity graph representation for multi-view clustering. *Information Sciences*, 544:155–167, 2021.

[40] GuangYu Zhang, YuRen Zhou, XiaoYu He, ChangDong Wang, and Dong Huang. One-step kernel multi-view subspace clustering. *Knowledge-Based Systems*, 189:105126, 2020.

[41] Lecheng Zheng, Yu Cheng, Hongxia Yang, Nan Cao, and Jingrui He. Deep co-attention network for multi-view subspace learning. In *Proceedings of the International Conference of World Wide Web*, pages 1528–1539, 2021.

[42] Runkun Lu, Jianwei Liu, and Xin Zuo. Attentive multi-view deep subspace clustering net. *Neurocomputing*, 435:186–196, 2021.

[43] Alan Edelman, Tomás A. Arias, and Steven Thomas Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[44] Misha E Kilmer, Karen Braman, Ning Hao, and Randy C Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013.

[45] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2006.

[46] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 977–986, 2014.

[47] R. H. Bartels and G. W. Stewart. Solution of the matrix equation ax + xb = c [f4]. *Communications of the ACM*, 15(9):820826, 1972.

[48] Wenrui Hu, Dacheng Tao, Wensheng Zhang, Yuan Xie, and Yehui Yang. The twist tensor nuclear norm for video completion. *IEEE Transactions on Neural Networks and Learning Systems*, 28(12):2961–2973, 2016.

[49] JianFeng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Matrix Analysis and Applications*, 20(4):1956–1982, 2010.

[50] Yongyong Chen, Xiaolin Xiao, Chong Peng, Guangming Lu, and Yicong Zhou. Low-rank tensor graph learning for multi-view subspace clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1):92–104, 2022.

[51] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[52] H. Wang, Y. Yang, and B. Liu. Gmc: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1116–1129, 2020.

[53] Xiaobo Wang, Zhen Lei, Xiaojie Guo, Changqing Zhang, Hailin Shi, and Stan Z Li. Multi-view subspace clustering with intactness-aware similarity. *Pattern Recognition*, 88:50–63, 2019.

[54] Shuai Chang, Jie Hu, Tianrui Li, Hao Wang, and Bo Peng. Multi-view clustering via deep concept factorization. *Knowledge-Based Systems*, 217:106807, 2021.

[55] Zhenglai Li, Chang Tang, Xinwang Liu, Xiao Zheng, Guanghui Yue, Wei Zhang, and En Zhu. Consensus graph learning for multi-view clustering. *IEEE Transactions on Multimedia*, 2021. doi=10.1109/TMM.2021.3081930.

[56] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Xifeng Guo, Marius Kloft, and Liangzhong He. Multiview subspace clustering via co-training robust data representation. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. doi=10.1109/TNNLS.2021.3069424.