This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Temporal Rate Reduction Clustering for Human Motion Segmentation

Xianghan Meng[†], Zhengyu Tong[†], Zhiyuan Huang, and Chun-Guang Li[*]
Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China
{mengxianghan,tongzhengyu,huangzhiyuan,lichunguang}@bupt.edu.cn

Abstract

Human Motion Segmentation (HMS), which aims to partition videos into non-overlapping human motions, has attracted increasing research attention recently. Existing approaches for HMS are mainly dominated by subspace clustering methods, which are grounded on the assumption that high-dimensional temporal data align with a Union-of-Subspaces (UoS) distribution. However, the frames in video capturing complex human motions with cluttered backgrounds may not align well with the UoS distribution. In this paper, we propose a novel approach for HMS, named Temporal Rate Reduction Clustering (TR^2C), which jointly learns structured representations and affinity to segment the sequences of frames in video. Specifically, the structured representations learned by TR^2C enjoy temporally consistency and are aligned well with a UoS structure, which is favorable for addressing the HMS task. We conduct extensive experiments on five benchmark HMS datasets and achieve state-of-the-art performances with different feature extractors. The code is available at: https://github.com/mengxianghan123/TR2C.

1. Introduction

Human motion recognition and analysis have been an active focus of research for around two decades [15, 36, 39]. As a preparatory step, Human Motion Segmentation (HMS) aims to divide sequences of frames in a video into distinct, non-overlapping segments, each representing a specific human motion [20]. Due to the labor-intensive nature of manually annotating sequences in video, researchers often regard HMS as an unsupervised time-series clustering task.

Roughly, the existing methods for HMS typically assume that the frames in a video capturing consecutive motions lie on a Union of low-dimensional Subspaces (UoS) embedded in high-dimensional data. Thus, subspace clustering methods have emerged as a dominated research line for the HMS task [13, 25, 30, 33, 57]. However, an important prior for the HMS task is that the temporally neighboring frames in a video are more likely belonging to the same human motion. To incorporate the temporal continuity between frames in a video, various temporal regularizer is introduced to encourage the temporally neighboring frames to be clustered into the same subspace [27, 48]. More recently, transfer learning-based subspace clustering methods, e.g., [50, 51, 62, 63], have been proposed to further improve the performance of HMS. Despite the flourish of developing these methods for the HMS task in the past decade, the clustering performance of HMS still faces a bottleneck.

For human activities recognition task, as explored in the prior works [21, 41], the frames in videos capture both complex human motions and cluttered backgrounds. Thus, it turns out to be more likely that the features of the frames in video can hardly align well with the UoS distribution at all. As a consequence, it is necessary to amend the representation of the frames in video to align with the UoS distribution while performing the motion segmentation.

In this paper, we attempt to jointly learn structured representations that align with the UoS distribution and simultaneously perform motion segmentation. To be specific, we propose a novel and effective approach for HMS, termed Temporal Rate Reduction Clustering (TR^2C), which integrates the Maximal Coding Rate Reduction (MCR^2) principle [58] and a temporal continuity regularization to jointly learn the temporally consistent representations that align with a UoS distribution and the affinity simultaneously. We solve the problem efficiently by introducing a neural network and leveraging differential programming. Extensive experiments are conducted on five HMS benchmark datasets and superior performance confirms the effectiveness of the proposed approach.

The contributions of the paper are highlighted as follows.

1. We propose a novel approach, named Temporal Rate Reduction Clustering (TR^2C), which is able to jointly learn temporally consistent structured representations and affinity to segment the sequences of frames in video.
2. We demonstrate the effectiveness of our proposed TR^2C with extensive experiments on five benchmark datasets

[*]Corresponding author. [†]These two authors are equally contributed.

and different feature extractors, achieving state-of-the-art performance.

To the best of our knowledge, it is for the first time to exploit the MCR^2 principle for clustering temporal sequences.

2. Related Work

In this section, we will review the previous works for HMS at first, and then introduce the relevant work on the principle of maximal coding rate reduction.

Probabilistic methods for HMS. Early human motion segmentation algorithms primarily relied on probabilistic models to model time series data, e.g., Hidden Markov Models [45], Dynamic Bayesian Networks [37] and Autoregressive Moving Average Models [55]. These methods typically employ the Expectation Maximization (EM) algorithm for effective optimization. Besides, there are also several effective frameworks which extend classical clustering algorithms (e.g., k-means) by combining Dynamic Time Warping [60, 61].

Subspace clustering based methods for HMS. Under the assumption that human motion data lie on a UoS, each motion corresponding to a subspace, it is appealing to apply subspace clustering methods to address the HMS task. To date, various temporal subspace clustering methods are proposed, e.g., Ordered Subspace Clustering (OSC) [48] and Temporal Subspace Clustering (TSC) [27], in which the temporal continuity information is exploited. In OSC, the $\|\cdot\|_{1,2}$ norm is introduced as a temporal continuity regularization; in TSC, the temporal continuity graph Laplacian is introduced to encourage neighboring frames to be grouped into the same subspace. Then, in [16], Gaussian Process is incorporated to handle missing data to enhance the robustness; in [53], minimum spanning tree is introduced to characterize the affinity between neighboring frames with less redundancy. In addition, transfer learning is also introduced to align the source domain and target domain by optimizing a projection [50, 51] or learning multi-mutual consistency and diversity across different domains [62, 63]. However, the performance of the methods mentioned above is still unsatisfactory due to the data deviating from the UoS distribution.

Representation learning based subspace clustering methods for HMS. To learn effective temporal representations for HMS, in [3], a dual-side auto-encoder is introduced to learn representations assisted with temporal consistency constraints; after that, in [4], a velocity guidance mechanism is leveraged for the better capturing of changes between different motions; in [7], non-local self-similarity is introduced to form the representations of each frame; in [8], graph consistency is introduced to regularize the learned representations. More recently, in [9], an approach termed Graph Constraint Temporal Subspace Clustering (GCTSC) is developed in which graph consistency-based representation learning is combined with temporal subspace clustering (TSC). Unfortunately, in these aforementioned methods, there is no evidence to demonstrate that the learned representations are suitable or well aligned with the UoS distribution.

MCR^2 principle. In supervised learning, a so-called Maximal Coding Rate Reduction (MCR^2) principle is proposed to learn discriminative and diverse features that conform to a UoS distribution [52, 58]. In the unsupervised learning field, the MCR^2 principle is leveraged to perform image clustering, assisted with contrastive learning in [29]. Then, an approach called Manifold Linearizing and Clustering (MLC) is presented in [11], which incorporates a doubly stochastic affinity into the MCR^2 framework, and in [5] MLC is further evaluated on visual datasets with pretrained CLIP features [40], achieving excellent performance. Nonetheless, there is no prior work to address the HMS task with the MCR^2 framework to date.

3. Our Method

In this section, we will first formulate a novel optimization problem for the HMS task to learn simultaneously efficient representations and affinity for segmentation. Then, we will develop a differential programming approach to solve the problem efficiently.

3.1. Problem Formulation

Given a video consisting of N frames $\mathcal{D} = \{\mathcal{I}_i\}_{i=1}^N$, HMS aims to group each frame into one of prescribed human motions. Denote $\boldsymbol{X} = [\boldsymbol{x}_i, \ldots, \boldsymbol{x}_N]$ as the sequence of extracted features from each frame, which are typically used as the input data of HMS frameworks.

In the HMS task, the extracted features from the sequence of frames corresponding to different motions are typically assumed to approximately lie on a Union of Subspaces (UoS), where each subspace is spanned by the features of the frames belonging to a specific motion. Based on such a UoS assumption, subspace clustering methods have emerged as the dominant approaches for the HMS task [13, 27, 30, 33, 48]. However, the segmentation performance of these methods seems to be stuck, primarily due to the misalignment between the data and the UoS assumption, especially in the scenarios involving complex motions and cluttered background [21, 41]. To address this limitation, it is crucial to learn structured representation of the data, i.e., a mapping function $\mathcal{F} : \boldsymbol{X} \to \boldsymbol{Z}$ that transforms the input $\boldsymbol{X}$ into representation $\boldsymbol{Z}$ with more favorable distribution, thereby enhancing segmentation performance.

Given the partition $\boldsymbol{\Pi}$, learning UoS representations. The principle of Maximal Coding Rate Reduction (MCR^2) guarantees to learn representations $\boldsymbol{Z}$ which align with UoS structure in supervised setting, where the coding rate quantifies the minimum average coding length required to com-

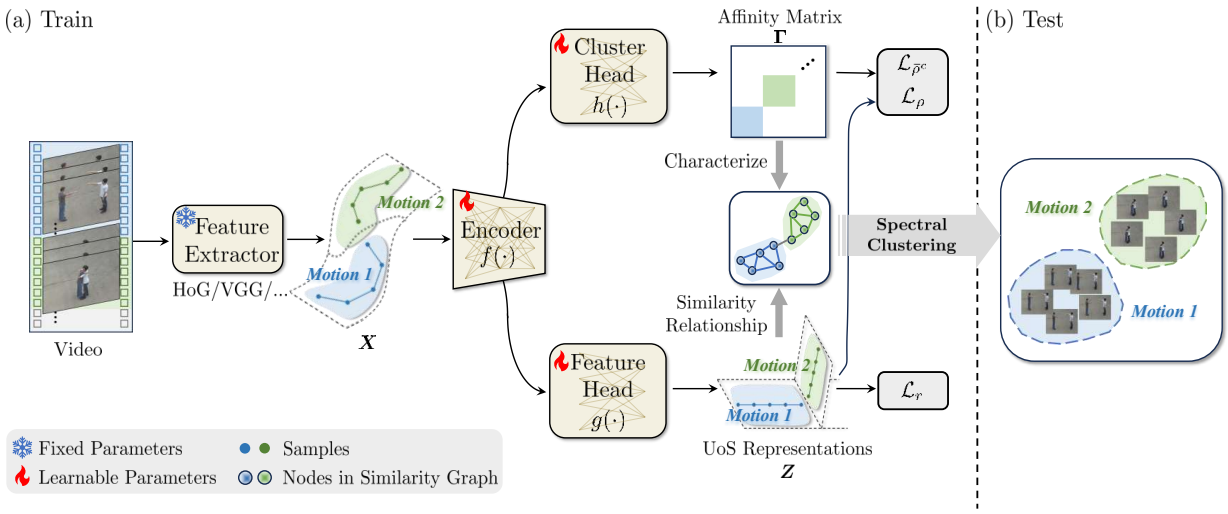


Figure 1. **The Framework of TR^{2}C.** Structured representations and affinity are jointly learned in TR^{2}C to facilitate motion segmentation.

press the representations which are drawn from a mixture of Gaussian distributions in lossy compression scenario [34]. To be specific, given a set of diagonal matrices $\mathbf{\Pi} = \{\mathbf{\Pi}_j \in \{0,1\}^{N\times N}\}_{j=1}^K$, where the m-th diagonal element of $\mathbf{\Pi}_j$ indicates whether the m-th sample belongs to the j-th class, the MCR^2 principle aims to optimize:

$$\begin{aligned} \max_{\boldsymbol{Z}} \quad & \Delta\rho(\boldsymbol{Z},\mathbf{\Pi},\epsilon) := \rho(\boldsymbol{Z},\epsilon) - \rho^c(\boldsymbol{Z},\epsilon \mid \mathbf{\Pi}), \\ \text{s.t.} \quad & \|\boldsymbol{z}_i\|_2^2 = 1, \quad \text{for } i = 1,\cdots,N, \end{aligned} \tag{1}$$

where

$$\rho(\boldsymbol{Z},\epsilon) := \frac{1}{2}\log\det(\boldsymbol{I} + \frac{d}{N\epsilon^2}\boldsymbol{Z}\boldsymbol{Z}^\top) \tag{2}$$

is the coding rate of the representations $\boldsymbol{Z} \in \mathbb{R}^{d\times N}$ with respect to precision $\epsilon > 0$ and

$$\rho^c(\boldsymbol{Z},\epsilon \mid \mathbf{\Pi}) := \sum_{j=1}^K \frac{\operatorname{tr}(\mathbf{\Pi}_j)}{2N}\log\det(\boldsymbol{I} + \frac{d}{\operatorname{tr}(\mathbf{\Pi}_j)\epsilon^2}\boldsymbol{Z}\mathbf{\Pi}_j\boldsymbol{Z}^\top) \tag{3}$$

is the sum of coding rate of the representations $\boldsymbol{Z}_j$ from each class indicated by $\mathbf{\Pi}_j$.

From a geometric perspective, the $\log\det(\cdot)$-based function, which serves as a concave relaxation of $\operatorname{rank}(\cdot)$, measures the volume of the representations. By jointly maximizing the holistic volume of the representations while minimizing their intra-class volumes, the representations naturally conform to a union of orthogonal subspaces distribution.[1]

Given $\boldsymbol{Z}$, learning the partition $\mathbf{\Pi}$. For the unsupervised HMS task, the representations are fixed and assumed lying on a UoS and we aim to find the assignment matrices set $\mathbf{\Pi}$. In such case, the correct assignment matrices set $\mathbf{\Pi}$ would sort these data into its own subspace and thus make the coding rate minimized [34]. Therefore, the task of learning $\mathbf{\Pi}$ can be formulated into an optimization problem as follows:

$$\min_{\mathbf{\Pi}} \quad \rho^c(\boldsymbol{Z},\epsilon \mid \mathbf{\Pi}). \tag{4}$$

However, for an unsupervised HMS task, the representations $\boldsymbol{Z}$ might not well align with a UoS distribution.

Jointly learning $\boldsymbol{Z}$ and $\mathbf{\Pi}$. When both $\boldsymbol{Z}$ and $\mathbf{\Pi}$ are jointly learned, we have an optimization problem as follows:

$$\begin{aligned} \min_{\boldsymbol{Z},\mathbf{\Pi}} \quad & \rho^c(\boldsymbol{Z},\epsilon \mid \mathbf{\Pi}), \\ \text{s.t.} \quad & \|\boldsymbol{z}_i\|_2^2 = 1, \quad \text{for } i = 1,\cdots,N. \end{aligned} \tag{5}$$

An important prior in the HMS task is that the temporally neighboring frames in the video are more likely belonging to the same motion. Therefore, it is helpful to introduce a temporal continuity regularizer, which encourages learning the representations that are of temporal consistency between neighboring frames and thus facilitates the segmentation task [27, 48]. However, the temporal consistency among frames in a video is ignored in problem (5).

Jointly learning temporally consistent representation $\boldsymbol{Z}$ and $\mathbf{\Pi}$. Analog to [27], we introduce a temporal Laplacian regularizer, which is defined as follows:

$$r(\boldsymbol{Z}) := \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N w_{ij}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2 = \operatorname{tr}(\boldsymbol{Z}\boldsymbol{L}\boldsymbol{Z}^\top), \tag{6}$$

where $\boldsymbol{L} = \operatorname{Diag}(\boldsymbol{W}\mathbf{1}_N) - \boldsymbol{W}$ is the graph Laplacian matrix, $\mathbf{1}_N$ is a column vector of dimension N consisting of 1, and the affinity $\boldsymbol{W} = \{w_{ij}\}_{i,j=1}^N$ is defined as:

$$w_{ij} := \begin{cases} 1, & \text{if } |i-j| \le \frac{s}{2}, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

[1] Please refer to [52, 58] for rigorous proofs.

where s is the size of a sliding window. The temporal Laplacian regularizer $r(\boldsymbol{Z})$ conforms the similarity relationship among the learned representations to the pre-defined affinity $\boldsymbol{W}$, which geometrically governs the smoothness of learned representations along the temporal dimension.

By taking into account the temporal continuity prior, we formulate an optimization problem as follows:

$$\begin{aligned} \min_{\boldsymbol{Z},\boldsymbol{\Pi}} \quad & \rho^c(\boldsymbol{Z},\epsilon \mid \boldsymbol{\Pi}) + \lambda r(\boldsymbol{Z}), \\ \text{s.t.} \quad & \|\boldsymbol{z}_i\|_2^2 = 1, \quad \text{for } i = 1, \cdots, N, \end{aligned} \tag{8}$$

where $\lambda > 0$ is a hyper-parameter. While problem (8) looks appealing, unfortunately, there exists undesired trivial solutions $(\boldsymbol{Z}_\star, \boldsymbol{\Pi}_\star)$ that all embeddings are collapsed.[2]

To prevent the collapsed solution, inspired by [35, 58], we add a maximizing total coding rate based regularization term into problem (8). Thus we have an optimization problem as follows:

$$\begin{aligned} \min_{\boldsymbol{Z},\boldsymbol{\Pi}} \quad & -\rho(\boldsymbol{Z},\epsilon) + \lambda_1 \rho^c(\boldsymbol{Z},\epsilon \mid \boldsymbol{\Pi}) + \lambda_2 r(\boldsymbol{Z}), \\ \text{s.t.} \quad & \|\boldsymbol{z}_i\|_2^2 = 1, \quad \text{for } i = 1, \cdots, N, \end{aligned} \tag{9}$$

where $\lambda_1, \lambda_2 > 0$ are two hyper-parameters. We call this framework in (9) a Temporal Rate Reduction Clustering (TR^{2}C).

Remarks. The total coding rate term $\rho(\boldsymbol{Z},\epsilon)$ in our TR^{2}C offers a tighter approximation to the $\operatorname{rank}(\boldsymbol{Z})$ [14, 32, 34, 58]. Minimizing $-\rho(\boldsymbol{Z},\epsilon)$ together with $\rho^c(\boldsymbol{Z},\epsilon \mid \boldsymbol{\Pi}) + r(\boldsymbol{Z})$ can help prevent over-compressing the learned representations. Although the TR^{2}C problem appears to be rational, it is still quite challenging to solve due to the combinatorial nature.

3.2. Optimization

Rather than directly optimizing $\boldsymbol{\Pi}$, in this paper, following [11], we introduce a doubly stochastic affinity matrix $\boldsymbol{\Gamma} \in \boldsymbol{\Xi}$, where $\boldsymbol{\Xi} := \{\boldsymbol{\Gamma} \in \mathbb{R}_+^{N\times N} \mid \boldsymbol{\Gamma}\mathbf{1} = \mathbf{1}, \boldsymbol{\Gamma}^\top \mathbf{1} = \mathbf{1}\}$, then the term ρ^c is relaxed to:

$$\bar{\rho}^c(\boldsymbol{Z},\epsilon \mid \boldsymbol{\Gamma}) := \frac{1}{N}\sum_{j=1}^{N} \log\det(\boldsymbol{I} + \frac{d}{\epsilon^2}\boldsymbol{Z}\operatorname{Diag}(\boldsymbol{\Gamma}_j)\boldsymbol{Z}^\top) \tag{10}$$

where $\boldsymbol{\Gamma}_j$ being the j-th column of $\boldsymbol{\Gamma}$.

Similar to [11, 59], we consult the differential programming approach to solve the continuously relaxed problem. We re-parameterize $\boldsymbol{Z}$ and $\boldsymbol{\Gamma}$ through properly designed neural networks and optimize over the parameters of the neural networks. To be specific, we introduce an encoder $f(\cdot)$, a feature head $g(\cdot)$ and a cluster head $h(\cdot)$ to form our implementation framework. Formally, the outputs of feature head and cluster head are computed by:

$$\begin{aligned} \boldsymbol{z}_i &= g\left(f\left(\boldsymbol{x}_i\right)\right), \\ \boldsymbol{y}_i &= h\left(f\left(\boldsymbol{x}_i\right)\right), \end{aligned} \tag{11}$$

for all $i \in \{1, \ldots, N\}$. Then, after the normalization of the outputs $\tilde{\boldsymbol{z}}_i = \boldsymbol{z}_i/\|\boldsymbol{z}_i\|_2$, $\tilde{\boldsymbol{y}}_i = \boldsymbol{y}_i/\|\boldsymbol{y}_i\|_2$, we compute the affinity $\boldsymbol{\Gamma}$ by:

$$\boldsymbol{\Gamma} = \mathcal{P}_{\Xi}(\tilde{\boldsymbol{Y}}^\top \tilde{\boldsymbol{Y}}), \tag{12}$$

where $\mathcal{P}_{\Xi}(\cdot)$ is a sinkhorn projection [6], which is a differentiable projection to doubly stochastic matrix. Note that owing to the normalization for $\boldsymbol{z}_i$ and the sinkhorn projection, the constraints in (9) and for defining the doubly stochastic Γ can be (automatically) satisfied.

Equipped with the reparameterization, rather than directly optimizing over $\boldsymbol{Z}$ and $\boldsymbol{\Gamma}$, we instead update the parameters of the networks by back-propagation. Specifically, we denote the parameters in networks $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ as $\boldsymbol{\theta}$. Then the parameters $\boldsymbol{\theta}$ can be updated by minimizing the following loss functions:

$$\mathcal{L} = -\mathcal{L}_\rho + \lambda_1 \mathcal{L}_{\bar{\rho}^c} + \lambda_2 \mathcal{L}_r, \tag{13}$$

where

$$\begin{aligned} \mathcal{L}_\rho &:= \frac{1}{2}\log\det(\boldsymbol{I} + \frac{d}{N\epsilon^2}\boldsymbol{Z}(\boldsymbol{\theta})\boldsymbol{Z}(\boldsymbol{\theta})^\top), \\ \mathcal{L}_{\bar{\rho}^c} &:= \sum_{j=1}^{N}\frac{1}{N}\log\det(\boldsymbol{I} + \frac{d}{\epsilon^2}\boldsymbol{Z}(\boldsymbol{\theta})\operatorname{Diag}(\boldsymbol{\Gamma}_j(\boldsymbol{\theta}))\boldsymbol{Z}(\boldsymbol{\theta})^\top), \\ \mathcal{L}_r &:= \operatorname{tr}(\boldsymbol{Z}(\boldsymbol{\theta})\boldsymbol{L}\boldsymbol{Z}(\boldsymbol{\theta})^\top). \end{aligned} \tag{14}$$

Finally, having the affinity $\boldsymbol{\Gamma}(\boldsymbol{\theta})$, we apply spectral clustering [43] to yield HMS results as in [9, 27]. For clarity, we illustrate the overall framework of our TR^{2}C in Figure 1 and summarize the whole training procedure in Algorithm 1.

4. Experiments

To evaluate the effectiveness of our proposed approach, following [4, 9, 27, 50, 51, 62, 63], we conduct experiments on five benchmark datasets, including Weizmann action dataset (Weiz) [17], Keck gesture dataset (Keck) [21], UT interaction dataset (UT) [41], Multi-model Action Detection dataset (MAD) [19], and UCF-11 YouTube action dataset (YouTube) [31]. We defer the description of these datasets to Appendix A.1.

4.1. Experimental Setups

For datasets Weiz, Keck, UT, and MAD, following the baselines, we conduct experiments based on 324-dimensional

[2] The existence of collapsed solutions often leads to an over-smoothing issue. For example, the over-smoothing issue in graph neural networks results in indistinguishable node embeddings [26]; the over-smoothing issue in deep subspace clustering causes catastrophically collapsed representations [18].

Algorithm 1 Temporal Rate Reduction Clustering (TR^{2}C)

Input: Input Features $\boldsymbol{X} \in \mathbb{R}^{D\times N}$, hyper-parameters λ_1, λ_2, number of iterations T, network parameters $\boldsymbol{\theta}$, learning rate η

Initialization: Randomly initialize parameters $\boldsymbol{\theta}$

for $t = 1, \ldots, T$ **do**
  # Forward propagation
  Compute $\boldsymbol{Z}(\boldsymbol{\theta})$ and $\boldsymbol{Y}(\boldsymbol{\theta})$ by (11)
  Compute affinty $\boldsymbol{\Gamma}(\boldsymbol{\theta})$ by (12)
  # Backward propagation
  Compute loss $\mathcal{L}$ by (13)
  Compute $\nabla_{\boldsymbol{\theta}} \doteq \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}$
  Set $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{\theta}}$
end for

Test: Apply spectral clustering on $\boldsymbol{\Gamma}(\boldsymbol{\theta})$.

HoG features [64] of each frame.[3] For the YouTube dataset, following [63], we conduct experiments on 1000-dimensional pretrained VGG-16 features [44]. To explore the limit of TR^{2}C, we also evaluate the performance with the features extracted from the image encoder of pretrained CLIP model [40]. Key information about datasets is summarized in Table 1.

Table 1. **Key information about datasets used by training.** We show the number of sequences, the number of motions, the maximal number of frames of all the sequences, dimension of HoG, VGG and CLIP features.

Datasets	#Seq	#Motions	#Frames	Dim (HoG/VGG)	Dim (CLIP)
Weiz	9	10	826	324 (HoG)	768
Keck	4	10	1245	324 (HoG)	768
UT	10	6	650	324 (HoG)	-
MAD	40	10	1379	324 (HoG)	-
YouTube	4	10	2572	1000 (VGG)	768

We use clustering accuracy (ACC) and normalized mutual information (NMI) as the evaluation metrics. The performance is reported by taking the mean and standard deviation after running the experiments with 5 different random seeds.

We choose a lightweight neural network architecture, where the encoder is a two-layer Multi-Layer Perceptron (MLP), with the clustering head and feature head being Fully Connected (FC) layers. The hyper-parameters λ_1 and λ_2 are tuned independently for each dataset. Sliding window size s is fixed as $s = 2$ for all datasets. The sensitivity to hyper-parameters is reported in Figure 5 and the hyper-parameter settings are summarized in the Appendix A.2.

[3]The HoG features are available at https://github.com/wanglichenxj/Low-Rank-Transfer-Human-Motion-Segmentation.

4.2. Comparative Results

We compare the performance of TR^{2}C on HoG features to subspace clustering algorithms, e.g., LRR [30], RSC [28], SSC [13], LSR [33], temporal regularized subspace clustering algorithms, e.g., OSC [48], TSC [27], transferable subspace clustering algorithms, e.g., TSS [50], LTS [51], MTS [62], CDMS [63], and representation learning assisted temporal clustering algorithms, e.g., DGE [8], DSAE [3], VSDA [4], GCTSC [9]. The results other than our TR^{2}C are cited from [9, 63].

As shown in Table 2, although TR^{2}C does not utilize additional data through a transfer learning strategy, yet it achieves a clustering accuracy that is 20% higher than that of the transfer learning-based approach. TR^{2}C also outperforms other representation learning assisted HMS algorithms, namely, DGE [8], DSAE [3], VSDA [4], GCTSC [9]. This may stem from the fact that these representation learning methods using self-similarity, auto-encoder, or graph consistency cannot substantially improve the structure of the data distribution. In contrast, TR^{2}C explicitly ensures that the learned representations exhibit an desirable distribution, contributing to improved separability. It is noteworthy that the conventional subspace clustering algorithms without using temporal continuity information perform well on the YouTube dataset, since the background of YouTube dataset shifts significantly among different motions and the pretrained VGG model extracts semantic meaningful features. TR^{2}C additionally integrates temporally consistency and yields satisfying segmentation result. Please refer to Appendix A.8 for the visualization of segmentation results of TR^{2}C.

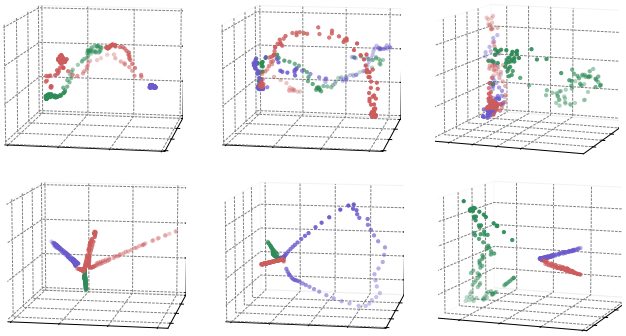
4.3. More Evaluations

Qualitative evaluation of representations. To have an intuitive comparison, we conduct experiments to visualize the input HoG data and the learned TR^{2}C representations on the Keck, UT and MAD datasets. The visualization results are displayed in Figure 2 (see Appendix A.1 for more results). For each dataset, we use a subset containing 3 different motions for better clarity. We apply Principal Component Analysis (PCA) for dimension reduction because it performs a linear projection on the input data, well preserving its structure.

As can be observed, the input data with raw HoG features (in the first row) lie on approximately one-dimensional manifolds without clear and separable a union of subspace structure. This observation accounts for the reason why the previous HMS approaches did not achieve satisfactory performance with the raw HoG feature. On the contrary, the output TR^{2}C representation (in the second row) exhibits a clear union-of-orthogonal-subspaces structure, making it easier to segment different motions. The clear contrast in PCA visualization reveals that the union-of-orthogonal-

Table 2. **The performance of TR^{2}C comparing to state-of-the-art algorithms.**

Method	Weiz		Keck		UT		MAD		YouTube	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
LRR [30]	43.82	36.38	48.62	42.97	40.51	41.62	22.49	23.97	-	-
RSC [28]	41.12	48.94	34.85	32.52	36.64	18.81	37.30	34.18	-	-
SSC [13]	60.09	45.76	38.58	31.37	49.98	43.89	47.58	38.17	-	-
LSR [33]	50.93	50.91	45.48	48.94	43.22	51.83	36.67	39.79	93.16	96.64
OSC [48]	70.47	52.16	59.31	43.93	68.77	58.46	55.89	43.27	-	-
TSC [27]	61.11	81.99	47.81	71.29	53.40	75.93	55.56	77.21	90.40	95.00
TSS [50]	62.08	85.09	53.95	80.49	59.44	78.78	57.92	82.86	62.94	88.20
LTS [51]	63.91	85.99	55.09	82.26	62.99	81.28	59.80	82.11	62.26	88.98
MTS [62]	64.36	83.71	60.10	82.70	64.33	82.39	61.63	83.14	64.40	81.41
CDMS [63]	65.05	83.75	62.07	80.40	66.43	83.06	65.36	82.51	67.98	91.33
MLC [11]	37.30	45.86	47.29	49.78	45.79	35.30	30.27	29.40	94.82	97.30
DGE [8]	-	-	72.00	83.00	-	-	67.00	82.00	-	-
DSAE [3]	61.99	78.79	57.53	74.07	60.06	79.50	55.48	77.34	-	-
VSDA [4]	62.87	79.92	58.04	73.97	62.03	82.26	56.06	77.70	-	-
GCTSC [9]	85.01	90.53	78.64	83.25	87.00	82.56	82.97	84.71	95.79	96.30
Our TR^{2}C	**94.12**±1.20	**95.91**±0.66	**83.50**±1.98	**85.63**±0.86	**93.54**±1.05	**91.83**±0.65	**83.08**±0.62	**86.86**±0.37	**97.96**±1.53	**98.96**±0.83



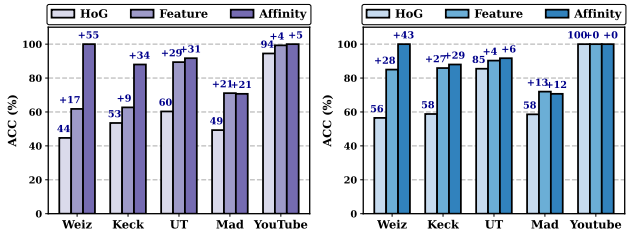
(a) Keck (b) UT (c) MAD

Figure 2. **Visualization of features via PCA.** First row: input HoG features. Second row: learned TR^{2}C representations. Experiments are conducted on the first sequence of each dataset.

subspaces distribution of features is a key factor contributing to the state-of-the-art performance of TR^{2}C.

Quantitative evaluation of representations. To quantitatively evaluate the effectiveness of the learned representation of TR^{2}C, we illustrate the clustering performance of using HoG feature and the learned representation in Figure 3. We perform spectral clustering (SC) [43] and Elastic Net Subspace Clustering (EnSC) [57] algorithms, and report the clustering accuracy in Figures 3a and 3b, respectively.[4]

Comparing to the HoG features, the clustering accuracy of using the learned representations improves significantly across all datasets and clustering algorithms, with particularly notable gains on the UT dataset using spectral clustering (a 29% improvement) and the Weizmann dataset using EnSC (a 28% improvement). Since the performance of clustering algorithms heavily depends on the underlying data distribution, these significant improvements highlight the enhanced quality of the learned representations. Assisted with these features, the clustering accuracy of the affinity matrix $\mathbf{\Gamma}$ is further improved, suggesting that the clustering head in TR^{2}C is more effective in revealing clusters than other classical clustering approaches. The superior performance of the clustering head may be attributed to the fact that the optimal $\mathbf{\Gamma}$ in Problem (9) is better at capturing the subspace membership of the learned representations.



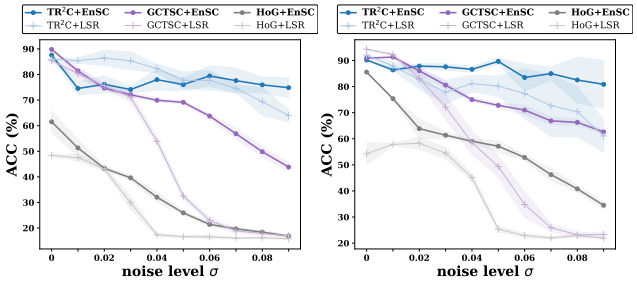
(a) SC (b) EnSC

Figure 3. **Clustering accuracy of using HoG features, learned features, and learned affinity.** Experiments are conducted on the first sequence of each dataset.

Robustness evaluation of representations. Intuitively, if the representations align with a UoS structure, they will enjoy strong robustness to the random noise corruption. To verify this, we corrupt the learned representations of TR^{2}C, GCTSC [9] and HoG features by the additive isotropic Gaussian noise $\mathcal{N}(\mathbf{0}, \sigma \boldsymbol{I})$, where $\sigma > 0$ is the noise level. We apply EnSC [57] and LSR [33] to cluster the corrupted features and plot the clustering accuracy along with the standard deviation after running with 5 different random seeds. As shown in Figure 4, although the clustering performance of GCTSC is highly competitive without noise

[4]For EnSC, we tune the hyper-parameter $\gamma \in \{1, 2, 5, 10, 20, 50, 100, 200, 400, 800, 1600, 3200\}$ and the hyper-parameter $\tau \in \{0.9, 0.95, 1\}$ and report the best clustering result.

($\sigma = 0$), it decreases significantly regardless of the clustering algorithms on both datasets. In contrast, the representations of TR^{2}C is more robust to the noise corruption. When clustering with EnSC (thick line), the average clustering accuracy of TR^{2}C's representations drops at most 15% and 10% on Weiz and UT dataset, respectively, which are significantly less than the 45% and 30% drop of GCTSC.



(a) Weiz (b) UT

Figure 4. **Clustering accuracy of features under noise corruption.** We test on features learned by TR^{2}C, GCTSC, and HoG features, using EnSC (thick) and LSC (thin) for clustering.

Ablation study. To study the effect of each component in the loss function, we conduct ablation study and report the results in Table 3 (see Appendix for more results). Clearly, we can read that both $\mathcal{L}_\rho$ and $\mathcal{L}_{\bar{\rho}^c}$ are indispensable for our TR^{2}C to learn structured representations to facilitate HMS. As illustrated, the absence of either $\mathcal{L}_\rho$ or $\mathcal{L}_{\bar{\rho}^c}$ will seriously detriment the performance, resulting in over-segmentation (line 3) or over-compactness (line 2) of representations. Additionally, the temporal regularizer $\mathcal{L}_r$ also contributes significantly to the clustering performance (line 1), which validates that temporal consistency of representations is an indispensable prior for HMS problem.

Table 3. **Ablation study.** We report the average performance of all the sequences in the Weiz, Keck and UT dataset.

Loss			Weiz		Keck		UT	
$\mathcal{L}_\rho$	$\mathcal{L}_{\bar{\rho}^c}$	$\mathcal{L}_r$	ACC	NMI	ACC	NMI	ACC	NMI
✓	✓		37.30	45.86	47.29	49.78	45.79	35.30
	✓	✓	53.14	61.51	47.91	51.39	63.13	59.51
✓		✓	64.68	74.67	58.60	65.21	65.67	66.09
✓			41.21	44.57	44.01	41.46	46.80	37.49
	✓		56.03	64.19	47.50	52.11	76.39	72.41
		✓	52.59	60.33	48.35	50.87	62.13	58.29
✓	✓	✓	**94.07**	**96.08**	**86.78**	**86.93**	**94.05**	**92.34**

Sensitivity to hyper-parameters. We study the sensitivity of our model with respect to the parameters λ_1 and λ_2, the window size s and the coding precision ϵ. As shown in Figure 5, λ_1 is recommended to be smaller than 0.35, as an over-large λ_1 might lead to over-compactness of the learned representations. In contrast, λ_2 and s can be selected from a wide range while maintaining optimal performance. The coding precision ϵ determines the level of distortion in data compression[5], which is recommended to be larger than 0.1. In general, our TR^{2}C framework is insensitive to these hyper-parameters.

Time cost comparison. To test the time consuming of TR^{2}C, we report the training time of the first sequence across different benchmarks and computing devices.[6] We defer the complexity analysis of TR^{2}C to Appendix A.6. For TSC and GCTSC along with its GPU implementation, we use the code provided by [9] and train for $T = 15$ and $T = 100$ iterations, respectively. For TR^{2}C, we train for $T = 500$ iterations. We conduct all the experiments with a single NVIDIA RTX 3090 GPU and Intel Xeon Platinum 8255C CPU. As shown in Table 4, the time cost of TR^{2}C is comparable to TSC, but significantly less than GCTSC, which is quite time-consuming. Since the parallel computing empowered by GPU speeds up particularly for large N, TR^{2}C+GPU outperforms TSC in terms of speed with a greater performance margin on the YouTube dataset.

Table 4. **Total training time (s) comparison.** The best time cost is marked in **bold** and the second best result is underlined.

	T	Weiz	Keck	UT	MAD	YouTube
TSC	15	20.0	20.4	**5.6**	**9.2**	116.5
GCTSC	100	1551.7	1554.1	415.4	810.3	12677.8
GCTSC+GPU	100	1122.2	1142.4	374.8	622.4	8474.7
TR^{2}C	500	91.0	228.2	82.2	138.4	929.1
TR^{2}C+GPU	500	**10.7**	**16.9**	10.4	12.8	**41.0**

TR^{2}C based on CLIP. The performance of TR^{2}C based on CLIP pretrained features (denoted as "TR^{2}C+CLIP") comparing to state-of-the-art approaches is shown in Table 5. As illustrated, the clustering accuracy of TR^{2}C+CLIP surpasses that of TR^{2}C on the Weizmann, Keck and YouTube dataset, with improvements of 2%, 7% and 1%, respectively. This performance enhancement is attributed to the superior representation capability of the CLIP pretrained model. Specifically, the pretrained CLIP image encoder captures more high-level semantic information from each frame, which is crucial for distinguishing different human motions. In contrast, HoG features primarily capture low-level information from each frame.

Besides, we also conduct experiments on the zero-shot learning of CLIP (please refer to Appendix A.9), which demonstrates that vanilla zero-shot classification is not suitable for HMS; whereas TR^{2}C+CLIP succeeds by learning temporally consistent representations that align with a union of orthogonal subspaces.

Comparison to Temporal Action Segmentation. Temporal Action Segmentation (TAS) is an unsupervised task

[5]Please kindly refer to [34] for more details.

[6]Since spectral clustering is adopted by all baselines for segmentation, our evaluation specifically focuses on the training time cost.

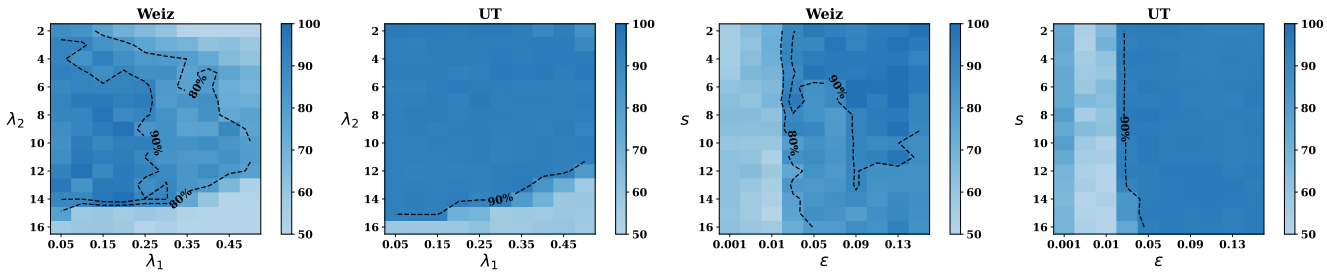


Figure 5. **Sensitivity to hyper-parameters.** The sensitivity of TR^{2}C with respect to λ_1, λ_2, s and ϵ is studied through experiments on the first sequence of the Weiz and UT dataset with three different random seeds.

Table 5. **The performance of TR^{2}C based on CLIP features comparing to state-of-the-art algorithms.** We denote "$\boldsymbol{A}$+CLIP" as the algorithm $\boldsymbol{A}$ based on CLIP features.

Method	Weiz		Keck		YouTube	
	ACC	NMI	ACC	NMI	ACC	NMI
TSC	61.11	81.99	47.81	71.29	90.40	95.00
TSC+CLIP	89.61	93.35	78.81	83.81	93.86	94.91
GCTSC	85.01	90.53	78.64	83.25	95.79	96.30
GCTSC+CLIP	89.39	89.90	83.31	84.55	96.64	97.41
TR^{2}C	94.12	95.91	83.50	85.63	97.96	98.96
TR^{2}C+CLIP	**96.21**	**97.12**	**90.10**	**89.63**	**99.35**	**99.48**

closely related to HMS, as both aim to partition videos into non-overlapping segments [10]. The key difference lies in the scale and the nature of the actions: HMS typically involves *macro-scale* motions (e.g., running, jumping) characterized by global and easily distinguishable movements, whereas TAS focuses on *micro-scale* manipulative actions (e.g., grasping a cup, pouring milk), which are more subtle and fine-grained. We conduct experiments on three benchmark dataset on TAS, including the Breakfast dataset [22], the YouTube Instructional dataset [2], and the 50 Salads dataset [47]. For the feature extractor selection of each dataset, we follow the baselines. Experimental details are described in Appendix A.10. As can be seen in Table 6, TR^{2}C performs comparable with the state-of-the-art TAS algorithms on the three benchmark datasets for TAS.

Table 6. **The action segmentation performance of TR^{2}C.** Other results are directly cited from their papers.

Method	Breakfast			YouTube Instr.			50 Salads		
	MoF	F1	mIoU	MoF	F1	mIoU	MoF	F1	mIoU
LSTM+AL [1]	42.9	-	-	-	39.7	-	-	-	-
TWF [42]	62.7	49.8	**42.3**	56.7	48.2	-	66.8	56.4	**48.7**
ABD [12]	**64.0**	52.3	-	67.2	49.2	-	**71.8**	-	-
CoSeg [54]	53.1	**54.7**	-	47.9	-	53.7	-	-	-
ASOT [56]	63.3	53.5	35.9	71.2	**63.3**	47.8	64.3	51.1	33.4
Our TR^{2}C	59.9	47.1	39.9	**71.8**	59.1	**57.8**	65.8	**58.4**	48.2

Conversely, we also evaluate the state-of-the-art TAS algorithms on Weizmann and Keck datasets with HoG, CLIP and DINOv2 [38] features. As shown in Table 7, our TR^{2}C achieves the best performance across different datasets and different features. Furthermore, we argue that the performance of our TR^{2}C faithfully reflects the quality of the input features, i.e., DINOv2>CLIP>HoG.

Table 7. **Evaluating state-of-the-art TAS algorithms** on Weiz and Keck datasets with HoG, CLIP and DINOv2 features.

Feature	Methods	Weiz		Keck		AVG	
		ACC	NMI	ACC	NMI	ACC	NMI
HoG	TWF [42]	66.1	84.3	48.8	63.7	57.5	74.0
	ASOT [56]	68.0	77.6	66.4	76.0	67.2	76.8
	HVQ [46]	66.7	61.5	63.5	75.2	65.1	68.4
	GCTSC [9]	85.0	90.5	78.6	83.3	81.8	86.9
	Our TR^{2}C	**94.1**	**95.9**	**83.5**	**85.6**	**88.8**	**90.8**
CLIP (ViT-L/14)	TWF [42]	76.8	89.3	70.4	79.4	73.6	84.4
	ASOT [56]	71.1	79.4	67.0	76.4	69.1	77.9
	HVQ [46]	72.6	85.2	73.5	78.5	73.1	81.9
	GCTSC [9]	89.4	89.9	83.3	84.6	86.4	87.3
	Our TR^{2}C	**96.2**	**97.1**	**90.1**	**89.6**	**93.2**	**93.4**
DINOv2 (ViT-L/14)	TWF [42]	66.8	83.1	65.2	70.8	66.0	77.0
	ASOT [56]	71.4	80.9	60.1	74.4	65.8	77.7
	HVQ [46]	73.0	85.5	67.2	77.1	70.1	81.3
	GCTSC [9]	90.8	91.8	82.8	84.8	86.8	88.3
	Our TR^{2}C	**98.6**	**98.5**	**90.2**	**89.7**	**94.4**	**94.1**

5. Conclusion

We have presented a novel framework for the HMS task, called TR^{2}C, which jointly learns structured representations and affinity to segment the frame sequences in video. Specifically, the structured representations learned by TR^{2}C maintain temporally consistency and align well with a UoS structure, which is favorable for the HMS task. We note that our TR^{2}C is an effective and efficient deep subspace clustering framework for HMS, where $\rho^c(\boldsymbol{Z}, \epsilon \mid \boldsymbol{\Pi})$ is for subspace detection, $r(\boldsymbol{Z})$ is a temporal continuity prior, and $-\rho(\boldsymbol{Z}, \epsilon)$ is to prevent representation collapse. As future work, it would be attempting to explore more sophisticated implementation by customizing different components.

Acknowledgments

The authors would like to thank the constructive comments from anonymous reviewers. This work is supported by the National Natural Science Foundation of China under Grants 61876022 and 62076031.

References

[1] Sathyanarayanan N Aakur and Sudeep Sarkar. A perceptual prediction framework for self supervised event segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1197–1206, 2019. 8

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 8, 5

[3] Yue Bai, Lichen Wang, Yunyu Liu, Yu Yin, and Yun Fu. Dual-side auto-encoder for high-dimensional time series segmentation. In *Proceedings of the IEEE International Conference on Data Mining*, pages 918–923, 2020. 2, 5, 6

[4] Yue Bai, Lichen Wang, Yunyu Liu, Yu Yin, Hang Di, and Yun Fu. Human motion segmentation via velocity-sensitive dual-side auto-encoder. *IEEE Transactions on Image Processing*, 32:524–536, 2022. 2, 4, 5, 6

[5] Tianzhe Chu, Shengbang Tong, Tianjiao Ding, Xili Dai, Benjamin David Haeffele, René Vidal, and Yi Ma. Image clustering via the principle of rate reduction in the age of pretrained models. In *International Conference on Learning Representations*, 2024. 2

[6] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26:2292–2300, 2013. 4

[7] Mariella Dimiccoli and Herwig Wendt. Enhancing temporal segmentation by nonlocal self-similarity. In *Proceedings of the IEEE International Conference on Image Processing*, pages 3681–3685, 2019. 2

[8] Mariella Dimiccoli and Herwig Wendt. Learning event representations for temporal segmentation of image sequences by dynamic graph embedding. *IEEE Transactions on Image Processing*, 30:1476–1486, 2020. 2, 5, 6

[9] Mariella Dimiccoli, Lluís Garrido, Guillem Rodriguez-Corominas, and Herwig Wendt. Graph constrained data representation learning for human motion segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1460–1469, 2021. 2, 4, 5, 6, 7, 8

[10] Guodong Ding, Fadime Sener, and Angela Yao. Temporal action segmentation: An analysis of modern techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):1011–1030, 2023. 8

[11] Tianjiao Ding, Shengbang Tong, Kwan Ho Ryan Chan, Xili Dai, Yi Ma, and Benjamin D. Haeffele. Unsupervised manifold linearizing and clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5450–5461, 2023. 2, 4, 6

[12] Zexing Du, Xue Wang, Guoqing Zhou, and Qing Wang. Fast and unsupervised action boundary detection for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2022. 8

[13] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, 2009. 1, 2, 5, 6

[14] Maryam Fazel, Haitham Hindi, and Stephen P Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *American Control Conference*, pages 2156–2162, 2003. 4

[15] Dariu M Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73 (1):82–98, 1999. 1

[16] Behnam Gholami and Vladimir Pavlovic. Probabilistic temporal subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3066–3075, 2017. 2

[17] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (12):2247–2253, 2007. 4, 1

[18] Benjamin D Haeffele, Chong You, and René Vidal. A critique of self-expressive deep subspace clustering. In *International Conference on Learning Representations*, 2021. 4

[19] Dong Huang, Shitong Yao, Yi Wang, and Fernando De La Torre. Sequential max-margin event detectors. In *European Conference on Computer Vision*, pages 410–424, 2014. 4, 1

[20] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3192–3199, 2013. 1

[21] Zhuolin Jiang, Zhe Lin, and Larry Davis. Recognizing human actions by learning and matching shape-motion prototype trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):533–547, 2012. 1, 2, 4

[22] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 780–787, 2014. 8, 5

[23] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12066–12074, 2019. 5

[24] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 5

[25] Chun-Guang Li, Chong You, and René Vidal. Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework. *IEEE Transactions on Image Processing*, 26(6):2988–3001, 2017. 1

[26] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 4

[27] Sheng Li, Kang Li, and Yun Fu. Temporal subspace clustering for human motion segmentation. In *Proceedings of*

the IEEE/CVF International Conference on Computer Vision, pages 4453–4461, 2015. 1, 2, 3, 4, 5, 6

[28] Yuanman Li, Jiantao Zhou, Xianwei Zheng, Jinyu Tian, and Yuan Yan Tang. Robust subspace clustering with independent and piecewise identically distributed noise modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8720–8729, 2019. 5, 6

[29] Zengyi Li, Yubei Chen, Yann LeCun, and Friedrich T Sommer. Neural manifold clustering and embedding. *arXiv preprint arXiv:2201.10000*, 2022. 2

[30] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, pages 663–670, 2010. 1, 2, 5, 6

[31] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1996–2003, 2009. 4, 1

[32] Xin Liu, Zhongdao Wang, Ya-Li Li, and Shengjin Wang. Self-supervised learning via maximum entropy coding. *Advances in Neural Information Processing Systems*, 35: 34091–34105, 2022. 4

[33] Canyi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. In *European Conference on Computer Vision*, pages 347–360, 2012. 1, 2, 5, 6

[34] Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1546–1562, 2007. 3, 4, 7

[35] Xianghan Meng, Zhiyuan Huang, Wei He, Xianbiao Qi, Rong Xiao, and Chun-Guang Li. Exploring a principled framework for deep subspace clustering. In *International Conference on Learning Representations*, 2025. 4

[36] Thomas B Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006. 1

[37] Kevin Patrick Murphy. *Dynamic bayesian networks: representation, inference and learning*. University of California, Berkeley, 2002. 2

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 8

[39] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108 (1-2):4–18, 2007. 1

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021. 2, 5

[41] Michael S Ryoo and Jake K Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1593–1600, 2009. 1, 2, 4

[42] Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. Temporally-weighted hierarchical clustering for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11225–11234, 2021. 8, 6

[43] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 4, 6

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 5

[45] Padhraic Smyth. Probabilistic model-based clustering of multivariate and sequential data. In *Proceedings of the International Workshop on AI and Statistics*, pages 299–304, 1999. 2

[46] Federico Spurio, Emad Bahrami, Gianpiero Francesca, and Juergen Gall. Hierarchical vector quantization for unsupervised action segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6996–7005, 2025. 8, 6

[47] Sebastian Stein and Stephen J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 729–738, 2013. 8, 5

[48] Stephen Tierney, Junbin Gao, and Yi Guo. Subspace clustering for sequential data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1019–1026, 2014. 1, 2, 3, 5, 6

[49] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3551–3558, 2013. 5

[50] Lichen Wang, Zhengming Ding, and Yun Fu. Learning transferable subspace for human motion segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 1, 2, 4, 5, 6

[51] Lichen Wang, Zhengming Ding, and Yun Fu. Low-rank transfer human motion segmentation. *IEEE Transactions on Image Processing*, 28(2):1023–1034, 2018. 1, 2, 4, 5, 6

[52] Peng Wang, Huikang Liu, Druv Pai, Yaodong Yu, Zhihui Zhu, Qing Qu, and Yi Ma. A global geometric analysis of maximal coding rate reduction. In *International Conference on Machine Learning*, 2024. 2, 3

[53] Xiumei Wang, Dingning Guo, and Peitao Cheng. Support structure representation learning for sequential data clustering. *Pattern Recognition*, 122:108326, 2022. 2

[54] Xiao Wang, Jingen Liu, Tao Mei, and Jiebo Luo. Coseg: Cognitively inspired unsupervised generic event segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):12507–12517, 2023. 8

[55] Yimin Xiong and Dit-Yan Yeung. Mixtures of arma models for model-based time series clustering. In *Proceedings of the IEEE International Conference on Data Mining*, pages 717–720, 2002. 2

[56] Ming Xu and Stephen Gould. Temporally consistent unbalanced optimal transport for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14618–14627, 2024. 8, 6

[57] Chong You, Chun-Guang Li, Daniel Robinson, and René Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3928–3937, 2016. 1, 6

[58] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Advances in Neural Information Processing Systems*, 33:9422–9434, 2020. 1, 2, 3, 4

[59] Shangzhi Zhang, Chong You, René Vidal, and Chun-Guang Li. Learning a self-expressive network for subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12393–12403, 2021. 4

[60] Feng Zhou, Fernando De la Torre, and Jeffrey F Cohn. Unsupervised discovery of facial events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2574–2581, 2010. 2

[61] Feng Zhou, Fernando De la Torre, and Jessica K Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):582–596, 2012. 2

[62] Tao Zhou, Huazhu Fu, Chen Gong, Jianbing Shen, Ling Shao, and Fatih Porikli. Multi-mutual consistency induced transfer subspace learning for human motion segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10277–10286, 2020. 1, 2, 4, 5, 6

[63] Tao Zhou, Huazhu Fu, Chen Gong, Ling Shao, Fatih Porikli, Haibin Ling, and Jianbing Shen. Consistency and diversity induced human motion segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):197–210, 2022. 1, 2, 4, 5, 6

[64] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1491–1498, 2006. 5