Consistency and Diversity induced Human Motion Segmentation

Tao Zhou, *Member, IEEE,* Huazhu Fu, *Senior Member, IEEE,* Chen Gong, Ling Shao, *Fellow, IEEE,* Fatih Porikli, *Fellow, IEEE*, Haibin Ling, Jianbing Shen, *Senior Member, IEEE*

Abstract—Subspace clustering is a classical technique that has been widely used for human motion segmentation and other related tasks. However, existing segmentation methods often cluster data without guidance from prior knowledge, resulting in unsatisfactory segmentation results. To this end, we propose a novel **C**onsistency and **D**iversity induced human **M**otion **S**egmentation (CDMS) algorithm. Specifically, our model factorizes the source and target data into distinct multi-layer feature spaces, in which transfer subspace learning is conducted on different layers to capture multi-level information. A multi-mutual consistency learning strategy is carried out to reduce the domain gap between the source and target data. In this way, the domain-specific knowledge and domain-invariant properties can be explored simultaneously. Besides, a novel constraint based on the Hilbert Schmidt Independence Criterion (HSIC) is introduced to ensure the diversity of multi-level subspace representations, which enables the complementarity of multi-level representations to be explored to boost the transfer learning performance. Moreover, to preserve the temporal correlations, an enhanced graph regularizer is imposed on the learned representation coefficients and the multi-level representations of the source data. The proposed model can be efficiently solved using the Alternating Direction Method of Multipliers (ADMM) algorithm. Extensive experimental results on public human motion datasets demonstrate the effectiveness of our method against several state-of-the-art approaches.

Index Terms—Subspace clustering, human motion segmentation, transfer learning, multi-level representation.

1 Introduction

Human motion segmentation has received widespread interest in industry and research communities due to its extensive applications in video retrieval, virtual reality, and smart surveillance for user interfaces and human action analysis [2], [3], [4]. The main goal of human motion segmentation is to partition temporal data sequences that depict human activities and actions into a set of non-overlapping and internally temporal segments. More importantly, it is often used as a preprocessing step before motion-related analytical tasks. Currently, there are several significant challenges for human motion segmentation, including the lack of well-defined activity fragments, and the ambiguity in motion primitives caused by temporal variability among different actions [5].

Human motion data contains temporal information, which is critical for motion segmentation. However, due to the high-dimensional structure of visual representations and complexity of temporal correlations, it is still challenging to capture such discriminative temporal information from time-series data [6]. Several algorithms have been proposed to address this challenge, including model-based [7], temporal proximity-based [6], and representation-based [8]. Moreover, since temporal data occurs in the form of consecutive frame samples, human motion segmentation can be treated as an unsupervised clustering task [9]. Among clustering-based motion segmentation methods, subspace clustering-based approaches [8], [10] have attracted increasing attention and obtained promising segmentation performance.

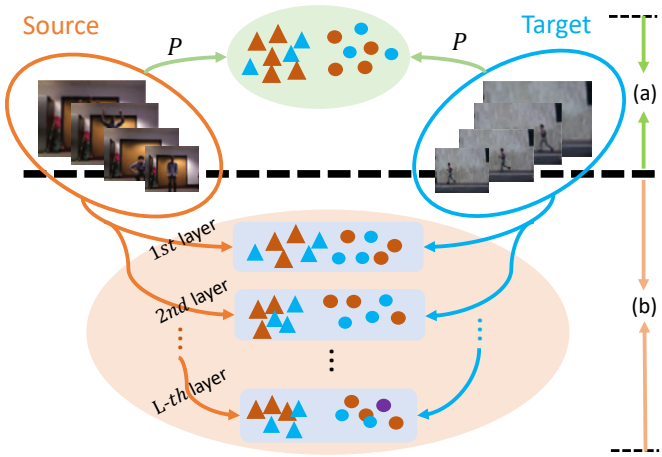

Figure 1: Comparison of different transfer subspace learning algorithms: (a) low-dimensional feature reconstruction based transfer subspace learning, and (b) multi-level feature reconstruction based transfer subspace learning. Different shapes denote the data points from the source or target domain, and the two colors denote different classes.

- *Tao Zhou and Chen Gong are with the PCA lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China. (e-mails: taozhou.dreams@gmail.com, chen.gong@njust.edu.cn).*
- *Huazhu Fu is with Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore (e-mail: hzfu@ieee.org).*
- *Ling Shao is with National Center for Artificial Intelligence (NCAI), Saudi Data and Artificial Intelligence Authority (SDAIA), Riyadh, Saudi Arabia (e-mail: ling.shao@ieee.org).*
- *Fatih Porikli is with the Research School of Engineering, the Australian National University (email: fatih.porikli@anu.edu.au).*
- *Haibin Ling is with the Department of Computer Science, Stony Brook University, Strony Brook, NY, USA (Email: hling@cs.stonybrook.edu).*
- *Jianbing Shen is with the State Key Laboratory of Internet of Things for Smart City, Department of Computer and Information Science, University of Macau, Macau, China. (e-mail: shenjianbingcg@gmail.com).*
- *A preliminary version of this work has appeared in CVPR 2020 [1].*
- *Corresponding author: Jianbing Shen.*

Subspace clustering is a popular technique to partition a given data set into different groups, based on the assumption that data points are drawn from multiple subspaces corresponding to different classes [11], [12]. In general, current subspace clustering approaches can be categorized into four types, including algebraic approaches [13], iterative methods [14], statistical methods, and spectral clustering methods [12]. Over the last few decades, several representative subspace clustering methods [20], [21], [22], [23] have been proposed to learn distinct and low-dimensional data representations, in which the learned representations can be fed to classic clustering algorithms (*e.g.*, spectral clustering [12]). However, these unsupervised subspace learning approaches often obtain unsatisfactory results due to the lack of some prior knowledge. Since labeled data from related tasks are often easy to obtain, transfer learning provides an effective solution to borrow knowledge from the related source data to improve the performance of target tasks [24], [25]. For the human motion segmentation task, transfer subspace learning-based approaches [26], [27] have been developed and demonstrated improved performance.

Although effectiveness has been achieved in transfer subspace learning-based human motion segmentation, several problems remain for existing methods. *First*, subspace clustering-based approaches tend to reconstruct data points (*e.g.*, a self-representation strategy) using low-dimensional feature representations (as shown in Fig. 1), however, few approaches conduct transfer subspace learning in multi-level feature spaces to simultaneously capture low- and high-level information. *Second*, transfer subspace learning forces the data distributions of two domains to be similar. To achieve this, one widely-used strategy is to project the original features of source and target data into a common low-dimensional feature space. Obviously, this strategy captures domain-invariant properties while ignoring some potentially useful domain-specific knowledge. Since both aspects are equally essential in transfer learning, it is important to balance them for boosting the model performance.

To this end, we propose a novel **C**onsistency and **D**iversity induced human **M**otion **S**egmentation (CDMS) algorithm, which incorporates transfer learning and multi-level subspace clustering in a unified framework, to enhance human motion segmentation (as shown in Fig. 2). Specifically, we first factorize the original features extracted from the source and target data into implicit multi-layer feature spaces by using a Non-negative Matrix Factorization (NMF), in which we conduct transfer subspace learning in multi-level feature spaces. A multi-mutual consistency learning strategy is proposed to reduce the distribution gap between the two domains. Moreover, we propose a novel constraint based on the Hilbert Schmidt Independence Criterion (HSIC) to enhance the diversity of the learned representation coefficients. Further, an enhanced graph regularizer is imposed on the learned representation coefficients and feature representations of the source data set to preserve the temporal correlations. Finally, we show that our model can be efficiently solved using the Alternating Direction Method of Multipliers (ADMM) algorithm. Experimental results on multiple benchmark datasets demonstrate the superiority of our model over other state-of-the-art approaches.

In summary, the key **contributions** are as follows:

- We present a novel human motion segmentation algorithm, which integrates multi-level subspace learning and transfer learning into a unified framework. Our model aims to transfer knowledge from related source data to boost the performance of target data in the human motion segmentation task. To the best of our knowledge, this work is the first to develop multi-level subspace learning for human motion segmentation.
- A deep NMF structure is built to capture the hidden information by leveraging the benefits of strong interpretability from the NMF model. Through this deep NMF structure, we obtain multi-level non-negative representations based on different dimensionalities of the dictionary atoms in multi-layer feature spaces, which reduce the distribution difference between the source and target domains.
- Our model explores domain-invariant properties by using a multi-mutual consistency learning strategy while preserving domain-specific knowledge. The main advantage of our approach is to automatically balance the two aspects for enhancing the ability of transfer subspace learning.
- We propose a novel constraint term to ensure that the learned multi-level subspace representation coefficients are diverse, which can help explore the complementary information from multi-level feature spaces to boost the segmentation performance.

This paper significantly extends our previous work in the conference paper [1], with multi-fold aspects. First, we reformulate a new multi-level transfer subspace learning framework with three key components, *i.e.*, multi-mutual consistency learning, diversity cross multi-level representation, and temporal correlation preservation, for human motion segmentation. Second, we propose to utilize the HSIC as a diversity constraint for explicitly encouraging the learned multi-level representations to be of sufficient diversity, which can enhance complementary information to boost transfer subspace learning performance. Third, we provide deeper insight into the proposed multi-level transfer subspace learning framework. In this regard, our model is a general approach, as both hand-crafted HOG features and deep CNN features can be fed into our model. Last but not least, more experiments and additional ablation studies are carried out to further investigate the effectiveness of the proposed model and different key components.

The rest of this paper is arranged as follows. Related works, including subspace clustering, human motion analysis, temporal data clustering, and transfer learning, are briefly introduced in Section 2. We present the details of the proposed method in Section 3. We provide the experimental settings, experimental results, and model study in Section 4. Finally, we conclude this paper in Section 5.

2 Related Work

There are four types of works that are most related to the proposed human motion segmentation method, including 1) subspace clustering, 2) human motion analysis, 3) temporal data clustering, and 4) transfer learning.

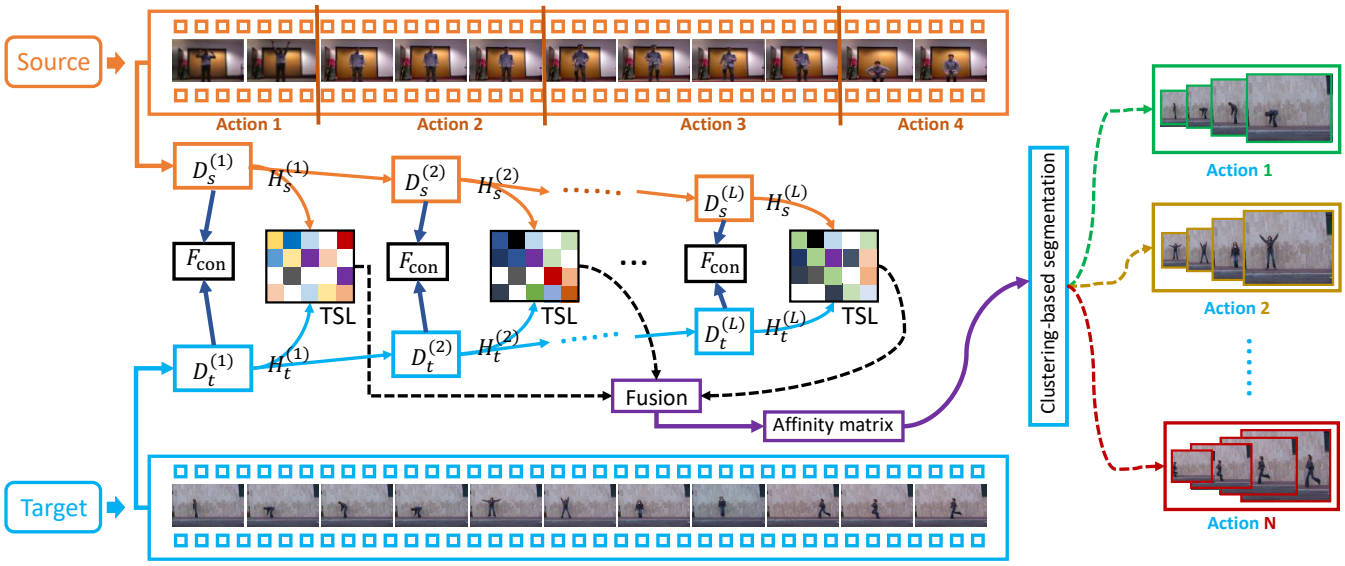

Figure 2: Overview of multi-level transfer subspace learning framework for human motion segmentation. Our model first factorizes the source and target data into multi-layer implicit feature spaces using a deep NMF model, in which multi-level transfer subspace learning (TSL) is carried out in different spaces (*i.e.,* layers) to capture multi-level information. Then, a multi-mutual consistency learning strategy is presented to reduce the difference in feature distribution between the two domains. After that, we construct a novel affinity matrix by fusing multi-level representation coefficients. Finally, the Normalized Cuts algorithm can be applied to the learned affinity matrix to obtain the segmentation (clustering) results.

2.1 Subspace Clustering

Subspace clustering [15], [16], [17], [18], [19] holds the assumption that data points can be drawn from multiple subspaces corresponding to different clustering groups. Currently, self-representation based subspace clustering is increasing attention, in which each data point is expressed using a linear combination of other data points. For example, sparse subspace clustering (SSC) [20] aims to find the sparsest representation among the infinitely possible representations based on ℓ_1-norm. Different from SSC, low-rank representation clustering (LRR) [21] attempts to uncover hidden structures with a low-rank representation. By introducing a graph regularizer, smooth representation clustering (SMR) [22] investigates the grouping effect of representation-based algorithms. Moreover, there are also several deep learning-based subspace clustering approaches [28], [29], [30], [32], [57]. However, these methods cannot be directly applied to motion segmentation since they do not consider the temporal correlations in successive frames.

2.2 Human Motion Analysis

Over the past decades, several methods have been developed for human motion analysis. For example, Zhong *et al.* [33] develop a bipartite graph co-clustering framework to segment unusual activities in video. Jenkins *et al.* [34] adopt the zero-velocity crossing points of the angular velocity to partition a stream of motion data into different sequences. Barbic *et al.* [35] develop to decompose human motion into multiple distinct actions using the probabilistic principal component analysis algorithm. Further, Beaudoin *et al.* [36] present a new framework to automatically distill a motion-motif graph from an arbitrary collection of motion data. Moreover, several clustering-based approaches have been proposed to segment a stream of human behavior into several activities [37].

2.3 Temporal Data Clustering

The goal of the temporal data clustering task is to segment data sequences into a set of non-overlapping parts. It has a wide range of applications, from facial analytics, and speech segmentation to human action recognition. To achieve this, a semi-Markov K-means clustering [38] model is often used to exploit repetitive patterns. For instance, Zhou *et al.* [3] develop a K-means kernel associated with a dynamic temporal alignment framework. Temporal subspace clustering (TSC) [8] method learns expressive coding coefficients based on a non-negative dictionary learning algorithm, which also introduces a temporal Laplacian regularization term to exploit the temporal correlations. Transfer subspace segmentation (TSS) [26] borrows knowledge from relevant source data to boost the target tasks. Low-rank transfer subspace (LTS) [27] method utilizes a domain-invariant projection to reduce the distribution gaps between the two domains, which constructs a graph regularizer to capture the temporal correlations. Sun *et al.* [39] develop an online multi-task clustering framework for multi-agent human motion segmentation, which formulates a linear encoder-decoder architecture. These temporal clustering methods are all formulated as unsupervised learning frameworks, some of which adopt a self-representation strategy to achieve the motion segmentation task.

2.4 Transfer Learning

Transfer learning aims to borrow prior knowledge from related tasks as source data to improve the performance of target tasks. Plenty of transfer learning models [40], [41], [42], [43], [44], [45] have been developed and obtain promising performance. Among these methods, domain-invariant feature learning is a popular strategy [45] to learn a common feature space where both the domain shift and distribution difference can be mitigated. Several works, such

Table 1: Main notations used in the proposed model.

Notation	Description
$\mathbf{X}_s$	Feature matrix of the source data
$\mathbf{X}_t$	Feature matrix of the target data
$\mathbf{D}_s^{(l)}$	Basis matrix of the source data in the l-th layer
$\mathbf{D}_t^{(l)}$	Basis matrix of the target data in the l-th layer
$\mathbf{H}_s^{(l)}$	Representation matrix of the source data in the l-th layer
$\mathbf{H}_t^{(l)}$	Representation matrix of the target data in the l-th layer
$\mathbf{Z}_s^{(l)}$	Representation coefficient of $\mathbf{X}_s$ in the l-th layer
$\mathbf{Z}_t^{(l)}$	Representation coefficient of $\mathbf{X}_t$ in the l-th layer
$\mathbf{L}$	Graph Laplace matrix
α, β, γ	Trade-off parameters

as dictionary learning [48], [49] and subspace learning [46], [47], explore the alignment of two domains. Moreover, deep learning has been introduced to integrate knowledge transfer and feature learning into a unified framework [50], [51], [52], [53], [54], [55]. However, most of these methods conduct the domain alignment using high-level features from top layers, while ignoring the low-level structural information. Besides, they mainly focus on domain-invariant feature learning without preserving domain-specific knowledge.

3 The Proposed Method

In this section, we first introduce the motivation (§ 3.1) and provide details of the proposed method (§ 3.2). Moreover, we describe the optimization steps for solving the proposed model (§ 3.3). Finally, we construct a new affinity matrix for clustering-based motion segmentation (§ 3.4), and provide complexity analysis (§ 3.5).

3.1 Motivation

As previously mentioned, three main challenges remain for human motion segmentation using transfer subspace learning, *i.e.*, *(1)* how to capture multi-level information to enhance the performance of transfer subspace learning; *(2)* how to decrease the distribution gap between the two domains while also preserving domain-specific knowledge; and *(3)* how to effectively preserve temporal correlations among motion data. To address these challenges, we propose a novel multi-level transfer subspace learning framework with three key components for human motion segmentation, and the details of each component will be provided in the following subsections.

Moreover, deep structure learning has proven its effectiveness in several machine learning and computer vision applications [56], [57], [58], [59]. To effectively capture multi-level structural information, we adopt a multi-layer decomposition process based on the deep NMF model, which can be formulated as:

$$\begin{aligned}\mathbf{X} &\approx \mathbf{D}^{(1)}\mathbf{H}^{(1)} \\ &\approx \mathbf{D}^{(1)}\mathbf{D}^{(2)}\mathbf{H}^{(2)} \\ &\vdots \\ &\approx \mathbf{D}^{(1)}\mathbf{D}^{(2)}\cdots\mathbf{D}^{(l)}\cdots\mathbf{D}^{(L)}\mathbf{H}^{(L)},\end{aligned} \tag{1}$$

where $\mathbf{D}^{(l)} \geq 0$ and $\mathbf{H}^{(l)} \geq 0$ $(l = 1, \ldots, L)$ represent the base matrix and feature representation in the l-th layer, respectively. And L denotes the number of layers. Through this deep NMF structure, we obtain multi-level non-negative representations based on different dimensionalities of the dictionary atoms, which can reduce the influence of the distribution gap between the two domains in a layer-wise fashion. It can be worth noting that the learned dictionary (base matrix) are not arbitrary in our NMF-based model with the non-negativity constraint.

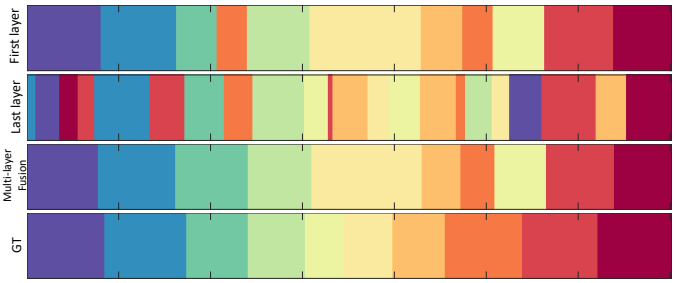

Figure 3: Comparison segmentation results of using different coefficient representations from the first layer, last layer, and multi-layer fusion to construct the affinity matrix on the Weiz dataset.

3.2 Multi-level Transfer Subspace Learning

Existing subspace clustering or subspace clustering-based motion segmentation approaches tend to reconstruct original data points using either shallow representations (*e.g.*, handcrafted features) or high-level representations (*e.g.*, features from the last layer of deep networks). However, the original data often contains complex structural information and hierarchical semantics, which are difficult to extract by only using a single-layer clustering strategy. In Fig. 3, we show the clustering results when constructing the affinity matrix using the coefficients from the first layer, last layer, and multi-layer fusion, respectively. As can be seen, the orange part exists in the results when using the first and last layer representation methods, which indicates that the two methods generate multiple fragments and cannot achieve meaningful or accurate segmentation. While the method using multi-level representations can accurately segment the current motion action. Motivated by these results, we propose a multi-level subspace learning strategy to effectively exploit the hierarchical semantics and structural information in a layer-wise fashion, which can be formulated as follows:

$$\begin{aligned}&\mathcal{L}\left(\mathbf{X}_s, \mathbf{X}_t; \mathbf{D}_s^{(l)}, \mathbf{D}_t^{(l)}, \mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}, \mathbf{Z}^{(l)}\right) = \\ &\|\mathbf{X}_s - \mathbf{D}_s^{(1)}\ldots\mathbf{D}_s^{(L)}\mathbf{H}_s^{(L)}\|_F^2 + \|\mathbf{X}_t - \mathbf{D}_t^{(1)}\ldots\mathbf{D}_t^{(L)}\mathbf{H}_t^{(L)}\|_F^2 \\ &\quad + \sum_{l=1}^{L}\|[\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{Z}^{(l)}\|_{2,1}, \\ &s.t.\ \ \mathbf{Z}^{(l)} \geq 0, \mathbf{1}^\top\mathbf{Z}^{(l)} = \mathbf{1}^\top, \forall l = 1, 2, \ldots, L,\end{aligned} \tag{2}$$

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. $\mathbf{D}_s^{(l)} \geq 0$ and $\mathbf{H}_s^{(l)} \geq 0$ $(l = 1, \ldots, L)$ denote the basis matrix and the feature representation matrix at the l-th layer for the source data, respectively (similar for the target data). In addition, d is the original feature dimension, and n_s and n_t are the number of the source and target data, respectively. Here, we have $\mathbf{H}_s \in \mathbb{R}^{d_l\times n_s}$ and $\mathbf{H}_t \in \mathbb{R}^{d_l\times n_t}$ with d_l denoting the feature dimension in the l-th layer, we obtain $\mathbf{Z}_l \in \mathbb{R}^{n_s\times n_s+n_t}, l = 1, 2, \ldots, L$. Moreover, $\mathbf{1}$ denotes a column vector whose elements are all set to one. The first two terms aim to explore the multi-level structures

in both the source and target data, and the third term is used to conduct the multi-level transfer subspace learning.

In addition, in Eq. (2), the non-negative constraint $\mathbf{Z}^{(l)} \geq 0$ enhances the discriminative ability of the learned representations. The constraint $\mathbf{1}^\top \mathbf{Z}^{(l)} = \mathbf{1}^\top$ ensures the sum of each coefficient vector to be one, resulting in the suppression of representation coefficients from different subspaces. In Eq. (2), the feature representations of the source data (*i.e.*, $\mathbf{H}_s^{(l)}$) are adopted as the subspace learning dictionary, which is used to reconstruct the feature representations of both the two domains (*i.e.*, the source and target). By using this reconstruction strategy, it enables that knowledge from the related source data to be effectively transferred to the target task. Besides, $\|\cdot\|_{2,1}$ denotes the $\ell_{2,1}$-norm, which is used to constrain the columns of a matrix to be zero [21], *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, there is an underlying assumption that some corruptions could be sample-specific, *i.e.* some data points may be corrupted while the others are clean.

Remarks: Since diverse information often exists in different domains, it is difficult to transfer useful knowledge from the related source to the target by only using a single subspace. Thus, we carry out subspace learning on multi-level feature spaces to obtain multiple subspace representations and then fuse them to seek an optimal affinity matrix. Moreover, the $\ell_{2,1}$-norm has been shown to be more robust to outliers than the classic Frobenius norm [16], [21].

3.2.1 Multi-mutual Consistency Learning

To decrease the distribution gaps between the two domains as well as preserve the knowledge from different domains, we propose a multi-mutual consistency learning strategy as follows:

$$\mathcal{R}_1 = \sum\nolimits_{l=1}^{L} F_{\text{con}} \left(\mathbf{D}_s^{(l)}, \mathbf{D}_t^{(l)} \right), \tag{3}$$

where the goal of the term $F_{\text{con}}(\cdot, \cdot)$ is to reduce the distribution gap between the two domains by penalizing the divergence of two basis matrices in different layers. In contrast, several methods directly project the original features from the source and target data into a common low-dimensional space using a domain-invariant projection matrix, which could result in a loss of an amount of domain-specific knowledge. Although there are different strategies to constrain the consistency between $\mathbf{D}_s^{(l)}$ and $\mathbf{D}_t^{(l)}$, a simple but effective strategy is adopted in our model, *i.e.*, $F_{\text{con}}(\mathbf{D}_s^{(l)}, \mathbf{D}_t^{(l)}) = \|\mathbf{D}_s^{(l)} - \mathbf{D}_t^{(l)}\|_F^2$.

Remarks: The proposed multi-mutual consistency learning strategy can reduce the distribution between source and target domains. By using the strategy, our transfer subspace learning model can transfer useful information from the source domain to benefit the segmentation while also preserving some domain-specific knowledge.

3.2.2 Diversity across Multi-level Representation

In the formulation of Eq. (2), we have presented a multi-level subspace learning strategy to effectively exploit the structural information in a layer-wise fashion. Further, to enhance the complementarity of multi-level representations, the encoded representation coefficients (*i.e.*, $\mathbf{Z}^{(l)}, l = 1, 2, \cdots, L$) of different levels are encouraged to be of sufficient diversity. To achieve this, a diversity regularization term is proposed. For convenience, we define a mapping $\phi(\mathbf{x})$ from $\mathbf{x} \in \boldsymbol{\mathcal{X}}$ to kernel space $\boldsymbol{\mathcal{F}}$, where $\phi(\cdot)$ is used to map original features into kernel space. In this kernel space, the inner product between vectors is given by a kernel function $k_1(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. Let $\boldsymbol{\mathcal{G}}$ be a second kernel space on $\boldsymbol{\mathcal{Y}}$, and we define a mapping function $\phi(\mathbf{y})$ from $\mathbf{y} \in \boldsymbol{\mathcal{Y}}$ to $\boldsymbol{\mathcal{G}}$ with $k_2(\mathbf{y}_i, \mathbf{y}_j) = \langle \phi(\mathbf{y}_i), \phi(\mathbf{y}_j) \rangle$. Inspired by [60], we adopt an empirical version of HSIC to enhance the diversity of representation coefficients in the proposed model, which is given below.

Definition 1. Given a series of n independent observations collected from p_{xy}, *i.e.*, $\mathbf{Z} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\} \subseteq \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{Y}}$, an estimator of HSIC$(\mathbf{Z}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{G}})$ is defined by

$$\text{HSIC}(\mathbf{Z}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{G}}) = (n-1)^{-2} \text{tr}(\mathbf{K}_1 \mathbf{H} \mathbf{K}_2 \mathbf{H}), \tag{4}$$

where $k_{1,ij} := k_1(\mathbf{x}_i, \mathbf{x}_j)$ and $k_{2,ij} := k_2(\mathbf{y}_i, \mathbf{y}_j)$. Here, $h_{ij} := \delta_{ij} - 1/n$ centralizes the two Gram matrices (*i.e.*, $\mathbf{K}_1$ and $\mathbf{K}_2$) in the feature space, ensuring that they have zero mean. Please refer to the details of HSIC in [60], [61].

Thus, to explore more complementary information from different feature spaces, we encourage the multi-level representation coefficients $\mathbf{Z}^{(l)}$ and $\mathbf{Z}^{w)}$ $(w \neq l)$ to be sufficiently diverse. The diversity regularization term can be defined by

$$\mathcal{R}_2 = \sum_{l=1}^{L} \sum_{m \neq l}^{L} \text{HSIC} \left(\mathbf{Z}^{(l)}, \mathbf{Z}^{(m)} \right). \tag{5}$$

Remarks: It is worth noting that the diversity across multi-level representations can help to explore more knowledge from source data, which could boost the performance of human motion segmentation.

3.2.3 Temporal Correlation Preservation

Temporal correlation is important for accurate clustering since human motion data is sequential and consecutive. Thus, it is expected to preserve the temporal information in representation coefficients $\mathbf{Z}$ and feature representations $\mathbf{H}$. To achieve this, we regulate the i-th coefficient's neighbors $[\mathbf{z}_{i-\tau/2}, \cdots, \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, \cdots, \mathbf{z}_{i+\tau/2}]$ to be close to $\mathbf{z}_i$, where τ denotes the number of sequential neighbors. Besides, to learn an effective reconstruction dictionary (*i.e.*, $\mathbf{X}_s$ in our method), we also regulate the i-th coefficient's neighbors $[\mathbf{h}_{i-\tau/2}, \cdots, \mathbf{h}_{i-1}, \mathbf{h}_{i+1}, \cdots, \mathbf{h}_{i+\tau/2}]$ to be close to $\mathbf{h}_i$ in the source data. Thus, this enhanced graph regularizer is imposed on both the learned representation coefficients and feature representations of the source dataset. By using it, our model can effectively uncover the temporal correlations residing in both the source and target data. To achieve this, we first build a weight matrix $\mathbf{S}$ [8], [26], where each element of $\mathbf{S}$ is given as:

$$s_{ij} = \begin{cases} 1, \text{ if } |i-j| \leq \tau, l(x_i) = l(x_j), \text{for source data;} \\ 1, \text{ if } |i-j| \leq \tau, \text{for target data;} \\ 0, \text{ otherwise,} \end{cases} \tag{6}$$

where $l(x_i)$ denotes the action label of the i-th sample x_i from the source data. After that, we have the temporal correlation preservation term as follows:

$$\mathcal{R}_3 = \sum_{l=1}^{L} \left(tr(\mathbf{Z}_l \mathbf{L} \mathbf{Z}_l^\top) + tr(\mathbf{H}_s^{(l)} \mathbf{L}_s \mathbf{H}_s^{(l)^\top}) \right), \tag{7}$$

where $tr(\cdot)$ represents the matrix trace. $\mathbf{L}$ denotes the Laplacian matrix with $\mathbf{L} = \mathbf{C} - \mathbf{S}$, where $\mathbf{C}$ is a diagonal degree matrix with $C_{ii} = \sum_j S_{ij}$, and $\mathbf{L}_s$ is the corresponding part to the source data.

Overall formulation: Finally, we formulate the proposed CDMS model (in Eq. 2) with three key components (in Eqs. (3)(5)(7)) into a unified framework as follows:

$$\begin{aligned}
&\min_{\Omega} \mathcal{L} + \alpha\mathcal{R}_1 + \beta\mathcal{R}_2 + \gamma\mathcal{R}_3 \\
&= \|\mathbf{X}_s - \mathbf{D}_s^{(1)} \dots \mathbf{D}_s^{(L)}\mathbf{H}_s^{(L)}\|_F^2 + \|\mathbf{X}_t - \mathbf{D}_t^{(1)} \dots \mathbf{D}_t^{(L)}\mathbf{H}_t^{(L)}\|_F^2 \\
&\quad + \sum_{l=1}^{L} \|[\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{Z}^{(l)}\|_{2,1} + \alpha \sum_{l=1}^{L} \|\mathbf{D}_s^{(l)} - \mathbf{D}_t^{(l)}\|_F^2 \\
&\quad + \beta \sum_{l=1}^{L}\sum_{m\neq l}^{L} \mathrm{HSIC}(\mathbf{Z}^{(l)}, \mathbf{Z}^{(m)}) \\
&\quad + \gamma \sum_{l=1}^{L} (tr(\mathbf{Z}^{(l)}\mathbf{L}\mathbf{Z}^{(l)\top}) + tr(\mathbf{H}_s^{(l)}\mathbf{L}_s\mathbf{H}_s^{(l)\top})), \\
&\quad s.t.\ \ \mathbf{Z}^{(l)} \geq 0, \mathbf{1}^\top\mathbf{Z}^{(l)} = \mathbf{1}^\top, \forall l = 1, 2, \dots, L,
\end{aligned} \tag{8}$$

where $\Omega = \{\mathbf{D}_s^{(l)} \geq 0, \mathbf{D}_t^{(l)} \geq 0, \mathbf{H}_s^{(l)} \geq 0, \mathbf{H}_t^{(l)z} \geq 0, \mathbf{Z}^{(l)}, \mathbf{W}^{(l)}\}$ $(l = 1, 2, \dots, L)$ is the variable set to be optimized, and α, β, and γ are trade-off parameters. For clarity, the main notations used in this paper are listed in Table 1.

3.3 Optimization

Although the proposed objective function in Eq. (8) is not jointly convex for all variables, we can employ the ADMM [62] algorithm to efficiently solve it. To adopt this strategy to our problem, we introduce two auxiliary variables $\mathbf{J}^{(l)}$ and $\mathbf{E}^{(l)}$ to replace $\mathbf{Z}^{(l)}$ and $[\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{Z}^{(l)}$, respectively. Therefore, we have the following equivalent problem:

$$\begin{aligned}
\mathcal{L}(\Omega) &= \|\mathbf{X}_s - \mathbf{D}_s^{(1)} \dots \mathbf{D}_s^{(L)}\mathbf{H}_s^{(L)}\|_F^2 \\
&\quad + \|\mathbf{X}_t - \mathbf{D}_t^{(1)} \dots \mathbf{D}_t^{(L)}\mathbf{H}_t^{(L)}\|_F^2 \\
&\quad + \textstyle\sum_{l=1}^{L} \|\mathbf{E}^{(l)}\|_{2,1} + \alpha \sum_{l=1}^{L} \|\mathbf{D}_s^{(l)} - \mathbf{D}_t^{(l)}\|_F^2 \\
&\quad + \beta \sum_{l=1}^{L}\sum_{m\neq l}^{L} \mathrm{HSIC}(\mathbf{Z}^{(l)}, \mathbf{Z}^{(m)}) \\
&\quad + \gamma \sum_{l=1}^{L} (tr(\mathbf{Z}^{(l)}\mathbf{L}\mathbf{Z}^{(l)\top}) + tr(\mathbf{H}_s^{(l)}\mathbf{L}_s\mathbf{H}_s^{(l)\top})) \\
&\quad + \textstyle\sum_{l=1}^{L} \Phi\left(\mathbf{\Lambda}_1^{(l)}, [\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{J}^{(l)} - \mathbf{E}^{(l)}\right) \\
&\quad + \textstyle\sum_{l=1}^{L} \Phi\left(\mathbf{\Lambda}_2^{(l)}, \mathbf{Z}^{(l)} - \mathbf{J}^{(l)}\right), \\
&\quad s.t.\ \mathbf{Z}^{(l)} \geq 0, \mathbf{1}^\top\mathbf{Z}^{(l)} = \mathbf{1}^\top, \forall l = 1, 2, \dots, L,
\end{aligned} \tag{9}$$

where $\Phi(\mathbf{\Lambda}, \mathbf{Q}) = \frac{\mu}{2}\|\mathbf{Q}\|_F^2 + \langle\mathbf{\Lambda}, \mathbf{Q}\rangle$, with $\langle\cdot,\cdot\rangle$ denoting the matrix inner product. $\mathbf{\Lambda}_1^{(l)}$ and $\mathbf{\Lambda}_2^{(l)}$ $(l = 1, 2, \dots, L)$ are Lagrangian multipliers, and μ is a penalty scalar. We describe the optimization steps for each subproblem below.

$\mathbf{D}_s$-subproblem: The optimization problem with respect to $\mathbf{D}_s$ is formulated as

$$\begin{aligned}
&\min_{\mathbf{D}_s \geq 0} \|\mathbf{X}_s - \mathbf{D}_s^{(1)} \dots \mathbf{D}_s^{(L)}\mathbf{H}_s^{(L)}\|_F^2 \\
&\quad + \alpha \textstyle\sum_{l=1}^{L} \|\mathbf{D}_s^{(l)} - \mathbf{D}_t^{(l)}\|_F^2, \quad \forall l = 1, 2, \dots, L.
\end{aligned} \tag{10}$$

We can update $\mathbf{D}_s^{(l)}$ in each layer one by one. By taking the derivative of Eq. (10) with respect to $\mathbf{D}_s^{(l)}$ and using the Karush-Kuhn-Tucker (KKT) condition [63], we have the updating rule:

$$\mathbf{D}_s^{(l)} \leftarrow \mathbf{D}_s^{(l)} \odot \frac{\Theta_s^{(l-1)\top}\mathbf{X}_s\mathbf{H}_s^{(L)\top}\Omega_s^{(l+1)\top} + \alpha\mathbf{D}_t^{(l)}}{\Theta_s^{(l-1)\top}\Theta_s^{(l-1)}\mathbf{D}_s^{(l)}\Omega_s^{(l+1)}\mathbf{H}_s^{(L)}\mathbf{H}_s^{(L)\top}\Omega_s^{(l+1)\top} + \alpha\mathbf{D}_s^{(l)}}, \tag{11}$$

where $\odot$ denotes element-wise product, and $\Theta_s^{(l-1)} = \mathbf{D}_s^{(1)}\mathbf{D}_s^{(2)} \cdots \mathbf{D}_s^{(l-1)}$ and $\Omega_s^{(l+1)} = \mathbf{D}_s^{(l+1)}\mathbf{D}_s^{(l+2)} \cdots \mathbf{D}_s^{(L)}$.

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

$$\mathbf{D}_t^{(l)} \leftarrow \mathbf{D}_t^{(l)} \odot \frac{\Theta_t^{(l-1)\top}\mathbf{X}_t\mathbf{H}_t^{(L)\top}\Omega_t^{(l+1)\top} + \alpha\mathbf{D}_s^{(l)}}{\Theta_t^{(l-1)\top}\Theta_t^{(l-1)}\mathbf{D}_t^{(l)}\Omega_t^{(l+1)}\mathbf{H}_t^{(L)}\mathbf{H}_t^{(L)\top}\Omega_t^{(l+1)\top} + \alpha\mathbf{D}_t^{(l)}}. \tag{12}$$

$\mathbf{H}_s$-subproblem: With the other variables fixed, the optimization problem associated with $\mathbf{H}$ is formulated as

$$\begin{aligned}
&\min_{\mathbf{H}_s \geq 0} \|\mathbf{X}_s - \mathbf{D}_s^{(1)} \dots \mathbf{D}_s^{(l)}\mathbf{H}_s^{(l)}\|_F^2 + \gamma \sum_{l=1}^{L} tr(\mathbf{H}_s^{(l)}\mathbf{L}_s\mathbf{H}_s^{(l)\top}) \\
&\quad + \sum_{l=1}^{L} \Phi\left(\mathbf{\Lambda}_1^{(l)}, [\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{Z}^{(l)} - \mathbf{E}^{(l)}\right).
\end{aligned} \tag{13}$$

It can be observed that the updates of $\mathbf{H}_s^{(l)}(l = 1, \dots, L)$ in each level are independent. Thus, $\mathbf{H}_s^{(l)}$ can be updated one by one. For $\mathbf{H}_s^{(l)}$, we optimize the following problem:

$$\begin{aligned}
&\min_{\mathbf{H}_s^{(l)} \geq 0} \|\mathbf{X}_s - \Theta_s^{(l)}\mathbf{H}_s^{(l)}\|_F^2 + \gamma tr(\mathbf{H}_s^{(l)}\mathbf{L}_s\mathbf{H}_s^{(l)\top}) \\
&\quad + \frac{\mu}{2}\|[\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{Z}^{(l)} - \mathbf{E}^{(l)} + \mathbf{\Lambda}_1^{(l)}/\mu\|_F^2.
\end{aligned} \tag{14}$$

Denoting $\mathbf{E}^{(l)} = [\mathbf{E}_s^{(l)}, \mathbf{E}_t^{(l)}]$, $\mathbf{Z}^{(l)} = [\mathbf{Z}_s^{(l)}, \mathbf{Z}_t^{(l)}]$, and $\mathbf{\Lambda}_1^{(l)} = [\mathbf{\Lambda}_{1,s}^{(l)}, \mathbf{\Lambda}_{1,t}^{(l)}]$, it is equivalent to optimizing

$$\begin{aligned}
&\min_{\mathbf{H}_s^{(l)} \geq 0} \|\mathbf{X}_s - \Theta_s^{(l)}\mathbf{H}_s^{(l)}\|_F^2 + \gamma tr(\mathbf{H}_s^{(l)}\mathbf{L}_s\mathbf{H}_s^{(l)\top}) \\
&\quad + \frac{\mu}{2}\|\mathbf{H}_s^{(l)} - \mathbf{H}_s^{(l)}\mathbf{Z}_s^{(l)} - \mathbf{E}_s^{(l)} + \mathbf{\Lambda}_{1,s}^{(l)}/\mu\|_F^2.
\end{aligned} \tag{15}$$

By taking the derivative of Eq. (15) *w.r.t.* $\mathbf{H}_s^{(l)}$ and using the KKT condition [63], we have the updating rule:

$$\mathbf{H}_s^{(l)} \leftarrow \mathbf{H}_s^{(l)} \odot \frac{2\Theta_s^{(l)\top}\mathbf{X}_s + \mu(\mathbf{E}_s^{(l)} - \mathbf{\Lambda}_{1,s}^{(l)}/\mu)(\mathbf{I} - \mathbf{Z}_s^{(l)})^\top}{2\Theta_s^{(l)\top}\Theta_s^{(l)}\mathbf{H}_s^{(l)} + \mu\mathbf{H}_s^{(l)}(\mathbf{I} - \mathbf{Z}_s^{(l)})(\mathbf{I} - \mathbf{Z}_s^{(l)})^\top + 2\gamma\mathbf{H}_s\mathbf{L}_s}, \tag{16}$$

where $\mathbf{I}$ denotes an identity matrix.

$\mathbf{H}_t$-subproblem: To update $\mathbf{H}_t$, the following objective function should be optimized:

$$\begin{aligned}
&\min_{\mathbf{H}_t \geq 0} \|\mathbf{X}_t - \mathbf{D}_t^{(1)} \dots \mathbf{D}_t^{(l)}\mathbf{H}_t^{(l)}\|_F^2 \\
&\quad + \sum_{l=1}^{L} \Phi\left(\mathbf{\Lambda}_{1,t}^{(l)}, \mathbf{H}_t^{(l)} - \mathbf{H}_s^{(l)}\mathbf{Z}_t^{(l)} - \mathbf{E}_t^{(l)}\right),
\end{aligned} \tag{17}$$

We also update $\mathbf{H}_t^{(l)}(l = 1, \dots, L)$ one by one. For $\mathbf{H}_t^{(l)}$, it is equivalent to optimizing the following problem:

$$\begin{aligned}
&\min_{\mathbf{H}_t^{(l)} \geq 0} \|\mathbf{X}_t - \Theta_t^{(l)}\mathbf{H}_t^{(l)}\|_F^2 \\
&\quad + \frac{\mu}{2}\|\mathbf{H}_t^{(l)} - \mathbf{H}_s^{(l)}\mathbf{Z}_t^{(l)} - \mathbf{E}_t^{(l)} + \mathbf{\Lambda}_{1,t}^{(l)}/\mu\|_F^2.
\end{aligned} \tag{18}$$

By taking the derivative of Eq. (18) *w.r.t.* $\mathbf{H}_t^{(l)}$ and using the KKT condition [63], the updating rule is given as:

$$\mathbf{H}_t^{(l)} \leftarrow \mathbf{H}_t^{(l)} \odot \frac{2\Theta_t^{(l)^\top}\mathbf{X}_t + \mu(\mathbf{H}_s^{(l)}\mathbf{Z}_t^{(l)} + \mathbf{E}_t^{(l)} - \frac{\mathbf{\Lambda}_{1,t}^{(l)}}{\mu})}{2\Theta_t^{(l)^\top}\Theta_t^{(l)}\mathbf{H}_t^{(l)} + \mu\mathbf{H}_t^{(l)}}. \tag{19}$$

J-subproblem: Dropping the unrelated terms, the optimization for $\mathbf{Z}^{(l)}$ yields

$$\begin{aligned}\min_{\mathbf{J}^{(l)}} \Phi\left(\mathbf{\Lambda}_1^{(l)}, [\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{J}^{(l)} - \mathbf{E}^{(l)}\right) \\ + \Phi\left(\mathbf{\Lambda}_2^{(l)}, \mathbf{Z}^{(l)} - \mathbf{J}^{(l)}\right).\end{aligned} \tag{20}$$

This is equivalent to optimizing the following problem:

$$\begin{aligned}\min_{\mathbf{J}^{(l)}} \|[\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \frac{\mathbf{\Lambda}_1^{(l)}}{\mu} - \mathbf{E}^{(l)} + \mathbf{H}_s^{(l)}\mathbf{J}^{(l)}\|_F^2 \\ + \|\mathbf{J}^{(l)} - \mathbf{Z}^{(l)} - \mathbf{\Lambda}_2^{(l)}/\mu\|_F^2.\end{aligned} \tag{21}$$

By taking the derivative of (21) *w.r.t* $\mathbf{J}^{(l)}$ and setting it to zero, its closed-form solution is given as follows:

$$\begin{aligned}\mathbf{J}^{(l)} = \left(\mathbf{H}_s^{(l)^\top}\mathbf{H}_s^{(l)} + \mathbf{I}\right)^{-1} \\ \mathbf{H}_s^{(l)^\top}\left([\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] + \frac{\mathbf{\Lambda}_1^{(l)}}{\mu} - \mathbf{E}^{(l)}\right) + \left(\mathbf{Z}^{(l)} + \frac{\mathbf{\Lambda}_2^{(l)}}{\mu}\right).\end{aligned} \tag{22}$$

Z-subproblem: By dropping the unrelated terms, we can optimize the following problem for updating $\mathbf{Z}^{(l)}$:

$$\begin{aligned}\min_{\mathbf{Z}^{(l)}} \beta\sum_{m\neq l}^{L}\text{HSIC}\left(\mathbf{Z}^{(l)}, \mathbf{Z}^{(m)}\right) + \gamma\left(tr(\mathbf{J}^{(l)}\mathbf{L}\mathbf{Z}^{(l)^\top}\right) \\ + \Phi(\mathbf{\Lambda}_2^{(l)}, \mathbf{Z}^{(l)} - \mathbf{J}^{(l)}), \;\; s.t.\; \mathbf{Z}^{(l)} \geq 0, \mathbf{1}^\top\mathbf{Z}^{(l)} = \mathbf{1}^\top.\end{aligned} \tag{23}$$

In this study, we adopt the inner product kernel for the HSIC constraint, *i.e.*, $\mathbf{K}_l = \mathbf{Z}^{(l)^\top}\mathbf{Z}^{(l)}$. Then, we have $\text{HSIC}(\mathbf{Z}^{(l)}, \mathbf{Z}^{(m)}) = tr(\mathbf{Z}^{(l)}\mathbf{K}\mathbf{Z}^{(l)^\top})$ with $\mathbf{K} = \sum_{m\neq l}^{L}\mathbf{M}\mathbf{K}_m\mathbf{M}$, where $m_{ij} = \delta_{ij} - 1/n$. Finally, by taking the derivative of (23) *w.r.t* $\mathbf{Z}^{(l)}$, we can obtain

$$\begin{aligned}\frac{\partial\mathcal{J}(\mathbf{Z}^{(l)})}{\partial\mathbf{Z}^{(l)}} = \beta\mathbf{Z}^{(l)}\mathbf{K} + \gamma\mathbf{Z}^{(l)}\mathbf{L} + \frac{\mu}{2}\mathbf{H}_s^{(l)^\top}\mathbf{H}_s^{(l)}\mathbf{Z}^{(l)} \\ - \mathbf{H}_s^{(l)^\top}\left([\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] + \frac{\mathbf{\Lambda}_1^{(l)}}{\mu} - \mathbf{E}^{(l)}\right).\end{aligned} \tag{24}$$

By setting $\frac{\partial\mathcal{J}(\mathbf{Z}^{(l)})}{\partial\mathbf{Z}^{(l)}}$ to zero, we have its closed-form solution. After that, an iterative algorithm [64] is used to obtain the optimal solution of $\mathbf{Z}^{(l)}$.

E-subproblem: Updating the error term $\mathbf{E}^{(l)}$ is equivalent to solving the following problem:

$$\min_{\mathbf{E}^{(l)}} \frac{1}{\mu}\|\mathbf{E}^{(l)}\|_{2,1} + \frac{1}{2}\|\mathbf{E}^{(l)} - \mathbf{G}\|_F^2, \tag{25}$$

where $\mathbf{G} = [\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{Z}^{(l)} + \mathbf{\Lambda}_1^{(l)}/\mu$. This optimization problem is solved using the algorithm in [65].

Multipliers updating: The multipliers $\mathbf{\Lambda}_1^{(l)}$ and $\mathbf{\Lambda}_2^{(l)}$ ($l = 1, 2, \ldots, L$) can be updated by

$$\begin{cases}\mathbf{\Lambda}_1^{(l)} := \mathbf{\Lambda}_1^{(l)} + \mu([\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{Z}^{(l)} - \mathbf{E}^{(l)}), \\ \mathbf{\Lambda}_2^{(l)} := \mathbf{\Lambda}_2^{(l)} + \mu(\mathbf{Z}^{(l)} - \mathbf{J}^{(l)}).\end{cases} \tag{26}$$

Initialization and implementation: Following previous deep NMF-based works [56], [57], we first pretrain a deep NMF model to obtain initial approximations for $\mathbf{D}_s^{(l)}$, $\mathbf{D}_t^{(l)}$, $\mathbf{H}_s^{(l)}$, and $\mathbf{H}_t^{(l)}$ ($l = 1, 2, \ldots, L$) in each layer. This pretraining process often reduces the training time of our model, and its effectiveness has also been proven in deep auto-encoder networks [66]. Let us consider the source data as an example. We decompose $\mathbf{X}_s \approx \mathbf{D}_s^{(1)}\mathbf{H}_s^{(1)}$ and further decompose $\mathbf{H}_s^{(1)} \approx \mathbf{D}_s^{(2)}\mathbf{H}_s^{(2)}$, until all layers are initialized. Then, we carry out the optimization steps for all variables and repeat this until convergence. The detailed steps for optimizing the proposed framework in Eq. (8) via the ADMM algorithm are summarized in Algorithm 1.

Algorithm 1: Optimizing the problem (8) via ADMM.

Input: Source data: $\mathbf{X}_s$ and target data $\mathbf{X}_t$, parameters α, λ, β, and γ.
Initialize: $\mathbf{\Lambda}_1^{(l)} = 0$, $\mathbf{\Lambda}_2^{(l)} = 0$, $\varepsilon = 10^{-4}$, $\rho = 1.5$, $\mu = 10^{-4}$, $max_\mu = 10^6$.
Output: $\mathbf{Z}^{(l)}, l = 1, 2, \ldots, L$.
while *not converged* **do**
  for *l=1,2,...,L* **do**
    Update $\mathbf{D}_s^{(l)}$, $\mathbf{D}_t^{(l)}$, $\mathbf{H}_s^{(l)}$, $\mathbf{H}_t^{(l)}$, $\mathbf{J}^{(l)}$, $\mathbf{Z}^{(l)}$, $\mathbf{E}^{(l)}$, $\mathbf{\Lambda}_1^{(l)}$, and $\mathbf{\Lambda}_2^{(l)}$ using Eqs. (11), (12), (16), (19), (22), (25), and (26), respectively.
  end
  Update the parameter μ via $\mu = \min(\rho\mu, max_\mu)$;
  Check the convergence conditions:
  $\|[\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{Z}^{(l)} - \mathbf{E}^{(l)}\|_\infty < \varepsilon$
  and $\|\mathbf{Z}^{(l)} - \mathbf{J}^{(l)}\|_\infty < \varepsilon$.
end

3.4 Clustering-based Motion Segmentation

With Algorithm 1, we can obtain the learned multi-level representations $\mathbf{Z}^{(l)}(l = 1, 2, \ldots, L)$. Then, we extract the corresponding target representations $\mathbf{Z}_t^{(l)} \in \mathbb{R}^{n_s\times n_t}$ from $\mathbf{Z}^{(l)} = [\mathbf{Z}_s^{(l)}, \mathbf{Z}_t^{(l)}]$. Inspired by [8], we develop a new similarity measurement to construct an affinity matrix $\mathbf{A}$ for our multi-level transfer subspace learning, which can explore intrinsic relationships among within-cluster samples from human motion data. Specifically, each element of $\mathbf{A}$ can be defined as the distance between a pair of the learned representation coefficients for target data, which is given by

$$a_{ij} = \frac{1}{L}\sum\nolimits_{l=1}^{L}\frac{\mathbf{z}_{t,i}^{(l)^\top}\mathbf{z}_{t,j}^{(l)}}{\|\mathbf{z}_{t,i}^{(l)}\|_2\|\mathbf{z}_{t,j}^{(l)}\|_2}, \tag{27}$$

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. Note that we compute each element a_{ij} by averaging the pair distance among multi-level feature spaces. After that, the Normalized Cuts [67] algorithm is applied to the learned affinity matrix $\mathbf{A}$ to produce the clustering results.

3.5 Complexity Analysis

The overall optimization procedure consists of two stages, *i.e.*, pretraining and model updating, thus we analyze the computational burden of the two parts separately. For clarity, we define p as the maximal layer size in all layers and n ($n = n_s + n_t$) as the total number of the source and target data. For the pretraining part, the computational complexity is of order $\mathcal{O}(Lt_p(n_s^2p + n_sp^2 + n_t^2p + n_tp^2))$, where t_p is the number of iterations. For the model updating part, the

computational cost lies in updating $\mathbf{D}_s^{(l)}$, $\mathbf{D}_t^{(l)}$, $\mathbf{H}_s^{(l)}$, $\mathbf{H}_t^{(l)}$, $\mathbf{J}^{(l)}$, $\mathbf{Z}^{(l)}$, and $\mathbf{E}^{(l)}$. Thus the computational complexity is of order $\mathcal{O}(Lt_u(n_s^2p + n_sp^2 + n_t^2p + n_tp^2 + p^3 + n^3 + n_s^2n + n_sn_tp))$, where t_u is the number of iterations in this stage. Finally, considering $n_s, n_t > p$ in our task, the overall computational cost is $\mathcal{O}(L((t_p+t_u)(n_s^2p+n_sp^2+n_t^2p+n_tp^2)+t_u(n^3+n_sn_tp)))$.

4 Experimental Results and Analysis

In this section, we first introduce the human motion datasets used in our experiments (§ 4.1), and provide the details of the experimental setup (§ 4.2). Then we show the comparison results with several state-of-the-art methods (§ 4.3) and conduct the model study (§ 4.4).

4.1 Human Motion Datasets

To comprehensively evaluate the proposed model, we conduct experiments on five benchmark human motion datasets (see Fig. 4 for some example frames). • **Keck Gesture Dataset (Keck)** [68] consists of 14 different actions from military signals, in which each subject is carried out 14 actions and gestures. Besides, the videos in this dataset were obtained by a fixed camera when these subjects stand out in a static background. • **Multi-Modal Action Detection Dataset (MAD)** [69] consists of actions captured from various modalities using a Microsoft Kinect V2 system, which includes RGB images, depth cues, and skeleton formats. Specifically, the RGB images and 3D depth cues are of a size of 240×320. Moreover, each subject performs 35 different actions within two indoor scenes. • **Weizmann Dataset (Weiz)** [70] is composed of 90 video sequences with 10 actions (running, walking, skipping, bending, etc.) captured by nine subjects in an outdoor environment. All videos have a size of 180×144 with 50 fps. • **UT-Interaction Dataset (UT)** [71] is composed of 20 videos, each of which includes six different action types of human-human interactions (such as punching, pushing, pointing, hugging, kicking, and handshaking). • **Youtube Dataset** [72] consists of 11 action categories, which are diving, biking/cycling, horse back riding, golf swinging, soccer juggling, swinging, walking, trampoline jumping, volleyball spiking, tennis swinging, and basketball shooting. In our experiment, we construct four sequences, each of which includes the first 10 actions.

4.2 Experimental Setup

4.2.1 Dataset Settings

Following the dataset preprocessing and settings in [27], in our experiments, we use the extracted 324-dimensional HOG features [73] for each frame. Due to the fact that different datasets consist of different numbers of actions, we randomly choose ten motions from the Keck, MAD, and Weiz datasets in our experiments. In addition, we extract deep CNN features from the Keck, Weiz, and Youtube datasets using the pretrained VGG-16 [74] model and obtain a 1000-dimension feature vector for each frame. For model evaluation, we select one dataset as the source and the remaining datasets as the target, and we obtain the mean results of motion segmentation on the target dataset.

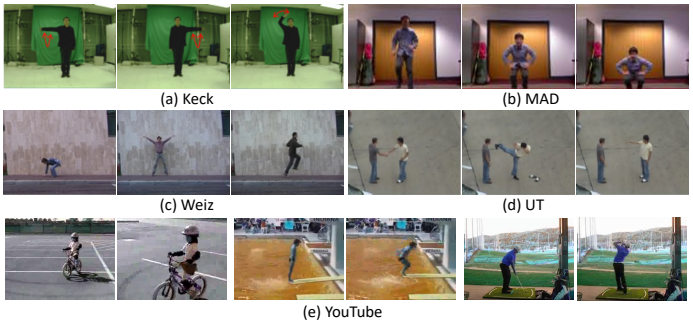

Figure 4: Sampling frames from five human motion benchmark datasets, *i.e.*, (a) Keck, (b) MAD, (c) Weiz, (d) UT, and (e) YouTube.

4.2.2 Compared Methods

We compare the proposed model with several state-of-the-art approaches, which include (1) Spectral clustering (**SC**) [12]. The feature vectors of target samples are directly fed into the standard spectral clustering algorithm [12] to obtain clustering results. (2) K-medoids (**KMD**) [75] selects target samples as centers and partitions them into different groups using the Manhattan Norm to measure the distance between data points. (3) Low-rank representation (**LRR**) [21] conducts subspace clustering by imposing a low-rank constraint on the target data. (4) Ordered subspace clustering (**OSC**) [76] utilizes a temporal correlation constraint to force representations of the temporal data to be closer. (5) Sparse subspace clustering (**SSC**) [20] assumes that there exists a dictionary that can be used to linearly represent all data points, and a sparse constraint is imposed on the representation coefficients. (6) Least square regression (**LSR**) [23] introduces the ℓ_2 regularization on the learned coefficient matrix, in which the grouping effect of LSR enables to group mostly correlated data together. (7) Temporal subspace clustering (**TSC**) [8] employs a non-negative dictionary learning algorithm to obtain an expressive coefficient matrix for temporal clustering, and a temporal Laplacian regularization is imposed on the coefficient matrix. (8) Transfer subspace segmentation (**TSS**) [26] is a transferable subspace clustering method that explores useful information from relevant source data to boost clustering performance on target human motion data. (9) Low-rank transfer subspace (**LTS**) [27] employs a graph regularizer to capture temporal correlation in both the source and target data as well as uncovers clustering structures within data by introducing a weighted low-rank constraint. For convenience, we denote the preliminary of our method, proposed in [1], as MCSTL.

4.2.3 Evaluation Metrics and Parameter Settings

To evaluate the model performance, we utilize two popular metrics, including *i.e.*, normalized mutual information (NMI) and accuracy (ACC). Specifically, NMI evaluates the mutual information between the ground truth and the recovered cluster labels, and ACC computes the classification score with the best map. Note that, higher values for both metrics indicate better performance. In addition, for our approach, we tune three trade-off parameters α, β, and γ in the range of $\{10^{-5}, 10^{-4}, \ldots, 10^{2}\}$. Furthermore, the number of layers for our CDMS model is set as 3, and the

Table 2: Clustering results of compared methods 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 first two best results are highlighted in **bold** and underlined when using the same source data.

(a) Results on Keck dataset

Method	NMI ↑	ACC ↑
SC [12]	0.4744	0.3886
KMD [75]	0.4702	0.3970
LRR [21]	0.4862	0.4297
OSC [76]	0.5931	0.4393
SSC [20]	0.3858	0.3137
LSR [23]	0.4548	0.4894
TSC(M) [8]	0.6935	0.4653
TSS(M) [26]	0.8049	0.5395
LTS(M) [27]	0.8226	0.5509
MCSTL(M) [1]	**0.8270**	0.6010
CDMS(M)	0.7891	**0.6044**
TSC(W) [8]	0.6862	0.4548
TSS(W) [26]	0.7928	0.5485
LTS(W) [27]	0.7983	0.5649
MCSTL(W) [1]	0.8196	0.5915
CDMS(W)	**0.8213**	**0.6085**
TSC (U) [8]	0.6797	0.4421
TSS(U) [26]	0.7937	0.4951
LTS(U) [27]	0.7947	0.5519
MCSTL(U) [1]	**0.8120**	0.6105
CDMS(U)	0.8040	**0.6207**

(b) Results on MAD dataset

Method	NMI ↑	ACC ↑
SC [12]	0.4369	0.3639
KMD [75]	0.3914	0.3226
LRR [21]	0.2249	0.2397
OSC [76]	0.5589	0.4327
SSC [20]	0.4758	0.3817
LSR [23]	0.3667	0.3979
TSC(K) [8]	0.7691	0.5473
TSS(K) [26]	**0.8286**	0.5792
LTS(K) [27]	0.8244	0.5874
MCSTL(K) [1]	0.8099	0.6125
CDMS(K)	0.8251	**0.6536**
TSC(W) [8]	0.7684	0.5418
TSS(W) [26]	0.8202	0.5736
LTS(W) [27]	0.8213	0.5906
MCSTL(W) [1]	**0.8307**	0.6158
CDMS(W)	0.8238	**0.6392**
TSC (U) [8]	0.7691	0.5315
TSS(U) [26]	0.8108	0.5479
LTS(U) [27]	0.8211	0.5980
MCSTL(U) [1]	**0.8314**	0.6163
CDMS(U)	0.8238	**0.6371**

(c) Results on Weizman dataset

Method	NMI ↑	ACC ↑
SC [12]	0.5435	0.4127
KMD [75]	0.5289	0.4441
LRR [21]	0.4382	0.3638
OSC [76]	0.7047	0.5216
SSC [20]	0.6009	0.4576
LSR [23]	0.5093	0.5091
TSC(K) [8]	0.7971	0.5931
TSS(K) [26]	0.8326	0.6030
LTS(K) [27]	0.8599	0.6391
MCSTL(K) [1]	0.8371	0.6436
CDMS(K)	**0.8601**	**0.6465**
TSC(M) [8]	0.8032	0.5961
TSS(M) [26]	0.8509	0.6208
LTS(M) [27]	**0.8579**	0.6156
MCSTL(M) [1]	0.8232	0.6348
CDMS(M)	0.8375	**0.6505**
TSC (U) [8]	0.7796	0.5402
TSS(U) [26]	0.8124	0.5865
LTS(U) [27]	0.8267	0.6122
MCSTL(U) [1]	0.8351	**0.6371**
CDMS(U)	**0.8616**	0.6266

(d) Results on UT dataset

Method	NMI ↑	ACC ↑
SC [12]	0.4894	0.4477
KMD [75]	0.5108	0.5122
LRR [21]	0.4051	0.4162
OSC [76]	0.6877	0.5846
SSC [20]	0.4998	0.4389
LSR [23]	0.4322	0.5183
TSC(K) [8]	0.7216	0.5213
TSS(K) [26]	0.7746	0.5371
LTS(K) [27]	0.7961	0.6127
MCSTL(K) [1]	0.8121	0.6148
CDMS(K)	**0.8267**	**0.6547**
TSC(M) [8]	0.7442	0.5288
TSS(M) [26]	0.7783	0.5335
LTS(M) [27]	0.8128	0.6299
MCSTL(M) [1]	0.8239	0.6433
CDMS(M)	**0.8306**	**0.6643**
TSC (W) [8]	0.7136	0.5111
TSS(W) [26]	0.7878	0.5944
LTS(W) [27]	0.8035	0.6296
MCSTL(W) [1]	0.8198	0.6463
CDMS(W)	**0.8288**	**0.6642**

number of sequential neighbors is tuned from the set of $\{9, 11, \ldots, 21\}$.

4.3 Performance Comparison

In all comparisons, 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 models, both source and target videos are fed for segmentation. The clustering results of different methods using HOG features are shown in Table 2, where **bold** and underlined denote the best two results.

From the results in Table 2, we make the following observations: (1) Compared with SC, KMD, LRR, OSC, SSC, and LRR, our method transfers more useful information from the source data to learn distinctive representations of the target data, resulting in improved segmentation performance. (2) Compared with transfer clustering-based segmentation methods (including TSC, TSS, and TSS), our method also obtains much better performance. This is because our approach explores domain-invariant features and preserves domain-specific knowledge simultaneously. These two aspects are equally important for transfer learning. More importantly, our model fuses multi-level representations to construct the affinity matrix for motion segmentation, which effectively exploits hierarchical semantics and structural information in a layer-wise fashion. Thus, these results validate the effectiveness of the proposed method against other state-of-the-art models in human motion segmentation.

In Fig. 5, we visualize the clustering results obtained by our model and other state-of-the-art methods on a sample video of the Weiz dataset. Different colors denote different

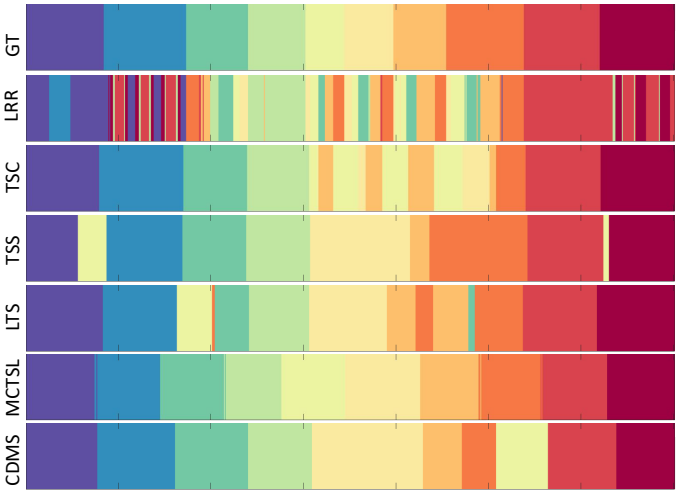

Figure 5: Visualization of clustering results on a sample video of the Weiz dataset. The ten colors denote ten different temporal clusters.

clusters in the action data. From the results, it can be observed that the LRR and SSC models generate multiple fragments and cannot achieve meaningful temporal segmentation. The main reason is likely that they do not consider the temporal correlations when conducting subspace clustering. Compared with the LRR and SSC methods, TSC obtains a relatively better segmentation but it still suffers from many unexpected fragments. LTS and TSS obtain much better performance in most cases; however, they occasionally generate some unexpected fragments. Overall, our methods (CDMS and MCTSL) achieve continuous and meaningful segments, providing much better segmentation results than other comparison methods.

Performance on Deep Features. As deep models have demonstrated promising performance in several computer

Table 3: Clustering results of compared methods in terms of NMI and ACC on three human motion datasets. Names in brackets denote the source datasets, where K, W, and Y denote Keck, Weizmann, and Youtube datasets, respectively. The first two best clustering results are highlighted in **bold** and underlined when using the same source data.

(a) Results on Keck dataset

Method	NMI ↑	ACC ↑
TSS(W) [26]	0.8595	0.5675
LTS(W) [27]	0.7906	0.5730
MCSTL(W) [1]	**0.8675**	**0.6423**
CDMS(W)	0.8663	0.5922
TSS(Y) [26]	0.8050	0.5471
LTS(Y) [27]	0.8167	0.5690
MCSTL(Y) [1]	0.8235	0.5875
CDMS(Y)	**0.8333**	**0.5940**

(b) Results on Weiz dataset

Method	NMI ↑	ACC ↑
TSS(K) [26]	0.8189	0.5913
LTS(K) [27]	0.8477	0.6346
MCSTL(K) [1]	0.7803	0.6175
CDMS(K)	**0.8539**	**0.6574**
TSS(Y) [26]	0.8194	0.5971
LTS(Y) [27]	0.8018	0.5906
MCSTL(Y) [1]	0.8111	0.6066
CDMS(Y)	**0.8579**	**0.6373**

(c) Results on Youtube dataset

Method	NMI ↑	ACC ↑
TSS(K) [26]	0.8444	0.6064
LTS(K) [27]	0.8898	0.6226
MCSTL(K) [1]	0.8141	0.6440
CDMS(K)	**0.9133**	**0.6798**
TSS(W) [26]	0.8820	0.6294
LTS(W) [27]	0.8695	0.6169
MCSTL(W) [1]	0.8839	0.6314
CDMS(W)	**0.8863**	**0.6420**

vision tasks, we further evaluate the performance of the proposed model using deep features as inputs. In this study, deep features are extracted using the pretrained VGG-16 [74] model for each frame in the source and target datasets, and then we test the performance using the same settings as the previous experiments. We compare the proposed models (MCSTL and CDMS) with two state-of-the-art subspace transfer methods (*i.e.*, TSS and LTS), and the comparison results are shown in Table 3. From this experiment, we have the following observations: (1) our MSTL model performs better than other comparison methods in most cases; (2) the models achieve similar or slightly better performance using deep features rather than HOG features. Moreover, it can be noted that there is a difference in computation cost between using the two types of features. When using deep features, it needs a large-scale dataset to train a pre-trained model, and then we can extract deep features as the inputs. The dimension of deep features is often high (*e.g.*, 1000 in this study), which requires many computation resources. While using hand-crafted (*e.g.*, HOG) features, it does not need extra training and requires fewer computation resources to process low-dimension features. Regardless of the input, our model is a general approach that can take hand-crafted features extracted from raw data or deep features.

4.4 Model Study

4.4.1 Parameter Sensitivity

In our model, τ denotes the number of sequential neighbors in the temporal correlation preservation term. To investigate the effect of this parameter, we test our model on the MAD dataset with Keck as the source for different values of τ. Fig. 6 shows the clustering performance for various values within $\{3, 5, \ldots, 21\}$. From the results, it can be observed that our model obtains relatively better performance when $\tau \in [15, 21]$. In addition, three key regularization parameters, *i.e.*, α, β and γ, 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. The experimental results on the MAD dataset are shown in Fig. 7 (a)(b)(c). From the results, it can be seen that our proposed method obtains much better NMI performance when $\alpha \in [0.001, 10]$, $\beta \in [1, 100]$, and $\gamma \in [10, 100]$. Moreover, the experimental results also indicate that each term in our framework is useful for boosting the segmentation results.

To validate the influence of three parameters for deep features, we conduct an experiment for parameter sensitivity study on MAD using Keck as the source. As shown in Fig. 8, our method obtains better performance when $\alpha \in [0.001, 1]$, $\beta \in [0.001, 10]$, and $\gamma \in [10, 100]$. Overall, three regularization parameters can be tuned to obtain good performance with different settings. Besides, we can follow the suggestions when applying our model to new datasets.

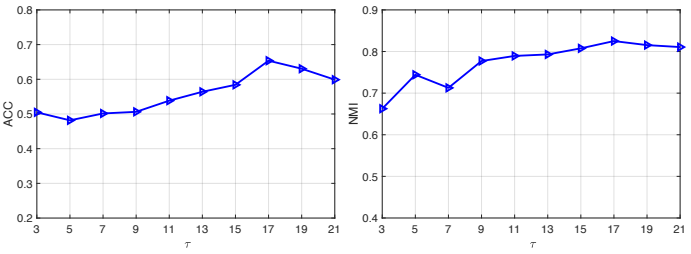

Figure 6: The segmentation performance in terms of NMI and ACC when using different numbers of sequential neighbors τ.

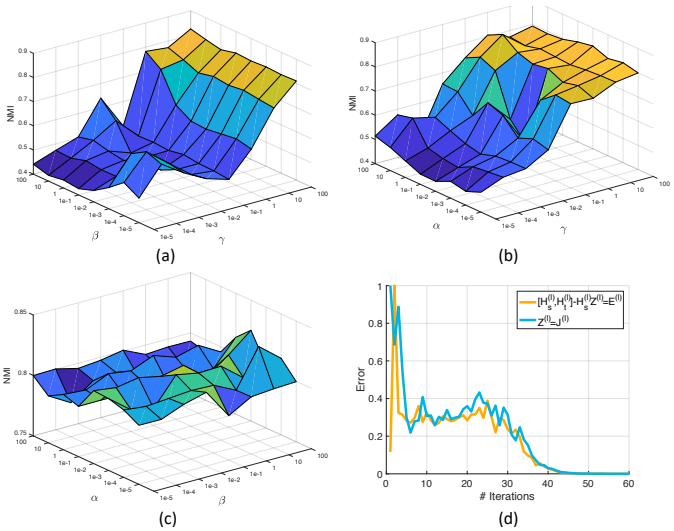

Figure 7: Parameter sensitivity study and convergence analysis on the Keck dataset using HOG features: (a) Sensitivity analysis for parameters β and γ, (b) Sensitivity analysis for parameters α and γ, (c) Sensitivity analysis for parameters α and β, and (d) Convergence curves.

4.4.2 Convergence Analysis

We compute the errors (*i.e.*, $\|[\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)}] - \mathbf{H}_s^{(l)}\mathbf{Z}^{(l)} - \mathbf{E}^{(l)}\|_\infty$ and $\|\mathbf{W}^{(l)} - \mathbf{J}^{(l)}\|_\infty$) to demonstrate the convergence of our optimization algorithm. We report the mean values of different layers, and the convergence curves on the MAD

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3147841, IEEE Transactions on Pattern Analysis and Machine Intelligence

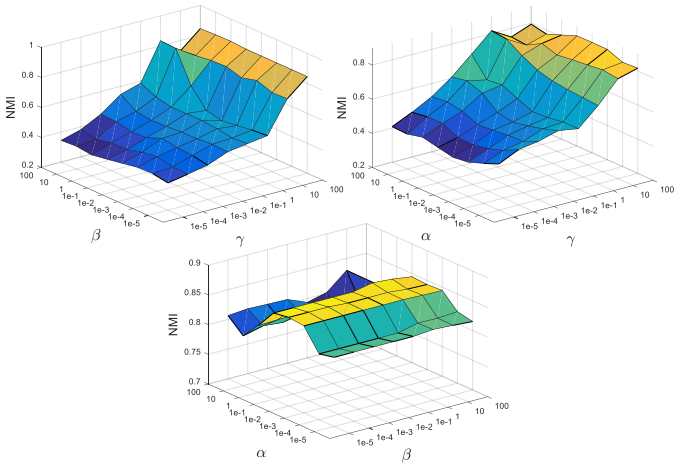

Figure 8: Parameter sensitivity study on MAD: (a) Sensitivity analysis for parameters β and γ, (b) Sensitivity analysis for parameters α and γ, and (c) Sensitivity analysis for parameters α and β.

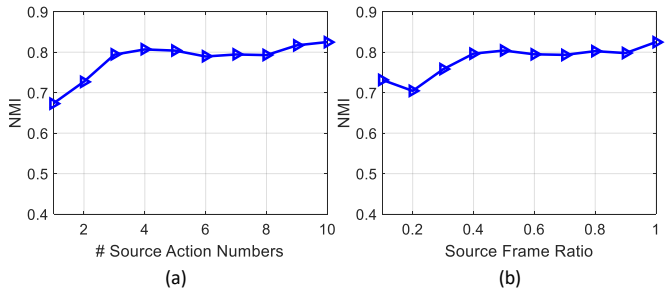

Figure 9: Segmentation results based on (a) using different action numbers and (b) using different frame ratios.

dataset (Keck as source) are presented in Fig. 7 (d). Note that, for better presentation, the errors are normalized into the range $[0, 1]$. As can be observed, our model converges within about 50 iterations.

4.4.3 Source Data Analysis

To evaluate the effectiveness of the source information for boosting the segmentation performance, we carry out an experiment on the MAD dataset (with Keck as the source) in which different numbers of actions are used. The results are shown in Fig. 9 (a). As can be observed, the performance increases in general when the number of actions increases. This suggests that diverse source data is helpful for boosting the segmentation performance in the target domain. The main reason could be that more actions in the source data can transfer more useful knowledge to ensure that our model learns the distinctive representations of the target data. In addition, we also test the effect of using different numbers of each action. To achieve this, we utilize the frames with different ratios (*i.e.*, $0.1, 0.2, \cdots, 1$) in each source action, while keeping the total number of actions fixed. We evaluate the performance on the MAD dataset, and the comparison results are shown in Fig. 9 (b). From the results, it can be observed that the performance of our model increases when the ratio of frames increases. This demonstrates that using more frames of each action in the source data can transfer useful knowledge to learn distinctive and effective representations of the target data. Thus, the effectiveness of the source data in improving the segmentation performance can be well validated.

It can be also noted that the results of our method with different source datasets are similar. If it needs to select a good source dataset to further improve the segmentation performance on the target data, we suggest selecting one source dataset that has a similar scenario to the target dataset. For example, we can obtain the best performance on MAD by using Keck as the source, since all these two datasets are under an indoor scenario.

Moreover, we have conducted an experiment by using two source datasets, and the comparison results are shown in Table 4. From the results, comparing using Weiz and Weiz & Keck as the sources, our method obtains similar performance in the two settings. Comparing using UT and UT & Weiz as the sources, our method gets a slightly better performance when using two datasets as sources than using single. Overall, our model can obtain similar and slightly better performance via using two datasets as the source than a single source, but the computation cost will increase when using more datasets. Thus, to balance the performance and computation cost, we just use one dataset as the source in this study. Additionally, our model is able to flexibly explore the complementarity among multi-level feature spaces, which can also be extended to other multi-modal learning tasks. In future work, we believe that the data size can be reduced by performing sampling techniques or replacing the original data set with a small number of points [77], [78], and binary representations [79] can also be considered to accelerate the computation speed. Moreover, another potential strategy is finding the most diverse and representative frames based on clustering methods, and then we can only use these frames as the source data for reducing the complexity.

Table 4: Comparisons on MAD when using one and two datasets as the source.

	ACC	NMI
Weiz (source)	0.6392	0.8238
Weiz + Keck (source)	0.6345	0.8325
UT (source)	0.6371	0.8238
UT + Weiz (source)	0.6440	0.8289

4.4.4 Dictionary Dimensionality Analysis

In the proposed model, we adopt a multi-layer deep matrix factorization structure, thus we need to set different dimensions for dictionary atoms at each layer. To investigate the effects of dimensionality of the dictionary atoms, we conduct a comparison experiment using different dimension sets. Specifically, we set ten cases in the dimension set, *i.e.*, {128-32; 128-16; 64-32; 64-16; 32-16; 128-64-16; 128-32-16; 64-32-16; 128-64-32-16; 128-64-32-16-8}. Fig. 10 shows the segmentation results obtained based on using different numbers of layers and dimensionalities of the dictionary atoms on MAD using Keck as the source data. From the results, the ACC results vary when using different dimensionalities of the dictionary atoms, while the NMI results are robust to dimensionalities of the dictionary atoms. Thus, in

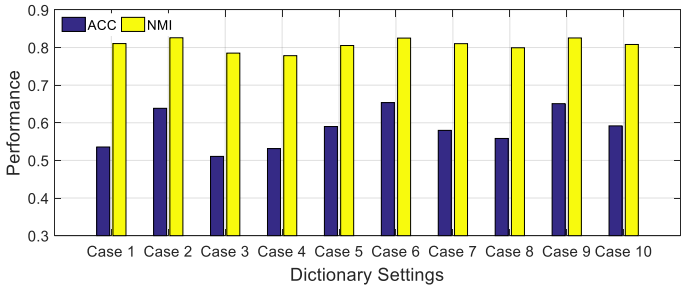

Figure 10: Segmentation results on MAD (Keck as source) via using different numbers of layers and dimensionalities of dictionary atoms.

this study, we adopt a three-layer deep NMF structure (*i.e.*, 128-64-16) for all comparisons as the default setting.

4.4.5 Ablation Study

Effectiveness of Multi-level Representations: To validate the effectiveness of fusing multi-level subspace representations from different feature spaces, we conduct an experiment to test the performance of our model when using the representations from the first layer, last layer, and multiple fused layers on the Weiz dataset. The results are shown in Fig. 11. We can see that our fusion strategy obtains much better performance than conducting subspace learning only on the representations from the first layer or last layer. This indicates the effectiveness of our model, which fuses the multi-level subspace representations for transfer learning.

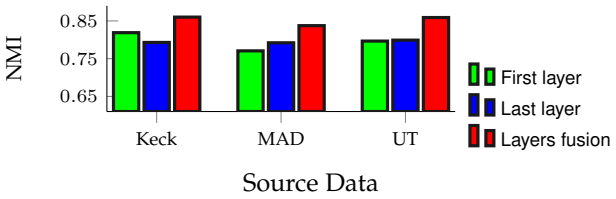

Figure 11: Performance comparison (NMI) when using representations from different layers or multi-layer fusion.

Effectiveness of Different Key Components: To validate the effectiveness of different key components (*i.e.*, the temporal correlation preservation term, diversity across multi-level representation term, and multi-mutual consistency learning term), we test our model when if moving one component and keeping the other two, on the Weiz dataset. Fig. 12 shows the comparison results of different key components. From these results, it can be observed that the temporal correlation preservation term plays an important role in human motion segmentation, which effectively preserves temporal correlations among consecutive frames. Moreover, we can see that the diversity constraint and consistency learning are also helpful for boosting the segmentation performance. Thus, these three key components work together to improve the model performance.

5 Conclusion

We have proposed a novel CDMS 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 carry out transfer subspace learning on different layers to capture multi-level structural information. A multi-mutual consistency learning strategy is carried out to reduce the distribution gap between the source and target domains. Further, we introduce a novel constraint term based on the HSIC to strengthen the diversity of multi-level subspace representations, which enables the complementarity of multi-level representations to be explored in order to boost the transfer learning performance. Moreover, we develop an enhanced graph regularizer term to preserve the temporal correlations. Finally, we obtain the optimized affinity matrix by fusing the multi-level subspace representation coefficients. Extensive experimental results on benchmark datasets demonstrate the effectiveness of the proposed model against several state-of-the-art human motion segmentation methods.

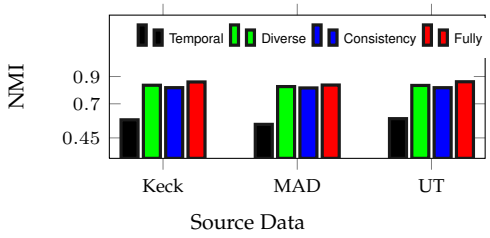

Figure 12: Results comparison of different key components.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Nos: 62172228, 61973162), Natural Science Foundation of Jiangsu Province (No: BZ2021013), the Fundamental Research Funds for the Central Universities (Nos: 30920032202, 30921013114), and National Key R&D Program of China (No: 2021YFA1001100).

References

[1] T. Zhou, H. Fu, C. Gong, J. Shen, L. Shao, and F. Porikli, "Multi-mutual consistency induced transfer subspace learning for human motion segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2020.

[2] R. Poppe, "Vision-based human motion analysis: An overview," *Comput. Vis. Image Underst.*, vol. 108, no. 1-2, pp. 4–18, 2007.

[3] 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, 2012.

[4] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 140–153, 2018.

[5] Jonathan Feng-Shun Lin, Michelle Karg, and Dana Kulić, "Movement primitive segmentation for human motion modeling: A framework for analysis," *IEEE Trans. Man-Mach. Syst.*, vol. 46, no. 3, pp. 325–339, 2016.

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

[7] Y. Xiong and D.-Y. Yeung, "Mixtures of arma models for model-based time series clustering," in *Proc. IEEE Int. Conf. Data Min.*. IEEE, pp. 717–720, 2002.

[8] S. Li, K. Li, and Y. Fu, "Temporal subspace clustering for human motion segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 4453–4461, 2015.

[9] F. Zhou, F. De la Torre, and J. K. Hodgins, "Aligned cluster analysis for temporal segmentation of human motion," in *Proc. IEEE Int. Conf. Automatic Face & Gesture Recognit.*, pp. 1–7, 2008.

[10] G. Xia, H. Sun, L. Feng, G. Zhang, and Y. Liu, "Human motion segmentation via robust kernel sparse subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 135–150, 2017.

[11] X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh, "Constrained multi-view video face clustering," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4381–4393, 2015.

[12] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 849–856, 2002.

[13] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 159–179, 1998.

[14] L. Lu and R. Vidal, "Combined central and subspace clustering for computer vision applications," in *Proc. Int. Conf. Mach. Learn.*, pp. 593–600, 2006.

[15] R. Vidal, "Subspace clustering," *IEEE Trans. Signal Process.*, vol. 28, no. 2, pp. 52–68, 2011.

[16] C. Zhang, Q. Hu, H. Fu, P. Zhu and X. Cao, "Latent multi-view subspace clustering," *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 4279–4287, 2017.

[17] T. Zhou, C. Zhang, X. Peng, H. Bhaskar and J. Yang, "Dual shared-specific multiview subspace clustering," *IEEE Trans. Cybern.*, vol 50, no. 8, pp. 3517–3530, 2019.

[18] J. Yang, J. Liang, K. Wang, P. L. Rosin and M.-H. Yang, "Subspace clustering via good neighbors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1537–1544, 2019.

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

[20] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, 2013.

[21] G. Liu, Z. Lin, and et al., "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, 2013.

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

[23] 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. Eur. Conf. Comput. Vis.*, pp. 347–360, 2012.

[24] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 4109–4118, 2018.

[25] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 3712–3722, 2018.

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

[27] L. Wang, Z. Ding, and Y. Fu, "Low-rank transfer human motion segmentation," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 1023–1034, 2018.

[28] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 24–33, 2017.

[29] X. Peng, J. Feng, S. Xiao, W. Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, 2018.

[30] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, 2018.

[31] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization.," in *Proc. AAAI Conf. Artif. Intell.*, pp. 2921–2927, 2017.

[32] P. Zhou, Y. Hou, and J. Feng, "Deep adversarial subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 1596–1604, 2018.

[33] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. II–II, 2004.

[34] A. Fod, M. J. Matarić, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, no. 1, pp. 39–54, 2002.

[35] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, "Segmenting motion capture data into distinct behaviors," in *Proceedings of Graphics Interface*, pp. 185–194, 2004.

[36] P. Beaudoin, S. Coros, M. Panne, and P. Poulin, "Motion-motif graphs," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 117–126, 2008.

[37] F. De la Torre, J. Campoy, Z. Ambadar, and J. F. Cohn, "Temporal segmentation of facial behavior," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1–8, 2007.

[38] M. W. Robards and P. Sunehag, "Semi-markov kmeans clustering and activity recognition from body-worn sensors," in *Proc. IEEE Int. Conf. Data Min.*, pp. 438–446, 2009.

[39] G. Sun, Y. Cong, L. Wang, Z. Ding, and Y. Fu, "Online multi-task clustering for human motion segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, pp. 970–979, 2019.

[40] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 692–699, 2013.

[41] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 361–368, 2013.

[42] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, 2009.

[43] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool, "Domain generalization and adaptation using low rank exemplar svms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1114–1127, 2017.

[44] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 1859–1867, 2017.

[45] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, pp. 222–230, 2013.

[46] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *Int. J. Comput. Vis.*, vol. 109, no. 1-2, pp. 74–93, 2014.

[47] Y. Xu, X. Fang, Ji. Wu, X. Li, and D. Zhang, "Discriminative transfer subspace learning via low-rank and sparse representation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 850–863, 2015.

[48] M. Geng, Y. Wang, T. Xiang, and Y. Tian, "Deep transfer learning for person re-identification," *arXiv preprint arXiv:1611.05244*, 2016.

[49] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *Int. J. Comput. Vis.*, vol. 109, no. 1-2, pp. 42–59, 2014.

[50] Z. Ding and Y. Fu, "Deep transfer low-rank coding for cross-domain learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1768–1779, 2018.

[51] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *Proc. Int. Conf. Mach. Learn.*, pp. 1180–1189, 2015.

[52] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, pp. 97–105, 2015.

[53] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 4068–4076, 2015.

[54] M. Long, J. Wang, Y. Cao, J. Sun, and S.Y. Philip, "Deep learning of transferable representation for scalable domain adaptation," in *IEEE Trans. Knowledge Data Eng.*, vol. 28, no. 8, pp. 2027–2040, 2016

[55] M. Long, Y. Cao, Z. Cao, J. Wang, and M.I. Jordan, "Transferable Representation Learning with Deep Adaptation Networks," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3071–3085, 2018.

[56] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller, "A deep matrix factorization method for learning attribute representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 417-429, 2016.

[57] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," *Proc. AAAI Conf. Artif. Intell.*, pp. 2921-2927, 2017.

[58] S. Jiang, Z. Ding, and Y. Fu, "Heterogeneous recommendation via deep low-rank sparse collective factorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1097-1111, 2019.

[59] P. Pan, Z. Xu, Y. Yang, and Y. Wu, F.and Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 1029–1038, 2016.

[60] A. Gretton, O. Bousquet, A. Smola, and B. Scholkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Proc. Int. Conf. Algo. Learn. Theory*, vol. 16, pp. 63–78, 2005.
[61] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 586–594, 2015.
[62] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 612–620, 2011.
[63] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.
[64] J. Huang, F. Nie, and H. Huang, "A new simplex sparse learning model to measure data similarity for clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, pp. 3569–3575, 2015.
[65] 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, 2012.
[66] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[67] J. Shi and J. Malik, "Motion segmentation and tracking using normalized cuts," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1154–1160, 1998.
[68] Z. Jiang, Z. Lin, and L. 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, 2012.
[69] D. Huang, S. Yao, Y. Wang, and Fe. De La Torre, "Sequential max-margin event detectors," in *Proc. Eur. Conf. Comput. Vis.*, pp. 410–424, 2014.
[70] 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, 2007.
[71] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: video structure comparison for recognition of complex human activities.," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1593–1600, 2009.
[72] J. Liu, Y. Yang, I. Saleemi, and M. Shah, "Learning semantic features for action recognition via diffusion maps," in *Comput. Vis. Image Underst.*, vol. 116, no. 3, pp. 361–377, 2012.
[73] N. Danal and B. Triggs, "Histgram of oriented gradients for human detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 886–893, 2005.
[74] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.
[75] L. Rdusseeun and P.-J. Kaufman, "Clustering by means of medoids," 1987.
[76] S. Tierney, J. Gao, and Y. Guo, "Subspace clustering for sequential data," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 1019–1026, 2014.
[77] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *Proc. AAAI Conf. Artif. Intell.*, pp. 313–318, 2011.
[78] N. Tremblay and A. Loukas, "Approximating spectral clustering via sampling: a review," in *Sampling Techniques for Supervised or Unsupervised Tasks*, pp. 129–183, 2020.
[79] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, 2018.