

Multi-mutual Consistency Induced Transfer Subspace Learning for Human Motion Segmentation

Tao Zhou¹, Huazhu Fu¹, Chen Gong², Jianbing Shen¹, Ling Shao¹, Fatih Porikli³

¹Inception Institute of Artificial Intelligence, Abu Dhabi, UAE.

²The Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, China.

³Australian National University, Australia.

Abstract

Human motion segmentation based on transfer subspace learning is a rising interest in action-related tasks. Although progress has been made, there are still several issues within the existing methods. First, existing methods transfer knowledge from source data to target tasks by learning domain-invariant features, but they ignore to preserve domain-specific knowledge. Second, the transfer subspace learning is employed in either low-level or high-level feature spaces, but few methods consider fusing multi-level feature representations for subspace learning. To this end, we propose a novel multi-mutual consistency induced transfer subspace learning framework for human motion segmentation. Specifically, our model factorizes the source and target data into distinct multi-layer feature spaces and reduces the distribution gap between them through a multi-mutual consistency learning strategy. In this way, the domain-specific knowledge and domain-invariant properties can be explored in different layers simultaneously. Our model also conducts the transfer subspace learning on different layers to capture multi-level structural information. Further, to preserve the temporal correlations, we project the learned representations into a block-like space. The proposed model is efficiently optimized by using the Augmented Lagrange Multiplier (ALM) algorithm. Experimental results on four human motion datasets demonstrate the effectiveness of our method over other state-of-the-art approaches.

1. Introduction

Human motion segmentation aims to partition visual data sequences that depict human actions and activities into a set of preferably non-overlapping and internally coherent temporal segments. It is an important preprocessing step before further motion and action related analytical tasks

[26, 38, 48, 59]. Human motion information is a key factor for temporal segmentation. However, due to the complexity of temporal correlations and the high-dimensional structure of visual representations, capturing such discriminative temporal information remains as a challenging task [23]. Therefore, several approaches have been developed to address this problem, including model-based [49], temporal proximity-based [23], representation-based [22, 25], and subspace clustering-based approaches [12, 25]. Among them, the subspace clustering-based methods have attracted notable attention and obtained promising results.

Subspace clustering is a powerful technique for partitioning data into multiple groups, which holds the assumption that data points are drawn from multiple subspaces corresponding to different classes [4, 24, 33]. Several representative subspace clustering methods [8, 16, 29, 32] have been developed to learn distinct and low-dimensional data representations, in which the learned representations are then fed into conventional clustering algorithms. However, it is often difficult for these unsupervised subspace learning methods to attain reasonable performance without prior knowledge. Fortunately, labeled data from related tasks are often easy to obtain. Thus, transfer learning is an ideal option for borrowing knowledge from relevant source data to improve the target tasks [5, 52]. In human motion segmentation, recent transfer subspace learning-based approaches [46, 47] have reported improved performance.

Although transfer subspace learning has achieved satisfactory results in human motion segmentation, there still exist several issues as follows. First, the transfer subspace learning based motion segmentation imposes the data distributions of two domains to be similar. To this end, one popular strategy is to project both the source and target data into a common feature space. This strategy explores domain-invariant properties but ignores the potentially useful domain-specific knowledge. However, both of these two aspects play essential roles, and it is challenging to balance

Corresponding author: Jianbing Shen (shenjianbingcg@gmail.com).

them for improved performance. Second, existing subspace clustering-based methods tend to reconstruct data points by using either the original or high-level features (*e.g.*, outputs of deep networks), with few conducting transfer subspace learning in multi-level feature spaces to capture low-level and high-level information simultaneously.

To address the above problems, we propose a novel method that incorporates transfer learning and multi-level subspace clustering into a unified framework to enhance human motion segmentation (as shown in Fig. 1). First, we factorize the original features of the source and the target data into implicit multi-layer feature spaces, in which a multi-mutual consistency learning strategy is used to reduce the distribution difference between the two domains. Second, we carry out the transfer subspace learning in different layers to fuse multi-level structural information effectively. Third, we project the learned representations into a block-like space to preserve the temporal correlations. Finally, we show that our model can be efficiently optimized using the Augmented Lagrange Multiplier (ALM) algorithm.

The main contributions are summarized as follows: We present a novel human motion segmentation algorithm, which integrates transfer learning and multi-level subspace learning into a unified framework. Our motion segmentation model explores domain-invariant properties by using a multi-mutual consistency learning strategy while preserving domain-specific knowledge. We conduct multi-level transfer subspace learning in different layers to simultaneously capture low- and high-level information. Extensive experiments on four public datasets demonstrate the superiority of our model over the state-of-the-art methods.

2. Related Work

Subspace clustering builds on the assumption that data points are drawn from multiple subspaces corresponding to different clusters. Recently, self-representation based subspace clustering, where each data point is expressed with a linear combination of other data points, has captured increasing attention [60, 53, 61]. For example, Sparse Subspace Clustering (SSC) [8] searches the sparsest representation among the infinitely many possible representations based on ℓ_1 -norm. Low-Rank Representation Clustering (LRR) [29] attempts to reveal cluster structure with a low-rank representation. SMOOTH Representation clustering (SMR) [16] analyzes the grouping effect of representation-based methods. There are also several deep learning-based subspace clustering approaches [19, 37, 53, 55, 57]. However, these methods cannot be directly applied in human motion segmentation since they ignore the temporal correlations between successive frames.

Temporal data clustering aims for segmenting data sequences into a set of non-overlapping parts. It has a wide range of applications, from facial analytics, speech segmen-

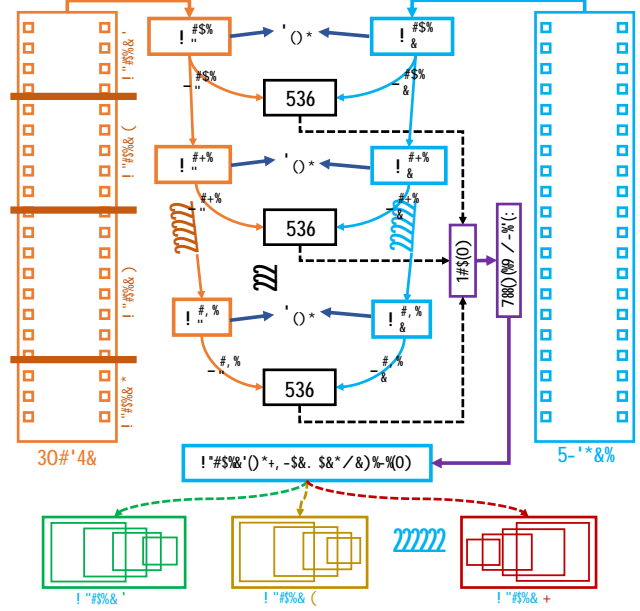


Figure 1: Overview of the proposed multi-mutual consistency induced transfer subspace learning framework for human motion segmentation. Our model first factorizes the source and target data into multi-layer implicit feature spaces, in which a multi-mutual consistency learning strategy is presented to reduce the distribution difference between the two domains. Then, it carries out a multi-level Transfer Subspace Learning (TSL) in different layers to capture multi-level structural information. After that, it fuses multi-level representations to construct an affinity matrix, and obtains the final segmentation results by using the Normalized Cuts algorithm.

tation to human action recognition. For this purpose, semi-Markov K-means clustering [39] attempts to exploit repetitive patterns. Zhou *et al.* [56] use a K-means kernel associated with a dynamic temporal alignment approach. Temporal Subspace Clustering (TSC) [25] learns a non-negative dictionary and data representation under the constraint of a temporal Laplacian regularization term. Transfer Subspace Segmentation (TSS) [46] adopts auxiliary data and transfers segmentation knowledge from source to target data. Low-rank Transfer Subspace (LTS) [47] employs a novel sequential graph to preserve temporal information residing in both the source and target data. These temporal clustering methods are all formulated as unsupervised learning scenarios, some of which adopt a self-representation strategy to achieve the motion segmentation task.

Transfer learning intends for leveraging on the prior knowledge from related source data to improve the results of target tasks. So far, plenty of transfer learning models have been proposed, such as domain-invariant feature learning [13, 34, 42] and classifier parameter adaptation [2, 27, 54]. Among these, domain-invariant feature learning [13] attempts to learn a common feature space where both the domain shift and distribution difference can be mitigated. Several works explore the alignment of two different

domains, for instance, subspace learning [41, 50] and dictionary learning [11, 62]. In the same vein, deep learning inspired techniques have been used to integrate knowledge transfer and learned features into one unified framework [7, 10, 31, 45]. However, most of these methods incorporate domain alignment strategies in their top layers, ignoring the low-level structural information.

3. The Proposed Method

3.1. Formulation

As aforementioned, three main challenges remain for human motion segmentation using transfer subspace learning, namely: (i) how to reduce the distribution difference while preserving domain-specific knowledge; (ii) how to capture multi-level information to enhance the performance of transfer subspace learning; and (iii) how to effectively capture the temporal correlations among motion data.

To address these challenges, we formulate our model with three strategies: 1) multi-mutual consistency learning, 2) multi-level subspace learning, and 3) temporal correlation preservation.

1) **Multi-mutual consistency learning** Deep structure learning has demonstrated its effectiveness in many real-world applications [20, 35, 36, 51]. To capture multi-level structural information, we use a multi-layer decomposition process based on the deep Non-negative Matrix Factorization (NMF) model as

$$\begin{aligned} \mathbf{X} &= \mathbf{D}^{(1)} \mathbf{H}^{(1)} \\ &\quad \mathbf{D}^{(1)} \mathbf{D}^{(2)} \mathbf{H}^{(2)} \\ &\quad \vdots \\ &\quad \mathbf{D}^{(1)} \mathbf{D}^{(2)} \dots \mathbf{D}^{(l)} \dots \mathbf{D}^{(L)} \mathbf{H}^{(L)}, \end{aligned} \quad (1)$$

where $\mathbf{D}^{(l)} \geq 0$ and $\mathbf{H}^{(l)} \geq 0$ ($l = 1, \dots, L$) denote the basis matrix and the feature representation matrix at the l -th layer, respectively, and L is the number of layers. It is worth noting that the feature representations in each layer capture different levels of information and knowledge from the original data.

To mitigate the difference in distribution between the source and target data, and at the same time, preserve the knowledge from different domains, we establish our multi-mutual consistency learning model as

$$\begin{aligned} L_1(\mathbf{X}_s, \mathbf{X}_t; \mathbf{D}_s^{(l)}, \mathbf{D}_t^{(l)}, \mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}) \\ = \|\mathbf{X}_s - \mathbf{D}_s^{(1)} \mathbf{D}_s^{(2)} \dots \mathbf{D}_s^{(L)} \mathbf{H}_s^{(L)}\|_F^2 \\ + \|\mathbf{X}_t - \mathbf{D}_t^{(1)} \mathbf{D}_t^{(2)} \dots \mathbf{D}_t^{(L)} \mathbf{H}_t^{(L)}\|_F^2 \\ + \sum_{l=1}^L F_{\text{con}}(\mathbf{D}_s^{(l)}, \mathbf{D}_t^{(l)}), \end{aligned} \quad (2)$$

where $\mathbf{X}_s \in \mathbb{R}^{d \times n_s}$ and $\mathbf{X}_t \in \mathbb{R}^{d \times n_t}$ denote the source and target data, respectively. d is the feature dimension, and

n_s and n_t are the number of the source and target data, respectively. $\gamma > 0$ is a trade-off parameter. The first two terms are used to explore the multi-level structures in both the source and target data. The third term $F_{\text{con}}(\cdot, \cdot)$ aims to decrease the distribution difference between two domains by penalizing the divergence of two basis matrices in different layers. In contrast, most existing methods directly project the source and target data into a common space using a domain-invariant projection matrix, resulting in a loss of domain-specific knowledge. Finally, although there are various strategies for constraining the consistency between $\mathbf{D}_s^{(l)}$ and $\mathbf{D}_t^{(l)}$, in this study, we utilize a simple but effective strategy, *i.e.*, $F_{\text{con}}(\mathbf{D}_s^{(l)}, \mathbf{D}_t^{(l)}) = \|\mathbf{D}_s^{(l)} - \mathbf{D}_t^{(l)}\|_F^2$.

2) **Multi-level transfer subspace learning** Existing subspace clustering or subspace clustering-based motion segmentation methods often reconstruct data points using either the shallow representations (*e.g.*, original features) or high-level representations (*e.g.*, features from the last layer of deep networks). Although the high-level representations have shown promising performance in clustering tasks, they omit a certain amount of useful information. Thus, we propose a multi-level subspace learning strategy to effectively exploit the structural information in different feature spaces, which we formulate as:

$$\begin{aligned} L_2(\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}; \mathbf{Z}^{(l)}) &= \sum_{l=1}^L [\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)} \mathbf{Z}^{(l)}\|_{2,1}, \\ \text{s.t. } \mathbf{Z}^{(l)} &\geq 0, \mathbf{1}^T \mathbf{Z}^{(l)} = \mathbf{1}, \quad l = 1, 2, \dots, L, \end{aligned} \quad (3)$$

where $\mathbf{1}$ denotes a column vector with all elements being one. The non-negative constraint $\mathbf{Z}^{(l)} \geq 0$ enhances the discriminative ability of the learned representations. The constraint $\mathbf{1}^T \mathbf{Z}^{(l)} = \mathbf{1}$ makes the sum of each coefficient vector to be one, therefore suppressing the representation coefficients from different subspaces. It is worth noting that, in Eq. (3), the feature representation of the source data (*i.e.*, $\mathbf{H}_s^{(l)}$) is regarded as a dictionary, which is then used to reconstruct the feature representations of both the source and target data. This enables knowledge from the source data to be transferred to the target task. Additionally, $\|\cdot\|_{2,1}$ denotes the $\ell_{2,1}$ -norm, which encourages the columns of a matrix to be zero [29], *i.e.*, $\|\mathbf{E}\|_{2,1} = \sum_{j=1}^N \sqrt{\sum_{i=1}^M [\mathbf{E}_{ij}]^2}$, where $\mathbf{E} \in \mathbb{R}^{M \times N}$. By using the $\ell_{2,1}$ -norm, an underlying assumption is that any corruptions are sample-specific, *i.e.*, some data vectors may be corrupted while the others are clean. Remarks: Our model learns multiple transferable subspaces within a layer-wise framework, which can capture multi-level structural information and provide more substantial knowledge to improve motion segmentation performance.

3) **Temporal correlation preservation** Temporal and structural information is crucial for accurate clustering since human motion data are consecutive and sequential.

Thus, it is essential to preserve the temporal information in the learned representation \mathbf{Z} . To achieve this, a popular strategy is to regulate the i -th coefficient's neighbors $[z_{i-2}, \dots, z_{i-1}, z_{i+1}, \dots, z_{i+2}]$ to be close to z_i , where l is the length of relevant frames. Here, we first build a weight matrix \mathbf{S} [25, 46], where we define each element as follows:

$$s_{ij} = \begin{cases} 1, & \text{if } |i - j| \leq l, l(x_i) = l(x_j), \text{ for source data;} \\ 1, & \text{if } |i - j| \leq l, \text{ for target data;} \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where $l(x_i)$ denotes the action label of the i -th sample x_i in the source data. We observe that the weight matrix has a block-like structure. To preserve the temporal correlations, we project the representation \mathbf{Z} into a block-like space, which we formulate as follows:

$$L_3(\mathbf{S}, \mathbf{Z}^{(l)}; \mathbf{W}^{(l)}) = \sum_{i=1}^L (\mathbf{S} - \mathbf{W}^{(l)} \mathbf{Z}^{(l)})^2_F + \mathbf{W}^{(l)}, \quad (5)$$

where λ is a trade-off parameter, and $\|\cdot\|_F$ is the matrix nuclear-norm [29]. Since there exist temporal correlations in the learned representation, we introduce the low-rank regularization on the projection matrix $\mathbf{W}^{(l)}$ by using the nuclear norm [29].

Overall formulation: Finally, we integrate the above three components (Eqs. (2)(3)(5)) into a single unified objective function as follows:

$$\begin{aligned} & \min L_1(\mathbf{X}_s, \mathbf{X}_t; \mathbf{D}_s^{(l)}, \mathbf{D}_t^{(l)}, \mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}) + \\ & \quad \text{Multi-mutual consistency learning} \\ & \quad L_2(\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}; \mathbf{Z}^{(l)}) + L_3(\mathbf{S}, \mathbf{Z}^{(l)}; \mathbf{W}^{(l)}) \\ & \quad \text{Multi-level transfer subspace learning} \quad \text{Temporal correlation preservation} \\ = \min & \quad \mathbf{X}_s - \mathbf{D}_s^{(1)} \mathbf{D}_s^{(2)} \dots \mathbf{D}_s^{(L)} \mathbf{H}_s^{(L)} \quad (6) \\ & + \mathbf{X}_t - \mathbf{D}_t^{(1)} \mathbf{D}_t^{(2)} \dots \mathbf{D}_t^{(L)} \mathbf{H}_t^{(L)} \quad (6) \\ & + \sum_{l=1}^L \mathbf{D}_s^{(l)} - \mathbf{D}_t^{(l)} \quad (6) \\ & + \sum_{l=1}^L [\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)} \mathbf{Z}^{(l)} \quad (6) \\ & + \sum_{l=1}^L \mathbf{S} - \mathbf{W}^{(l)} \mathbf{Z}^{(l)} \quad (6) + \sum_{l=1}^L \mathbf{W}^{(l)}, \\ & \text{s.t. } \mathbf{Z}^{(l)} \geq 0, \mathbf{1}^T \mathbf{Z}^{(l)} = 1, \quad l = 1, 2, \dots, L, \end{aligned}$$

where $\mathbf{Z}^{(l)} = \{\mathbf{D}_s^{(1)} \geq 0, \mathbf{D}_t^{(1)} \geq 0, \mathbf{H}_s^{(1)} \geq 0, \mathbf{H}_t^{(1)} \geq 0, \mathbf{Z}^{(l)}, \mathbf{W}^{(l)}\}$ ($l = 1, 2, \dots, L$) is the variable set to be optimized, and $\lambda, \mu, \gamma, \eta$ are trade-off parameters.

3.2. Clustering

By using Eq. (6), we can obtain the learned multi-level representations $\mathbf{Z}^{(l)}$ ($l = 1, 2, \dots, L$), and then the corresponding target representations $\mathbf{Z}_t^{(l)} \in \mathbb{R}^{n_s \times n_t}$ can be extracted from $\mathbf{Z}^{(l)} = [\mathbf{Z}_s^{(l)}, \mathbf{Z}_t^{(l)}]$. To exploit the intrinsic relationships among within-cluster samples in human motion

data, we utilize the strategy from [25] and introduce another similarity measurement to construct an affinity matrix \mathbf{A} . Each element of \mathbf{A} can be defined as the distance between any pair of the learned target representations, which is:

$$a(i, j) = \frac{1}{L} \sum_{l=1}^L \frac{\mathbf{z}_{t,i}^{(l)} \mathbf{z}_{t,j}^{(l)}}{\mathbf{z}_{t,i}^{(l)} \mathbf{z}_{t,j}^{(l)}}, \quad (7)$$

where $\mathbf{z}_{t,i}^{(l)}$ and $\mathbf{z}_{t,j}^{(l)}$ denote the i -th and j -th columns of $\mathbf{Z}_t^{(l)}$, respectively. Then, the Normalized Cut [43] algorithm is applied to the learned affinity matrix \mathbf{A} to produce the temporal segmentation results.

3.3. Optimization

The objective function in Eq. (6) is not jointly convex with respect to all variables. Thus, we utilize the ALM [28] algorithm to efficiently solve it. To adopt the ALM strategy to our problem, we introduce one auxiliary variable $\mathbf{J}^{(l)}$ to replace $\mathbf{W}^{(l)}$ in the nuclear term of our objective function. Then, we solve the previous optimization function by minimizing the following ALM problem:

$$\begin{aligned} L(\mathbf{Z}) = & \mathbf{X}_s - \mathbf{D}_s^{(1)} \mathbf{D}_s^{(2)} \dots \mathbf{D}_s^{(L)} \mathbf{H}_s^{(L)} \quad (6) \\ & + \mathbf{X}_t - \mathbf{D}_t^{(1)} \mathbf{D}_t^{(2)} \dots \mathbf{D}_t^{(L)} \mathbf{H}_t^{(L)} \quad (6) \\ & + \sum_{l=1}^L \mathbf{D}_s^{(l)} - \mathbf{D}_t^{(l)} \quad (6) + \sum_{l=1}^L \mathbf{E}^{(l)} \quad (6) \\ & + \sum_{l=1}^L \mathbf{S} - \mathbf{W}^{(l)} \mathbf{Z}^{(l)} \quad (6) + \sum_{l=1}^L \mathbf{J}^{(l)} \quad (6) \\ & + \sum_{l=1}^L [\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)} \mathbf{Z}^{(l)} - \mathbf{E}^{(l)} \\ & + \sum_{l=1}^L \mathbf{W}^{(l)} - \mathbf{J}^{(l)}, \\ & \text{s.t. } \mathbf{Z}^{(l)} \geq 0, \mathbf{1}^T \mathbf{Z}^{(l)} = 1, \quad l = 1, 2, \dots, L, \end{aligned} \quad (8)$$

where $(\mathbf{A}, \mathbf{Q}) = \frac{\mu}{2} \|\mathbf{Q}\|_F^2 + \langle \mathbf{A}, \mathbf{Q} \rangle$, with $\langle \cdot, \cdot \rangle$ denoting the matrix inner product. μ is a positive penalty scalar, and $\mathbf{E}^{(l)}$ and $\mathbf{J}^{(l)}$ ($l = 1, 2, \dots, L$) are Lagrangian multipliers. We describe the optimization steps for each subproblem below.

\mathbf{D}_s -subproblem: The optimization problem associated with \mathbf{D}_s can be written as

$$\begin{aligned} \min_{\mathbf{D}_s} & \quad \mathbf{X}_s - \mathbf{D}_s^{(1)} \mathbf{D}_s^{(2)} \dots \mathbf{D}_s^{(L)} \mathbf{H}_s^{(L)} \quad (9) \\ & + \sum_{l=1}^L \mathbf{D}_s^{(l)} - \mathbf{D}_t^{(l)} \quad (9), \quad l = 1, 2, \dots, L. \end{aligned}$$

By taking the derivative of Eq. (9) w.r.t. $\mathbf{D}_s^{(l)}$ and using the Karush-Kuhn-Tucker (KKT) condition [1], we obtain the following updating rule:

$$\mathbf{D}_s^{(l)} = \frac{\mathbf{D}_s^{(l-1)} \mathbf{X}_s \mathbf{H}_s^{(L)} \mathbf{D}_s^{(l+1)} + \mathbf{D}_t^{(l)}}{\frac{\mathbf{D}_s^{(l-1)}}{\mathbf{D}_s^{(l-1)}} \mathbf{D}_s^{(l)} \frac{\mathbf{D}_s^{(l+1)}}{\mathbf{D}_s^{(l+1)}} \mathbf{H}_s^{(L)} \mathbf{H}_s^{(L)} + \frac{\mathbf{D}_t^{(l)}}{\mathbf{D}_s^{(l)}}}, \quad (10)$$

where $\mathbf{D}_s^{(1-1)} = \mathbf{D}_s^{(1)} \mathbf{D}_s^{(2)} \dots \mathbf{D}_s^{(l-1)}$, and $\mathbf{D}_s^{(l+1)} = \mathbf{D}_s^{(l+2)} \dots \mathbf{D}_s^{(L)}$.

Similarly, we have the updating rule for $\mathbf{D}_t^{(l)}$ as follows

$$\mathbf{D}_t^{(l)} = \frac{\mathbf{D}_t^{(l-1)} \mathbf{X}_t \mathbf{H}_t^{(L)} \mathbf{D}_t^{(l+1)} + \mathbf{D}_s^{(l)}}{\mathbf{D}_t^{(l-1)} \mathbf{D}_t^{(l-1)} \mathbf{D}_t^{(l)} \mathbf{D}_t^{(l+1)} \mathbf{H}_t^{(L)} \mathbf{H}_t^{(L)} \mathbf{D}_t^{(l+1)} + \mathbf{D}_t^{(l)}}. \quad (11)$$

\mathbf{H}_s -subproblem: The optimization problem associated with $\mathbf{H}_s^{(l)}$ can be written as

$$\min_{\mathbf{H}_s^{(l)}} \mathbf{X}_s - \mathbf{D}_s^{(1)} \mathbf{D}_s^{(2)} \dots \mathbf{D}_s^{(l)} \mathbf{H}_s^{(l)} \mathbf{F}^2 + \frac{1}{2} [\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)} \mathbf{Z}^{(l)} - \mathbf{E}^{(l)}. \quad (12)$$

By taking the derivative of Eq. (12) w.r.t. $\mathbf{H}_s^{(l)}$ and using the KKT condition [1], we then obtain the following updating rule:

$$\mathbf{H}_s^{(l)} = \mathbf{H}_s^{(l-1)} + \frac{2 \mathbf{D}_s^{(l)} \mathbf{X}_s + \mu (\mathbf{E}_s^{(l)} - \frac{1}{\mu} (\mathbf{I} - \mathbf{Z}_s^{(l)}))}{2 \mathbf{D}_s^{(l)} \mathbf{D}_s^{(l)} \mathbf{H}_s^{(l)} + \mu \mathbf{H}_s^{(l)} (\mathbf{I} - \mathbf{Z}_s^{(l)}) (\mathbf{I} - \mathbf{Z}_s^{(l)})}. \quad (13)$$

where $\mathbf{E}^{(l)} = [\mathbf{E}_s^{(l)}, \mathbf{E}_t^{(l)}]$, $\mathbf{Z}^{(l)} = [\mathbf{Z}_s^{(l)}, \mathbf{Z}_t^{(l)}]$, and $\frac{1}{\mu} = [\frac{1}{\mu_s}, \frac{1}{\mu_t}]$. $\mathbf{E}_s^{(l)}$, $\mathbf{Z}_s^{(l)}$ and $\frac{1}{\mu_s}$ denote the corresponding parts to $\mathbf{H}_s^{(l)}$, and \mathbf{I} is an identity matrix.

Similarly, we have the updating rule for $\mathbf{H}_t^{(l)}$ as follows:

$$\mathbf{H}_t^{(l)} = \mathbf{H}_t^{(l-1)} + \frac{2 \mathbf{D}_t^{(l)} \mathbf{X}_t + \mu (\mathbf{H}_s^{(l)} \mathbf{Z}_t^{(l)} + \mathbf{E}_t^{(l)} - \frac{1}{\mu} \mathbf{I})}{2 \mathbf{D}_t^{(l)} \mathbf{D}_t^{(l)} \mathbf{H}_t^{(l)} + \mu \mathbf{H}_t^{(l)}}. \quad (14)$$

\mathbf{W} -subproblem: $\mathbf{W}^{(l)}$ can be optimized by solving

$$\min_{\mathbf{W}^{(l)}} \mathbf{S} - \mathbf{W}^{(l)} \mathbf{Z}^{(l)} \mathbf{F}^2 + \frac{1}{2} [\mathbf{W}^{(l)}, \mathbf{W}^{(l)} - \mathbf{J}^{(l)}]. \quad (15)$$

Taking the derivative of the above objective with respect to $\mathbf{W}^{(l)}$, we obtain the closed-form solution

$$\mathbf{W}^{(l)} = \mathbf{S} \mathbf{Z}^{(l)} + \frac{\mu \mathbf{J}^{(l)} - \frac{1}{2} \mathbf{Z}^{(l)} \mathbf{Z}^{(l)}}{2} + \frac{\mu \mathbf{I}}{2}^{-1}. \quad (16)$$

\mathbf{J} -subproblem: The optimization problem associated with $\mathbf{J}^{(l)}$ can be written as

$$\min_{\mathbf{J}^{(l)}} \frac{1}{\mu} \mathbf{J}^{(l)} + \frac{1}{2} \mathbf{J}^{(l)} - (\mathbf{W}^{(l)} + \frac{1}{2} \mathbf{Z}^{(l)} / \mu) \mathbf{F}^2. \quad (17)$$

The above problem can be solved via using a singular value thresholding operator [3].

\mathbf{Z} -subproblem: Dropping the unrelated terms with respect to $\mathbf{Z}^{(l)}$ yields

$$\begin{aligned} \min_{\mathbf{Z}^{(l)}} \quad & \mathbf{S} - \mathbf{W}^{(l)} \mathbf{Z}^{(l)} \mathbf{F}^2 \\ & + \frac{1}{2} [\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)} \mathbf{Z}^{(l)} - \mathbf{E}^{(l)}, \end{aligned} \quad (18)$$

s.t. $\mathbf{Z}^{(l)} \geq 0, \mathbf{1} \mathbf{Z}^{(l)} = \mathbf{1}$.

Algorithm 1: Solving problem (6) via ALM.

```

1 Input: Source data:  $\mathbf{X}_s$  and target data  $\mathbf{X}_t$ , parameters  $\mu, \mu_s, \mu_t$ , and  $\mathbf{J}^{(0)}$ .
2 Initialize:  $\mathbf{D}_s^{(1)} = \mathbf{0}$ ,  $\mathbf{D}_t^{(1)} = \mathbf{0}$ ,  $\mu = 10^{-4}$ ,  $\mu_s = 1.5$ ,  $\mu_t = 10^{-4}$ ,  $\max_{\mu} = 10^6$ .
3 Output:  $\mathbf{Z}^{(l)}$ ,  $l = 1, 2, \dots, L$ .
4 while not converged do
5   for  $l=1, 2, \dots, L$  do
6     Update  $\mathbf{D}_s^{(l)}$ ,  $\mathbf{D}_t^{(l)}$ ,  $\mathbf{H}_s^{(l)}$ ,  $\mathbf{H}_t^{(l)}$ ,  $\mathbf{W}^{(l)}$ ,  $\mathbf{J}^{(l)}$ ,  $\mathbf{E}^{(l)}$ ,  $\frac{1}{\mu}$ , and  $\frac{1}{\mu_s}$  using Eqs. (10), (11), (13), (14), (16), (17), (18), (19), and (20), respectively.
7   end
8   Update the parameter  $\mu$  via  $\mu = \min(\mu, \max_{\mu})$ ;
9   Check the convergence conditions:
10    $[\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)} \mathbf{Z}^{(l)} - \mathbf{E}^{(l)} < \epsilon$ 
11   and  $\mathbf{W}^{(l)} - \mathbf{J}^{(l)} < \epsilon$ .
12 end

```

By taking the derivative of (18) w.r.t. $\mathbf{Z}^{(l)}$ and setting it to zero, we can obtain its closed-form solution. After that, we apply an efficient iterative algorithm [18] to obtain the final solution of $\mathbf{Z}^{(l)}$.

\mathbf{E} -subproblem: The error term $\mathbf{E}^{(l)}$ can be updated by solving the following problem:

$$\min_{\mathbf{E}^{(l)}} \frac{1}{\mu} \mathbf{E}^{(l)} + \frac{1}{2} \mathbf{E}^{(l)} - \mathbf{G} \mathbf{F}^2, \quad (19)$$

where $\mathbf{G} = [\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)} \mathbf{Z}^{(l)} + \frac{1}{\mu} \mathbf{I}$. This subproblem can be efficiently solved by using the algorithm in [30].

Multipliers updating: The multipliers $\frac{1}{\mu}$ and $\frac{1}{\mu_s}$ can be updated by using the following equation:

$$\begin{aligned} \frac{1}{\mu} &:= \frac{1}{\mu} + \mu ([\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)} \mathbf{Z}^{(l)} - \mathbf{E}^{(l)}), \\ \frac{1}{\mu_s} &:= \frac{1}{\mu_s} + \mu (\mathbf{W}^{(l)} - \mathbf{J}^{(l)}). \end{aligned} \quad (20)$$

Note that we pretrain each of the layers to obtain initial approximations for $\mathbf{D}_s^{(l)}$, $\mathbf{D}_t^{(l)}$, $\mathbf{H}_s^{(l)}$, and $\mathbf{H}_t^{(l)}$. This pre-training process can reduce the training time of our model, and its effectiveness has also been proven in deep auto-encoder networks [15]. Taking the source data as an example, we decompose $\mathbf{X}_s = \mathbf{D}_s^{(1)} \mathbf{H}_s^{(1)}$ and then decompose $\mathbf{H}_s^{(1)} = \mathbf{D}_s^{(2)} \mathbf{H}_s^{(2)}$ until all layers are initialized. Then, we repeat the updating steps until convergence. The details for solving Eq. (6) via the ALM algorithm are summarized in Algorithm 1.

3.4 Complexity Analysis

The major computational burden of Algorithm 1 lies in two stages, *i.e.*, pretraining and model updating, so we analyze them separately. The computational complexity for the pretraining step is of order $O(L t_p (n_s^2 p + n_s p^2 + n_t^2 p + n_t p^2))$, where t_p is the number of iterations, and p is the maximal layer size out of all layers. In the model updating stage, the updates for $\mathbf{D}_s^{(l)}$, $\mathbf{D}_t^{(l)}$, $\mathbf{H}_s^{(l)}$, $\mathbf{H}_t^{(l)}$, and $\mathbf{J}^{(l)}$ are the most time-consuming parts, leading to a computational complexity of

order $O(Lt_u(n_s^2p + n_s p^2 + n_t^2p + n_t p^2 + p^3 + n^3))$, where t_u is the number of iterations in this step, and $n = n_s + n_t$. Finally, considering $n_s, n_t > p$ for the current task, the overall computational complexity of the proposed model is $O(L((t_p + t_u)(n_s^2p + n_s p^2 + n_t^2p + n_t p^2) + t_u n^3))$.

4. Experimental Results

4.1. Human Motion Datasets

We conduct the comparison experiments on four typical human motion datasets (see Fig. 2 for some example frames) as follows: • **Keck Gesture Dataset (Keck)** [21] consists of 14 different actions based on military signals with a frame size of 640×480 . In this dataset, subjects perform 14 gestures and actions. The videos were obtained by using a fixed camera with the subjects standing in front of a static and simple background. • **Multi-Modal Action Detection Dataset (MAD)** [17] consists of actions captured in multiple modalities by using a Microsoft Kinect V2 system in RGB, depth and skeleton formats. Specifically, the RGB frames are with the size of 240×320 , and 3D depth images are with the size of 240×320 . Besides, each subject performs 35 actions in two different indoor environments. • **Weizmann Dataset (Weiz)** [14] consists of 90 video sequences, which include 10 actions (running, walking, skipping, bending, etc.) performed by nine subjects in an outdoor environment. All videos have the size of 180×144 with 50 fps. • **UT-Interaction Dataset (UT)** [40] consists of 20 videos, each of which includes six classes of human-human interactions (*e.g.*, punching, kicking, pushing, hugging, pointing, and handshaking). All video sequences are around 60 seconds long.

4.2. Experimental Setup

Dataset settings. Following the dataset preprocessing in [47], we utilize the extracted HOG features [6] with a 324-dimensional feature vector for each frame. To make segmentation results comparable across different datasets, all input videos are modified to be a sequence of 10 actions using the same settings as in [47]. In model evaluations, we randomly select five sequences as the source data and then report the average performance.

Compared methods. We compare the proposed model with the following state-of-the-art methods: (1) Spectral Clustering (SC) [33]. The features of target samples are fed into the standard spectral clustering algorithm [33] to obtain the clustering results. (2) K-medoids (KMD) selects target samples as centers and clusters them using a generalization of the Manhattan Norm to measure the distance between points. (3) Low-Rank Representation (LRR) [29] incorporates a low-rank constraint on the representation coefficients. (4) Ordered Subspace Clustering (OSC) [44] takes a temporal constraint and forces representations of the tem-



Figure 2: Sampling frames of four human motion datasets.

poral data to be similar. (5) Sparse Subspace Clustering (SSC) [8] assumes that there exists a dictionary that can represent all data points by using a sparse combination. It also applies a sparse constraint to the representation coefficients. (6) Least Square Regression (LSR) [32] utilizes the Frobenius norm to encourage a grouping effect which tends to cluster highly correlated data together. (7) Temporal Subspace Clustering (TSC) [25] presents a temporal Laplacian regularization and a jointly learned dictionary to learn distinctive codes for human motion data. (8) Transfer Subspace Segmentation (TSS) [46] utilizes auxiliary data and transfers segmentation knowledge from a source to target dataset. (9) Low-rank Transfer Subspace (LTS) [47] presents a novel sequential graph to preserve temporal information residing in both the source and target data.

Evaluation metrics and parameters settings. To comprehensively compare our proposed method with other state-of-the-art methods, we utilize two popular metrics to evaluate the segmentation quality, *i.e.*, Normalized Mutual Information (NMI) and Accuracy (ACC). Note that, higher values indicate better performance for the two metrics. We first tune α in the range of $\{10^{-5}, 10^{-4}, \dots, 10^2\}$ by fixing the other parameters, obtaining a better performance when $\alpha = 0.1$. Thus, we empirically set α to be 0.1, and tune the parameters β , γ , and δ in the range of $\{10^{-5}, 10^{-4}, \dots, 10^2\}$. Furthermore, the number of layers for our model is set as 4, and the correlated frame distance τ is set to 11.

4.3. Performance Comparison

In all comparison experiments, we set one sequence as the source and another one as the target. As we use four datasets for our evaluations, we report the segmentation results when testing on one dataset at one time, using the remaining three as the source domains. Besides, since SC, KMD, LRR, OSC, SSC, and LSR are not designed to utilize source information, we only employ target videos as input for these methods. For the TSC, TSS, and LRT methods, we input both source and target videos for segmentation. The comparison segmentation results are shown in Table 1, where **bold** indicates the best performance. Compared with SC, KMD, LRR, OSC, SSC, and LRR, our method transfers useful information from source data to learn distinctive representations of the target data, resulting in improving the segmentation performance. Compared with trans-

Table 1: Clustering comparison results in terms of NMI and ACC on four human motion datasets. Names in brackets indicate the source datasets. M, K, W, and U denote MAD, Keck, Weizmann, and UT-interaction, respectively. The best clustering results are denoted in bold when using the same source data.

(a) Results on Keck dataset			(b) Results on MAD dataset			(c) Results on Weizman dataset			(d) Results on UT dataset		
Method	NMI	ACC	Method	NMI	ACC	Method	NMI	ACC	Method	NMI	ACC
SC	0.4744	0.3886	SC	0.4369	0.3639	SC	0.5435	0.4127	SC	0.4894	0.4477
KMD	0.4702	0.3970	KMD	0.3914	0.3226	KMD	0.5289	0.4441	KMD	0.5108	0.5122
LRR	0.4862	0.4297	LRR	0.2249	0.2397	LRR	0.4382	0.3638	LRR	0.4051	0.4162
OSC	0.5931	0.4393	OSC	0.5589	0.4327	OSC	0.7047	0.5216	OSC	0.6877	0.5846
SSC	0.3858	0.3137	SSC	0.4758	0.3817	SSC	0.6009	0.4576	SSC	0.4998	0.4389
LSR	0.4548	0.4894	LSR	0.3667	0.3979	LSR	0.5093	0.5091	LSR	0.4322	0.5183
TSC(M)	0.6935	0.4653	TSC(K)	0.7691	0.5473	TSC(K)	0.7971	0.5931	TSC(K)	0.7216	0.5213
TSS(M)	0.8049	0.5395	TSS(K)	0.8286	0.5792	TSS(K)	0.8326	0.6030	TSS(K)	0.7746	0.5371
LTS(M)	0.8226	0.5509	LTS(K)	0.8244	0.5874	LTS(K)	0.8599	0.6391	LTS(K)	0.7961	0.6127
Ours(M)	0.8270	0.6010	Ours(K)	0.8099	0.6125	Ours(K)	0.8371	0.6436	Ours(K)	0.8121	0.6148
TSC(W)	0.6862	0.4548	TSC(W)	0.8202	0.5736	TSC(M)	0.8032	0.5961	TSC(M)	0.7442	0.5288
TSS(W)	0.7928	0.5485	TSS(W)	0.8202	0.5736	TSS(M)	0.8509	0.6208	TSS(M)	0.7783	0.5335
LTS(W)	0.7983	0.5649	LTS(W)	0.8213	0.5906	LTS(M)	0.8579	0.6156	LTS(M)	0.8128	0.6299
Ours(W)	0.8196	0.5915	Ours(W)	0.8307	0.6158	Ours(M)	0.8232	0.6348	Ours(M)	0.8239	0.6433
TSC(U)	0.6797	0.4421	TSC(U)	0.7691	0.5315	TSC(U)	0.7796	0.5402	TSC(W)	0.7136	0.5111
TSS(U)	0.7937	0.4951	TSS(U)	0.8108	0.5479	TSS(U)	0.8124	0.5865	TSS(W)	0.7878	0.5944
LTS(U)	0.7947	0.5519	LTS(U)	0.8211	0.5980	LTS(U)	0.8267	0.6122	LTS(W)	0.8035	0.6296
Ours(U)	0.8120	0.6105	Ours(U)	0.8314	0.6163	Ours(U)	0.8351	0.6371	Ours(W)	0.8198	0.6463



Figure 3: Visualization of clustering results on a sample video of the Keck dataset. The ten colors denote ten different temporal clusters.

fer clustering-based segmentation methods (including TSC, TSS, and TSS), our method also obtains much better performance. This is because our approach simultaneously explores domain-invariant features and preserves domain-specific knowledge. These two aspects are equally important for transfer learning. Additionally, our method fuses multi-level representations to construct the affinity matrix for motion segmentation, which effectively preserves the structural information from different layers.

In Fig. 3, we visualize the clustering results rendered by our method as well as other comparison methods on a sample video of the Keck dataset. Different colors in-

dicating different action clusters. As can be seen, the LRR and SSC methods generate multiple fragments and cannot achieve meaningful and accurate segmentation. This is because they do not consider the temporal information. Compared with LRR and SSC, TSC performs better but it still generates some unexpected fragments. LTS and TSS obtain relative better performance in most cases, but they occasionally generate fragments in segmentation results. Overall, our method obtains continuous segments and achieves much better segmentation results than other methods.

4.4 Model Study

Parameter sensitivity. In our approach, three key regularization parameters, *i.e.*, λ , γ , and β , need to be manually tuned. To investigate the effects of the three parameters on the model output, we fix the value of one parameter and change the other two parameters. The experimental results on the Keck dataset are shown in Fig. 4 (a)(b)(c). From the results, it can be observed that our proposed method obtains much better NMI performance when $\lambda \in [0.001, 1]$, $\gamma \in [0.001, 0.1]$, and $\beta \in [0.01, 1]$. Moreover, the experimental results also indicate that every term in our model is useful for improving the segmentation results.

Convergence analysis. We compute the relative errors (*i.e.*, $[\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)} \mathbf{Z}^{(l)} - \mathbf{E}^{(l)}$ and $\mathbf{W}^{(l)} - \mathbf{J}^{(l)}$) to demonstrate the convergence of our optimization algorithm. We report the mean values of different layers in the two terms, and the convergence curves on the Keck dataset are

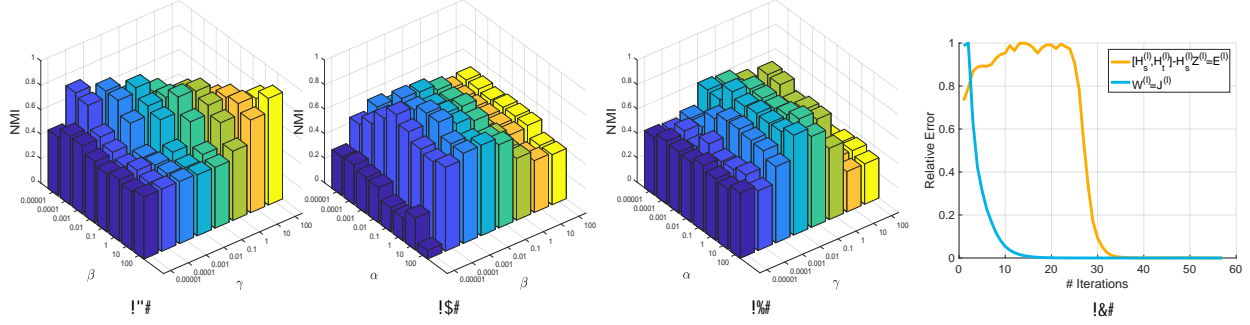


Figure 4: Parameter sensitivity and convergence analysis on Keck dataset. (a) Sensitivity analysis for parameters α and β , (b) Sensitivity analysis for parameters γ and δ , (c) Sensitivity analysis for parameters ϵ and ζ , and (d) Convergence curves.

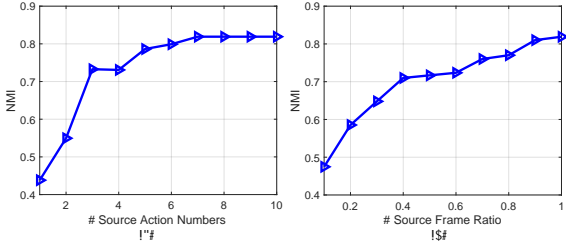


Figure 5: Segmentation results based on (a) different action numbers and (b) different frame ratios in each action.

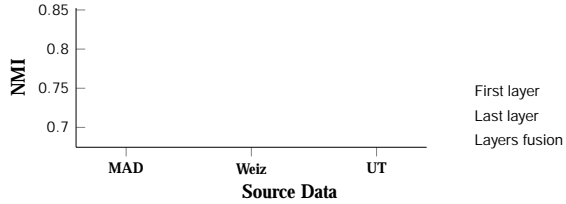


Figure 6: Performance comparison (NMI) when using representations from different layers or multi-layer fusion.

presented in Fig. 4 (d). Note that, for a better presentation, the errors are normalized into the range $[0, 1]$. As can be observed, our model converges within about 50 iterations.

Source data analysis. To evaluate the effectiveness of the source information for the segmentation task, we first test the source action video (UT as an example) that contains different numbers of actions, and the results on the Keck dataset are shown in Fig. 5 (a). As can be observed, the performance increases when the number of actions increases. This indicates that the diversity of the source data is crucial for improving the performance. More actions in the source data can transfer much useful knowledge to ensure that our model learns the distinctive representation of the target data. Besides, we utilize the frames with different ratios (*i.e.*, 0.1, 0.2, \dots , 1) of each action, while keeping the number of actions to be consistent. We evaluate the performance on the Keck dataset, as shown in Fig. 5 (b). The results indicate that the performance of our model increases

when the ratio of frames increases.

Ablation study. To validate the effectiveness of fusing the multi-level subspace representations from different feature spaces, we show the results of our method on the Keck dataset when using the representations from the first layer, the last layer and the fused multi-layers in Fig. 6. It can be observed that our fusion strategy obtains much better performance than conducting subspace learning only on the representation from the first layer or the last layer. This indicates the effectiveness of our model which fuses the multi-level subspace representations for transfer learning.

5. Conclusion

We have proposed a multi-mutual consistency induced transfer subspace learning framework for human motion segmentation. Our model first factorizes the original features of the source and target data into implicit multi-layer feature spaces, in which we use a mutual consistency learning strategy to reduce the distribution difference between the two domains. Then, we carry out the transfer subspace learning in multi-level feature spaces to effectively exploit different-level structural information. Furthermore, we present a temporal correlation preservation term to improve the effectiveness of learned representations. We obtain the final representation by fusing multiple subspace representations from different layers. Experimental results on benchmark datasets show that our method can significantly outperform the state-of-the-art methods. In the future, we can apply our multi-level feature representations to other related tasks, such as multi-modal learning [58], multi-source object detection [9], etc.

Acknowledgements: This research was supported in part by NSF of China (No:61973162), NSF of Jiangsu Province (No:BK20171430), the Fundamental Research Funds for the Central Universities (No:30918011319), Zhejiang Lab’s Open Fund (No:2019KD0AB04), CCF-Tencent Open Fund, the “Young Elite Scientists Sponsorship Program” by Jiangsu Province, and the “Young Elite Scientists Sponsorship Program” by CAST (No:2018QNR001).

References

- [1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge university press, 2004. 4, 5
- [2] L. Bruzzone and M. Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *IEEE TPAMI*, 32(5):770–787, 2009. 2
- [3] J.-F. Cai, Emmanuel J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010. 5
- [4] X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh. Constrained multi-view video face clustering. *IEEE TIP*, 24(11):4381–4393, 2015. 1
- [5] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*, pages 4109–4118, 2018. 1
- [6] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 6
- [7] Z. Ding and Y. Fu. Deep transfer low-rank coding for cross-domain learning. *IEEE TNNLS*, 30(6):1768–1779, 2018. 3
- [8] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE TPAMI*, 35(11):2765–2781, 2013. 1, 2, 6
- [9] K. Fu, D.-P. Fan, G.-P. Ji, and Q. Zhao. JI-dcf: Joint learning and densely-cooperative fusion framework for rgb-d salient object detection. In *CVPR*, 2020. 8
- [10] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *ICML*, pages 1180–1189, 2015. 3
- [11] M. Geng, Y. Wang, T. Xiang, and Y. Tian. Deep transfer learning for person re-identification. *arXiv preprint arXiv:1611.05244*, 2016. 3
- [12] B. Gholami and V. Pavlovic. Probabilistic temporal subspace clustering. In *CVPR*, pages 3066–3075, 2017. 1
- [13] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, pages 222–230, 2013. 2
- [14] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE TPAMI*, 29(12):2247–2253, 2007. 6
- [15] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 5
- [16] H. Hu, Z. Lin, J. Feng, and J. Zhou. Smooth representation clustering. In *CVPR*, pages 3834–3841, 2014. 1, 2
- [17] D. Huang, S. Yao, Y. Wang, and Fe. De La Torre. Sequential max-margin event detectors. In *ECCV*, pages 410–424. Springer, 2014. 6
- [18] J. Huang, F. Nie, and H. Huang. A new simplex sparse learning model to measure data similarity for clustering. In *IJCAI*, 2015. 5
- [19] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid. Deep subspace clustering networks. In *NIPS*, pages 24–33, 2017. 2
- [20] S. Jiang, Z. Ding, and Y. Fu. Heterogeneous recommendation via deep low-rank sparse collective factorization. *IEEE TPAMI*, 2019. 3
- [21] Z. Jiang, Z. Lin, and L. Davis. Recognizing human actions by learning and matching shape-motion prototype trees. *IEEE TPAMI*, 34(3):533–547, 2012. 6
- [22] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. In *ACM Sigmod Record*, volume 30, pages 151–162. ACM, 2001. 1
- [23] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and Knowledge Discovery*, 7(4):349–371, 2003. 1
- [24] H. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM TKDD*, 3(1):1, 2009. 1
- [25] S. Li, K. Li, and Y. Fu. Temporal subspace clustering for human motion segmentation. In *ICCV*, pages 4453–4461, 2015. 1, 2, 4, 6
- [26] T. Li, Z. Liang, S. Zhao, J. Gong, and J. Shen. Self-learning with rectification strategy for human parsing. In *CVPR*, 2020. 1
- [27] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool. Domain generalization and adaptation using low rank exemplar svms. *IEEE TPAMI*, 40(5):1114–1127, 2017. 2
- [28] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In *NIPS*, pages 612–620, 2011. 4
- [29] G. Liu, Z. Lin, and et al. Robust recovery of subspace structures by low-rank representation. *IEEE TPAMI*, 35(1):171–184, 2013. 1, 2, 3, 4, 6
- [30] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE TPAMI*, 35(1):171–184, 2012. 5
- [31] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015. 3
- [32] 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 *ECCV*, pages 347–360. Springer, 2012. 1, 6
- [33] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2002. 1, 6
- [34] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *CVPR*, pages 692–699, 2013. 2
- [35] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. Hierarchical multimodal lstm for dense visual-semantic embedding. In *ICCV*, pages 1881–1889, 2017. 3
- [36] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, pages 1029–1038, 2016. 3
- [37] X. Peng, J. Feng, S. Xiao, W. Y. Yau, J. T. Zhou, and S. Yang. Structured autoencoders for subspace clustering. *IEEE TIP*, 27(10):5076–5086, 2018. 2
- [38] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, pages 401–417, 2018. 1

- [39] M. W. Robards and P. Suneag. Semi-markov kmeans clustering and activity recognition from body-worn sensors. In *ICDM*, pages 438–446. IEEE, 2009. 2
- [40] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: video structure comparison for recognition of complex human activities. In *ICCV*, volume 1, page 2. Citeseer, 2009. 6
- [41] M. Shao, D. Kit, and Y. Fu. Generalized transfer subspace learning through low-rank constraint. *IJCV*, 109(1-2):74–93, 2014. 3
- [42] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. Generalized domain-adaptive dictionaries. In *CVPR*, pages 361–368, 2013. 2
- [43] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, pages 1154–1160. IEEE, 1998. 4
- [44] S. Tierney, J. Gao, and Y. Guo. Subspace clustering for sequential data. In *CVPR*, pages 1019–1026, 2014. 6
- [45] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, pages 4068–4076, 2015. 3
- [46] L. Wang, Z. Ding, and Y. Fu. Learning transferable subspace for human motion segmentation. In *AAAI*, 2018. 1, 2, 4, 6
- [47] L. Wang, Z. Ding, and Y. Fu. Low-rank transfer human motion segmentation. *IEEE TIP*, 28(2):1023–1034, 2018. 1, 2, 6
- [48] W. Wang, Z. Zhang, S. Qi, J. Shen, Y. Pang, and L. Shao. Learning compositional neural information fusion for human parsing. In *ICCV*, pages 5703–5713, 2019. 1
- [49] Y. Xiong and D.-Y. Yeung. Mixtures of arma models for model-based time series clustering. In *ICDM*, pages 717–720. IEEE, 2002. 1
- [50] Y. Xu, X. Fang, Ji. Wu, X. Li, and D. Zhang. Discriminative transfer subspace learning via low-rank and sparse representation. *IEEE TIP*, 25(2):850–863, 2015. 3
- [51] M. Ye and J. Shen. Probabilistic structural latent representation for unsupervised embedding. In *CVPR*, 2020. 3
- [52] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, pages 3712–3722, 2018. 1
- [53] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu. Generalized latent multi-view subspace clustering. *IEEE TPAMI*, 42(1):86–99, 2018. 2
- [54] J. Zhang, W. Li, and P. Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. In *CVPR*, pages 1859–1867, 2017. 2
- [55] H. Zhao, Z. Ding, and Y. Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927, 2017. 2
- [56] F. Zhou, F. De la Torre, and J. K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE TPAMI*, 35(3):582–596, 2012. 2
- [57] P. Zhou, Y. Hou, and J. Feng. Deep adversarial subspace clustering. In *CVPR*, pages 1596–1604, 2018. 2
- [58] T. Zhou, H. Fu, G. Chen, J. Shen, and L. Shao. Hi-Net: Hybrid-fusion network for multi-modal MR image synthesis. *IEEE TMI*, 2020. 8
- [59] T. Zhou, W. Wang, S. Qi, H. Ling, and J. Shen. Cascaded human-object interaction recognition. In *CVPR*, 2020. 1
- [60] T. Zhou, C. Zhang, C. Gong, H. Bhaskar, and J. Yang. Multiview latent space learning with feature redundancy minimization. *IEEE TCYB*, 2018. 2
- [61] T. Zhou, C. Zhang, X. Peng, H. Bhaskar, and J. Yang. Dual shared-specific multiview subspace clustering. *IEEE TCYB*, 2019. 2
- [62] F. Zhu and L. Shao. Weakly-supervised cross-domain dictionary learning for visual recognition. *IJCV*, 109(1-2):42–59, 2014. 3