This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2021.3055625, IEEE Transactions on Circuits and Systems for Video Technology

Low-Rank Tensor Graph Learning for Multi-view Subspace Clustering

Yongyong Chen, Xiaolin Xiao, Chong Peng, Guangming Lu, and Yicong Zhou, *Senior Member, IEEE,*

***Abstract*—Graph and subspace clustering methods have become the mainstream of multi-view clustering due to their promising performance. However, (1) since graph clustering methods learn graphs directly from the raw data, when the raw data is distorted by noise and outliers, their performance may seriously decrease; (2) subspace clustering methods use a "two-step" strategy to learn the representation and affinity matrix independently, and thus may fail to explore their high correlation. To address these issues, we propose a novel multi-view clustering method via learning a Low-Rank Tensor Graph (LRTG). Different from subspace clustering methods, LRTG simultaneously learns the representation and affinity matrix in a single step to preserve their correlation. We apply Tucker decomposition and $l_{2,1}$-norm to the LRTG model to alleviate noise and outliers for learning a "clean" representation. LRTG then learns the affinity matrix from this "clean" representation. Additionally, an adaptive neighbor scheme is proposed to find the K largest entries of the affinity matrix to form a flexible graph for clustering. An effective optimization algorithm is designed to solve the LRTG model based on the alternating direction method of multipliers. Extensive experiments on different clustering tasks demonstrate the effectiveness and superiority of LRTG over seventeen state-of-the-art clustering methods.**

***Index Terms*—Multi-view clustering, low-rank, tensor approximation, graph learning**

I. INTRODUCTION

Subspace clustering has become an important tool to discover the underlying structure of high-dimensional data [1]. It aims to simultaneously group the data points into their essential clusters and find a low-dimensional subspace representation [1–4]. To yield the representation matrix with the block diagonal property, many works considered different regularizers, such as sparsity [5], low-rankness [6], smooth representation [7], and block diagonal representation [8] under the assumption that high-dimensional data can be modeled as samples drawn from the union of multiple low-dimensional subspaces.

This work was funded by the Science and Technology Development Fund, Macau SAR (File no. 189/2017/A3), and by University of Macau (File no. MYRG2018-00136-FST). (Corresponding author: Yicong Zhou)

Y. Chen and G. Lu are with the Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China. Y. Chen is also with the Shenzhen Key Laboratory of Visual Object Detection and Recognition, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China (Emails: YongyongChen.cn@gmail.com and luguangm@hit.edu.cn).

X. Xiao is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (Email: shellyxiaolin@gmail.com).

C. Peng is with the College of Computer Science and Technology, Qingdao University, Qingdao 266071, China (Email: pchong1991@163.com).

Y. Zhou is with the Department of Computer and Information Science, University of Macau, Mac266071au 999078, China (Email: yicongzhou@um.edu.mo).

With the advance of technology, it has been increasingly common to capture multi-view data for dimension reduction [9], outlier detection [10], subspace learning [11, 12], 3D position estimation [13], and vehicle re-identification [14]. For example, face images with different view features such as color, textures and edges greatly contribute to high recognition rate; Action sequences could be characterized by RGB, depth, thermal and skeleton sensors. Therefore, how to design effective methods for multi-view clustering has attracted research attention in recent years [15–18].

A large number of methods have been proposed for multi-view clustering, which can be roughly classified into four categories: co-training style algorithms, multi-kernel learning, multi-view graph clustering, and multi-view subspace clustering (MVSC). Due to their promising performance and easy understanding, multi-view graph clustering [19–21] and MVSC methods [16, 17, 22, 23] have become popular. By representing each data point as a vertex and the pairwise similarity by edges, the clustering task is transformed into a graph partition problem [19, 24, 25], and the quality of the constructed graph dominates the clustering performance. The works in [25, 26] constructed a graph for each view independently and then learned a unified affinity matrix shared by all graphs. Nie *et al.* [27] proposed to learn the similarity matrix and clustering structure simultaneously, while it is not flexible to handle noise or outliers. However, these graph clustering methods directly construct the graph from the raw data which may be easily distorted by noise and outliers. Kang *et al.* [28] used the adaptive neighbor idea to build a low-rank graph by the low-rank and sparse decomposition. When some views of a sample are absent [29], many incomplete multi-view clustering methods [30, 31] based on graph learning have been proposed. Similar ideas *i.e.*, learning one graph with more consistency and complementary information were exploited to improve clustering performance. The authors in [32] used the fundamental sampling patterns to complete the multi-view data. The work in [29] used group sparsity and alternation for subspace clustering with missing data.

Two common schemes have been explored to extend the existing single-view subspace clustering methods into the multi-view setting: (1) learn a common latent space shared by all views to explore the consistency; (2) introduce a low-rank tensor constraint to capture the high order correlations among multiple views. Following the first line, Xia *et al.* [20] proposed to decompose pre-defined graphs into a shared low-rank transition probability matrix and a sparse error matrix. One shortcoming of the first line is that this category lies in the lose of view-specific information [16, 17]. Unlike the first line, the studies in [15–18] investigated the low-rank tensor representation to fit the multi-view clustering. To well capture the high order correlations among multiple views, Zhang *et al.* [16] proposed to store each representation matrix as a slice of a third-order tensor, called representation tensor, and then imposed the low-rank tensor constraint on it. Following [16], the work in [17] used the tensor multi-rank to obtain the representation tensor with clear physical meaning. Wu *et al.* [18] developed an advanced version of [20] by learning an essential tensor for multi-view clustering. In summary, under the self-expressiveness assumption, most existing MVSC approaches followed a "two-step" strategy, *i.e.*, first exploring different regularizers to impose specific structural constraints on the representation matrix, and then constructing the graph by the learned representation matrices. Consequently, the learned affinity matrix may be sub-optimal and have difficulty in capturing the true relationship among all data points. Two recent methods in [24, 33] have addressed this limitation. However, these methods focus on single-view clustering and ignore the informative multi-view features.

To overcome the above limitations, we propose a Low-Rank Tensor Graph (LRTG) for multi-view subspace clustering. The main idea of LRTG is to learn an adaptive affinity matrix from the "clean" representation tensor instead of the contaminated raw data. The representation tensor is imposed with the low-rank property by the Tucker decomposition. The main contributions of this work can be summarized as follows:

- We propose LRTG as a unified model to learn a low-rank tensor graph for multi-view subspace clustering. LRTG is able to simultaneously learn the representation and affinity matrix in one single step. It is also robust to noise and outliers.
- We use Tucker decomposition and $l_{2,1}$-norm to the LRTG model to explore the low-rank property and remove noise and outliers, respectively. A "clean" representation is then obtained to learn the affinity matrix.
- An adaptive neighbor scheme is proposed to the LRTG model to find the K largest entries of the affinity matrix and then obtain a flexible graph for clustering.
- To solve LRTG, we introduce an effective optimization algorithm based on the alternating direction method of multipliers. Extensive experiments on several benchmark databases demonstrate the effectiveness and superiority of the proposed LRTG over seventeen state-of-the-art methods.

The remainder of this paper is organized as follows. Section II briefly discusses related work. In Section III, we introduce the proposed LRTG model and solve this model by the alternating direction method of multipliers. The experimental results and model analysis are provided in Section IV. Finally, Section V concludes this paper.

II. Related Work and Preliminaries

In this section, we briefly review the closely related work including multi-view graph clustering and MVSC. Notations are listed in Table I. Finally, we review Tucker decomposition which is used to describe the low-rank tensor property of the representation tensor.

TABLE I
Explanation of notation in this paper.

Notation	Meaning
$\mathcal{X}$, X, x	tensor, matrix, vector
n	the number of samples
V	the number of views
d_v	feature dimension of the v-th view
$X^v \in \mathbb{R}^{d_v \times n}$	feature matrix of the v-th view
$\mathcal{Z} \in \mathbb{R}^{n \times n \times V}$	representation tensor
$A \in \mathbb{R}^{n \times n}$	affinity matrix
$E^{(v)} \in \mathbb{R}^{d_v \times n}$	corruption
$\|\cdot\|_{2,1}$	the $l_{2,1}$-norm
$\|\cdot\|_F$	the Frobenius norm
$\|\cdot\|_\infty$	the infinity norm
$\mathbf{1}$	vector with all entries as 1
$Z_{(m)}$	mode-m unfolding
$\times_m$	mode-m product

A. Multi-view Graph Clustering

Multi-view graph clustering methods directly use the raw data to construct the affinity matrix. For example, k-nearest neighbor [34] used cosine or heat kernel distances to measure the similarity. Other graph construction methods include local linear similarity graph [35] and local discriminant graph [36]. Nie *et al*. [19] and Zhan *et al*. [26] first generate an initial graph for each view and then learn a unified representation for clustering. By combining the above two steps into a single one, the work in [19] proposed an auto-weighted multi-view learning method. Inspired by the essential connection between spectral clustering and Markov chain, Xia *et al*. [20] and Wu *et al*. [18] exploited robust principal component analysis for multi-view clustering from the matrix and tensor aspects, respectively.

B. Multi-view Subspace Clustering

Inspired by the fact that the high-dimensional data usually lie in a union of several low-dimensional subspaces, subspace clustering is to simultaneously cluster the data points into multiple subspaces and find a low-dimensional subspace to fit each group [6, 37]. The representation matrix Z can be generally learned from:

$$\begin{aligned} &\min_{Z,E} \mathcal{R}(Z) + \alpha \|E\|_l \\ &s.t.\ X = XZ + E,\ diag(Z) = 0, \end{aligned} \tag{1}$$

where X is the feature matrix and E denotes noise. The l-norm is specific with respect to different types of noise. The main difference lies in the different regularizers $\mathcal{R}$. SSC [5] used the l_1-norm ($\mathcal{R}(Z) = \|Z\|_1$) while LRR used the nuclear norm ($\mathcal{R}(Z) = \|Z\|_*$). Zhang *et al*. proposed the block-diagonal adaptive locality-constrained representation [4] and adaptive structure-constrained robust latent low-rank coding [38], respectively. To improve the representation ability, the mutual-manifold regularized robust fast latent LRR [39] was developed. Jia *et al*. [40] proposed to learn the graph and the clustering result simultaneously and the dissimilarity propagation-guided graph-Laplacian principal component analysis [41]. They also proposed a semi-supervised spectral clustering method [42] based on the structured sparsity regularization. Due to the superior power of subspace clustering [30], Zhang *et al.* [16] and Xie *et al.* [17] extended the single-view subspace clustering in Eq. (1) into multi-view setting by the following model:

$$\begin{aligned} &\min_{\mathcal{Z},E} \mathcal{R}(\mathcal{Z}) + \alpha \sum_{v=1}^{V} \|E^v\|_{2,1} \\ &s.t.\ X^v = X^v Z^v + E^v, v = 1, 2, \cdots, V, \\ &\mathcal{Z} = \Phi(Z^1, Z^2, \cdots, Z^V), \end{aligned} \tag{2}$$

where V is the number of views. The last row of Eq. (2) aims to construct $\mathcal{Z}$ by storing each representation matrix Z^v as a frontal slice. The regularizer $\mathcal{R}$ in Eq. (2) usually differs from the one in Eq. (1). This is because variable $\mathcal{Z}$ has a new dimension (the view dimension) over variable Z in Eq. (1) [43]. After the representation tensor $\mathcal{Z}$ is learned from Eq. (2), the affinity matrix is constructed by $A = \frac{1}{V}\sum_{v=1}^{V}\left(|Z^v| + |Z^{v^T}|\right)$. Many MVSC methods [16, 17, 23, 44] follow the "two-step" strategy to construct the affinity matrix. For example, the tensorial t-product representation was developed in [45] to solve the MVSC in the third-order tensor space. The works in [46, 47] solved MVSC by the low-rank matrix factorization and tailored tensor low-rank representation, respectively. Recently, several deep multi-view subspace clustering methods have been developed. For example, the work in [48] used the convolution neural networks for multi-modal subspace clustering while the study in [49] integrated the latent representation [50] with auto-encoder framework for MVSC. Zhang *et al.* [51] proposed the deep partial multi-view networks to handle the missing multi-view features. Yin *et al.* [52] proposed to learn a shared generative latent representation for MVSC. Xie *et al*. proposed the deep multi-view joint clustering framework [53].

C. Tucker Decomposition

Definition 1 (Mode-m unfolding) [54] The mode-m unfolding (also known as matricization) of tensor $\mathcal{X}$ is a matrix denoted by $X_{(m)}$, whose entries are obtained by arranging (lexicographically in the indices other than the m-th index) the mode-m fibers as the columns of the matrix. $X_{(m)} \in \mathbb{R}^{i_m \times \Pi_{m^* \neq m} i_{m^*}}$ is the mode-m unfolding of the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$.

Definition 2 (Mode-m product) [54] A mode-m product of $\mathcal{Z} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ and $U \in \mathbb{R}^{J_m \times I_m}$ is denoted by $\mathcal{Y} = \mathcal{Z} \times_m U \in \mathbb{R}^{I_1 \times \cdots \times I_{m-1} \times J_m \times I_{m+1} \cdots \times I_M}$, with entries given by

$$\mathcal{Y}_{i_1,\cdots,i_{m-1},j_m,i_{m+1},\cdots,i_M} = \sum_{i_m} \mathcal{Z}_{i_1,\cdots,i_{m-1},i_m,i_{m+1},\cdots,i_M} U_{i_m,j_m}, \tag{3}$$

and $Y_{(m)} = U * Z_{(m)}$. Here $*$ denotes the multiplication.

Definition 3 (Tucker decomposition) Given a tensor $\mathcal{Z} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$, its Tucker decomposition is defined as the multiplication of a core tensor and M factor matrices, *i.e.*,

$$\mathcal{Z} = \mathcal{C} \times_1 U_1 \times_2 U_2 \cdots \times_M U_M, \tag{4}$$

where $\mathcal{C} \in \mathbb{R}^{R_1 \times \cdots \times R_M}$ is the core tensor with lower dimension, and $\{U_m \in \mathbb{R}^{R_m \times I_m}, m = 1, \cdots, M$, and $R_m \leq I_m\}$ are factor matrices with orthonormal columns.

III. Proposed LRTG Algorithm

In this section, we introduce the LRTG method. LRTG learns the affinity matrix A from the "clean" representation tensor $\mathcal{Z}$ which is encoded by Tucker decomposition. In addition, an adaptive-neighbor strategy is adopted to seek K largest neighbors for constructing the affinity matrix. Based on the alternating direction method of multipliers, an effective algorithm is designed for solving LRTG in Section III-B.

A. The Proposed LRTG

Assume that there are V multiple views $\{X^v \in \mathbb{R}^{d_v \times n}\}$, where d_v is the feature dimension of the v-th feature and n is the number of data points. For multi-view graph clustering methods, they usually compute the affinity matrix A by using the following model:

$$\begin{aligned} &\min_{A} \sum_{v=1}^{V}\sum_{i,j}^{n} \|X_i^v - X_j^v\|_2^2 a_{ij} + \beta\|A\|_F^2 \\ &s.t.\ A^T\mathbf{1} = \mathbf{1},\ \mathbf{0} \leq A \leq \mathbf{1}, \end{aligned} \tag{5}$$

where a_{ij} is the (i, j)-th entry of A. $\mathbf{1}$ and $\mathbf{0}$ represent the column vectors with all entries as 1 and 0, respectively. Raw data are directly used to pursue the affinity matrix. However, the raw data are inevitably corrupted by noise and outliers [6, 55, 56], which would cause compromised performance. Inspired by the noise robustness of subspace clustering methods, we learn the affinity matrix from the representation tensor instead of the raw data. That is, the closeness of original data points is measured by the relationship of their representation coefficients [33, 57]. The proposed LRTG can be formulated as follows:

$$\left\{\begin{aligned} &\min_{\mathcal{Z},E,A} \|E\|_{2,1} + \alpha \sum_{v=1}^{V} tr\big(Z^{v^T} L_A Z^v\big) + \beta\|A\|_F^2 \\ &s.t.\ X^v = X^v Z^v + E^v, \quad v = 1, 2, \cdots, V, \\ &\quad \mathcal{Z} = \Phi(Z^1, Z^2, \cdots, Z^V), \\ &\quad \mathcal{Z} = \mathcal{C} \times_1 U_1 \times_2 U_2 \times_3 U_3,\ U_i^T * U_i = I, (i = 1, 2, 3), \\ &\quad E = [E^1; E^2; \cdots; E^V],\ A^T\mathbf{1} = \mathbf{1},\ \mathbf{0} \leq A \leq \mathbf{1}, \end{aligned}\right\} \tag{6}$$

where $L_A = D - (A + A^T)/2$ is the graph Laplacian matrix of A; D is a diagonal matrix whose i-th diagonal entry is $\sum_j (a_{ij} + a_{ji})/2$. $\Phi(\cdot)$ merges all representation matrices $\{Z^v\}$ to a 3-order tensor $\mathcal{Z}$, named representation tensor.

- Due to the simplicity, we investigate Tucker decomposition to describe the low-rank property of $\mathcal{Z}$, that is, $\mathcal{Z} = \mathcal{C} \times_1 U_1 \times_2 U_2 \times_3 U_3$. $\mathcal{C} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is the core tensor and U_i $(i = 1, 2, 3)$ is the orthogonal factor matrix. $R_1, R_2 \ll n$ and $R_3 \leq V$ are used to insure the low-rank property of $\mathcal{Z}$;
- Eq. (6) uses Tucker decomposition and $l_{2,1}$-norm to generate a "clean" $\mathcal{Z}$. Then, we learn the affinity matrix A from $\mathcal{Z}$, *i.e.*, $\sum_{v=1}^{V} tr\big(Z^{v^T} L_A Z^v\big) = \sum_{v=1}^{V}\sum_{i,j}^{n} \|Z_i^v - Z_j^v\|_2^2 a_{ij}$;
- A "good" affinity matrix should have the following property: high intra-cluster similarity and low inter-cluster similarity. Thus, we adopt the adaptive neighbor scheme

to make coefficients over intra-class data points larger than those over inter-class data points;
- Unlike previous subspace clustering methods, LRTG simultaneously learns $\mathcal{Z}$ and A in one single step such that their correlation can be well preserved.
- The proposed LRTG is different from the work in [58]. The graph Laplacian matrix L_A in LRTG is learned from $\mathcal{Z}$ while that of [58] is constructed in the k-nearest neighbor fashion [57]. Thus, L_A in LRTG is adaptive while that of [58] is fixed. On the other hand, the probability property *i.e.*, $A^T\mathbf{1} = \mathbf{1},\ \mathbf{0} \le A \le \mathbf{1}$, cannot be guaranteed in [58].

By combining the above aspects, the proposed LRTG in Eq. (6) can yield a flexible graph for multi-view clustering.

B. Optimization of LRTG

Here we use the alternating direction method of multipliers (ADMM) to solve the proposed LRTG model in Eq. (6). Since variable $\mathcal{Z}$ is coupled with the objective function and three constraints, we introduce one auxiliary variable $\mathcal{Y}$ to make $\mathcal{Z}$ separable and further transform Model (6) into the following model:

$$\begin{aligned}
&\min_{\mathcal{Z},\mathcal{Y},E,A} \|E\|_{2,1} + \alpha \sum_{v=1}^{V} tr\big(Y^{v^T} L_A Y^v\big) + \beta \|A\|_F^2 \\
&s.t.\ X^v = X^v Y^v + E^v, \quad v = 1, 2, \cdots, V, \\
&\quad \mathcal{Z} = \Phi(Z^1, Z^2, \cdots, Z^V), \\
&\quad \mathcal{Z} = \mathcal{C} \times_1 U_1 \times_2 U_2 \times_3 U_3,\ U_i^T * U_i = I, (i = 1, 2, 3), \\
&\quad E = [E^1; E^2; \cdots; E^V],\ A^T\mathbf{1} = \mathbf{1},\ \mathbf{0} \le A \le \mathbf{1}, \\
&\quad \mathcal{Z} = \mathcal{Y}.
\end{aligned} \tag{7}$$

To cooperate with ADMM, we firstly adopt the variable splitting technique by introducing an auxiliary variable $\mathcal{Y}$ to make $\mathcal{Z}$ decoupled [59]. Borrowing the idea of ADMM, the constraint problem (7) is solved by minimizing the following augmented Lagrangian function of Eq. (7):

$$\begin{aligned}
&\hbar(\mathcal{Z}, \mathcal{Y}, E, A; \Theta, \Pi) = \|E\|_{2,1} + \beta \|A\|_F^2 + \\
&\sum_{v=1}^{V} \Big(\alpha tr\big(Y^{v^T} L_A Y^v\big) + \langle \Theta^v, X^v - X^v Y^v - E^v \rangle + \\
&\tfrac{\rho}{2} \|X^v - X^v Y^v - E^v\|_F^2\Big) + \langle \Pi, \mathcal{Z} - \mathcal{Y} \rangle + \tfrac{\rho}{2} \|\mathcal{Z} - \mathcal{Y}\|_F^2
\end{aligned} \tag{8}$$

where Θ and Π are Lagrangian multipliers corresponding to two equations. $\langle \cdot, \cdot \rangle$ denotes the inner product. ρ is called the penalty parameter.

To solve Eq. (8), we alternately update each variable by fixing the other variables. We first learn a low-rank representation tensor $\mathcal{Z}$ based on the efficient Tucker decomposition. Second, the auxiliary variable $\mathcal{Y}$ is updated by solving a convex quadratic problem. Third, the noise matrix and the affinity matrix are updated in parallel. Finally, two Lagrangian multipliers and the penalty parameter are updated. All subproblems are listed as follows:

Step 1 update $\mathcal{Z}$: Dropping out the irrelevant terms in Eq. (8), the optimal solution of $\mathcal{Z}$ in the $(t+1)$-th iteration can be obtained by solving

$$\begin{aligned}
&\mathcal{Z}_{t+1} = \operatorname*{argmin}_{\mathcal{Z}} \|\mathcal{Z} - (\mathcal{Y}_t - \frac{\Pi_t}{\rho_t})\|_F^2, \\
&s.t.\ \mathcal{Z} = \mathcal{C} \times_1 U_1 \times_2 U_2 \times_3 U_3,\ U_i' * U_i = I.
\end{aligned} \tag{9}$$

The above problem can be easily solved by the classic higher order orthogonal iteration algorithm [60] to obtain the core tensor $\mathcal{C}'$ and the orthogonal factor matrices $U_i',\ i = 1, 2, 3$ of tensor $\mathcal{Y}_t - \frac{\Pi_t}{\rho_t}$. Then, the low-rank representation tensor $\mathcal{Z}_{t+1}$ is obtained by

$$\mathcal{Z}_{t+1} = \mathcal{C}' \times_1 U_1' \times_2 U_2' \times_3 U_3'. \tag{10}$$

Step 2 update $\mathcal{Y}$: With the other variables fixed, $\mathcal{Y}$ is updated by

$$\begin{aligned}
\mathcal{Y}_{t+1} = \operatorname*{argmin}_{\mathcal{Y}} \sum_{v=1}^{V} \Big(\frac{\rho_t}{2} \|X^v - X^v Y^v - E_t^v + \frac{\Theta_t^v}{\rho_t}\|_F^2 + \\
\alpha\ tr\big(Y^{v^T} L_A Y^v\big)\Big) + \frac{\rho_t}{2} \|\mathcal{Z}_{t+1} - \mathcal{Y} + \frac{\Pi_t}{\rho_t}\|_F^2.
\end{aligned} \tag{11}$$

It is obvious that Eq. (11) is independent with respect to each Y^v, and thus can be separated into V subproblems. Specifically, the v-th subproblem is

$$\begin{aligned}
Y_{t+1}^v = \operatorname*{argmin}_{Y^v} \frac{\rho_t}{2} \|X^v - X^v Y^v - E_t^v + \frac{\Theta_t^v}{\rho_t}\|_F^2 + \\
\alpha\ tr\big(Y^{v^T} L_A Y^v\big) + \frac{\rho_t}{2} \|Z_{t+1}^v - Y^v + \frac{\Pi_t^v}{\rho_t}\|_F^2.
\end{aligned} \tag{12}$$

The above subproblem is convex with respect to Y^v. Therefore, by setting the derivative of Eq. (12) with respect to Y^v to zero, the closed-form solution Y_{t+1}^v is

$$\begin{aligned}
Y_{t+1}^v = \big(\rho_t (I + X^{v^T} X^v) + 2\alpha L_A\big)^{-1} \big(\rho_t Z_{t+1}^v + \\
\Pi_t^v + \rho_t X^{v^T} (X^v - E_t^v + \frac{\Theta_t^v}{\rho_t})\big).
\end{aligned} \tag{13}$$

Step 3 update E: To optimize E, we have

$$\begin{aligned}
E_{t+1} &= \operatorname*{argmin}_{E} \frac{1}{\rho_t} \|E\|_{2,1} + \frac{1}{2} \sum_{v=1}^{V} \|E^v - F_t^v\|_F^2, \\
&= \operatorname*{argmin}_{E} \frac{1}{\rho_t} \|E\|_{2,1} + \frac{1}{2} \|E - F_t\|_F^2,
\end{aligned} \tag{14}$$

where $F_t^v = X^v - X^v Y_{t+1}^v + \frac{\Theta_t^v}{\rho_t}$ and F_t is constructed by vertically concatenating the matrices $\{F^v\}$. The j-th column of E_{t+1} is

$$E_{t+1}(:, j) = \begin{cases} \frac{\|F_t(:,j)\|_2 - \frac{1}{\rho_t}}{\|F_t(:,j)\|_2} F_t(:, j), & \text{if } \frac{1}{\rho_t} < \|F_t(:, j)\|_2; \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Step 4 update A: By ignoring terms irrelevant of A, Eq. (8) can be reformulated as

$$\begin{aligned}
&A_{t+1} = \operatorname*{argmin}_{A} \sum_{v=1}^{V} \alpha\ tr\big(Y_{t+1}^{v^T} L_A Y_{t+1}^v\big) + \beta \|A\|_F^2, \\
&s.t.\ A^T\mathbf{1} = \mathbf{1},\ \mathbf{0} \le A \le \mathbf{1}.
\end{aligned} \tag{16}$$

Note that the graph Laplacian matrix L_A is defined as $D - (A + A^T)/2$. We rewrite Eq. (16) as[1]

$$\begin{aligned}
&A_{t+1} = \operatorname*{argmin}_{A} \sum_{j=1}^{n} \sum_{v=1}^{V} \frac{\alpha}{2} \|Y_i^v - Y_j^v\|_2^2 a_{ij} + \beta a_{ij}^2, \\
&s.t.\ a_i^T\mathbf{1} = 1,\ \mathbf{0} \le a_i \le \mathbf{1}.
\end{aligned} \tag{17}$$

[1]For the sake of simplicity, the iteration number t is omitted in the update of A.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2021.3055625, IEEE Transactions on Circuits and Systems for Video Technology

By denoting $g_{ij} = \sum_{v=1}^{V} \|Y_i^v - Y_j^v\|_2^2$ and $g_i \in \mathbb{R}^{n\times 1}$, Eq. (17) can be solved by a set of independently small-scale subproblems and the i-th problem is

$$\begin{aligned} a_i = \underset{a_i}{\operatorname{argmin}} \|a_i + \frac{\alpha}{4\beta} g_i\|_2^2 \\ s.t.\ a_i^T \mathbf{1} = 1,\ \mathbf{0} \le a_i \le \mathbf{1}. \end{aligned} \tag{18}$$

Using the Lagrangian method, we have the Lagrangian function of Eq. (18)

$$\|a_i + \frac{\alpha}{4\beta_i} g_i\|_2^2 - \eta(a_i^T \mathbf{1} - 1) - \gamma^T a_i, \tag{19}$$

where η and γ are the Lagrangian multipliers. According to the Karush-Kuhn-Tucker condition [61], we have $a_i = \max\{\frac{\eta}{2} - \frac{\alpha g_i}{4\beta_i}, 0\}$.

As stated in [62], a "good" affinity matrix should have the following property: high intra-cluster similarity and low inter-cluster similarity. This means that the coefficients over intra-class data points are larger than those over inter-class data points. Following [28, 33], we use the adaptive neighbor scheme to keep K largest entries in a_i and set the others to zero, that is, a_i has K positive entries and $a_{ij} = 0$ for $j > K$. Thus, we have

$$\begin{cases} a_{i,j} = \dfrac{\eta}{2} - \dfrac{\alpha g_{ij}}{4\beta_i} > 0, \quad j \le K & \text{(20a)} \\ a_{i,j} = \dfrac{\eta}{2} - \dfrac{\alpha g_{ij}}{4\beta_i} \le 0, \quad j > K & \text{(20b)} \\ a_i^T \mathbf{1} = \displaystyle\sum_{j=1}^{K} \Big(\frac{\eta}{2} - \frac{\alpha g_{ij}}{4\beta_i}\Big) = 1, & \text{(20c)} \end{cases}$$

and

$$\begin{cases} \eta = \dfrac{2}{K} - \dfrac{\alpha}{2K\beta_i} \displaystyle\sum_{j=1}^{K} g_{ij} & \text{(21a)} \\ \beta_i = \dfrac{\alpha}{4} \Big(K g_{i,K+1} - \displaystyle\sum_{j=1}^{K} g_{ij}\Big) & \text{(21b)} \\ a_{ij} = \dfrac{g_{i,K+1} - g_{ij}}{K g_{i,K+1} - \sum_{r=1}^{K} g_{ir}}. & \text{(21c)} \end{cases}$$

In Eqs. (21a), (21b) and (21c), we set β_i to its maximum. Eq. (21a) is obtained from Eq. (20c). Using Eqs. (20b) and (21a), we have Eq. (21b). By substituting η and β_i into Eq. (20a), we have Eq. (21a). It is worth noting that parameter β in Eq. (6) is determined by the number of adaptive neighbors. This means that we need only to pre-define parameter α and the number of neighbors K in Eq. (6).

Step 5 update Θ, Π and ρ: Fixing variables $\mathcal{Z}$, $\mathcal{Y}$, and E in the $(t+1)$-th iteration, the Lagrangian multipliers Θ, Π and penalty parameter ρ are updated by

$$\begin{aligned} \Theta_{t+1}^v &= \Theta_t^v + \rho_t (X^v - X^v Y_{t+1}^v - E_{t+1}^v); \\ \Pi_{t+1} &= \Pi_t + \rho_t (\mathcal{Z}_{t+1} - \mathcal{Y}_{t+1}); \\ \rho_{t+1} &= \min\{\lambda * \rho_t, \rho_{max}\}. \end{aligned} \tag{22}$$

where $\lambda > 1$ is to fasten the convergence speed [63]. ρ_{max} is the maximum value of ρ. For clarity, the whole optimization procedure of Eq. (7) is summarized in Algorithm 1. Once the affinity matrix A is yielded by Algorithm 1, A is input into the spectral clustering algorithm to yield the final clustering results.

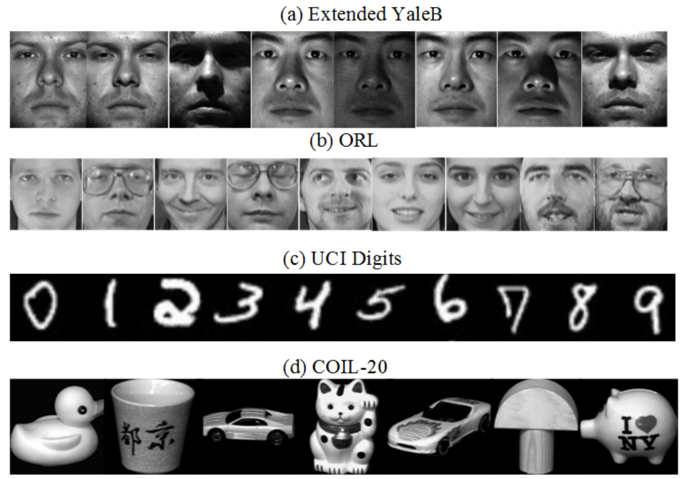


Fig. 1. Samples of databases. (a) Extended YaleB, (b) ORL, (c) UCI-Digits, and (d) COIL-20.

Algorithm 1 LRTG for multi-view subspace clustering

Input: multi-view features: $\{X^v\}$; parameter: α; the number of nearest neighbors K;

Initialize: $\mathcal{Y}$, $\mathcal{Z}$, E, A, Θ, Π initialized to $\mathbf{0}$; $\rho = 1$, $\lambda = 1.5$, $\epsilon = 10^{-7}$, iteration $t = 0$;

while not converged **do**
  Update $\mathcal{Z}_{t+1}$ by Eq. (10);
  for $v = 1$ to V **do**
    Update Y_{t+1}^v by Eq. (13);
  end for
  Update E_{t+1}, A_{t+1}, Θ_{t+1}, Π_{t+1}, and ρ_{t+1} by Eqs. (15), (21c), and (22), respectively;
  Check the convergence condition

$$\max \left\{ \begin{array}{c} \|X^{(v)} - X^{(v)} Y_{t+1}^{(v)} - E_{t+1}^{(v)}\|_\infty \\ \|\mathcal{Z}_{t+1} - \mathcal{Y}_{t+1}\|_\infty \end{array} \right\} \le tol;$$

end while

Output: Affinity matrix A_t.

C. Complexity Analysis

For the optimization problem with three or more variable blocks, it is intractable to analyze the gloabl convergence [6, 59]. In the proposed LRTG, there are four variable blocks and thus we cannot give a theoretical guarantee. But as stated in IV-D-3), the proposed LRTG has fast empirical convergence. The detailed complexity of each subproblem is presented as follows: Step 1 would cost $\mathcal{O}(RVn^2)$ for Tucker decomposition, where R is the rank of $\mathcal{Z}$; Step 2 contains matrix inverse and matrix multiplication with cost $\mathcal{O}(Vn^3)$; For Step 3, it costs $\mathcal{O}(Vn^2)$ operations; The cost of Step 4 is $\mathcal{O}(n^2)$. Thus, the total complexity of LRTG is $\mathcal{O}\big(TVn^2(R + n + 1)\big)$, where T is the number of iterations.

TABLE II
CLUSTERING RESULTS (MEAN±STANDARD DEVIATION) ON *Extended YaleB* AND *ORL*.

Data	Type	Method	ACC	NMI	AR	F-score	Precision	Recall
Extended YaleB ($K=5,\ \alpha=2$)	SVC	$\mathrm{SSC}_{\mathbf{best}}$[5]	0.587±0.003	0.534±0.003	0.430±0.005	0.487±0.004	0.451±0.002	0.509±0.007
		$\mathrm{LRR}_{\mathbf{best}}$[6]	0.615±0.013	0.627±0.040	0.451±0.002	0.508±0.004	0.481±0.002	0.539±0.001
		$\mathrm{RSS}_{\mathbf{best}}$[33]	0.742±0.001	0.787±0.000	0.685±0.001	0.717±0.001	0.704±0.001	0.730±0.000
		$\mathrm{rBDLR}_{\mathbf{best}}$[4]	0.224±0.020	0.141±0.031	0.038±0.009	0.144±0.010	0.129±0.008	0.165±0.018
	MVC	MLAP[64]	0.278±0.002	0.231±0.002	0.119±0.002	0.207±0.001	0.204±0.001	0.211±0.001
		RMSC[20]	0.210±0.013	0.157±0.019	0.060±0.014	0.155±0.012	0.151±0.012	0.159±0.013
		DiMSC[23]	0.615±0.003	0.636±0.002	0.453±0.005	0.504±0.006	0.481±0.004	0.534±0.004
		LT-MSC[16]	0.626±0.010	0.637±0.003	0.459±0.030	0.521±0.006	0.485±0.001	0.539±0.002
		MVCC[65]	0.179±0.013	0.150±0.017	0.044±0.006	0.156±0.006	0.131±0.005	0.192±0.016
		MLAN[19]	0.346±0.011	0.352±0.015	0.093±0.009	0.213±0.023	0.159±0.018	0.321±0.013
		ECMSC[22]	0.783±0.011	0.759±0.012	0.544±0.008	0.597±0.010	0.513±0.009	0.718±0.006
		t-SVD[17]	0.652±0.000	0.667±0.004	0.500±0.003	0.550±0.002	0.514±0.004	0.590±0.004
		GMC[21]	0.434±0.000	0.449±0.000	0.157±0.000	0.265±0.000	0.204±0.000	0.378±0.000
		LMSC[49]	0.598±0.005	0.568±0.004	0.354±0.007	0.423±0.006	0.390±0.006	0.463±0.005
		GLTA_Tucker[58]	0.540±0.003	0.566±0.005	0.428±0.006	0.486±0.006	0.473±0.006	0.499±0.006
		GLTA_TSVD[58]	0.614±0.004	0.631±0.006	0.439±0.007	0.497±0.006	0.473±0.006	0.524±0.006
		SCMV-3DT[45]	0.410±0.001	0.413±0.002	0.185±0.002	0.276±0.001	0.244±0.002	0.318±0.001
		UGLTL[66]	0.338±0.006	0.344±0.005	0.152±0.003	0.242±0.002	0.224±0.002	0.264±0.003
	Ours	LRTG	**0.954±0.000**	**0.905±0.000**	**0.899±0.000**	**0.909±0.000**	**0.908±0.000**	**0.911±0.000**
ORL ($K=10,\ \alpha=2$)	SVC	$\mathrm{SSC}_{\mathbf{best}}$[5]	0.765±0.008	0.893±0.007	0.694±0.013	0.682±0.012	0.673±0.007	0.764±0.005
		$\mathrm{LRR}_{\mathbf{best}}$[6]	0.773±0.003	0.895±0.006	0.724±0.020	0.731±0.004	0.701±0.001	0.754±0.002
		$\mathrm{RSS}_{\mathbf{best}}$[33]	0.846±0.024	0.938±0.007	0.798±0.023	0.803±0.023	0.759±0.030	0.852±0.017
		$\mathrm{rBDLR}_{\mathbf{best}}$[4]	0.684±0.035	0.853±0.012	0.560±0.042	0.572±0.041	0.498±0.055	0.676±0.026
	MVC	MLAP[64]	0.789±0.021	0.895±0.010	0.714±0.025	0.720±0.024	0.686±0.027	0.759±0.024
		RMSC[20]	0.723±0.007	0.872±0.012	0.645±0.003	0.654±0.007	0.607±0.009	0.709±0.004
		DiMSC[23]	0.838±0.001	0.940±0.003	0.802±0.000	0.807±0.003	0.764±0.012	0.856±0.004
		LT-MSC[16]	0.795±0.007	0.930±0.003	0.750±0.003	0.768±0.004	0.766±0.009	0.837±0.005
		MVCC[65]	0.625±0.031	0.813±0.009	0.507±0.028	0.520±0.027	0.456±0.037	0.607±0.013
		MLAN[19]	0.705±0.02	0.854±0.018	0.384±0.010	0.376±0.015	0.254±0.021	0.721±0.020
		ECMSC[22]	0.854±0.011	0.947±0.009	0.810±0.012	0.821±0.015	0.783±0.008	0.859±0.012
		t-SVD[17]	**0.970±0.003**	**0.993±0.002**	**0.967±0.002**	**0.968±0.003**	**0.946±0.004**	**0.991±0.003**
		GMC[21]	0.633±0.000	0.857±0.000	0.337±0.000	0.360±0.000	0.232±0.000	0.801±0.000
		LMSC[49]	0.877±0.024	0.949±0.006	0.839±0.022	0.843±0.021	0.806±0.027	0.884±0.017
		GLTA_Tucker[58]	0.855±0.025	0.936±0.006	0.804±0.022	0.809±0.021	0.770±0.028	0.852±0.015
		GLTA_TSVD[58]	0.905±0.025	0.969±0.007	0.890±0.023	0.892±0.023	0.859±0.029	0.929±0.018
		SCMV-3DT[45]	0.839±0.012	0.908±0.007	0.763±0.018	0.769±0.017	0.747±0.020	0.792±0.016
		UGLTL[66]	0.924±0.028	0.970±0.013	0.912±0.033	0.913±0.032	0.887±0.041	0.941±0.024
	Ours	LRTG	0.933±0.003	0.970±0.002	0.905±0.005	0.908±0.005	0.888±0.004	0.928±0.007

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed LRTG for multi-view clustering on eight popular real-world databases. The code will be released at https://cyyhit.github.io/.

A. Experimental Settings

1) Databases: Eight commonly used real-world databases with five different categories: face images, news stories, generic object, handwritten digits, and Prokaryotic are selected to investigate the effectiveness of the proposed LRTG. Some examples of these databases are shown in Fig. 1. *Extended YaleB*[2] *and ORL*[3]*:* The extended YaleB database contains 2414 face images of 38 individuals, each of which has 64 near frontal images under different lighting conditions. Following [16], 3 types of features *i.e.*, $2500d$ (dimension, d) intensity, $3304d$ LBP, and $6750d$ Gabor are extracted of the first 10 classed. The ORL database includes 400 face images with 40 classes under different times, lighting, facial expressions, and facial details. For ORL, 3 types of features *i.e.*, $4096d$ intensity, $3304d$ LBP, and $6750d$ Gabor are explored; *BBC4view, BBCSport*[4]*, and 3Sources*[5]*:* They are news stories databases. BBC4view contains 685 documents from BBC Sport website about sports news on 5 topics and 4 different types of features are extracted. BBCSport consists of 544 documents and 2 different types of features are extracted. 3Sources consists of 416 distinct news stories from 6 classes with 3 views; *COIL_20*[6]*:* For COIL_20, 3 view features including $1024d$ intensity, $3304d$ LBP, and $6750d$ Gabor are employed. *UCI-Digits*[7]*:* UCI-Digits contains 10 classes of handwritten digits from the UCI repository. Each class has 200 examples. Thus there are 2000 samples in total. Following [21], three features including $240d$ Fourier coefficients, $76d$ pixel averages and $6d$ morphological features are extracted. *Prokaryotic:* Prokaryotic consists of 551 prokaryotic species with two features: textual

[2]http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html
[3]http://www.uk.research.att.com/facedatabase.html
[4]http://mlg.ucd.ie/datasets/segment.html
[5]http://mlg.ucd.ie/datasets/3sources.html
[6]http://www.cs.columbia.edu/CAVE/software/softlib/
[7]http://archive.ics.uci.edu/ml/datasets/Multiple+Features

TABLE III
CLUSTERING RESULTS (MEAN±STANDARD DEVIATION) ON *3Sources* AND *UCI-Digits*.

Data	Type	Method	ACC	NMI	AR	F-score	Precision	Recall
3Sources ($K = 10,\ \alpha = 50$)	SVC	$\text{SSC}_{\textbf{best}}$[5]	0.762±0.003	0.694±0.003	0.658±0.004	0.743±0.003	0.769±0.001	0.719±0.005
		$\text{LRR}_{\textbf{best}}$[6]	0.647±0.033	0.542±0.018	0.486±0.028	0.608±0.033	0.594±0.031	0.636±0.096
		$\text{RSS}_{\textbf{best}}$[33]	0.722±0.000	0.601±0.000	0.533±0.000	0.634±0.000	0.679±0.000	0.595±0.000
		$\text{rBDLR}_{\textbf{best}}$[4]	0.615±0.049	0.546±0.050	0.422±0.080	0.564±0.053	0.544±0.083	0.593±0.046
	MVC	MLAP[64]	0.805±0.000	0.756±0.000	0.688±0.000	0.762±0.000	0.751±0.000	0.773±0.000
		RMSC[20]	0.583±0.022	0.630±0.011	0.455±0.031	0.557±0.025	0.635±0.029	0.497±0.028
		DiMSC[23]	0.795±0.004	0.727±0.010	0.661±0.005	0.748±0.004	0.711±0.005	0.788±0.003
		LT-MSC[16]	0.781±0.000	0.698±0.003	0.651±0.003	0.734±0.002	0.716±0.008	0.754±0.005
		MVCC[65]	0.761±0.016	0.698±0.016	0.626±0.010	0.731±0.008	0.607±0.009	**0.916±0.008**
		MLAN[19]	0.775±0.015	0.676±0.005	0.580±0.008	0.666±0.007	0.756±0.003	0.594±0.009
		ECMSC[22]	0.346±0.025	0.132±0.029	0.011±0.031	0.295±0.013	0.240±0.019	0.391±0.043
		t-SVD[17]	0.781±0.000	0.678±0.000	0.658±0.000	0.745±0.000	0.683±0.000	0.818±0.000
		GMC[21]	0.693±0.000	0.622±0.000	0.443±0.000	0.605±0.000	0.484±0.000	0.804±0.000
		LMSC[49]	0.912±0.006	0.826±0.007	0.842±0.011	0.887±0.008	0.873±0.007	0.877±0.012
		GLTA_Tucker[58]	0.846±0.000	0.728±0.000	0.665±0.000	0.736±0.000	0.805±0.000	0.678±0.000
		GLTA_TSVD[58]	0.859±0.008	0.753±0.015	0.713±0.014	0.775±0.011	0.827±0.009	0.730±0.013
		SCMV-3DT[45]	0.440±0.020	0.386±0.009	0.226±0.012	0.411±0.009	0.399±0.012	0.425±0.016
		UGLTL[66]	0.388±0.020	0.082±0.007	0.036±0.012	0.341±0.015	0.250±0.006	0.535±0.042
	Ours	LRTG	**0.947±0.000**	**0.865±0.000**	**0.881±0.000**	**0.909±0.000**	**0.911±0.000**	0.906±0.000
UCI-Digits ($K = 15,\ \alpha = 50$)	SVC	$\text{SSC}_{\textbf{best}}$[5]	0.815±0.011	0.840±0.001	0.770±0.005	0.794±0.004	0.747±0.010	0.848±0.004
		$\text{LRR}_{\textbf{best}}$[6]	0.871±0.001	0.768±0.002	0.736±0.002	0.763±0.002	0.759±0.002	0.767±0.002
		$\text{RSS}_{\textbf{best}}$[33]	0.819±0.000	0.863±0.000	0.787±0.000	0.810±0.000	0.756±0.000	0.872±0.000
		$\text{rBDLR}_{\textbf{best}}$[4]	0.711±0.069	0.714±0.035	0.608±0.064	0.649±0.056	0.614±0.064	0.690±0.049
	MVC	MLAP[64]	0.822±0.001	0.775±0.001	0.713±0.001	0.742±0.001	0.729±0.001	0.756±0.001
		RMSC[20]	0.915±0.024	0.822±0.008	0.789±0.014	0.811±0.012	0.797±0.017	0.826±0.006
		DiMSC[23]	0.703±0.010	0.772±0.006	0.652±0.006	0.695±0.006	0.673±0.005	0.718±0.007
		LT-MSC[16]	0.803±0.001	0.775±0.001	0.725±0.001	0.753±0.001	0.739±0.001	0.767±0.001
		MVCC[65]	0.914±0.001	0.871±0.001	0.832±0.001	0.849±0.001	0.839±0.001	0.858±0.001
		MLAN[19]	0.874±0.000	0.910±0.000	0.847±0.000	0.864±0.000	0.797±0.000	0.943±0.000
		ECMSC[22]	0.718±0.001	0.780±0.001	0.672±0.001	0.707±0.001	0.660±0.001	0.760±0.001
		t-SVD[17]	0.955±0.000	0.932±0.000	0.924±0.000	0.932±0.000	0.930±0.000	0.934±0.000
		GMC[21]	0.736±0.000	0.815±0.000	0.678±0.000	0.713±0.000	0.644±0.000	0.799±0.000
		LMSC[49]	0.893±0.000	0.815±0.000	0.783±0.000	0.805±0.000	0.798±0.000	0.812±0.000
		GLTA_Tucker[58]	0.815±0.000	0.768±0.001	0.707±0.001	0.737±0.001	0.731±0.001	0.743±0.001
		GLTA_TSVD[58]	0.996±0.000	0.989±0.000	0.991±0.000	0.992±0.000	0.992±0.000	0.992±0.000
		SCMV-3DT[45]	0.930±0.001	0.861±0.001	0.846±0.001	0.861±0.001	0.859±0.001	0.864±0.001
		UGLTL[66]	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	Ours	LRTG	0.981±0.000	0.953±0.000	0.957±0.000	0.961±0.000	0.961±0.000	0.962±0.000

data and genomic representations.

2) Competitors: For a comprehensive comparison, we compare the proposed LRTG with the following 17 state-of-the-art competitors:

- *Single-view competitors:* Four representative single-view clustering (SVC) methods: sparse subspace clustering(SSC) [5], low-rank representation (LRR) [6], robust subspace segmentation (RSS) [33], and block-diagonal adaptive locality-constrained representation (rBDLR) [4] are adopted. To construct the affinity matrix, SSC uses the l_1-norm, LRR uses the nuclear norm, RSS explores the l_2-norm, while rBDLR exploits the block diagonal representation. In our experiments, we conduct these four single-view clustering approaches on each view feature independently and report the best clustering results;
- *Multi-view competitors:* We compare thirteen multi-view clustering (MVC) methods, including multi-task low-rank affinity pursuit (MLAP) [64], low-rank and sparse decomposition (RMSC) [20], diversity-induced multi-view subspace clustering (DiMSC) [23], low-rank tensor constrained multi-view subspace clustering (LT-MSC) [16], multi-view clustering via concept factorization with local manifold regularization (MVCC) [65], auto-weighted multi-view learning (MLAN) [19], exclusivity-consistency regularized multi-view subspace clustering (ECMSC) [22], multi-view clustering by tensor multi-rank minimization (t-SVD) [17], graph-based multi-view clustering (GMC) [21], latent multi-view subspace clustering (LMSC) [49], graph-regularized low-rank tensor approximation (GLTA) [58], tensorial t-product representation (SCMV-3DT) [45], and unified graph low-rank tensor learning (UGLTL) [66].

We carried out two versions of GLTA, that is, GLTA_Tucker and GLTA_TSVD. Parameter settings of all competitors are followed original papers. MLAP, RMSC, LT-MSC, ECMSC, t-SVD, LMSC, SCMV-3DT, GLTA, and UGLTL are the recently proposed multi-view subspace clustering while MLAN and GMC belong to the multi-view graph clustering.

3) Evaluation Metrics: To fully and quantitatively examine the performance of the proposed LRTG, six popular metrics are used including accuracy (ACC), normalized mutual information (NMI), adjusted rank index (AR), F-score, Precision,

TABLE IV
CLUSTERING RESULTS (MEAN±STANDARD DEVIATION) ON *BBC4view* AND *BBCSport*.

Data	Type	Method	ACC	NMI	AR	F-score	Precision	Recall
BBC4view ($K = 15,\ \alpha = 50$)	SVC	SSC$_{\mathbf{best}}$[5]	0.660±0.002	0.494±0.005	0.470±0.001	0.599±0.001	0.578±0.001	0.622±0.001
		LRR$_{\mathbf{best}}$[6]	0.802±0.000	0.568±0.000	0.621±0.000	0.712±0.000	0.697±0.000	0.727±0.000
		RSS$_{\mathbf{best}}$[33]	0.837±0.000	0.621±0.000	0.665±0.000	0.747±0.000	0.720±0.000	0.775±0.000
		rBDLR$_{\mathbf{best}}$[4]	0.714±0.084	0.560±0.049	0.539±0.102	0.652±0.075	0.627±0.083	0.682±0.072
	MVC	MLAP[64]	0.872±0.000	0.725±0.000	0.751±0.000	0.808±0.000	0.824±0.000	0.793±0.000
		RMSC[20]	0.775±0.003	0.616±0.004	0.560±0.002	0.656±0.002	0.703±0.003	0.616±0.001
		DiMSC[23]	0.892±0.001	0.728±0.002	0.752±0.002	0.810±0.002	0.811±0.002	0.810±0.002
		LT-MSC[16]	0.591±0.000	0.442±0.005	0.400±0.001	0.546±0.000	0.525±0.000	0.570±0.001
		MVCC[65]	0.745±0.001	0.587±0.001	0.550±0.000	0.656±0.001	0.654±0.001	0.658±0.000
		MLAN[19]	0.853±0.007	0.698±0.010	0.716±0.005	0.783±0.004	0.776±0.003	0.790±0.004
		ECMSC[22]	0.308±0.028	0.047±0.009	0.008±0.018	0.322±0.017	0.239±0.009	0.497±0.064
		t-SVD[17]	0.858±0.001	0.685±0.002	0.725±0.002	0.789±0.001	0.800±0.001	0.778±0.002
		GMC[21]	0.693±0.000	0.563±0.000	0.479±0.000	0.633±0.000	0.501±0.000	0.860±0.000
		LMSC[49]	0.883±0.000	0.699±0.000	0.746±0.000	0.806±0.000	0.797±0.000	0.816±0.000
		GLTA_Tucker[58]	0.910±0.000	0.771±0.000	0.810±0.000	0.854±0.000	0.864±0.000	0.845±0.000
		GLTA_TSVD[58]	**0.996±0.000**	**0.983±0.000**	**0.990±0.000**	**0.993±0.000**	**0.996±0.000**	**0.990±0.000**
		SCMV-3DT[45]	0.485±0.001	0.479±0.001	0.264±0.001	0.471±0.001	0.393±0.001	0.589±0.001
		UGLTL[66]	0.827±0.002	0.722±0.001	0.639±0.003	0.717±0.002	0.779±0.003	0.664±0.002
	Ours	LRTG	0.894±0.000	0.769±0.000	0.791±0.000	0.839±0.000	0.857±0.000	0.822±0.000
BBCSport ($K = 10,\ \alpha = 2$)	SVC	SSC$_{\mathbf{best}}$[5]	0.627±0.003	0.534±0.008	0.364±0.007	0.565±0.005	0.427±0.004	0.834±0.004
		LRR$_{\mathbf{best}}$[6]	0.836±0.001	0.698±0.002	0.705±0.001	0.776±0.001	0.768±0.001	0.784±0.001
		RSS$_{\mathbf{best}}$[33]	0.878±0.000	0.714±0.000	0.717±0.000	0.784±0.000	0.787±0.000	0.782±0.000
		rBDLR$_{\mathbf{best}}$[4]	0.719±0.102	0.596±0.090	0.542±0.139	0.671±0.089	0.592±0.124	0.793±0.063
	MVC	MLAP[64]	0.868±0.001	0.763±0.003	0.791±0.003	0.842±0.002	0.827±0.002	0.858±0.003
		RMSC[20]	0.826±0.001	0.666±0.001	0.637±0.001	0.719±0.001	0.766±0.001	0.677±0.001
		DiMSC[23]	0.922±0.000	0.785±0.000	0.813±0.000	0.858±0.000	0.846±0.000	0.872±0.000
		LT-MSC[16]	0.460±0.046	0.222±0.028	0.167±0.043	0.428±0.014	0.328±0.028	0.629±0.053
		MVCC[65]	0.928±0.000	0.816±0.000	0.831±0.000	0.870±0.000	0.889±0.000	0.853±0.000
		MLAN[19]	0.721±0.000	0.779±0.000	0.591±0.000	0.714±0.000	0.567±0.000	0.962±0.000
		ECMSC[22]	0.285±0.014	0.027±0.013	0.009±0.011	0.267±0.020	0.244±0.007	0.297±0.045
		t-SVD[17]	0.879±0.000	0.765±0.000	0.784±0.000	0.834±0.000	0.863±0.000	0.807±0.000
		GMC[21]	0.807±0.000	0.760±0.000	0.722±0.000	0.794±0.000	0.727±0.000	0.875±0.000
		LMSC[49]	0.847±0.003	0.739±0.001	0.749±0.001	0.810±0.001	0.799±0.001	0.822±0.001
		GLTA_Tucker[58]	0.939±0.000	0.825±0.000	0.849±0.000	0.885±0.000	0.890±0.000	0.880±0.000
		GLTA_TSVD[58]	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
		SCMV-3DT[45]	0.980±0.000	0.929±0.000	0.935±0.000	0.950±0.000	0.959±0.000	0.942±0.000
		UGLTL[66]	0.367±0.001	0.087±0.002	0.061±0.004	0.387±0.001	0.264±0.002	0.993±0.006
	Ours	LRTG	0.943±0.005	0.869±0.009	0.840±0.012	0.879±0.010	0.866±0.006	0.892±0.014

and Recall. Generally, the higher values of these six measures mean the better clustering quality. For detailed definitions of these metrics, please refer to [17]. Except for MLAN, all methods including the proposed LRTG perform K-means to obtain the final clustering indicators. We perform all methods 10 repetitions and the average results with the standard deviation are reported.

B. Clustering Performance Comparison

The detailed clustering results are reported in Tables II to V, in which the best results are highlighted in bold and the second-best ones are underlined. More specifically, we have the following observations:

- On Extended YaleB, 3Sources, and Prokaryotic databases, the proposed LRTG outperforms all competitors, while on the other databases, LRTG is the second or third best algorithm. Specifically, the improvements of LRTG are around 17.1%, 11.8%, 21.4%, 19.2%, 20.4%, and 18.1% with respect to six measures over the second-best method on Extended YaleB, and around 3.5%, 3.9%, 3.9%, 2.2%, 3.8%, and 2.9% on 3Sources, respectively. This demonstrates the superiority of LRTG over all counterparts.
- Three single-view clustering methods (SSC, LRR, rBDLR) cannot achieve competitive results than most of multi-view clustering ones. While another single-view clustering approach RSS has superior performance than several multi-view ones, such as MLAP, RMSC, MVCC, especially on Extended YaleB database. Among these four single-view clustering methods, RSS performs better than SSC, rBDLR and LRR. The reason may be that RSS well captures the underlying low-dimensional structure within data by jointly learning the representation matrix and the affinity matrix;
- RMSC, LT-MSC, LMSC and t-SVD methods follow two steps to learn the representation tensor and the affinity matrix. However, they achieved unstable performance on different databases. t-SVD achieves the best results on ORL, while it has difficulty in obtaining comparable performance on the other databases, especially Extended YaleB and 3Sources. There is a similar conclusion for MLAN. Specifically, MLAN performs even worse

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2021.3055625, IEEE Transactions on Circuits and Systems for Video Technology

TABLE V
CLUSTERING RESULTS (MEAN±STANDARD DEVIATION) ON *COIL_20* AND *Prokaryotic*.

Data	Type	Method	ACC	NMI	AR	F-score	Precision	Recall
COIL-20 ($K = 5,\ \alpha = 10$)	SVC	$\text{SSC}_{\textbf{best}}$[5]	0.803±0.022	0.935±0.009	0.798±0.022	0.809±0.013	0.734±0.027	0.804±0.028
		$\text{LRR}_{\textbf{best}}$[6]	0.761±0.003	0.829±0.006	0.720±0.020	0.734±0.006	0.717±0.003	0.751±0.002
		$\text{RSS}_{\textbf{best}}$[33]	0.837±0.012	0.930±0.006	0.789±0.005	0.800±0.005	0.717±0.012	0.897±0.017
		$\text{rBDLR}_{\textbf{best}}$[4]	0.681±0.047	0.821±0.015	0.599±0.045	0.621±0.042	0.549±0.059	0.720±0.018
	MVC	MLAP[64]	0.738±0.020	0.825±0.009	0.685±0.023	0.701±0.021	0.688±0.027	0.715±0.016
		RMSC[20]	0.685±0.045	0.800±0.017	0.637±0.044	0.656±0.042	0.620±0.057	0.698±0.026
		DiMSC[23]	0.778±0.022	0.846±0.002	0.732±0.005	0.745±0.005	0.739±0.007	0.751±0.003
		LT-MSC[16]	0.804±0.011	0.860±0.002	0.748±0.004	0.760±0.007	0.741±0.009	0.776±0.006
		MVCC[65]	0.732±0.018	0.845±0.007	0.675±0.022	0.692±0.021	0.647±0.034	0.744±0.013
		MLAN[19]	0.862±0.011	0.961±0.004	0.835±0.006	0.844±0.013	0.758±0.008	0.953±0.007
		ECMSC[22]	0.782±0.001	0.942±0.001	0.781±0.001	0.794±0.001	0.695±0.002	0.925±0.001
		t-SVD[17]	0.830±0.000	0.884±0.005	0.786±0.003	0.800±0.004	0.785±0.007	0.808±0.001
		GMC[21]	0.791±0.001	0.941±0.000	0.782±0.000	0.794±0.000	0.694±0.000	0.929±0.000
		LMSC[49]	0.806±0.013	0.862±0.007	0.765±0.014	0.776±0.013	0.770±0.013	0.783±0.013
		GLTA_Tucker[58]	0.878±0.008	0.945±0.001	0.869±0.007	0.875±0.007	0.856±0.013	0.895±0.001
		GLTA_TSVD[58]	0.903±0.006	0.946±0.001	0.891±0.007	0.897±0.006	0.893±0.013	0.900±0.001
		SCMV-3DT[45]	0.701±0.028	0.810±0.009	0.635±0.003	0.654±0.029	0.614±0.039	0.702±0.018
		UGLTL[66]	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	Ours	LRTG	0.927±0.000	0.976±0.000	0.928±0.000	0.932±0.000	0.905±0.000	0.961±0.000
Prokaryotic ($K = 15,\ \alpha = 2$)	SVC	$\text{SSC}_{\textbf{best}}$[5]	0.466±0.000	0.242±0.000	0.083±0.000	0.439±0.000	0.446±0.000	0.432±0.000
		$\text{LRR}_{\textbf{best}}$[6]	0.499±0.000	0.245±0.000	0.115±0.000	0.410±0.000	0.485±0.000	0.355±0.000
		$\text{RSS}_{\textbf{best}}$[33]	0.523±0.000	0.307±0.000	0.125±0.000	0.456±0.000	0.476±0.000	0.437±0.000
		$\text{rBDLR}_{\textbf{best}}$[4]	0.708±0.049	0.444±0.048	0.435±0.039	0.623±0.023	0.741±0.038	0.537±0.017
	MVC	MLAP[64]	0.583±0.000	0.243±0.000	0.203±0.000	0.479±0.000	0.546±0.000	0.436±0.000
		RMSC[20]	0.461±0.049	0.315±0.041	0.198±0.044	0.447±0.027	0.567±0.038	0.369±0.023
		DiMSC[23]	0.395±0.001	0.070±0.000	0.053±0.000	0.346±0.000	0.441±0.000	0.284±0.000
		LT-MSC[16]	0.431±0.007	0.156±0.020	0.051±0.016	0.401±0.006	0.429±0.011	0.376±0.003
		MLAN[19]	0.712±0.002	0.387±0.003	0.425±0.003	0.618±0.002	0.728±0.002	0.537±0.002
		ECMSC[22]	0.432±0.001	0.193±0.001	0.078±0.001	0.383±0.002	0.457±0.002	0.329±0.001
		t-SVD[17]	0.523±0.000	0.197±0.000	0.137±0.000	0.486±0.000	0.474±0.000	0.500±0.000
		GMC[21]	0.496±0.000	0.193±0.000	0.091±0.000	0.461±0.000	0.447±0.000	0.476±0.000
		LMSC[49]	0.686±0.002	0.306±0.001	0.262±0.001	0.603±0.001	0.514±0.001	0.728±0.001
		GLTA_Tucker[58]	0.709±0.002	0.376±0.001	0.312±0.002	0.633±0.002	0.535±0.001	**0.775±0.004**
		GLTA_TSVD[58]	0.731±0.000	0.408±0.000	0.371±0.000	0.650±0.000	0.577±0.000	0.744±0.000
		SCMV-3DT[45]	0.619±0.003	0.432±0.003	0.331±0.002	0.556±0.003	0.651±0.001	0.485±0.004
	Ours	LRTG	**0.788±0.000**	**0.484±0.000**	**0.492±0.000**	**0.671±0.000**	**0.750±0.000**	0.607±0.000

than single-view clustering methods. For example, RSS achieves improvements around 14.1%, 8.4%, 41.4% with respect to ACC, NMI, AR over MLAN on ORL.

Since each entry of the affinity matrix denotes the similarity of pairwise data points, a good affinity matrix should have high intra-cluster similarity and low inter-cluster similarity, that is, block diagonal structure. To show this, we also give a comparison of the affinity matrices obtained by eight popular clustering methods on Extended YaleB database as shown in Fig. 2. We can see that all competing methods especially RMSC construct affinity matrices with a roughly block-diagonal structure while LRTG can learn a better affinity matrix over them. This further demonstrates the superiority of the proposed LRTG. The reason is that the proposed LRTG learns an adaptive affinity matrix from the clean representation tensor instead of the raw data.

C. Clustering Performance Comparison on Noisy Databases

Because the original images may be corrupted by noise, such as Gaussian noise, we conduct new experiments on noisy COIL_20 and Extended YaleB databases to investigate the robustness of the proposed LRTG. The zero-mean Gaussian noise with variance 0.5 was added into original COIL_20 and Extended YaleB databases and then we extracted Intensity, LBP and Gabor features to generate the noisy multi-view datasets. Table VI gives the clustering results of all methods on two noisy COIL_20 and Extended YaleB databases. One can see that the proposed LRTG still outperforms the other methods on two noisy datasets.

D. Model Discussion

1) Ablation Study on the "two-step" Strategy: Differing from most existing multi-view subspace clustering which construct the affinity matrix by two separate steps, LRTG directly learns the a unified and adaptive affinity matrix. To further investigate the influence of the "two-step" strategy, we report the clustering results *i.e.*, ACC and NMI of LRTG as shown in Table VII. Here $\text{LRTG}_{\mathcal{Z}}$ and LRTG_{A} denote LRTG using the representation tensor $\mathcal{Z}$ and the unified affinity matrix A, respectively. It is observed that LRTG_{A} outperforms $\text{LRTG}_{\mathcal{Z}}$ on all databases, indicating that directly learning a flexible affinity matrix is better than that by two separate steps.

2) Parameter Selection: As in Eq. (21b), parameter β is determined by the number of adaptive neighbors K. Thus, there are two free parameter α and K.

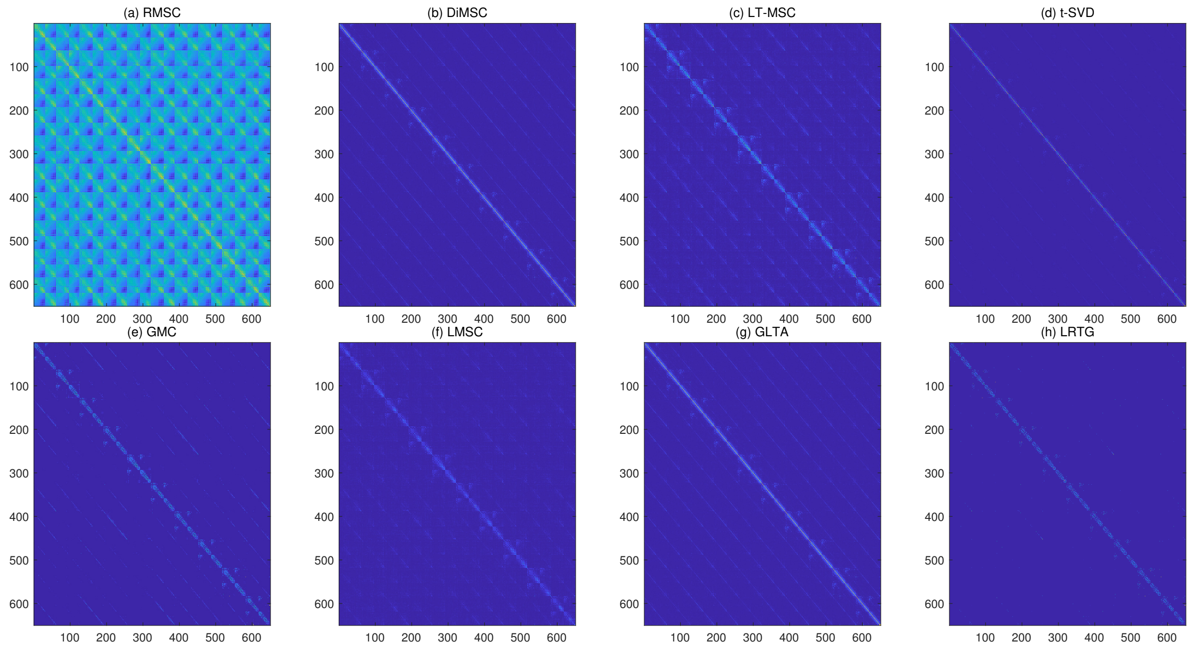


Fig. 2. Comparison of affinity matrix on Extended YaleB by (a) RMSC, (b) DiMSC, (c) LT-MSC, (d) t-SVD, (e) GMC, (f) LMSC, (g) GLTA_TSVD, (h) LRTG.

TABLE VI
CLUSTERING RESULTS (MEAN±STANDARD DEVIATION) ON NOISY FEATURES WITH GAUSSIAN NOISE.

Type	Method	*COIL-20*			*Extended YaleB*		
		ACC	NMI	AR	ACC	NMI	AR
SVC	$\text{SSC}_{\textbf{best}}$[5]	0.779±0.001	0.924±0.001	0.742±0.001	0.571±0.017	0.570±0.003	0.331±0.007
	$\text{LRR}_{\textbf{best}}$[6]	0.724±0.011	0.817±0.009	0.656±0.017	0.614±0.001	0.633±0.001	0.452±0.001
	$\text{RSS}_{\textbf{best}}$[33]	0.787±0.008	0.917±0.001	0.674±0.010	0.723±0.005	0.679±0.003	0.535±0.011
	$\text{rBDLR}_{\textbf{best}}$[4]	0.629±0.055	0.787±0.025	0.549±0.050	0.293±0.035	0.256±0.042	0.098±0.022
MVC	MLAP[64]	0.721±0.019	0.806±0.007	0.669±0.015	0.335±0.003	0.338±0.003	0.191±0.002
	RMSC[20]	0.665±0.043	0.777±0.016	0.618±0.041	0.259±0.014	0.214±0.023	0.108±0.015
	DiMSC[23]	0.756±0.023	0.830±0.010	0.694±0.018	0.347±0.007	0.347±0.006	0.194±0.003
	LT-MSC[16]	0.739±0.033	0.817±0.014	0.664±0.031	0.503±0.018	0.511±0.016	0.318±0.009
	MVCC[65]	0.706±0.033	0.804±0.011	0.628±0.032	0.244±0.007	0.226±0.007	0.104±0.004
	MLAN[19]	0.842±0.000	0.942±0.000	0.802±0.000	0.505±0.000	0.425±0.000	0.130±0.000
	t-SVD[17]	0.809±0.008	0.889±0.006	0.757±0.010	0.644±0.001	0.637±0.001	0.469±0.001
	GMC[21]	0.864±0.000	0.957±0.000	0.833±0.000	0.655±0.000	0.689±0.000	0.441±0.000
	LMSC[49]	0.805±0.001	0.855±0.001	0.754±0.001	0.557±0.001	0.548±0.001	0.311±0.001
	GLTA_Tucker[58]	0.859±0.017	0.912±0.009	0.824±0.015	0.582±0.017	0.535±0.003	0.281±0.005
	GLTA_TSVD[58]	0.880±0.011	0.942±0.003	0.867±0.012	0.560±0.006	0.588±0.010	0.436±0.011
Ours	LRTG	**0.917±0.000**	**0.967±0.000**	**0.893±0.000**	**0.939±0.000**	**0.906±0.000**	**0.874±0.000**

TABLE VII
PERFORMANCE OF LRTG.

Database	$\text{LRTG}_{\mathcal{Z}}$		$\text{LRTG}_{\mathcal{A}}$	
	ACC	NMI	ACC	NMI
Extended YaleB	0.561	0.565	0.954	0.905
ORL	0.915	0.963	0.933	0.970
COIL_20	0.766	0.880	0.927	0.976
BBC4view	0.885	0.750	0.894	0.769
BBCSport	0.864	0.796	0.943	0.869
3Sources	0.811	0.786	0.947	0.865
UCI-Digits	0.968	0.925	0.981	0.953
Prokaryotic	0.478	0.148	0.788	0.484
Average	0.7810	0.7266	0.9209	0.8489

α and K are empirically selected from the sets of $[0.001, 0.01, 0.1, 0.5, 1, 2, 5, 10, 50, 100]$ and $[5 : 15]$, respectively. Due to page limitation, we only report ACC and NMI values with different combinations of α and K as shown in Fig. 3. It is observed that LRTG can achieve promising performance when α is set to a relatively large value ($\alpha \geq 1$). We also investigated the sensitivity of the proposed LRTG with different estimated ranks in Table IX. We can observe that LRTG can achieve promising performance when the estimated ranks R_1, R_2, R_3 were set as $(50, 100, V)$, where V is the number of views.

3) Empirical Convergence: As stated in III-C, it is chal-

TABLE VIII
AVERAGE RUNNING TIME (IN SECONDS) ON ALL DATABASES.

Method	DiMSC	LT-MSC	ECMSC	LMSC	tSVD	GLTA_Tucker	GLTA_TSVD	SCMV-3DT	LRTG
Extended YaleB	30.54	128.15	705.60	39.74	54.19	28.06	26.55	300.78	60.37
ORL	13.16	65.28	379.67	45.75	34.85	16.04	17.29	157.58	36.26
COIL-20	617.29	874.91	1305.32	321.05	169.10	324.14	143.21	1412.23	706.91
BBC4view	207.21	335.51	1238.70	180.38	97.99	50.09	54.38	6082.81	62.75
BBCSport	38.15	77.23	266.86	59.28	19.59	54.63	16.53	61.95	50.59
3Sources	1.07	9.47	454.56	9.25	6.61	4.45	4.49	31.83	2.52
UCI-digits	296.66	725.50	305.59	634.30	158.26	58.58	251.02	1604.51	229.37
Prokaryotic	17.65	29.34	659.61	11.84	7.32	14.17	9.93	111.01	10.19

TABLE IX
CLUSTERING RESULTS OF LRTG WITH DIFFERENT RANKS ON ORL/EXTENDED YALEB DATABASES.

(R_1, R_2, R_3)	ACC	NMI	AR	F-score	Precision	Recall
$(50, 50, V)$	0.928/0.860	0.969/0.878	0.895/0.823	0.897/0.841	0.866/0.787	0.931/0.905
$(50, 100, V)$	0.933/0.954	0.970/0.905	0.905/0.899	0.908/0.909	0.888/0.908	0.928/0.911
$(100, 50, V)$	0.923/0.868	0.966/0.891	0.894/0.836	0.897/0.854	0.877/0.797	0.917/0.918
$(100, 100, V)$	0.923/0.961	0.966/0.918	0.885/0.914	0.888/0.923	0.855/0.921	0.922/0.924

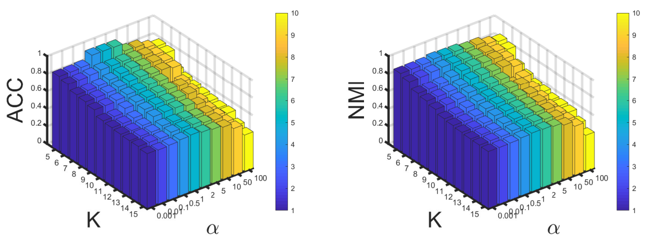


Fig. 3. Parameters tuning (α and K) in terms of ACC and NMI on Extended YaleB.

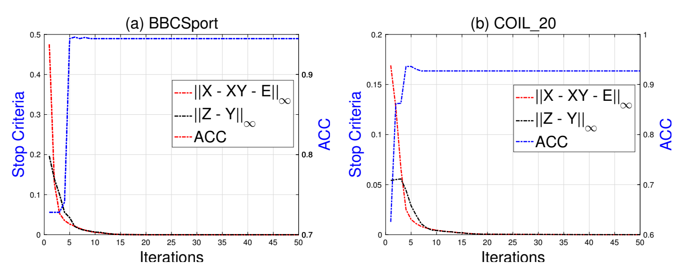


Fig. 4. The convergence curves and ACC versus iterations on (a) BBCSport and (b) COIL_20.

lenging to prove the theoretical convergence of the proposed LRTG since it contains four variable blocks. Thus, in this subsection, we aim to investigate the empirical convergence of LRTG. In Fig. 4, we plot the stop criterion defined in the line 8 of Algorithm 1 and ACC in each iteration. Our LRTG has fast convergence property.

4) Running Times: We also provide the running times of several representative multi-view clustering methods to investigate their efficiency in Table VIII. Each time is the average value of 10 experiments. The running times of ECMSC and SCMV-3DT are much higher while the other methods cost the running time roughly on the same magnitude. For SCMV-3DT, the reason may be that it constructs a large size tensor and its computational complexity is highly related to the size of multi-view features. As stated in Section III-C, the complexity of our LRTG is cubic to the number of samples.

V. CONCLUSIONS

In this paper, we proposed a novel multi-view clustering method via learning a robust low-rank tensor graph (LRTG). LRTG integrates the Tucker decomposition, $l_{2,1}$-norm, and the adaptive neighbor scheme into a joint optimization framework to learn a flexible graph. Specifically, the Tucker decomposition is explored to model the global structure and the $l_{2,1}$-norm is to remove noise and outliers from the raw data. Both of them are devoted to producing a "clean" representation. Moreover, the "clean" representation tensor is used to learn a reliable affinity matrix with adaptive neighbors. Experimental results on eight real-world databases demonstrate that the proposed LRTG can stably yield superior performance over several state-of-the-art competitors. In the future, we consider to extend the proposed method based on kernel theory to handle the non-linearity problem. One the other hand, other tensor decomposition techniques have also been developed, such as CP decomposition and Tensor Train decomposition. Thus, it is an interesting research direction to investigate the best tensor decomposition for multi-view clustering.

REFERENCES

[1] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, 2011.
[2] L. Xing, B. Chen, J. Wang, S. Du, and J. Cao, "Robust high-order manifold constrained low rank representation for subspace clustering," *IEEE Trans. Circuits Syst. Video Technol.*, 2020.
[3] Y.-P. Zhao, L. Chen, and C. P. Chen, "Laplacian regularized nonnegative representation for clustering and dimensionality reduction," *IEEE Trans. Circuits Syst. Video Technol.*, 2020.
[4] Z. Zhang, J. Ren, S. Li, R. Hong, Z. Zha, and M. Wang, "Robust subspace discovery by block-diagonal adaptive locality-constrained representation," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 1569–1577.
[5] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, 2013.
[6] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, 2013.

[7] H. Hu, Z. Lin, J. Feng, and J. Zhou, "Smooth representation clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3834–3841.

[8] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 487–501, 2018.

[9] C. Zhang, H. Fu, Q. Hu, P. Zhu, and X. Cao, "Flexible multi-view dimensionality co-reduction," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 648–659, 2016.

[10] H. Zhao, H. Liu, Z. Ding, and Y. Fu, "Consensus regularized multi-view outlier detection," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 236–248, 2017.

[11] Y. Xie, W. Zhang, Y. Qu, L. Dai, and D. Tao, "Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 572–586, 2018.

[12] J. Lei, X. Li, B. Peng, L. Fang, N. Ling, and Q. Huang, "Deep spatial-spectral subspace clustering for hyperspectral image," *IEEE Trans. Circuits Syst. Video Technol.*, 2020.

[13] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou, "Fast and robust multi-person 3d pose estimation from multiple views," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7792–7801.

[14] S. Teng, S. Zhang, Q. Huang, and N. Sebe, "Multi-view spatial attention embedding for vehicle re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, 2020.

[15] Y. Chen, X. Xiao, and Y. Zhou, "Multi-view clustering via simultaneously learning graph regularized low-rank tensor representation and affinity matrix," in *Proc. Int. Conf. Multimedia Expo.* IEEE, 2019, pp. 1348–1353.

[16] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1582–1590.

[17] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, and Y. Qu, "On unifying multi-view self-representations for clustering by tensor multi-rank minimization," *Int. J. Comput. Vis.*, vol. 126, no. 11, pp. 1157–1179, 2018.

[18] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5910–5922, 2019.

[19] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for image clustering and semi-supervised classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1501–1511, 2018.

[20] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition." in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2149–2155.

[21] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, 2019.

[22] X. Wang, X. Guo, Z. Lei, C. Zhang, and S. Z. Li, "Exclusivity-consistency regularized multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 923–931.

[23] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 586–594.

[24] M. Yin, S. Xie, Z. Wu, Y. Zhang, and J. Gao, "Subspace clustering via learning an adaptive low-rank graph," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3716–3728, 2018.

[25] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. Joint Conf. Artif. Intell.*, 2017, pp. 2564–2570.

[26] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2887–2895, 2017.

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

[28] Z. Kang, H. Pan, S. C. Hoi, and Z. Xu, "Robust graph learning from noisy data," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1833–1843, 2019.

[29] D. Pimentel-Alarcón, L. Balzano, R. Marcia, R. Nowak, and R. Willett, "Group-sparse subspace clustering with missing data," in *Proc. Statist. Signal Process. Workshop*. IEEE, 2016, pp. 1–5.

[30] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, 2018.

[31] J. Wen, Z. Zhang, Z. Zhang, L. Fei, and M. Wang, "Generalized incomplete multiview clustering with flexible locality structure diffusion," *IEEE Trans. Cybern.*, 2020.

[32] M. Ashraphijuo, X. Wang, and V. Aggarwal, "Fundamental sampling patterns for low-rank multi-view data completion," *Pattern Recognit.*, p. 107307, 2020.

[33] X. Guo, "Robust subspace segmentation by simultaneously learning data representations and their affinity matrix," in *Proc. Joint Conf. Artif. Intell.*, 2015, pp. 3547–3553.

[34] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[35] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[36] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image clustering using local discriminant models and global integration," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2761–2773, 2010.

[37] Y. Chen, S. Wang, F. Zheng, and Y. Cen, "Graph-regularized least squares regression for multi-view subspace clustering," *Knowl-Based Syst.*, p. 105482, 2020.

[38] Z. Zhang, L. Wang, S. Li, Y. Wang, Z. Zhang, Z. Zha, and M. Wang, "Adaptive structure-constrained robust latent low-rank coding for image recovery," in *Proc. IEEE Int. Conf. Data Min.* IEEE, 2019, pp. 846–855.

[39] X. Li, Z. Zhang, L. Zhang, and M. Wang, "Mutual-manifold regularized robust fast latent lrr for subspace recovery and learning," *Neural Comput. Appl.*, pp. 1–14, 2019.

[40] Y. Jia, H. Liu, J. Hou, and S. Kwong, "Clustering-aware graph construction: A joint learning perspective," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 6, pp. 357–370, 2020.

[41] Y. Jia, J. Hou, and S. Kwong, "Constrained clustering with dissimilarity propagation-guided graph-laplacian pca," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020.

[42] Y. Jia, S. Kwong, and J. Hou, "Semi-supervised spectral clustering with structured sparsity regularization," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 403–407, 2018.

[43] Y. Chen, X. Xiao, and Y. Zhou, "Jointly learning kernel representation tensor and affinity matrix for multi-view clustering," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 1985–1997, 2019.

[44] Y. Xie, J. Liu, Y. Qu, D. Tao, W. Zhang, L. Dai, and L. Ma, "Robust kernelized multiview self-representation for subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020.

[45] M. Yin, J. Gao, S. Xie, and Y. Guo, "Multiview subspace clustering via tensorial t-product representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 851–864, 2018.

[46] S. Gao, Z. Yu, T. Jin, and M. Yin, "Multi-view low-rank matrix factorization using multiple manifold regularization," *Neurocomputing*, vol. 335, pp. 143–152, 2019.

[47] Y. Jia, H. Liu, J. Hou, S. Kwong, and Q. Zhang, "Multi-view spectral clustering tailored tensor low-rank representation," *arXiv preprint arXiv:2004.14705*, 2020.

[48] M. Abavisani and V. M. Patel, "Deep multimodal subspace clustering networks," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 6, pp. 1601–1614, 2018.

[49] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu, "Generalized latent multi-view subspace clustering,"

IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 1, pp. 86–99, 2018.

[50] R. Li, C. Zhang, H. Fu, X. Peng, T. Zhou, and Q. Hu, "Reciprocal multi-layer subspace learning for multi-view clustering," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 8172–8180.

[51] C. Zhang, Y. Cui, Z. Han, J. T. Zhou, H. Fu, and Q. Hu, "Deep partial multi-view learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[52] M. Yin, W. Huang, and J. Gao, "Shared generative latent representation learning for multi-view clustering." in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 6688–6695.

[53] Y. Xie, B. Lin, Y. Qu, C. Li, W. Zhang, L. Ma, Y. Wen, and D. Tao, "Joint deep multi-view learning for image clustering," *IEEE Trans. Knowl. Data Eng.*, 2020.

[54] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2014.

[55] Y. Chen, Y. Guo, Y. Wang, D. Wang, C. Peng, and G. He, "Denoising of hyperspectral images using nonconvex low rank matrix approximation," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5366–5380, 2017.

[56] Y. Chen, S. Wang, and Y. Zhou, "Tensor nuclear norm-based low-rank approximation with total variation regularization," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 6, pp. 1364–1377, 2018.

[57] M. Yin, J. Gao, and Z. Lin, "Laplacian regularized low-rank representation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 504–517, 2016.

[58] Y. Chen, X. Xiao, and Y. Zhou, "Multi-view subspace clustering via simultaneously learning the representation tensor and affinity matrix," *Pattern Recognit.*, vol. 106, p. 107441, 2020.

[59] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[60] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[61] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[62] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, 2016.

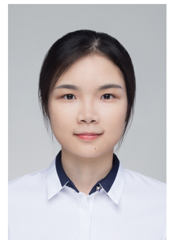
[63] Y. Chen, X. Xiao, and Y. Zhou, "Low-rank quaternion approximation for color image processing," *IEEE Trans. Image Process.*, vol. 29, pp. 1426–1439, 2019.

[64] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan, "Multi-task low-rank affinity pursuit for image segmentation," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2439–2446.

[65] H. Wang, Y. Yang, and T. Li, "Multi-view clustering via concept factorization with local manifold regularization," in *Proc. IEEE Int. Conf. Data Min.* IEEE, 2016, pp. 1245–1250.

[66] J. Wu, X. Xie, L. Nie, Z. Lin, and H. Zha, "Unified graph and low-rank tensor learning for multi-view clustering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 04, 2020, pp. 6388–6395.

Yongyong Chen received the B.S. and M.S. degrees from the Shandong University of Science and Technology, Qingdao, China, in 2014 and 2017, respectively, and the Ph.D. degree from the University of Macau, Macau, in 2020. He is currently an Assistant Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include image processing, data mining, and computer vision.



Xiaolin Xiao received the B.E. degree from Wuhan University, China, in 2013 and the Ph.D. degree from University of Macau, Macau, China, in 2019. Currently, she is a Postdoc Fellow with the School of Computer Science and Engineering, South China University of Technology, China. Her research interests include superpixel segmentation, saliency detection, and color image processing and understanding.

Chong Peng received his PhD in Computer Science from Southern Illinois University, Carbondale, IL, USA in 2017. Currently, he is an associate professor at College of Computer Science and Technology, Qingdao University. He has published more than 40 research papers in top-tier conferences and journals, including AAAI, ICDE, CVPR, SIGKDD, ICDM, CIKM, ACM TIST, TKDD, TOMM, IEEE TNNLS, TIP, TMM, TGRS, Pattern Recognition, Neural Networks, Information Sciences, etc. His research interests include pattern recognition, data mining, and machine learning.

Guangming Lu (Member, IEEE) received the bachelors degree in electrical engineering, the masters degree in control theory and control engineering, and the Ph.D. degree in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 1998, 2000, and 2005, respectively. From 2005 to 2007, he was a Postdoctoral Fellow with Tsinghua University. He is currently a Professor with the Harbin Institute of Technology, Shenzhen, China. He has published over 100 technical papers at some international journals and conferences, including IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, NeurIPS, CVPR, AAAI, and IJCAI. His research interests include computer vision, pattern recognition, and machine learning

Yicong Zhou (M07-SM14) received the B.S. degree in electrical engineering from Hunan University in Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA.

In 2011, he joined University of Macau as an Assistant Professor with the Department of Computer and Information Science, where he is currently an Associate Professor and a Director of the Vision and Image Processing Laboratory. His research interests include image processing, computer vision, machine learning, and multimedia security. He is a Fellow of the Society of Photo-Optical Instrumentation Engineers (SPIE). He received the Third Price of Macao Natural Science Award as a sole winner in 2020 and a co-winner in 2014. Since 2015, he has been a leading Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He serves as an Associate Editor for *IEEE Transactions on Neutral Networks and Learning Systems, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Geoscience and Remote Sensing*, and four other journals. He received the Best Editor Award in 2020 to recognize his contribution to Journal of Visual Communication and Image Representation. He was recognized as the "Highly Cited Researcher" in Web of Science in 2020.