Low-Rank Transfer Human Motion Segmentation

Lichen Wang, Zhengming Ding, *Member, IEEE*,
and Yun Fu, *Senior Member, IEEE*

***Abstract*— Human motion segmentation has a great potential in real-world applications. Conventional segmentation approaches cluster data with no guidance from prior knowledge, which could easily cause unpredictable segmentation output and decrease the performance. To this end, we seek to improve the human-motion segmentation performance by fully utilizing pre-existing well-labeled source data. Specifically, we design a new transfer subspace clustering method for motion segmentation with a weighted rank constraint. Specifically, our proposed model obtains the representations of both source and target sequences by mitigating their distribution divergence, which allows for more effective knowledge transfer to the target. To guide new representation learning, we designed a novel sequential graph to preserve temporal information residing in both the source and the target. Furthermore, a weighted low-rank constraint is added to enforce the graph regularizer and uncover clustering structures within data. Experiments are evaluated on four human motion databases, which prove the enhanced performance and increased stability of our model compared with state-of-the-art baselines.**

***Index Terms*— Human motion segmentation, low-rank learning, temporal data clustering.**

I. Introduction

THE goal of human motion segmentation is to cluster a long sequential motion into several short, non-overlapping sections. Such a segmentation is an important preprocessing step for a wide range of motion/action related analytical/recognition tasks. In a lot of real applications including security surveillance, motion analysis, and action recognition. Video sequences usually contain tens or even hundreds of continuous actions; however, most conventional action recognition approaches cannot handle these scenarios [1], [2], since they assume that each video only contains a single action. Thus, action segmentation approaches are required in order to divide the long videos before performing other analytical processes on the videos.

Manuscript received January 16, 2018; revised May 31, 2018 and July 24, 2018; accepted August 28, 2018. Date of publication September 18, 2018; date of current version October 26, 2018. This research is supported in part by the NSF IIS Award 1651902 and U.S. Army Research Office Award W911NF-17-1-0367. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jocelyn Chanussot. *(Corresponding author: Lichen Wang.)*

L. Wang is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: wanglichenxj@gmail.com).

Z. Ding is with the Department of Computer, Information and Technology Indiana University–Purdue University Indianapolis, Indianapolis, IN 46202 USA (e-mail: zd2@iu.edu).

Y. Fu is with the Department of Electrical and Computer Engineering, College of Computer and Information Science, Northeastern University, Boston, MA 02115 USA (e-mail: yunfu@ece.neu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2870945

Temporal data clustering is a challenging task [3], [4] which seeks robust and accurate clustering strategies to group coherent sequences together without the guidance of prior information. Compared with independent data, human motion data contain temporal information, which is critical for segmentation. A comprehensive survey [5] reveals that time series data clustering is difficult due to its complex temporal correlation and high dimensional data structure. Based on the categorization mentioned in [6], there are three lines of temporal clustering methods, including model-based [7], [8], temporal proximity based [5] and representation based algorithms [9]–[11]. Among them, the representation based approaches, especially subspace clustering algorithms, are most popular. Following this trend, our approach is also a representation based method through subspace clustering.

The major goal of subspace clustering is obtaining a distinctive and effective representation of the target data for clustering tasks. It achieves promising performances in a lot of clustering applications, including action recognition [12] and image clustering [13]. Several representative clustering approaches in subspace were designed recently, such as low-rank representation (LRR) [12] and sparse subspace clustering (SSC) [13]. The learned representation is used as input for pre-existing clustering algorithms (e.g. normalized-cut). Several modification methods are proposed to obtain representations for specific clustering tasks. Reference [14] added a dictionary into the model, which is simultaneously updated in the learning process to achieve distinctive coding performance. Reference [15] designed a new divide-and-conquer approach for data segmentation in large scale data. An active subspace clustering approach to solve the nuclear norm regularized with high efficiency is used to reduce computational complexity [16]. However, since subspace clustering methods are usually based on self-representation, insufficient or incorrectly applied data may hinder the clustering performance.

These clustering methods are based on an unsupervised learning scenario, and sometimes it is difficult to obtain reasonable and expected output without prior knowledge. On the contrary, supervised learning strategy which learns prior knowledge from labeled datasets is costing and expensive; hence, it is not ideal to solve the problem by labeling the data manually. Utilizing the information of related data to enhance the clustering result of the target data is a crucial approach. Fortunately, related and labeled data are easy to achieve. Transfer learning, an effective strategy when encountering learning performance with limited prior knowledge, is well explored in this scenario [17], [18]. It borrows

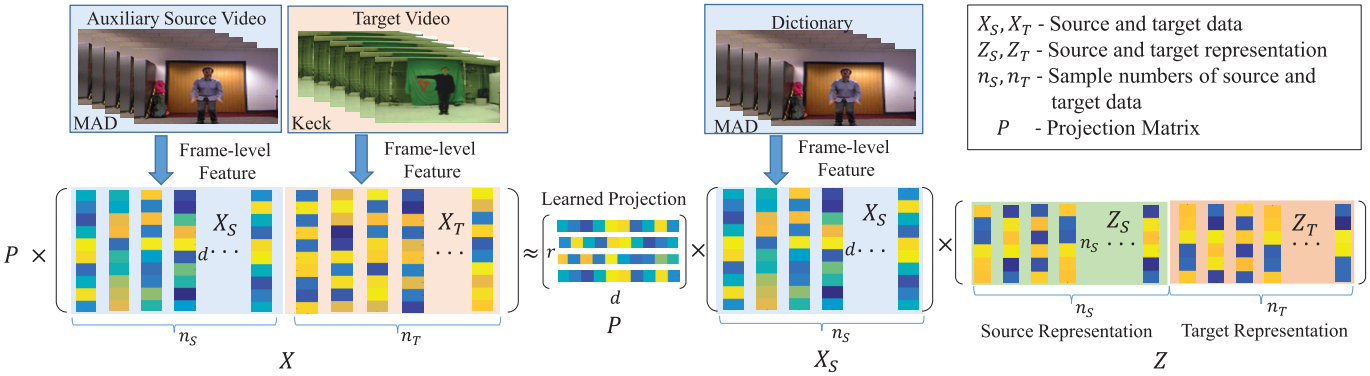


Fig. 1. Framework of our approach, where well-labeled source data X_S are set as a basis for our model to learn representations of both X_S and X_T under a reconstruction scheme. This model transfers the data structure from the source to effectively learn the target representation. Since the distribution of source and target data are different, a domain-invariant mapping P is learned simultaneously to align different distributions into a low-dimensional and distinctive common space. Furthermore, a weighted rank constraint is utilized to enhance the effectiveness of the model and eliminate noise or outlier influences in the sequential graph.

knowledge from relevant source information to improve the target tasks.

To this end, we further explore transfer learning and propose a novel human motion clustering approach (Fig. 1). The core idea is to adapt well-labeled source knowledge from relevant source datasets to improve the clustering performance on the target motion data. To the best of our knowledge, we are the first try to explore transfer learning in unsupervised clustering problems. The novelties of this work are listed as follows:

- A novel transfer learning based subspace clustering approach is proposed. We aim to transfer knowledge from relevant source data to improve the clustering performance of target data.
- A graph regularizer is built to uncover the temporal and structural information residing in both the target and source dataset. This approach obtains more distinctive and effective representations for subspace clustering. Furthermore, we explore a rank constraint on the graph regularizer to uncover additional structural information.
- A domain-invariant projection is introduced and simultaneously trained during knowledge transfer, which aligns the source and target features, which have different distributions, into a new and common space.

II. Related Work

We introduce several related works including subspace clustering, temporal data clustering, as well as transfer learning in this section.

Subspace Clustering is an extension of traditional clustering methods. Traditional methods group similar data points into the same cluster based directly on feature similarity. Subspace clustering searches a subspace and finds the clusters from a database and groups data inside a new and more distinctive space. It has attracted increasing attention since it achieved high clustering results in a lot of challenging tasks. Sparse subspace clustering (SSC) [13] enforces sparse constraints on the coefficients and learns a sparse representation from row data. Low-rank representation learning (LRR) [12] considers the data structure globally. It seeks the lowest-ranked representation in a given dictionary and usually achieves better performance directly on the learned representations. Least-square regression (LSR) [19] tries to group data samples with high correlation together by deploying the Frobenius norm. Discriminative subspace clustering method (DSC) [20] deploys a quadratic classifier which is trained by unlabeled data to obtain the discriminative information. A new affinity measurement is also proposed which is more effective than the commonly used one. In addition, many efforts have been devised to reduce the computational cost [15], [21], [22]. Unfortunately, these methods are not well designed for temporal data clustering, especially for human motion segmentation.

They do not consider spatial connection or consider the global structural information, neglecting the temporal clues residing inside the data. Our approach utilizes a graph regularizer as well as rank constraint to preserve the temporal structural information in the learned representations, which is more accurate for time series clustering tasks.

Temporal Data Clustering aims to cut long sequential data and achieve several short groups without any overlaps. It is a necessary technique for a lot of real-world scenarios, including natural language processing, human action recognition and facial behavior analysis. Few methods have been developed for human motion segmentation. An extension of Dynamic Bayesian Networks (DBNs) [23] adds a truncated approximation to the Dirichlet process to improve the flexibility of the clustering model. Semi-Markov K-means clustering [24] is designed to explore repetitive patterns in the temporal data for clustering. Hierarchical cluster analysis [25] proposes a K-means kernel associated with a dynamic temporal alignment approach for temporal data clustering. Maximum-margin temporal clustering [26] obtains each segment position and learns a multiple-class Support Vector Machine simultaneously. Temporal Subspace Clustering (TSC) [14] learns both a dictionary and data representation on the constraint of a temporal regulation. These temporal clustering methods belong to an unsupervised learning scenario, which utilizes a self-representation strategy and focuses on digging clustering

information from the target data itself, which is difficult to properly cluster temporal data into a reasonable, meaningful and expected result without any supervision or guidance, so we propose a transfer learning based segmentation approach which digs information from labeled source data and transforms the knowledge to facilitate the clustering performance.

Transfer Learning transfers prior knowledge from similar source data to enhance the result of target tasks. It is effective in solving problems involving tasks which have only limited training samples. Details about transfer learning are introduce in the work of [27]. Three settings are arranged based on [27]: unsupervised transfer learning [28], inductive transfer learning [29], and transductive transfer learning [30], [31]. Our approach belongs to the transductive transfer scenario, where the tasks of the source and target data but the domain (or data distribution region) are different.

Transductive transfer learning transforms the information from the source domain by manipulating the representation [32]. Domain shift is the biggest challenge of transfer learning because the samples of the source and target distribute in inconsistent regions in the feature space. One solution is for the model to use a well-aligned data representation which attempts to project the source and target into another common space where the distribution gap can be mitigated. In [33], external source data is set as a dictionary for the model to learn a different representation of target data for object recognition tasks. Intermediate representations of data between the target and source domains are used in unsupervised domain adaptation object recognition [34]. A low-rank constraint based reconstruction approach which is proposed to mitigate the domain distribution difference is proposed to transfer the information source data into an intermediate space [35]. Our approach also belongs to this line, but rather than classifying individual tasks, which aims to obtain a more distinctive representation for clustering tasks.

This work is the extension of our previous paper [36]. Compared with [36], we implement the previous temporal graph regularizer with a weighted rank constraint, which tends to uncover the global structure within data while preserving more temporal and sequential knowledge. Furthermore, we involve one more new action dataset, evaluate it with more cross-domain tasks, and achieve more quantitative results. These results show a comprehensive and convincing demonstration of the effectiveness and stability of our approach. Extensive experiments and discussion on modified source and target data also illustrate the detailed property of our approach. The results indicate that the proposed approach achieves better performance than previous models.

III. The Proposed Approach

We first present the motivation and problem definition. Then, the details of our model are provided. In the end, we give the solutions of the proposed model. Table I summarizes the notations utilized throughout this work for better illustration. Lowercase letters represent scale values and matrices are represented by uppercase letters. Assume $X = [x_1, x_2, x_3, \cdots, x_n]$ is the data samples, and n is sample number which belong to a set of several subspaces $\{\mathbb{S}\}_{i=1}^{k}$. k represents cluster number. The goal is to segment X_T to the corresponding clusters.

TABLE I
Symbol Description

Symbol	Description
X_S	Feature of labeled source data.
X_T	Feature of unlabeled target data.
X	Concatenated source and target data.
Z_S	Learned representation of X_S.
Z_T	Learned representation of X_T.
W	Generated weight regulation matrix.
L_W	Graph Laplace matrix.
P	Learned feature projection.
λ_1, λ_2	Trade-off parameter.

A. Motivation

Conventional subspace clustering approaches [14], [37] explore self-representation strategy, where it is difficult to achieve meaningful and expected output without any prior knowledge guidance. These supervised approaches are not ideal since achieving labeled data for specific tasks is costing and labor intensive. Therefore, we aim to utilize motion knowledge residing in source dataset to enhance the target data clustering performance. Since temporal information residing inside the temporal motion data samples, we propose a graph regularizer to preserve the temporal structure. Moreover, a low-rank constraint is explored to reveal the distinctive structure in the learned latent structure and improve the clustering performance.

B. Learning Transferable Representation

Segmenting the target data directly in the original feature space is challenging since the distribution structure in the feature space is not distinctive; because of this fact, the performance of the segmentation would be low. For this, we set labeled source data as a dictionary to reconstruct both source and target data. The result is that the source and target would lie in the same feature space. The reconstruction formulation is written as follows:

$$X \approx X_S Z, \tag{1}$$

where X is the concatenation of source and target samples, and $X = [X_S, X_T] \in \mathbb{R}^{d \times n}$. Each column represents a sample, and $X_S \in \mathbb{R}^{d \times n_S}$, $X_T \in \mathbb{R}^{d \times n_T}$ are feature matrix. feature dimension is represented by d, X_S and X_T have the sample number of n_S and n_T. $n = n_S + n_T$. Z is the learned representation of X and $Z = [z_1, z_2, z_3, \cdots, z_n]$, where each z_i represents corresponding sample. $Z = [Z_S, Z_T] \in \mathbb{R}^{n_S \times n}$ is the learned representations of both source and target based on Eq. (1), i.e., $Z_T \in \mathbb{R}^{n_S \times n_T}$, $Z_S \in \mathbb{R}^{n_S \times n_S}$.

As we mentioned before, X_S and X_T have different feature distributions since they are achieved from different sources. If X_S is directly used for coding the target data, high reconstruction error would be involved in the learned representation Z. Therefore, we further explore a domain-invariant

projection P to project X_S and X_T for knowledge transfer. $P \in \mathbb{R}^{r\times d}$ where r regulate the dimension of P and also the dimension of the projected common space. The purpose of P is to search a common and discriminative space which can project the data samples to the space and reduce the distribution difference between the target and source data. Thus, we rewrite the new reconstruction formulation by seeking a new feature space and transferring knowledge, shown as follows:

$$PX \approx PX_S Z. \tag{2}$$

To implement the model of Eq. (2), we propose a least-square regression based formulation as follows:

$$\min_{P,Z} \|PX - PX_S Z\|_{\mathrm{F}}^2 + \lambda_1 \|P\|_{\mathrm{F}}^2, \tag{3}$$

where $\|P\|_{\mathrm{F}}$ is the Frobenius norm of P with a trade-off parameter λ_1, and $\|P\|_{\mathrm{F}}^2 = \sum_{i=1}^{r}\sum_{j=1}^{d} |P_{i,j}|^2$ specifically. $\|PX - PX_S Z\|_{\mathrm{F}}^2$ is used to minimize the reconstruction error, $\lambda_1\|P\|_{\mathrm{F}}^2$ is designed to constrain the variable scale. There is a simple format for Eq. (2) which directly obtains Z_T without concatenating X_S, such as $PX_T \approx PX_S Z_T$. However, we expect to transfer more effective information between the source and target data. Projecting both the target and source could obtain more general P. In Eq. (3), the first term can be decomposed to $\|P[X_S, X_T] - PX_S[Z_S, Z_T]\|_{\mathrm{F}}^2 = \|PX_S - PX_S Z_S\|_{\mathrm{F}}^2 + \|PX_T - PX_S Z_T\|_{\mathrm{F}}^2$. $\|PX_S - PX_S Z_S\|_{\mathrm{F}}^2$ is necessary which is another crucial term to transfer knowledge to P to obtain distinctive subspace. Quantitative evaluation in following section will further prove this claim. Previous work [19] has demonstrated that the Frobenius norm is an effective process to preserve structural information in Z, which is the key for clustering.

C. Temporal Graph Regularizer

Since human motion data are consecutive and sequential, the temporal and structural information is a crucial clue for accurate clustering. We expect to further preserve the temporal information in Z for more accurate and robust clustering performance. To this end, a graph regularizer $T(Z)$ was designed to incorporate the temporal information into Z. A graph based regularizer strategy [38] is an effective method to obtain an effective representation. It respects the intrinsic geometric structure and reveals the hidden semantic structures. The purpose of $T(Z)$ is to make the neighbors of learned representation samples be close. By adding a rank constraint, we further make the data structure of Z more distinctive. $T(Z)$ would regulate that its neighbors $[z_{i-s/2}, \cdots, z_{i-3}, z_{i-2}, z_{i-1}, z_{i+1}, z_{i+2}, \cdots, z_{i+s/2}]$ be close to z_i, where s is the length of relevant frames. We first propose a graph regularizer $T(Z)$ to pull the similarities of nearby representation points and the regularizer is shown below:

$$T(Z) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\|z_i - z_j\|_2^2 = \mathrm{tr}(ZL_W Z^\top). \tag{4}$$

In Eq. (4), tr(.) represents a matrix trace and it is defined to be the main diagonal elements sum. The graph Laplacian matrix $L_W = D - W$ [39], where $D_{ii} = \sum_{i=1}^{n} w_{ij}$ and $W \in \mathbb{R}^{n\times n}$ is the weight regularization matrix. Each element of W is shown below:

$$w_{ij} = \begin{cases} 1, & \text{if } |i-j| \le s, l(x_i) = l(x_j), \text{for source} \\ 1, & \text{if } |i-j| \le s, \text{for target} \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $l(x_i)$ represents the action class/label of z_i in the source data. The model assigns the samples which belong to the same group get the temporal constraints in the source, so that they are able to fully utilize labeled information from the source. Furthermore, there is no requirement that the specific action classes in the source and target should be overlapped. The model still works well even if no overlaps exist between the two action datasets. In addition, since segmentation labels in the target data are unknown, we can assume the whole target data representation has temporal property.

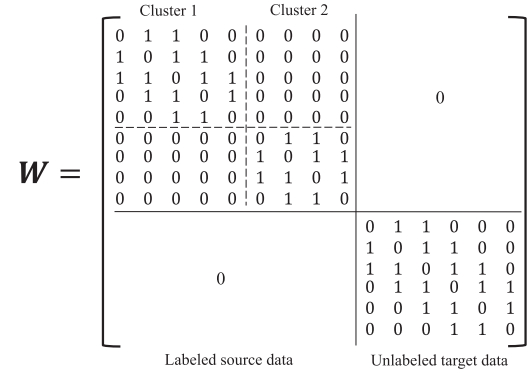


Fig. 2. The structure of W in a simple case. If we have $s = 2$, $n_S = 9$ and $n_T = 6$. We can see that when the distance of ith and jth frame-level features is less than s, the $W_{ij} = 1$ would regulate the learned representation to be close. Furthermore, the correlation weight between two different groups in source data is totally zeros to involve the labeled information to the model.

From Eq. (5), we observe that when the distance of the ith and jth frame features is less than s, $T(Z)$ would regulate the learned representation to be close. If the distance is greater than s, there is no regularization between the two samples. To better illustrate how $T(Z)$ encodes temporal data structure, we generate a sample of W in a simple case. When $s = 2$, $n_S = 9$ and $n_T = 6$, the generated weight matrix W is shown in Figure 2. The correlation weight between two different groups in the source data are all zeros, so we further involve segmented information into the model to constrain the coding result. We only deployed binary weights to construct W to show the core idea of temporal constraint. We consider the frames to have the same level of similarity within s neighbors no matter the distance between them. The other sophisticated continuous value W for temporal preservation is also applicable to have higher segmentation performance. For example, a continuous version $W_{ij} = \alpha e^{-\frac{|i-j|}{2\delta^2}}$, if $|i-j| \le s$ can also be utilized. It assumes that the closer the two frames are, the more similar their representations should be. We briefly test the performance based on continuous value W and achieve roughly 0.5% performance improvement; the result proves the effectiveness of this strategy. However, considering this

is not our major research topic and this strategy requires a sophisticated parameter tuning process, we still utilize binary W in our implementation.

We transform Eq. (4) using the Eigen-decomposition technique on L_W. After matrix manipulation, we obtain the equation below:

$$\begin{aligned}\mathrm{tr}(ZL_WZ^T) &= \mathrm{tr}(ZU_WS_WU_W^\top Z)\\ &= \mathrm{tr}(ZU_WS_W^{\frac{1}{2}}S_W^{\frac{1}{2}}U_W^\top Z) = \|ZA_W\|_{\mathrm{F}}^2, \quad (6)\end{aligned}$$

where $A_W = U_WS_W^{\frac{1}{2}}$. Based on several previous works [40], real-world data, especially human related data (human face and gesture) [40], has strong rank structure. The rank constraint could further reveal the subspace structure. Thus, this term could achieve more effective and distinctive representation than Frobenius norm.

Directly seeking the lowest rank of ZA_W is difficult since rank function has discrete nature. We follow the work of [12] and replace Frobenius norm on ZA_W by the nuclear norm and modify the challenging problem to a convex optimization problem. Then we obtain $\|ZA_W\|_{\mathrm{F}}^2 \rightarrow \|ZA_W\|_*$ where $\|\cdot\|_*$ demotes nuclear norm. This term can involve low-rank property into the learned representation.

Moreover, the work of Cai *et al.* [38] indicates that a non-negative matrix factorization could further improve the discriminating power of the learned representations. A non-negative constraint $Z \geq 0$ was added to the objective function. Thus, we achieve our objective function below:

$$\begin{aligned}&\min_{P,Z}\ \|PX - PX_sZ\|_{\mathrm{F}}^2 + \lambda_1\|P\|_{\mathrm{F}}^2 + \lambda_2\|ZA_W\|_*,\\ &\text{s.t. } Z \geq 0, \quad PXHX^\top P^\top = I, \quad (7)\end{aligned}$$

where the trade-off parameters, λ_1 and λ_2, are used to balance weights of the terms. The constraint $PXHX^\top P^\top = I$ would preserve the data variance after projection, which could further enhance the discriminability of the learned representation. $H = \mathrm{I} - \frac{1}{n}\mathbf{1}$ is centering matrix, where I is an identity matrix and $\mathbf{1} \in \mathbb{R}^{n\times n}$ is the matrix of ones.

D. Clustering

After achieving the learned representation Z, the corresponding target representation Z_T is cut out from Z where $Z = [Z_S, Z_T]$. And we cluster Z_T to achieve our final goal. A graph G is generated for existing clustering method in the final stage. Previous clustering methods such as LRR [12] and SSC [13] commonly defined the weights of clustering graph as $(|Z| + |Z^\top|)/2$; however, the within-cluster samples in human motion data are always highly correlated with each other [41] and the commonly used G cannot exploit the intrinsic relationships well. To take advantage of this property, we follow the strategy of [14] and devise another similarity measurement to construct the graph $G \in \mathbb{R}^{n\times n}$. The weight matrix W_G of the graph is defined on the distance between each pair of the representation samples as follows:

$$W_G(i,j) = \frac{z_i^\top z_j}{\|z_i\|_2\|z_j\|_2}. \quad (8)$$

After the weight matrix W_G is obtained, we utilize an effective conventional clustering approach, Normalized Cuts (NCut) [42], is used to obtain the segment results. The cluster number is known in our implementation.

E. Optimization

Solving Eq. (7) is a challenging problem since it is hard to directly get the explicit solutions. Thus, we utilize the alternating direction method of multipliers (ADMM) [43]. It optimizes each variable by fixing others. Two auxiliary variable $V \in \mathbb{R}^{n_S\times n}$ and $U \in \mathbb{R}^{n\times n}$ are used in the optimization algorithm. We transform Eq. (7) as follows:

$$\begin{aligned}&\min_{P,V,Z,U}\ \|PX - PX_SV\|_{\mathrm{F}}^2 + \lambda_1\|P\|_{\mathrm{F}}^2 + \lambda_2\|U\|_*,\\ &\text{s.t. } V = Z, \quad U = ZA_W, \ Z \geq 0, \ PXHX^\top P^\top = I. \quad (9)\end{aligned}$$

Eq. (9) is converted to an augmented Lagrangian function [44], and the expression is shown below:

$$\begin{aligned}\mathcal{L} = &\frac{1}{2}\|PX - PX_SV\|_{\mathrm{F}}^2 + \lambda_1\|P\|_{\mathrm{F}}^2 + \lambda_2\|U\|_*\\ &+ \langle\Lambda_1, V - Z\rangle + \frac{\mu}{2}\|V - Z\|_{\mathrm{F}}^2\\ &+ \langle\Lambda_2, U - VA_W\rangle + \frac{\mu}{2}\|U - VA_W\|_{\mathrm{F}}^2,\\ &\text{s.t. } U = ZA_W, V = Z, PXHX^\top P^\top = I, Z \geq 0, \quad (10)\end{aligned}$$

where $\Lambda_1 \in \mathbb{R}^{n_S\times n}$ and $\Lambda_2 \in \mathbb{R}^{n\times n}$ are Lagrangian multipliers. μ is the parameter for penalty. Eq. (9) can be derived by alternatively minimizing the value of $\mathcal{L}$ with respect to Z, V and P. During minimization process, other variables are fixed and we only update one variable each time until the equation is convergent. P and Z are initialized with random value. V, Λ_1 and Λ_2 are initialized with zero matrix.

Update V: By ignoring other fixed variables, the Lagrangian equation (10) can be written as follow:

$$\begin{aligned}\min_V\ &\frac{1}{2}\|PX - PX_SV\|_{\mathrm{F}}^2 + \langle\Lambda_1, V - Z\rangle + \frac{\mu}{2}\|V - Z\|_{\mathrm{F}}^2\\ &+ \langle\Lambda_2, U - VA_W\rangle + \frac{\mu}{2}\|U - VA_W\|_{\mathrm{F}}^2. \quad (11)\end{aligned}$$

We set the derivation of $\mathcal{L}$ with respect of V to 0, as $\frac{\partial\mathcal{L}}{\partial V} = 0$. The equation is shown below:

$$\begin{aligned}&(-PX_S)^T(-PX_SV + PX) + \Lambda_1 + \mu(V - Z)\\ &\quad + \Lambda_2(-A_W) + \mu(VA_W - U)A_W^\top = 0. \quad (12)\end{aligned}$$

Then we can get the following equation:

$$\begin{aligned}&[-(PX_S)^TPX + \Lambda - \mu Z - \Lambda_1A_W^T - \mu UA_W^T]\\ &\quad + [(PX_S)^TPX_S + \mu I]V + V[\mu A_WA_W^T] = 0. \quad (13)\end{aligned}$$

We deploy Bartels-Stewart algorithm [45] to solve Eq. (13). It is an effective way to solve a standard Sylvester equation.

Update P: We simplify the equation by converting Eq. (10) from Frobenius norm to trace format. The transformed equation is below:

$$\begin{aligned} P = & \underset{PXHX^\top P^\top = I}{\arg\min} \|PX - PX_S Z\|_{\mathrm{F}}^2 + \lambda_1 \|P\|_{\mathrm{F}}^2 \\ = & \underset{PXHX^\top P^\top = I}{\arg\min} \operatorname{tr}(P[(X - X_S Z)(X - X_S Z)^\top \\ & + \lambda_1 I]P^\top). \end{aligned} \tag{14}$$

We utilize the generalized Eigen-decomposition to solve Eq. (14) which is shown below:

$$[(X - X_S Z)(X - X_S Z)^\top + \lambda_1 I]\rho = \gamma XHX^\top \rho, \tag{15}$$

where ρ is the eigenvector from the Eigen-decomposition result, and γ is the value eigenvalue of ρ. The eigenvector vectors $\rho_i (i = 0, 1, 2, \cdots, p-2, p-1)$ which can minimize Eq. (10) are obtained from the minimum eigenvalue solutions to the eigenvalue problem. Moreover, $P = [\rho_0, \rho_1, \cdots, \rho_{p-2}, \rho_{p-1}]^\top$.

Update U: By ignoring other variables, we separate U and obtain the following equation:

$$\min_Z \ \lambda_2 \|U\|_* + \langle \Lambda_2, U - ZA_W \rangle + \frac{\mu}{2} \|U - ZA_W\|_{\mathrm{F}}^2. \tag{16}$$

We utilize Singular Value Thresholding (SVT) [46] operator to effectively solve the equation.

Update Z: Eq. (10) can be written as follows when ignoring other variables:

$$\min_Z \ \langle \Lambda_1, V - Z \rangle + \frac{\mu}{2} \|V - Z\|_{\mathrm{F}}^2. \tag{17}$$

Eq. (17) has a closed-formed solution which is $Z = V + \frac{\Lambda_1}{\mu}$. In order to satisfy the non-negative constraint for new representation Z, the update rule is written as follows:

$$Z = F_+(V + \frac{\Lambda_1}{\mu}), \tag{18}$$

where $F_+(.)$ is a non-negative function, $(F_+(A))_{ij} = \max(A_{ij}, 0)$. And A_{ij} is an element in matrix A. The complete clustering approach and solution steps are shown in Algorithm 1. The update steps are iteratively executed several times until the equation is convergent.

F. Complexity Analysis

There are three time-consuming processes in optimization. The first is process is Step 3 (Updating V) by using Bartels Stewart algorithm, its complexity is $\mathbf{O}(n_S^2\, n)$. Second is Step 4 (Eigen-decomposition) which costs $\mathbf{O}(d^3)$, where d is the representation dimension. Third is Step 5 (Updating U) by deploying the ALM approach. ALM approach is viable for large scale dataset, based on the discussion of [12], the computational complexity is $\mathbf{O}(n_S^2\, n + n_S^3)$. To this end, the major computational section of our approach is $\mathbf{O}(td^3 + 2tn_S^2 + tn_S^3)$ where t is the iterations number. There is still space to further reduce the complexity. First, the source data can be evenly reduced to a smaller size to speed up the optimization process without losing the segmentation performance; we will discuss the details in the next section. Second, Coppersmith-Winograd algorithm [47] can reduce the Eigen-decomposition process to $O(d^{2.376})$. Thus, our approach is efficient and scalable for real-world applications.

Algorithm 1 Human Action Subspace Clustering

Input: Source and target feature matrix X_S and X_T, step size η, cluster number k, parameters $\lambda_1, \lambda_2, s, \mu$
Output: Index vector of clustering result Y
1: Generate temporal matrix W, L_W, and A_W
2: **while** have not converged **do**
3: Update $V_{(k+1)}$ based on (13), others variables are fixed;
4: Update $P_{(k+1)}$ based on (14), others variables are fixed;
5: Update $U_{(k+1)}$ based on (16), others variables are fixed;
6: Update $Z_{(k+1)}$ based on (17), others variables are fixed;
7: Update $\Lambda_{1(k+1)}$, $\Lambda_{1(k+1)} = \Lambda_{1(k)} + \eta\mu(V_{k+1} - Z_{k+1})$;
8: Update $\Lambda_{2(k+1)}$, $\Lambda_{2(k+1)} = \Lambda_{2(k)} + \eta\mu(U_{k+1} - Z_{k+1}A_W)$;
9: $k = k + 1$
10: **end while**
11: Building an undirected graph G based on Eq. (8)
12: Obtain k clusters based on G by NCut and output Y

IV. Experiment

We evaluate our method associate with several state-of-the-art representative subspace clustering methods on human motion datasets.

A. Human Motion Datasets

Four human motion datasets were used in the experiments, including the Weizmann dataset [48], Keck dataset [49], Multi-Modal Action Detection dataset [50] and UT-Interaction dataset [51]. Brief introductions are listed below:

Multi-Modal Action Detection Dataset (MAD) [50] contains actions captured in multiple modals by a Microsoft Kinect V2 system in RGB, depth and skeleton formats. The RGB frames are in the resolution of 240 × 320f and 3D depth images are in the resolution of 240 × 320. The human skeleton information captured 20 body joints into a coordinate. All formats were captured at 30 fps. Each subject performed 35 actions in two different indoor environments.

Weizmann Dataset (Weiz) [48] has 90 video sequences including 10 actions performed by nine subjects in an outdoor environment. The video resolution is 180 × 144 with at 50 fps. All subjects perform ten everyday actions including running, walking, skipping, bending and so on.

Keck Gesture Dataset (Keck) [49] includes 14 different actions based on military signals. RGB frames are in resolution of 640 × 480. Three subjects performed 14 gestures and actions. The videos were recorded using a fixed camera with the subjects standing in front of a static and simple background.

UT-Interaction Dataset (UT) [51] has 20 videos and each videos contain 6 classes of human-human interactions including punching, kicking, pushing, hugging, pointing, and hand shaking. Each video sequence is around 1 minute. The resolution of the videos is 720 × 480 at 30 fps.

B. Experimental Setup

Low-level HoG features [52] are extracted from each frame of the corresponding videos to obtain 324-dimensional feature vectors. The major reason we selected HoG feature is that HoG feature is a frame-level feature. This kind of feature could provide clustering approaches with more flexibility when performing the segmentation tasks.

Due to the differences across training datasets, we standardized the input data to make a more accurate model to segment the given video. To achieve this, all input videos were modified so that each video became a sequence of 10 actions. Both the Weizmann and Keck datasets only contained a single action per video, and had to be concatenated using the same setting as [26]. The MAD dataset contained videos with sequences of greater than 10 actions, which was cut down to fit the standard used for testing. Once these sequences were created, they were randomly chosen in groups of 5 sequences, tested against the target sequence, and their performances were recorded. The final reported clustering performance is based on the average of every tested sequence group.

λ_1 and λ_2 are set to be 0.1 and 0.2 as default, the length of correlated frames s is set to 7 and the projection size r is 80. The parameter sensitivity will be evaluated in the next section. We evaluate our approach associate with three classical as well as six state-of-the-art clustering methods. Brief introductions of compared methods are listed below:

- K-means (KMS) [53]. K-means tries to cluster each observation to the group based on the nearest mean, thus minimizing the within-cluster sum of squares.
- K-medoids (KMD) [54]. Unlike K-means approach, k-medoids selects target samples as centers and cluster with a generalization of the Manhattan Norm to define the distance between points instead of l_2.
- Spectral Clustering (SPE) [55]. the spectrum of the similarity matrix of the target samples are utilized by spectral clustering to implement dimensionality reduction to achieve high clustering performance.
- Low-Rank Representation (LRR) [12]. LRR learns and obtains the lowest rank representation of the data samples. LRR effectively obtains the global structure of data samples, delivering more robust segmentation performance from corrupted data associated with high level outliers.
- Ordered Subspace Clustering (OSC) [37]. OSC proposed a temporal constraint and explicitly enforces the temporal data representation to be similar.
- Sparse Subspace Clustering (SSC) [13]. SSC assumes that a dictionary exists which can represent all data points by a sparsely combination. It proposes a sparse constraint to obtain the coefficients. It learns an effective sparse representation for clustering.
- Least Square Regression (LSR) [56]. LSR utilizes the Frobenius norm to encourage a grouping effect which tends to group highly correlated data together.
- Temporal Subspace Clustering (TSC) [14]. TSC proposes a temporal Laplacian regularization as well as a jointly learned dictionary to obtain expressive and distinctive codings for time series data.
- Transfer Subspace Segmentation (TSS) [36]. TSS approach also utilizes auxiliary data and transfers segmentation knowledge from source to target dataset.

In our experiments, the codes of the methods are obtained from authors and the parameters are tuned to achieve the best performances. Normalized Mutual Information (NMI) and Accuracy (ACC), two widely used metrics for cluster validity [57], are utilized as metrics for evaluating our model. ACC comes from classification with the best mapping. NMI evaluates the mutual information across the ground truth and the recovered cluster labels based on a normalization operation. The expressions are shown as follows:

$$ACC = \sum_{i=1}^{n} \delta(s_i \cdot map(r_i))/n,$$
$$NMI = \frac{\sum_{i,j} n_{ij} \log \frac{n n_{ij}}{n_{i+} n_{+j}}}{\sqrt{(\sum_i n_{i+} \log \frac{n_{i+}}{n})(\sum_j n_{j+} \log \frac{n_{+j}}{n})}}, \quad (19)$$

where $map(r_i)$ is the permutation mapping function which maps each cluster label r_i to the ground truth s_i. $\delta(x, y)$ is the Kronecker delta function. $\delta(x, y) = 1$ if $x = y$, and $\delta(x, y) = 0$ otherwise. Both ACC and NMI are positive measurements which means the higher the number is the better the performance.

C. Performance Comparison

In the experiments, we set one sequence as source and another one as target. Since four datasets were used for evaluation, we segmented test videos based on the other three datasets as source. For comparable methods such as LRR, OSC, SSC and LSR, we only input target videos, since the methods are not designed to utilize source information. For TSC and TSS methods, we input both source and target videos for segmentation. The results are listed in Table II. The results show that our approach outperforms other methods. Compared with the second best approach, TSS, our approach achieves averagely 5% higher performance in terms of accuracy. We also concatenate source and target data as input to TSC approach, and we can observe that the clustering result drops about 1% in TSC. These results indicate that simply increasing data samples cannot improve the clustering performance. Second, the clustering performance would reduce if the model cannot align the source and target effectively. The result demonstrates that our approach is able to align different distributions of two datasets and transfer useful information to improve the segmentation performance.

We visualize segmentation results from one sample of our approach and other compared methods in Figure 3. Different colors indicate different action clusters. From visualization results, we notice that the results of SSC, LRR and LSR are unacceptable. They generate multiple fragments due to the lack of the consideration for time sequential connections across near neighbor frames. OSC performs better but still suffers from rhythmed actions such as working and waving hand. The fragments are still significant. TSC and TSS have much better performance, however, they are still not sensitive and occasionally generate fragments in segmentation results. However, from Figure 3 we find out that our approach

TABLE II

CLUSTERING PERFORMANCE ACCURACIES (ACC) AND NMI. EACH SUBTABLE CORRESPONDING TO A DATASET. NAMES IN BRACKETS INDICATE THE SOURCE DATASETS. M, K, W AND U REPRESENT MAD, KECK, WEIZMANN AND UT-INTERACTION, RESPECTIVELY. BOLD FONT DENOTES THE BEST PERFORMANCE COMPARED WITH THE METHODS USING THE SAME SOURCE. (a) RESULT OF MAD DATASET. (b) RESULT OF KECK DATASET. (c) RESULT OF WEIZMANN DATASET. (d) RESULT OF UT DATASET

(a)

Method	ACC	NMI
KMS	0.3541	0.4188
KMD	0.3226	0.3914
SPE	0.3639	0.4369
LRR	0.2397	0.2249
OSC	0.4327	0.5589
SSC	0.3817	0.4758
LSR	0.3979	0.3667
TSC	0.5556	0.7721
TSC(W)	0.5418	0.7684
TSC(K)	0.5473	0.7691
TSC(U)	0.5315	0.7691
TSS(W)	0.5736	0.8202
TSS(K)	0.5792	**0.8286**
TSS(U)	0.5479	0.8108
Ours(W)	**0.5906**	**0.8213**
Ours(K)	**0.5874**	0.8244
Ours(U)	**0.5980**	**0.8211**

(b)

Method	ACC	NMI
KMS	0.3510	0.4553
KMD	0.3970	0.4702
SPE	0.3886	0.4744
LRR	0.4297	0.4862
OSC	0.4393	0.5931
SSC	0.3137	0.3858
LSR	0.4894	0.4548
TSC	0.4781	0.7129
TSC(M)	0.4653	0.6935
TSC(W)	0.4548	0.6862
TSC(U)	0.4421	0.6797
TSS(M)	0.5395	0.8049
TSS(W)	0.5485	0.7928
TSS(U)	0.4951	0.7937
Ours(M)	**0.5509**	**0.8226**
Ours(W)	**0.5649**	**0.7983**
Ours(U)	**0.5519**	**0.7974**

(c)

Method	ACC	NMI
KMS	0.4081	0.5562
KMD	0.4441	0.5289
SPE	0.4127	0.5435
LRR	0.3638	0.4382
OSC	0.5216	0.7047
SSC	0.4576	0.6009
LSR	0.5091	0.5093
TSC	0.6111	0.8199
TSC(M)	0.5961	0.8032
TSC(K)	0.5931	0.7971
TSC(U)	0.5402	0.7796
TSS(M)	**0.6208**	0.8509
TSS(K)	0.6030	0.8326
TSS(U)	0.5865	0.8124
Ours(M)	0.6156	**0.8579**
Ours(K)	**0.6391**	**0.8599**
Ours(U)	**0.6122**	**0.8267**

(d)

Method	ACC	NMI
KMS	0.4712	0.5677
KMD	0.5122	0.5108
SPE	0.4477	0.4894
LRR	0.4162	0.4051
OSC	0.5846	0.6877
SSC	0.4389	0.4998
LSR	0.5183	0.4322
TSC	0.5340	0.7593
TSC(M)	0.5288	0.7442
TSC(K)	0.5213	0.7216
TSC(W)	0.5111	0.7136
TSS(M)	0.5535	0.7783
TSS(K)	0.5371	0.7746
TSS(W)	0.5944	0.7878
Ours(M)	**0.6299**	**0.8128**
Ours(K)	**0.6127**	**0.7961**
Ours(W)	**0.6296**	**0.8035**

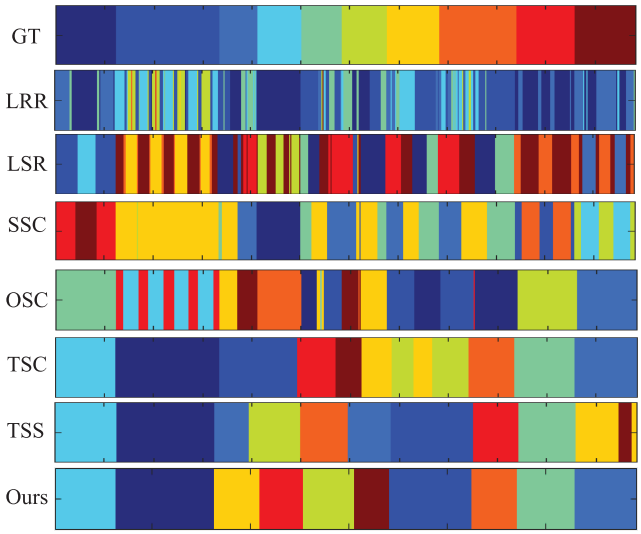


Fig. 3. Visualization of clustering results. The 10 colors denote 10 different temporal clusters. The first row is the ground truth of 10 clusters. The result illustrates that LRR, LSR and SSC are unable to segment temporal data well because no temporal information is preserved in the model. The performances are better for OSC and TSC but the results still contain fragments and inaccurate parts. (Please view the color figure for better visualization).

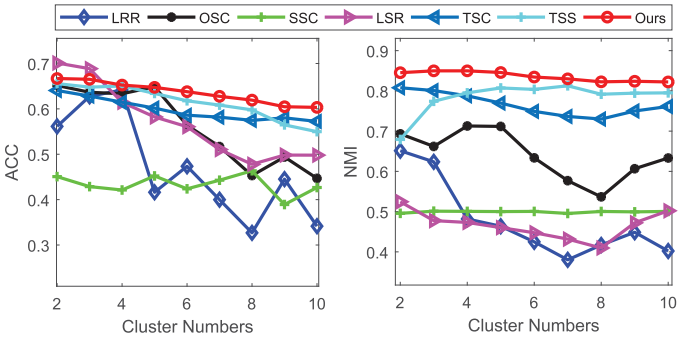


Fig. 4. Segmentation performance of Weizmann dataset based on different action numbers in a video. It indicates that our approach is stable and obtains high accuracy in a wide range of cluster numbers.

TABLE III

PERFORMANCE COMPARISON BASED ON HOG FEATURE [52] AND GOOGLENET (G-NET) FEATURE [59]

Feature	Method	Weiz(M)		Method	UT(K)	
		ACC	NMI		ACC	NMI
HoG	Ours(U)	0.6122	0.8267	Ours(W)	**0.6296**	**0.8035**
G-Net	Ours(U)	**0.6219**	**0.8291**	Ours(W)	0.6213	0.8013

accurately recognizes similar but different actions without increasing fragments. It is a more effective, sensitive, and robust approach.

Considering the fact that deep models achieve significant high performance in the computer vision field [58], we also evaluated the performance when deep features are deployed. We extracted GoogLeNet feature [59] frame by frame to obtain the feature matrix X_S and X_T, then tested the performance in the same setting. The results are shown in Table III, where Weizmann and UT datasets are set as the source and target respectively. We observe that a deep model can achieve similar or slightly higher performance. This fact illustrates that hand-craft features or deep features have similar performance based on our approach, thus, both types of features are suitable for this segmentation scenario.

Moreover, we changed the action numbers of each video from the Weizmann dataset and tested the clustering performance on all methods. We removed some actions and set action numbers of each video from 2 to 10 and evaluated the clustering performance. The results are illustrated in Figure 4. We can see that our approach outperforms almost all other

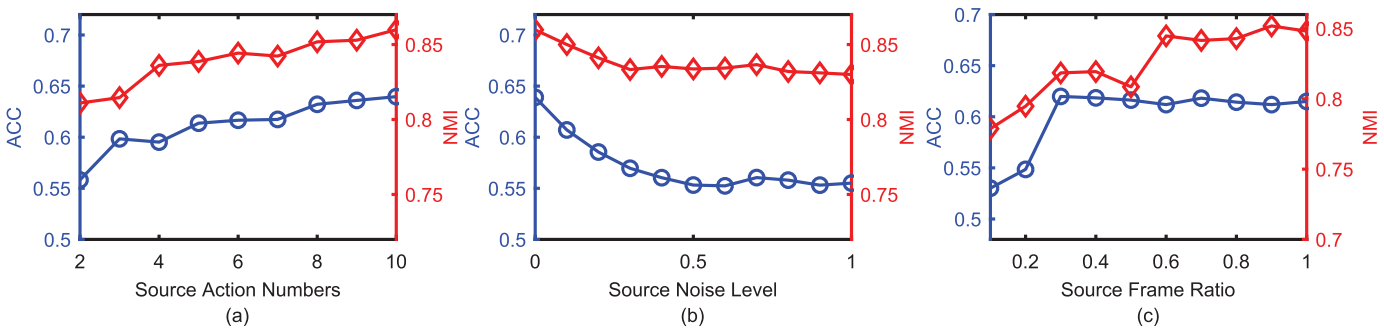


Fig. 5. (a) Segmentation performance of the Weizmann dataset based on different action numbers contained in dictionaries. This figure indicates that the source action is crucial for improving the segmentation performance and that the temporal knowledge is indeed transferred. (b) Segmentation result when different level of Gaussian noise is combined with the source data. It demonstrates the effectiveness of the knowledge transfer from source to target. (c) Segmentation performance based on different action numbers contained in dictionaries. This figure indicates that the source is crucial to improve the segmentation performance. The knowledge is indeed transferred.

methods. In addition, we also notice that our approach performs more stably so that it achieves similar performances in different cluster numbers. Other methods have more fluctuant performance in different cluster numbers.

D. Source Data Analysis

Since our approach transfers structure information between source and target, we evaluated the effectiveness of the source information for the segmentation task.

We first tested the source action video which contained different numbers of actions and the results are shown in Figure 5 (a). We randomly selected the actions, ran the test 5 times, and reported the average performance. From Figure 5 (a), we observe that as the action number increases, the performance also increases. This fact demonstrates that the diversity of the source data is crucial to improving the performance. More actions in the source video could transfer useful information to learn distinctive representations of the target video. This indicates the effectiveness of the source data in the segmentation process.

We also reduced the frame numbers per action while keeping the action numbers consistent. We utilized the frames of original videos from the ratio of 10%, 20% to 100%. The frames are evenly removed from the complete video and the performance is illustrated in Figure 5 (c). We evaluate the performance on Weizmann datasets. From this, we observe that our model can still keep the performance when the frame ratio is greater than 50%. The accuracy metric is robust and stable enough to hold the performance when only 30% of the frames are available. However, if the frames are less than 30%, the performance would drop significantly. The result indicates that our model does not require entire frames to achieve high performance since the closest frames are redundant and cannot provide diverse information to further improve the performance. Thus, in the real-world applications, we can reduce the source video frame numbers to further increase the segmentation speed without losing accuracy.

Moreover, we added a different scale of Gaussian noise in the source video and the result is shown in Figure 5 (b). The performance was evaluated with different noise 5 times and the average performance was reported. Figure 5 (b) clearly shows that the model achieves the highest performance when no noise is added in source data. As the noise increases, the performance drops in the beginning but gradually becomes stable at the end. It can be assumed that adding noise would destroy data structure, which would implicatively eliminate the learned representation structure. These results denote that clean data without a large amount of noise are important to further enhance the performance.

TABLE IV
PERFORMANCE COMPARISON BASED ON DIFFERENT CLUSTERING STRATEGIES FOR THE LEARNED Z_T

Methods	Spectral Clustering	K-means	K-medoids	Ours
ACC	0.5100	0.5604	0.5933	**0.6122**
NMI	0.8109	0.8210	0.8125	**0.8267**

We further evaluate our model if we shuffle the dictionary sequence. The ACC and NMI results are 0.5510 and 0.8203 in the same setting. These performances indicate that the dictionary sequence is uncorrelated to segmentation performance. Since X_S is set as dictionary to reconstruct source and target data, shuffling the dictionary sequence would only change the feature positions in learned representation Z, but the feature position has no effect on clustering performance.

In our implementation, we deploy Eq. (8) for the final clustering procedure on the learned target representation Z_T. We further compared other clustering methods and the result is listed in Table IV. We observe that our clustering strategy can achieve the best performance, which proves the effectiveness and accuracy of the strategy.

E. Model Analysis

To prove the effectiveness of the proposed constraints in our model, the model was further modified by removing and utilizing other graph regularizer and testing the performance under the same setting. The result is shown in Table II. In Model-1, we removed the P Frobenius norm, $\|P\|_{\mathrm{F}}^2$. The performance dropped significantly, indicating that the constraint of P is crucial and that controlling the scale of P avoids overfitting to the learned representation. In Model-2,

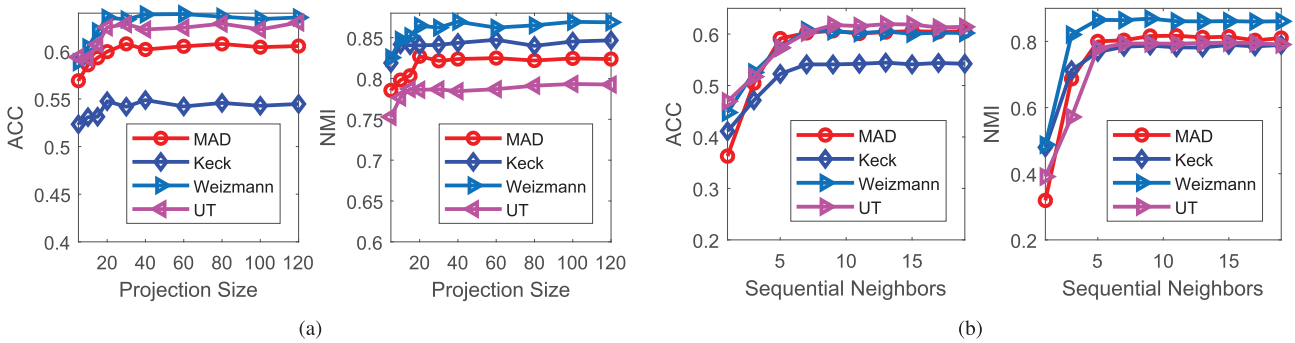


Fig. 6. Parameter sensitivity analysis. (a) Segmentation performance with various project sizes r. (b) Segmentation performance with various values of s.

TABLE V
SEGMENTATION RESULT IN MODIFIED MODELS

Dataset	Keck(M)		Weiz(K)	
Model	ACC	NMI	ACC	NMI
Model-1	0.2798	0.4115	0.2948	0.3844
Model-2	0.5266	0.7881	0.5985	0.8267
Model-3	0.5481	0.7999	0.5712	0.8172
Model-4	0.5395	0.8049	0.6030	0.8326
Model-5	0.5369	0.8186	0.6134	0.8533
Model-6	**0.5519**	**0.8211**	**0.6391**	**0.8599**

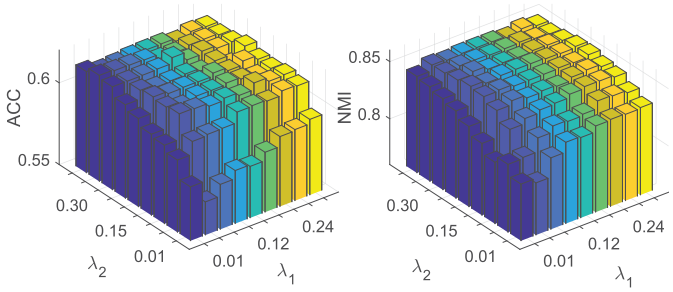


Fig. 7. Parameter analysis in different value of λ_1 and λ_2.

$\|ZAw\|_*$, the temporal and rank constraint, was removed. The performance is lower than other models but still achieves high performance compared with other compared methods. This result indicates that the source data do improve the clustering performance; however, it is not the optimized result without the temporal and rank constraint. In Model-3 and Model-4, we switch the nuclear norm to $l1$ and F-norm as $\|ZAw\|_1$ and $\|ZAw\|_{\mathrm{F}}^2$ in the model. The performance becomes better but still not ideal. In Model-5, the model only learned the target representation X_T instead of both source and target, $X = [X_S, X_T]$, to obtain Z_T. It still could not achieve the best performance compared with our proposed approach in Model-6. Compared with other constraints, rank constraint associated with the temporal graph in both source and target data is the most accurate model to preserve the data structure and achieve the highest segmentation performance.

F. Parameter Analysis

Figure 6(a) shows the parameter sensitivity of projection size r. The figure denotes that the performance is not high when $r < 20$, and if $r \geqslant 20$, although there is fluctuation as r increases, the result is still relatively stable. Another parameter s, the length of relevant frames, is also a major parameter in our algorithm. Figure 6(b) illustrates that clustering performance in various values of s. The figure shows that when $s \geqslant 5$, the performance is accurate and stable.

There are two parameters, λ_1 and λ_2, where λ_1 constrains the scale of P, and λ_2 controls the weight of the temporal regularizer and rank constraint. We deploy various values to test the parameter sensitivity of our algorithm on the Weizmann dataset. The result is illustrated in Figure 7. The result denotes that our method can achieve more accurate results when both λ_1 and λ_2 are greater than 0.1. The range of λ_1 and λ_2 are large, which reflects the fact that our approach is robust and insensitive to parameters. The experimental results demonstrate that every term in our approach is necessary and contributes for improving the segmentation performance. In summary, our approach is highly accurate, robust, stable and parameter insensitive.

V. CONCLUSION

We introduced a novel transfer learning based human motion segmentation approach. This approach explores clustering knowledge from other similar well-labeled human motion dataset, and transfers the knowledge to target data. Specifically, source data was set as a dictionary, then a projection and representation were jointly learned to align source information and target data together. A temporal graph regularizer and a rank constraint further improved the effectiveness of the learned representation. Extensive experiments demonstrated that our approach outperformed state-of-the-art temporal subspace clustering methods on four human motion datasets. Further experiments indicated that our approach was robust, accurate and parameter insensitive. Since the learned representations are distinctive across different actions, thus, it is possible to further expand this model to more comprehensive action segmentation as well as classification scenario. In the future, we will explore on this direction.

REFERENCES

[1] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE CVPR*, Jun. 2015, pp. 4305–4314.
[2] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018.
[3] J. Kleinberg, "An impossibility theorem for clustering," in *Proc. 15th NIPS*, 2002, pp. 463–470.

[4] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. ACM SIGKDD*, 2010, pp. 333–342.

[5] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining Knowl. Discovery*, vol. 7, no. 4, pp. 349–371, 2003.

[6] Y. Yang and K. Chen, "Temporal data clustering via weighted clustering ensemble with different representations," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 2, pp. 307–320, Feb. 2011.

[7] P. Smyth, "Probabilistic model-based clustering of multivariate and sequential data," in *Proc. Workshop AI Statist.*, 1999, pp. 299–304.

[8] Y. Xiong and D.-Y. Yeung, "Mixtures of ARMA models for model-based time series clustering," in *Proc. IEEE Data Mining*, Dec. 2002, pp. 717–720.

[9] N. Dimitrova and F. Golshani, "Motion recovery for video content classification," *ACM Trans. Inf. Syst.*, vol. 13, no. 4, pp. 408–439, 1995.

[10] W. Chen and S.-F. Chang, "Motion trajectory matching of video objects," *Storage and Retrieval for Media Databases 2000*, vol. 3972. Bellingham, WA, USA: SPIE, 1999, pp. 544–554. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/3972/0000/Motion-trajectory-matching-of-video-objects/10.1117/12.373587.full?SSO=1

[11] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Locally adaptive dimensionality reduction for indexing large time series databases," *ACM SIGMOD Rec.*, vol. 30, no. 2, pp. 151–162, 2001.

[12] 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, Jan. 2013.

[13] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE CVPR*, Jun. 2009, pp. 2790–2797.

[14] S. Li, K. Li, and Y. Fu, "Temporal subspace clustering for human motion segmentation," in *Proc. IEEE ICCV*, Dec. 2015, pp. 4453–4461.

[15] A. Talwalkar, L. Mackey, Y. Mu, S.-F. Chang, and M. I. Jordan, "Distributed low-rank subspace segmentation," in *Proc. IEEE ICCV*, Dec. 2013, pp. 3543–3550.

[16] G. Liu and S. Yan, "Active subspace: Toward scalable low-rank learning," *Neural Comput.*, vol. 24, no. 12, pp. 3371–3394, 2012.

[17] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[18] C. Zhang, L. Zhang, and J. Ye, "Generalization bounds for domain adaptation," in *Proc. NIPS*, 2012, pp. 3320–3328.

[19] H. Zhao, Z. Ding, and Y. Fu, "Block-wise constrained sparse graph for face image representation," in *Proc. IEEE FG*, vol. 1, May 2015, pp. 1–6.

[20] V. Zografos, L. Ellis, and R. Mester, "Discriminative subspace clustering," in *Proc. IEEE CVPR*, Jun. 2013, pp. 2107–2114.

[21] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *Proc. IEEE CVPR*, Jun. 2013, pp. 430–437.

[22] S. Wang, B. Tu, C. Xu, and Z. Zhang, "Exact subspace clustering in linear time," in *Proc. AAAI*, 2014, pp. 2113–2120.

[23] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Nonparametric Bayesian learning of switching linear dynamical systems," in *Proc. NIPS*, 2009, pp. 457–464.

[24] M. W. Robards and P. Sunehag, "Semi-Markov kmeans clustering and activity recognition from body-worn sensors," in *Proc. IEEE Data Mining*, Dec. 2009, pp. 438–446.

[25] F. Zhou, F. De la Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 582–596, Mar. 2013.

[26] M. Hoai and F. De la Torre, "Maximum margin temporal clustering," in *Proc. 15th Int. Conf. Artif. Intell. Statist.*, 2012, pp. 1–9.

[27] Z. Ding, M. Shao, and Y. Fu, "Robust multi-view representation: A unified perspective from multi-view learning to domain adaption," in *Proc. AAAI*, 2018, pp. 5434–5440.

[28] N. D. Lawrence and J. C. Platt, "Learning to learn with the informative vector machine," in *Proc. ACM ICML*, 2004, p. 65.

[29] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. ACM ICML*, 2007, pp. 759–766.

[30] L. Wang, Z. Ding, and Y. Fu, "Adaptive graph guided embedding for multi-label annotation," in *Proc. IJCAI*, 2018, pp. 2798–2804.

[31] Z. Ding, N. M. Nasrabadi, and Y. Fu, "Task-driven deep transfer learning for image classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2016, pp. 2414–2418.

[32] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *Proc. ACM SIGKDD*, 2007, pp. 210–219.

[33] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *Int. J. Comput. Vis.*, vol. 109, nos. 1–2, pp. 74–93, 2014.

[34] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. IEEE ICCV*, Nov. 2011, pp. 999–1006.

[35] I.-H. Jhuo, D. Liu, D. T. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *Proc. IEEE CVPR*, Jun. 2012, pp. 2168–2175.

[36] L. Wang, Z. Ding, and Y. Fu, "Learning transferable subspace for human motion segmentation," in *Proc. AAAI*, 2018, pp. 4195–4202.

[37] S. Tierney, J. Gao, and Y. Guo, "Subspace clustering for sequential data," in *Proc. IEEE CVPR*, Jun. 2014, pp. 1019–1026.

[38] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[39] R. Merris, "Laplacian matrices of graphs: A survey," *Linear Algebra Appl.*, vols. 197–198, pp. 143–176, 1994.

[40] Z. Ding, S. Suh, J.-J. Han, C. Choi, and Y. Fu, "Discriminative low-rank metric learning for face recognition," in *Proc. IEEE FG*, vol. 1, May 2015, pp. 1–6.

[41] S. Li and Y. Fu, "Learning balanced and unbalanced graphs via low-rank coding," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1274–1287, May 2015.

[42] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

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

[44] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. Philadelphia, PA, USA: SIAM, 1989.

[45] R. H. Bartels and G. Stewart, "Solution of the matrix equation AX + XB = C [F4]," *Commun. ACM*, vol. 15, no. 9, pp. 820–826, 1972.

[46] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[47] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," in *Proc. ACM STC*, 1987, pp. 1–6.

[48] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[49] Z. Jiang, Z. Lin, and L. S. Davis, "Recognizing human actions by learning and matching shape-motion prototype trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 533–547, Mar. 2012.

[50] D. Huang, S. Yao, Y. Wang, and F. De La Torre, "Sequential max-margin event detectors," in *Proc. ECCV*. Zürich, Switzerland: Springer, 2014, pp. 410–424.

[51] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *Proc. IEEE ICCV*, Sep./Oct. 2009, pp. 1593–1600.

[52] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. IEEE CVPR*, vol. 2, Jun. 2006, pp. 1491–1498.

[53] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[54] L. Kaufman and P. Rousseeuw, *Clustering by Means of Medoids*. Amsterdam, The Netherlands: North Holland, 1987.

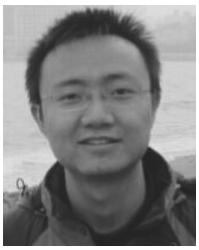
[55] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. NIPS*, 2002, pp. 849–856.

[56] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. ECCV*. Florence, Italy: Springer, 2012, pp. 347–360.

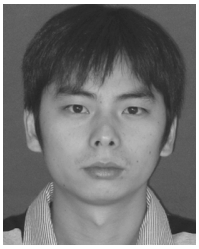
[57] J. Wu, H. Xiong, and J. Chen, "Adapting the right measures for K-means clustering," in *Proc. ACM SIGKDD*, 2009, pp. 877–886.

[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[59] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, 2015, pp. 1–9.



Lichen Wang received the B.Eng. degree in automation from the Harbin Institute of Technology, Harbin, China, in 2013, and the M.Eng. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA. His research interests include computer vision, machine learning, and deep learning. He has served as a PC Member for AAAI. He is a Student Member of AAAI. He was a recipient of the Student Travel Grant from AAAI 18. He has served as a Reviewer for CVPR, IJCAI, and NN.



Zhengming Ding (S'14–M'18) received the B.Eng. degree in information security and the M.Eng. degree in computer software and theory from the University of Electronic Science and Technology of China, China, in 2010 and 2013, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Northeastern University, USA, in 2018. Since 2018, he has been a Faculty Member with the Department of Computer, Information and Technology, Indiana University–Purdue University Indianapolis. His research interests include machine learning and computer vision. Specifically, he devotes himself to develop scalable algorithms for challenging problems in transfer learning and deep learning scenario.

Yun Fu (S'07–M'08–SM'11) received the B.Eng. degree in information engineering and the M.Eng. degree in pattern recognition and intelligence systems from Xi'an Jiaotong University, China, and the M.S. degree in statistics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign. He has been an Interdisciplinary Faculty Member with the College of Engineering and the College of Computer and Information Science, Northeastern University, since 2012. He has extensive publications in leading journals, books/book chapters, and international conferences/workshops. His research interests are machine learning, computational intelligence, big data mining, computer vision, pattern recognition, and cyber-physical systems. He is a fellow of IAPR, OSA, and SPIE, a Lifetime Senior Member of ACM, a Lifetime Member of AAAI and the Institute of Mathematical Statistics, a member of the ACM Future of the Computing Academy, the Global Young Academy, AAAS, INNS, and a Beckman Graduate Fellow from 2007 to 2008. He received seven Prestigious Young Investigator Award from NAE, ONR, ARO, IEEE, INNS, UIUC, and the Grainger Foundation; nine Best Paper Award from the IEEE, IAPR, SPIE, and SIAM; and many major Industrial Research Awards from Google, Samsung, and Adobe. He serves as an associate editor, the chair, a PC member, and a reviewer for many top journals and international conferences/workshops. He is currently an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEANING SYSTEMS.