Unsupervised Hierarchical Dynamic Parsing and Encoding for Action Recognition

Bing Su, Jiahuan Zhou, *Student Member, IEEE*, Xiaoqing Ding, *Life Fellow, IEEE*, and Ying Wu, *Fellow, IEEE*

***Abstract*— Generally, the evolution of an action is not uniform across the video, but exhibits quite complex rhythms and non-stationary dynamics. To model such non-uniform temporal dynamics, in this paper, we describe a novel hierarchical dynamic parsing and encoding method to capture both the locally smooth dynamics and globally drastic dynamic changes. It parses the dynamics of an action into different layers and encodes such multi-layer temporal information into a joint representation for action recognition. At the first layer, the action sequence is parsed in an unsupervised manner into several smooth-changing stages corresponding to different key poses or temporal structures by temporal clustering. The dynamics within each stage are encoded by mean-pooling or rank-pooling. At the second layer, the temporal information of the ordered dynamics extracted from the previous layer is encoded again by rank-pooling to form the overall representation. Extensive experiments on a gesture action data set (Chalearn Gesture) and three generic action data sets (Olympic Sports, Hollywood2, and UCF101) have demonstrated the effectiveness of the proposed method.**

***Index Terms*— Action recognition, temporal clustering, hierarchical modeling, dynamic encoding.**

I. Introduction

APPEARANCES and dynamics are two important components of actions. Effectively encoding these spatial-temporal information into representations is crucial for various action-based applications. Especially, the performance of action recognition methods depends heavily on the representation of video data. For this reason, many recent efforts focus on developing various feature encoding methods and action representations in different levels. The widely-used Bag-of-Visual-Words (BoW) [1] based action representation is able to well encode appearance information of the video. The BoW framework includes three steps: local descriptors extraction, codebook learning, and feature encoding or descriptors pooling. The raw local descriptors themselves are noisy and the discriminative power of the distributed BoW representation comes from the efficient encoding of these local descriptors. Generally, most encoding methods such as BoW and Fisher vector [2], aggregate all local appearance descriptors without considering their temporal positions. As a result, the temporal dependencies and dynamics of the video are seriously neglected.

Dynamics characterize the inherent global temporal dependencies of actions. Existing dynamics-based approaches generally view the video as a sequence of observations and model it with temporal models. The models can either be state-space-based such as Hidden Markov Model (HMM) [3] and Conditional Random Field (CRF) [4] or exemplar-based such as Dynamic Time Warping (DTW) [5]. Such models generally not only require a large amount of training data to exactly estimate parameters, statistics and temporal alignments but also cannot directly lead to vector representations with a fixed dimension. Recently, Fernando *et al.* [6] propose to pool frame-wise features via learning to rank within the BoW framework, which encodes the temporal evolution of appearances in a principled manner and results in a representation with the same dimension of the frame-wise features. The dynamics are considered as the ordering relations of frame-wise features and the changes of all successive frames are treated equally.

The dynamic behind an action is time-varying and not easy to be figuratively expressed. However, for a specifically given action video, the dynamic does have some intuitive rhythms or regularities. One clue is that humans can recognize an action from some ordered key frames. Typically each frame captures a key pose, and the number of key poses is much smaller than the number of frames in the whole video. Taking an example of Fig. 1, a video recording an action "jump" may contain up to hundreds of frames, but only three key poses can represent the drastic changes in the dynamics: running approach, body stay flew in the air and touch down. There may be many similar frames corresponding to each key pose. These key poses segment the whole action into different divisions or stages, and each stage consists of the frames related to a key pose. Therefore, the dynamics of an action can also be viewed as a hierarchy. The dynamics within each stage are relatively stable, and the dynamics of the sequence of the stages or key poses represent the essential

Manuscript received January 11, 2017; revised June 10, 2017 and July 23, 2017; accepted August 13, 2017. Date of publication August 25, 2017; date of current version September 15, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61603373, Grant 61032008, and Grant 61471214, in part by the National Basic Research Program of China (973 program) under Grant 2013CB329403, in part by the National Science Foundation under Grant IIS-1217302 and Grant IIS-1619078, and in part by the Army Research Office under Grant W911NF-16-1-0138. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Lei Zhang. *(Corresponding author: Bing Su.)*

B. Su is with the Science and Technology on Integrated Information System Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China (e-mail: subingats@gmail.com).

J. Zhou and Y. Wu are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: jzt011@eecs.northwestern.edu; yingwu@eecs.northwestern.edu).

X. Ding is with the State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: dingxq@tsinghua.edu.cn).

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

Digital Object Identifier 10.1109/TIP.2017.2745212

evolution of the action. Encoding the dynamics at different levels indiscriminately may bury the essential evolution, and be sensitive to noises and temporal distortions.

In this paper, we incorporate the dynamics in both levels into a joint representation for action recognition. We build an unsupervised hierarchical structure for each action video to parse the dynamic of appearances into different levels and encode them in different layers. In the first layer, we parse the sequence of frame-wise features into different stages and encode the dynamic and appearances into a feature vector within each stage. In the second layer, we extract high-level dynamic encoding representation by rank pooling the encoded features produced in the first layer.

The contributions of this work include: 1) The proposed hierarchical parsing and encoding is a new unsupervised representation learning method. It hierarchically abstracts the prominent dynamic and generates a representation that is robust to speed and local variations and captures the high-level semantic information for a video. 2) We propose an unsupervised temporal clustering method to achieve efficient dynamic parsing. It is built on a single action sequence and no annotations or training are needed to perform parsing. 3) The extracted representations from multi-scale parsings provide complementary discriminative information and hence can be combined.

This paper is an extension of our previous conference paper [7]. The major extensions include: 1) For video-based action video, the sequence of frame-wise features is first modeled by linear dynamic system and the state sequence is used instead of the observation sequence to obtain the parsing segments; 2) The proposed method is experimentally evaluated with deep-learning-based frame-wise features on the large scale UCF101 dataset, and with the Fisher Vector encoding in addition to the BOW encoding on the other datasets; 3) The influences of parameters and the effects of temporal clustering are experimentally evaluated on more datasets with more illustrations; 4) The relationships and comparisons of the proposed method with rank pooling [6] and improved dense trajectories [8] are discussed in detail; 5) "Deeper" hierarchical model is built with more than two layers and the representations from these layers are combined.

The rest of this paper is organized as follows: Section II briefly reviews the existing work on action recognition; Section III presents the proposed hierarchical dynamic parsing and encoding method; We evaluate the proposed method in Section IV and draw conclusions in Section V.

II. Related Work

Since appearance and dynamics are two important aspects of actions, most previous work on action recognition can be accordingly categorized into two groups: appearance-based and dynamic-based. We briefly review the related methods in both categories.

A. Appearance-Based Action Representation Methods

Various features have been proposed to represent the appearances of frames, which can be either hand-crafted or deeply learned. For extracting hand-crafted features, BoW framework is widely used. Different BoW-based methods differ in the local visual descriptors and the coding scheme. HOG [9], [10], HOF [11] and MBH [12] are typical low-level descriptors used in video-based action recognition. These descriptors can be computed either sparsely at local space-time cuboids [13] or by dense sampling scheme [8], [14]. HOG/HOF descriptors are extracted around STIPs in [13]. Several descriptors such as HOG and MBH are fused to encode the densely sampled trajectories in [12], and the dense trajectories are improved to correct camera motion in [8]. Various coding variants have also been proposed to encode these local descriptors, such as local soft assignment [15], sparse coding [16], locality-constrained linear coding [17], super vector coding [18], multi-view super vector [19], super sparse coding with spatial-temporal awareness [20], Fisher vector [2], vector of locally aggregated descriptors [21], [22], and rank pooling [6].

Efforts have also been made to construct hierarchical feature representations based on BoW to capture context information and high-level concepts. Three levels of spatial-temporal context hierarchy are modeled in ascending order of abstraction in [23]. At multiple spatial-temporal scales, the most discriminative class-specific shapes of space-time feature neighborhoods are learned in [24] and the contextual interactions between interest points are encoded to augment local features in [25]. A two-layer nested Fisher vector encoding is stacked in [26].

Another way to represent appearance is to exploit mid-level representation through mining discriminative action parts. For depth videos, the parts can be conjunctive structures [27], [28]. For RGB videos, the parts can be subvolumes [29], [30], salient spatio-temporal structures [31], intermediate concepts related to underlying sub-modalities [32], tight clusters with coherent appearance and motion features [33], spatio-temporal patches [34]. In [35], atomic parts of action called motion atoms are discovered by discriminative clustering, and a temporal composite of motion atoms called motion phrases are mined in a bottom-up manner. The activations of motion features, atoms and phrases are stacked to form the final representation.

Different from hand-crafted features, deep neural networks have been applied to learn representations directly from videos. 3D Convolutional Neural Network (CNN) is proposed to capture the motion information in adjacent frames by performing 3D convolutions in [36]. A thorough evaluation of CNNs on large-scale video classification is presented in [37]. In [38], convolutional features generated by deep architectures are aggregated by trajectory-constrained pooling. In [39], appearance and motion-based CNN features are computed from all the tracks of body joints. In [40], the sequence of convolutional net-based frame-wise features for a video is mapped by multi-layer Long Short Term Memory (LSTM) networks into a fixed length representation, which is decoded by another LSTM to produce a target sequence for unsupervised training. In [41], deep 3D ConvNets (C3D) are trained on large-scale supervised video datasets to learn spatio-temporal features.

Fig. 1. The action "jump" can be roughly parsed into three divisions: running approach, body stay flew in the air and touch down. Each division can also be parsed into different sub-divisions.

B. Dynamic-Based Action Modeling Methods

Both deterministic models and generative models have been studied to model and represent dynamics and motions in action recognition. For deterministic models, the temporal structures or alignments are explicitly modeled. Dynamic time warping (DTW) is used to align action sequences for recognition in [5]. Maximum margin temporal warping is proposed in [42] to learn temporal action alignments and phantom action templates. Latent SVM is employed to learn segment classifiers with the corresponding temporal displacements in [43], the latent hierarchical model in [44], and temporal segmental grammars in [45] for modeling the temporal structures of motion segments or sub-activities. Actom sequence model [46], [47] represents an action by a sequence of pre-defined temporal parts called actoms. Graphs [48], [49] are also used to model temporal structures and relationships among local features. Order-preserving optimal transport is proposed to match action sequences in [50].

Recently deep neural architectures are employed for modeling actions. In [51], spatial and temporal nets are incorporated into a two-stream ConvNet. In [52], salient dynamics of actions are modeled by the differential recurrent neural networks. Factorized spatio-temporal convolutional networks ($F_{ST}CV$) proposed in [53] factorizes the 3D convolution kernel learning as a cascade of 2D spatial kernel learning and 1D temporal kernel learning, and multiple clips sampling strategy is used to handle alignment. In [54], temporal segment network (TSN) models the dynamics throughout the entire video by applying two-stream ConvNets on a sequence of sparsely sampled short snippets.

Generative models are typically based on temporal (hidden) state-space, such as HMM [3], [55]–[57], coupled HMM [58], semi-Markov model that incorporates prior knowledge on state duration [59], CRF [4], HCRF [60], dynamic Bayes nets [61], temporal AND-OR graph [62], and linear dynamic systems [63]. Hierarchical sequence summarization is achieved in [64] by recursively learning hidden spatio-temporal dynamics based on latent variables of CRFs and grouping observations with similar latent states. The hierarchical combinatorial structures of cross-view actions are represented by a compositional multi-view AND-OR model in [65] via explicitly modeling the geometry, appearance and motion variations.

The proposed HDPE method incorporates both appearances and dynamics. The input video is represented by a sequence of appearance features, which is further parsed into different stages. The local appearances within each stage and the dynamics at different levels of the hierarchy are encoded into a compact representation in a bottom-up manner. The parsing is achieved by a novel temporal clustering algorithm. We also briefly discuss the differences with some other methods for parsing a sequence into segments or clusters.

C. Temporal Clustering for Sequence Parsing

Aligned cluster analysis [66] divides a sequence by minimizing the similarities among the segments, where the similarity between two segments is measured by a dynamic time alignment kernel. As the dynamics of each segment may not be stable, the segments do not correspond to stable action stages. In contrast, our method divides a sequence into segments by minimizing the within-segment variances so that the frames within each segment are similar. As each segment shows a stable dynamic, it can be viewed as a stage of an action. In MMTC [67], features in sequences are clustered into several common clusters, and a multi-class SVM is trained to assign clusters using all training sequences. Our method acts on each individual sequence independently and no training is needed, and the segments from different sequences are different and only account for the evolution of the specific sequence.

III. Hierarchical Dynamic Parsing and Encoding

Video-wise temporal evolution modeling method proposed in [6] aggregates the frame-wise features into a functional representation via a ranking machine. This representation captures the evolution of appearances over frames and hence provides the video-wise temporal information. However, the ranking function within the learning to rank machine attempts to rank all the frames in the video and these frames are equally treated, which ignores the non-stationary evolution of dynamic within different stages and cannot directly exploit the complex hierarchical temporal structures. Hierarchical architecture has the ability to learn a higher-level semantic representation by pooling local features in the lower layer and refining the features from the lower layer to the higher layer. In this section, we propose a hierarchical temporal evolution modeling method, namely *Hierarchical Dynamic Parsing and Encoding or HDPE*, to take the "rhythmic" of stage-varying dynamics into account. The pipeline of HDPE is shown in Fig. 2.

A. Unsupervised Temporal Clustering

In order to capture the temporal structures corresponding to relatively-uniform local dynamics, we first propose an

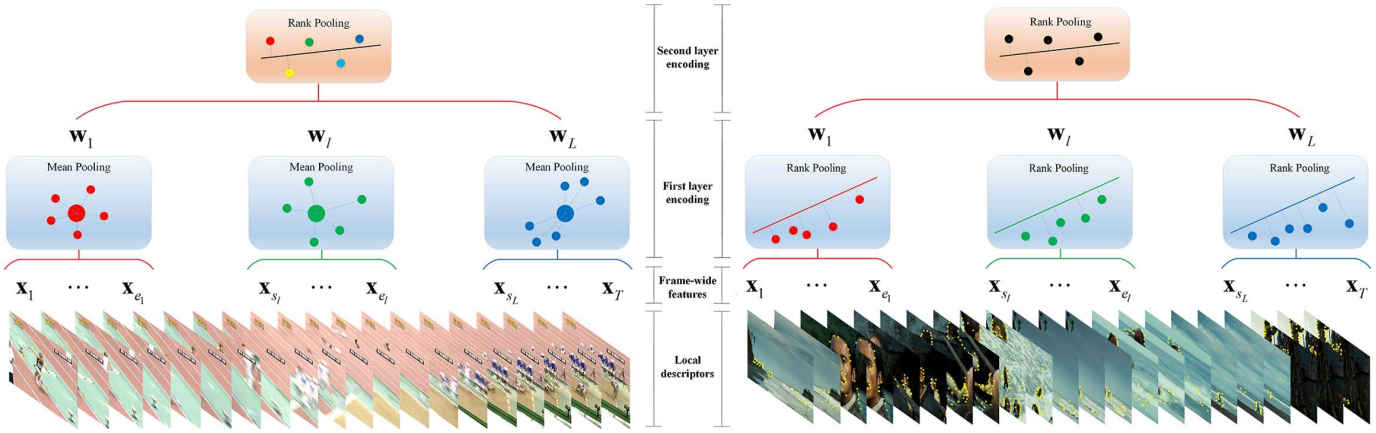


Fig. 2. The pipeline of the proposed method. The first layer can either adopt mean pooling (left) or rank pooling (right).

unsupervised temporal clustering method that learns the parse of an action sequence only from the sequence itself.

For each action video, we extract a feature vector from each frame. Thus the action video can be represented as a sequence of such features. We denote the video by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$, where $\mathbf{x}_t$ is the feature vector extract from the t-th frame, and T is the number of frames in the whole video. We denote the partition of $\mathbf{X}$ by a segmentation path $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_L]$, where L is the number of divisions, typically $L < T$. $\mathbf{p}_t = [s_t, e_t]^T$ provides the range $\{s_t, s_t + 1, \cdots, e_t\}$ of the t-th division, s_t and e_t are the start and end indexes of the frames in this division. The number of frames divided into the t-th division is $l_t = e_t - s_t + 1$. We hope that each division contains a set of steady evolving frames corresponding to the same key pose or temporal structure. We require $\mathbf{P}$ being a non-overlapping and completing partition that covers the whole video. The term "non-overlap" means no frame can be simultaneously divided into two divisions, the term "complete" means that every frame in the sequence must be divided into one and only one division, hence the elements of $\mathbf{P}$ satisfy the following constraints: $s_1 = 1, e_L = T$, $s_{t+1} = e_t + 1, \forall t = 1, \cdots, L-1$, $e_t \geq s_t, \forall t = 1, \cdots, L$. There may be noisy or outlier frame in the sequence, which is significantly different with its successive neighbor frames. To avoid assigning such outlier frame into a separate division and prevent extremely unbalance divisions, we make the restriction on the number of elements in each division. Specifically, we limit the maximum number of elements within one division by $l_m = f \cdot l_{ave}$, where f is the band factor, and $l_{ave} = \frac{T}{L}$ is the average number of elements in each division by uniform segmentation.

Given a partition $\mathbf{P}$, we define a sequence $\mathbf{U} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_L]$, where $\boldsymbol{\mu}_j$ is the mean of frame-wise features of the frames in the j-th division. $\mathbf{U}$ only contains the key atoms that reflect the basic and drastic evolutions of $\mathbf{X}$. Each atom can be viewed as relating to a key pose and is an inevitable stage of the action. Therefore, we call $\mathbf{U}$ the essential sequence of $\mathbf{X}$. Once $\mathbf{U}$ is given, the partition $\mathbf{P}$ can be obtained by computing the optimal alignment path along which the sum of distances between the aligned elements in $\mathbf{X}$ and the warped $\mathbf{U}$ is minimal among all possible paths:

$$\min_{\mathbf{P}} \sum_{j=1}^{L} \sum_{i=s_j}^{e_j} \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|_2^2 \tag{1}$$

Consider a partial path that assigns the first i-th elements in $\mathbf{X}$ to the first j-th elements in $\mathbf{U}$, and the last l elements of the first i-th elements in $\mathbf{X}$ are assigned to the j-th element of $\mathbf{U}$. We denote the sum of element-wise distances along this partial path by the partial distance $d(i, j, l)$. The minimal partial distance can be determined recurrently:

$$d(i,j,l) = \begin{cases} \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|_2^2, l = 1, i = j = 1 \\ \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|_2^2 + \min\limits_{k=1}^{l_m} d(i-1, j-1, k), l = 1 \\ \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|_2^2 + d(i-1, j, l-1), l \leq l_m \\ Inf, otherwise \end{cases} \tag{2}$$

Eq. (2) does not have aftereffect, hence Eq. (2) can be effectively solved by dynamic programming. When both partial sequences reach the end, the minimal distance along the optimal path is determined by $\min\limits_{l=1}^{l_m} d(T, L, l)$ and the optimal partition path $\mathbf{P}$ can be obtained by back tracking.

Given the partition $\mathbf{P}$ of the sequence $\mathbf{X}$, the essential sequence $\mathbf{U}$ can be obtained by computing the mean of each division. The essential sequence in turn can be used to parse the sequence $\mathbf{X}$ into different divisions. Determining the essential sequence $\mathbf{U}$ and computing the partition $\mathbf{P}$ rely on each other. We develop an unsupervised temporal clustering method to jointly mine temporal structures in the sequence $\mathbf{X}$ and learn the partition $\mathbf{P}$ that parses $\mathbf{X}$ into stages with respect to these temporal structures.

We first initialize the partition $\mathbf{P}$ to be a uniform partition that divides the sequence $\mathbf{X}$ into L equal segments. For example, if $L = 3, T = 9$, i.e. we divide a sequence $\mathbf{X}$ with 9 elements into 3 segments, the initial partition $\mathbf{P} = [[1, 3]^T, [4, 6]^T, [7, 9]^T]$. Then we compute the essential sequence $\mathbf{U} = [\mu_1, \mu_2, \cdots, \mu_L]$, whose elements are the

Algorithm 1 Unsupervised Action Parsing by Temporal Clustering

Input: a sequence $\mathbf{X}$, the number of divisions L, the maximal number of iterations Ite, the band factor f;
Output: the partition $\mathbf{P}$ of $\mathbf{X}$;
Initialize the partition path $\mathbf{P}$ to be a uniform partition;
while $\mathbf{P}$ has not converged and the number of iterations is less than Ite **do**
  Compute the essential sequence $\mathbf{U}$ using (3);
  Update the partition path $\mathbf{P}$ by solving the dynamic programming problem (2) with the band factor f;
end while

means of elements in the corresponding divisions:

$$\boldsymbol{\mu}_j = \frac{1}{l_j} \sum_{k=s_j}^{e_j} \mathbf{x}_k, \, j = 1, \cdots, L \tag{3}$$

After that, we update the partition $\mathbf{P}$ by aligning the elements in $\mathbf{X}$ to those in $\mathbf{U}$ to parse $\mathbf{X}$ using the dynamic programming algorithm. The essential sequence $\mathbf{U}$ is recomputed in turn with the updated $\mathbf{P}$. The two procedures are continued until the partition is unchanged with the previous iteration or a pre-fixed number of iterations is reached. We summarize the joint partition learning and temporal clustering algorithm in Alg. 1.

1) Convergency: Given $\mathbf{P}$, computing the essential sequence $\mathbf{U}$ by using Eq. (3) is equivalent to the solution of minimum mean square error problem: $\min_{\boldsymbol{\mu}} \sum_{i=s_j}^{e_j} \|\mathbf{x}_i - \boldsymbol{\mu}\|_2^2, \, j = 1, \cdots, L$. Given $\mathbf{U}$, computing $\mathbf{P}$ directly minimizes Eq. (1). Both procedures reduce the objective of Eq. (1). Eq. (1) has a trivial lower bound $\sum_{j=1}^{L} \sum_{i=s_j}^{e_j} \left\|\mathbf{x}_i - \boldsymbol{\mu}_j\right\|_2^2 \geq 0, \forall \mathbf{P}, \mathbf{U}$. Thus the partition learning algorithm will at least converge to a local minimum.

2) Computational Complexity: The complexities of dynamic programming Eq. (2) and calculating Eq. (3) are $O(LNd)$ and $O(Nd)$, L, N and d are the number of segments, the length of the input sequence and the dimension of the frame-wise features. Similar with k-means clustering, in practice, temporal clustering also converges fast and the improvement on the results is small after the first dozen iterations, especially when the sequence does show a segmentation structure. We fix the maximum number of iterations to 30 in our experiments. Hence the complexity of the temporal clustering Alg. 1 is $O(LNd)$. As the method processes each sequence separately, parallel speedup can be easily performed.

B. State Sequence Extraction Via Linear Dynamic Systems

For video-based actions, the state-of-the-art features extracted from each frame are generally very high-dimensional and contain redundant and noisy information. For example, for the improved dense trajectories-based feature [8] discussed in Sec. II, which is perhaps the most widely used hand craft features for action videos, when encoding the MBH descriptors, the dimensionality of frame-wise features is 49 152 if the Fisher Vector encoding method is used with 256 number of Gaussians and a compression factor of 0.5 for PCA. As analyzed in Sec. III-A, the computation complexity of the temporal clustering algorithm 1 is linearly proportional to the dimensionality of the frame-wise features d. Directly parsing the sequence of such high-dimensional features introduces a heavy computation overhead. Moreover, the temporal clustering algorithm 1 relies on the Euclidean distance between frame-wise features, but the distance measure may be meaningless in such high-dimensional space.

A straightforward way to handle such high-dimensional data is to perform dimensionality reduction. However, supervised dimensionality reduction methods such as [55] and [68] not only introduce supervision information but also need large amounts of training samples and high space and time complexities to train the transformation. Even for the simple unsupervised PCA, a covariance matrix of huge size need to be calculated and processed to obtain the projection, the temporal dependencies of frame-wise features are totally lost and all the training sequence samples need to be available for training.

The goal of temporal clustering is to segment a sequence into several dynamic-coherent partitions. It is desired that the dynamics of human motions are gradual within each partition. However, for RGB video data, it is difficult to detect and represent human motion dynamics in clustered background. What can be observed are the hand-crafted or learned features representing the global appearances of frames. The essential human poses or motions cannot be directly represented, but can only be considered as latent states. The frame-wise features can then be viewed as the observations generated by the corresponding states. The viewpoint changes and the clustered background can be viewed as the observation noises introduced in the generation process. Therefore, it is natural to model the temporal dynamics of a feature sequence by the linear dynamic system (LDS). It has been shown in [63] that the motion dynamics of a feature sequence can be represented by the state trajectory sequence with only a few dimensions per frame, where the state sequence can be obtained from the sequence itself by a LDS.

For an action video, the evolutions of its frame-wise features can be modeled by an LDS as follows:

$$\begin{cases} \mathbf{s}_{t+1} = \mathbf{A}\mathbf{s}_t + \gamma_t \\ \mathbf{x}_t = \mathbf{B}\mathbf{s}_t + \eta_t \end{cases} \tag{4}$$

where the sequence $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$ of frame-wise features is also called the observation sequence, and $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_T]$ is the latent state sequence. $\mathbf{s}_t \in \mathbb{R}^{d'}$ is the state or latent variable corresponding to the frame-wise observation $\mathbf{x}_t \in \mathbb{R}^d$ at frame t. d is the dimensionality of the frame-wise features, and d' is the dimensionality of the frame-wise states. The state sequence is modeled as a first-order Markov process, that is, the next state $\mathbf{s}_{t+1}$ is determined by the current state $\mathbf{s}_t$, and the current observation $\mathbf{x}_t$ is determined by the current state $\mathbf{s}_t$. $\gamma_t \sim N(0, \Sigma_\gamma)$ and $\eta_t \sim N(0, \Sigma_\eta)$ are the system noise and the observation

noise, respectively, which are modeled by two zero-mean i.i.d. Gaussian processes. Σ_γ and Σ_η are the co-variances of the corresponding Gaussian distributions, respectively.

Given the observation sequence $\mathbf{X} \in \mathbb{R}^{d\times T}$, the parameters in Eq. (4) and the state sequence $\mathbf{S} \in \mathbb{R}^{d'\times T}$ have closed-form least squares estimations [69]. The singular value decomposition (SVD) is first applied to $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\Lambda\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{d\times d}$ and $\mathbf{V} \in \mathbb{R}^{T\times T}$ are orthogonal matrices, and $\Lambda \in \mathbb{R}^{d\times T}$ is a rectangular diagonal matrix. The state sequence can be estimated as:

$$\mathbf{S} = \tilde{\Lambda}\mathbf{V}^T \tag{5}$$

where $\tilde{\Lambda} \in \mathbb{R}^{d'\times T}$ is the truncated rectangular diagonal matrix that only preserves the rows of Λ with respect to the d' largest diagonal values. The other parameters can be estimated as:

$$\mathbf{B} = \mathbf{U}, \mathbf{A} = \mathbf{S}_{2:T}\mathbf{S}^{+}_{1:T-1}$$

where $+$ is the Moore-Penrose inverse. $\mathbf{A}$ is in fact the least squares estimation from: $\mathbf{A} = \arg\min_{\mathbf{A}} \|\mathbf{A}\mathbf{S}_{1:T-1} - \mathbf{S}_{2:T}\|_F^2$. Σ_γ and Σ_η can then be estimated from the residuals.

$\mathbf{A}$ and $\mathbf{B}$ can be viewed as the system dynamic matrix that controls the transition from the current state to the next state and the appearance mapping matrix that maps the latent state to the output observation, respectively. In this way, LDS decouples the sequence into latent dynamics and appearance observations, and the state sequence $\mathbf{S}$ thus reveals the basic dynamic evolution of the sequence. The intrinsic dimensionality d' of the state space is usually quite small. In [63], d' is set to 3, and $\mathbf{S}$ can still reflect the basic dynamic behavior. In this paper, we set d' to 15 or 30 in our experiments. Performing temporal clustering on $\mathbf{S}$ instead of $\mathbf{X}$ with thousands of dimensions not only generally produces more accurate parsing segments because the disruption of the large appearance variations and noises is weakened, but also leads to a significant speedup since the complexity of the temporal clustering increases linearly with d'.

C. The First Layer Modeling

For an action sequence sample $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$, we first parse it into L divisions using Alg. 1. We denote the parsing result of $\mathbf{X}$ by $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_L]$. The evolution within each division is relatively steady and hence the frames in each division can be equally treated. An abstract feature vector can be extracted from each division via mean pooling or rank pooling [6].

Mean pooling simply uses the mean of the frame-wise features as the output of the division. For the l-th division, we denote the segmentation fragment as $\mathbf{X}^{[l]} = [\mathbf{x}_{s_l}, \mathbf{x}_{s_l+1}, \cdots, \mathbf{x}_{e_l}]$. The mean pooling result of the division can be calculated as:

$$\mathbf{w}_l = \frac{1}{e_l - s_l + 1}\sum_{\tau=0}^{e_l-s_l} \mathbf{x}_{s_l+\tau}$$

Rank pooling learns a linear ranking function to order the frame-wise features in each division via learning to rank and uses the parameters of the function as the representation of the temporal structure associated with the division. Since video frames usually exhibit large variations, the low-level and noisy hand-crafted features in the same video sequence can vary significantly, and abrupt changes or distortions often occur. Therefore, the connection between the features and the frame index is quite weak, and hence the learned ranking functions may be disturbed by the undesirable abrupt variations or distortions. To smooth hand-crafted features and enhance their correlations along the evolution, a vector valued function that transforms each element $\mathbf{x}_{s_l+t}$ to the corresponding time varying mean vector $\mathbf{v}_{s_l+t} = \frac{\mathbf{u}_{s_l+t}}{\|\mathbf{u}_{s_l+t}\|}$, where $\mathbf{u}_{s_l+t} = \frac{1}{t+1}\sum_{\tau=0}^{t}\mathbf{x}_{s_l+\tau}$, is first applied to $\mathbf{X}^{[l]}$, resulting in $\mathbf{V}^{[l]} = [\mathbf{v}_{s_l}, \mathbf{v}_{s_l+1}, \cdots, \mathbf{v}_{e_l}]$.

The high-level deep-learning-based frame-wise features such as C3D [41] are extracted from local spatial-temporal volumes and hence have already been smoothed. The dependency between such high-level features and the frame index can be directly learned by a ranking function. Therefore, for such features, we apply rank pooling directly on the normalized independent representations $\mathbf{v}_{s_l+t} = \frac{\mathbf{x}_{s_l+t}}{\|\mathbf{x}_{s_l+t}\|}$.

A linear function $f(\mathbf{w}_l; \mathbf{v}) = \mathbf{w}_l^T \cdot \mathbf{v}$ is used to predict the ranking score for each $\mathbf{v}_{s_l+t}$. The parameters $\mathbf{w}_l$ of the linear function is learned to rank the orders of the elements in the division, such that $f(\mathbf{w}_l; \mathbf{v}_{s_l}) > f(\mathbf{w}_l; \mathbf{v}_{s_l+1}) > \cdots > f(\mathbf{w}_l; \mathbf{v}_{e_l})$.

$$\begin{aligned}\arg\min_{\mathbf{w}_l} &\frac{1}{2}\|\mathbf{w}_l\|^2 + C\sum_{0\le a<b\le e_l-s_l}\varepsilon_{ab}\\ s.t.\ &\mathbf{w}_l^T\cdot(\mathbf{v}_{s_l+a} - \mathbf{v}_{s_l+b}) \ge 1-\varepsilon_{ab},\\ &\varepsilon_{ab}\ge 0, \forall 0\le a<b\le e_l-s_l\end{aligned} \tag{6}$$

$\mathbf{w}_l$ is used as the representation of the l-th temporal structure. After the first layer modeling, the original sequence $\mathbf{X}$ is mapped to the sequence of key temporal structures $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_L]$, which contains high-level abstract information based on the original representation.

For simple actions and fine-grained actions, compared with the dynamic of divisions, the dynamic within each division is quite uniform and contributes little to the discrimination of the whole actions. Changing the orders of frames in a division does not influence the understanding of the action. Mean pooling is suitable for such cases, which is equivalent to extracting key frames. The key frames are more robust to individual frames and local distortions since each key frame is the mean of a division. For complex activities, the dynamics in divisions may be complex so that the orders of frames in each division cannot be changed, and hence it is better to apply rank pooling.

D. The Second Layer Modeling

The output sequence $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_L]$ from the first layer reflects the essential temporal evolution of the sequence, which can be thought as the sequence of key poses, each pose is a pooling of the frames in the corresponding stage and captures the stage-wide temporal evolution. The second layer extracts the video-wise temporal evolution from these ordered

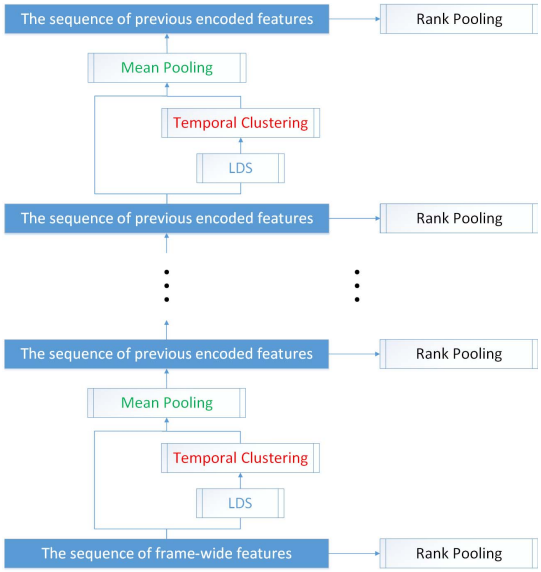


Fig. 3. The multi-layer HDPE model.

stage-wide temporal evolutions. The learning-to-rank modeling used in each division of the first layer is applied to $\mathbf{W}$. A ranking function $f(\mathbf{y}; \mathbf{w}') = \mathbf{y}^T \cdot \mathbf{w}'$ that aims at providing the orders of the time varying mean vectors $\mathbf{w}_1', \mathbf{w}_2', \cdots, \mathbf{w}_L'$ by applying vector valued function to elements of $\mathbf{W}$ such that $f(\mathbf{y}; \mathbf{w}_l') > f(\mathbf{y}; \mathbf{w}_k'), \forall 1 \leq k < l \leq L$. The parameter vector $\mathbf{y}$ of $f(\mathbf{y}; \mathbf{w}')$ serves as the final representation of the video sequence $\mathbf{X}$.

Several advantages of the proposed HDPE method are as follows. First, the method is totally unsupervised, simple and easy to perform. The parsing, the state sequence extraction and the hierarchical encoding are all built on a single action sequence. No annotations are needed to perform parsing or encoding, and no labels or negative data are needed for training. Second, the method is robust to local distortions and individual outliers or noisy frames. The abstract feature produced by the first layer for each division is a pooling of all the frame-wise features in the division, and few outliers or distortions have little effect on the pooling result. Third, the learned representation implicitly combines local appearances and global dynamic in a principled hierarchical manner. The orders within the parsed divisions are not so important, hence the pooling of the first layer focuses on capturing the locally averaged appearances. The temporal orders among the divisions are crucial and reflect the inherent dynamic of the video. The encoding of the second layer focuses on capturing such global high-level dynamic.

E. Stacking More Layers

In the aforementioned sections, we have constructed a two-layer hierarchy. Further, our method can be easily generalized to more layers as shown in Fig. 3. The input action video is firstly represented by a sequence of frame-wise features, which serves as the input sequence of the first layer. Within the layer, the state sequence is extracted from the input sequence and temporal clustering is performed to the state sequence to parse it into segments. In the next, the frame-wise features within each segment are readily encoded into a vector by mean pooling or rank pooling operation, and all the encoded feature vectors of all the ordered segments are gathered together to form a new sequence representation, which is the output of this layer and also serves as the input sequence of the next layer. The above parsing and encoding process can be repeated to build a multi-layer HDPE model. At last, a vector representation can be extracted by rank pooling from the encoded output sequence of any layer. In order to obtain the final HDPE representation of the video, either the rank pooled vector of the last layer or the combination of vectors of all the layers can be adopted. The length of the encoded sequence of a layer is shorter than the input sequence of this layer. Therefore, for a video with T frames, at most $T - 1$ layers can be constructed.

IV. Experiments

In this section we evaluate the performance of the proposed method on one gesture recognition dataset, i.e. the Chalearn gesture dataset, and three challenging generic action recognition datasets, including the Olympic Sports dataset, the Hollywood2 dataset and the UCF101 dataset.

A. Datasets

1) ChaLearn Gesture Recognition Dataset [70], [71]: This dataset consists of Kinect video data from 20 Italian gestures. There are 955 videos in total, and each video contains 8 to 20 continuous gestures. The recordings and annotations include RGB, depth, foreground segmentation and Kinect skeletons. The dataset is split into training, validation and test sets. Following [72], [73], and [6], since we focus on individual gesture action recognition, we perform experiments on the segmented video clips each contain one gesture instance using the ground truth segments, and report the multi-class (the mean over all classes) precision, recall and F-score measures on the validation set.

2) Olympic Sports Dataset [43]: This dataset contains 783 video sequences from 16 sports actions. The videos are collected from YouTube and annotated using Amazon Mechanical Turk. The dataset is split into training and test sets. The training set includes 649 video sequences and the test set includes the remaining 134 video sequences. We report the mean average precision over all classes (mAP) as in [8] and the accuracy as in [42].

3) Hollywood2 Dataset [9]: This dataset contains RGB-video data from 12 generic action classes. There are in total 1,707 video clips in the dataset, which are collected from 69 different Hollywood movies. The dataset is split into training and test sets. The training set includes 823 videos and the test set includes the remaining 884 videos. The videos in the two sets are selected from different movies. We report mAP as in [9] and [8].

4) UCF101 Dataset [74]: This dataset contains 13 320 video clips from 101 realistic action classes. The videos are collected from YouTube. The dataset is challenging for the diversity in actions and variations in viewpoint, camera motion, scale, illumination, cluttered background, etc. The dataset provides three training/testing splits, and we report the average accuracy over the splits as in [74].

B. Experimental Setup

1) Frame-Wise Features: For each action video, we extract a high-dimensional feature vector from each frame and represent the video by a sequence of frame-wise features. For the Olympic Sports dataset and the Hollywood2 dataset, we use the improved dense trajectories descriptors [8], which have achieved state-of-the-art results. We extract trajectory, HOG, HOF and MBH descriptors from the trajectories corresponding to a dense regular grid for all frames. We follow the same settings with [8] when extracting descriptors and the dimensionalities of the trajectory, HOG, HOF and MBH descriptors are 30, 96, 108 and 192, respectively. The square-root trick is applied on these descriptors except trajectory descriptors.

We use two methods to aggregate these descriptors: bag-of-visual-words (BoW) encoding and fisher vector (FV) encoding. For BoW, we learn a codebook with a size of (4 000) for each type of descriptors by k-means clustering as in [8] and quantize the descriptors to their nearest visual words in the codebook. The histogram of the quantized descriptors in one frame is used as the frame-wise feature of the frame. Hence the dimensionality of the frame-wise features is $4,000$. For FV, we first reduce the dimensionality of each type of descriptors by a factor of two using PCA. We then train a Gaussian Mixture Model with $K = 256$ Gaussian components for each type of descriptors. The dimensionality of the frame-wise features by FV is $d = 2KD$ for a type of descriptors, where D is the reduced dimension of the descriptors after PCA.

For the Chalearn Gesture recognition dataset, we employ the skeleton features provided by Fernando *et al.* [6]. The normalized relative locations of body joints w.r.t the torso joints are calculated and clustered into a codebook with a size of 100. The histogram of the quantized relative locations in one frame is employed as the frame-wise feature with a dimensionality of 100.

For the UCF101 dataset, we extract the deep-learning-based features by using the 3D convolutional (C3D) network [41]. For each video, a 3D-window with a width of 16 frames is sliding along the frame axis with a movement of 2 frames each time, resulting in a sequence of clips. The C3D network pre-trained on the Sports-1M dataset [37] is applied to these clips to extract 4096-dimensional feature vectors. Each video is then represented by a sequence of C3D features.

2) Implementation Details: The order in Eq. (6) can also be inverse, i.e., the rank value computed from the linear function of the previous frame is forced to be smaller than that of the current frame. If the first layer adopts rank pooling, the second layer encodes the results of the first layer with the same order and combines them together. If the first layer adopts mean-pooling, the second layer encodes the results of the first layer in both forward and inverse orders and combines them together. Following [6], we also use the SVR solver of liblinear [75] to solve Eq. (6) and fix the value of C to 1.

On the ChaLearn dataset and the Hollywood2 dataset, when using the BoW encoding, we apply chi-squared kernel map on each time varying mean vector in the second layer and apply the L_2 normalization on the output representation of the second layer. The average kernel strategy is adopted to fuse the representations generated from different types of descriptors. When using the FV encoding, we apply the square-root trick on each time varying mean vector in the second layer. The representations of different descriptors are concatenated and we apply L_2-normalization to the final representation. On the UCF101 dataset, we apply chi-squared kernel map followed by the L_2 normalization to the final representation. For all datasets, we train linear SVMs for classification, where C is fixed to 100.

TABLE I

COMPARISON OF PERFORMANCES USING THE TWO POOLING METHODS ON THE CHALEARN GESTURE DATASET

Pooling Method	Precision	Recall	F-score
M-HDPE	**78.34**	**78.18**	**78.15**
R-HDPE	75.95	75.83	75.79

C. Comparison of Pooling in the First Layer

In the first layer modeling, the encoding of each division could either be mean pooling or rank pooling as mentioned in III-C. We compare the two pooling methods on the Chalearn Gesture dataset, the Olympic Sports dataset, the Hollywood2 dataset, and the UCF101 dataset, in Tab. I to Tab. VI, respectively. Both the BoW-based and FV-based frame-wise features are evaluated on the Olympic Sports dataset and the Hollywood2 dataset. M-HDPE and R-HDPE denote that the mean pooling and the rank pooling are used in the first layer modeling in HDPE, respectively.

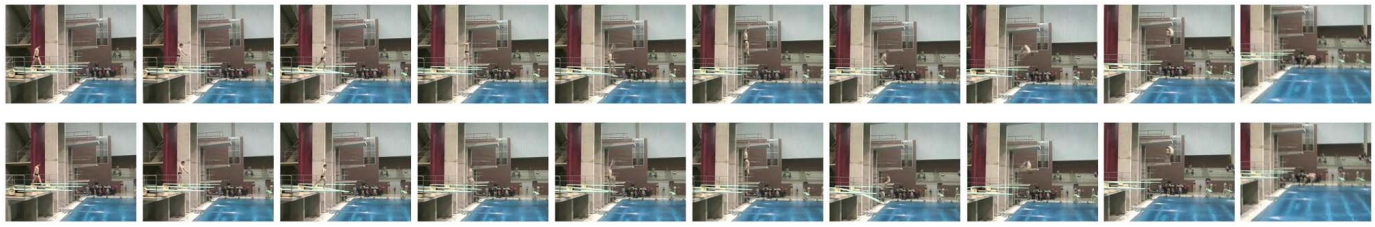
Generally, the mean pooling outperforms the rank pooling on the ChaLearn gesture dataset, the Olympic Sports dataset and the UCF101 dataset, while the rank pooling achieves better results on the Hollywood2 dataset. This verifies the explanation in III-C. That is, for fine-grained actions such as gestures, since the evolution within each division is quite uniform, the within-division dynamic can be ignored, and the local appearance information is enhanced by mean-pooling. In the Olympic Sports dataset and the UCF101 dataset, each video only contains a single action performed by a single subject in the same scene. There is no shot cut or change between sub-lenses. As shown in Fig. 4(a), such one-fold actions are relatively simple, and the divisions parsed from such short actions correspond to quite uniform stages of the action. Thus mean pooling within divisions performs better.

However, as shown in Fig. 4(b), the video in the Hollywood2 dataset generally contains more than one scenes and actions. There are usually multiple subjects performing different motions, or performing the same action in different time intervals or scenes. The viewpoint often switches several times among different sub-lenses or storyboards. Although a subject action exists, it is often accompanied by many erratic motions. Such action is made up of several short components and hence is more complex. As can be observed, many divisions parsed from such complex action actually correspond to different scenes, viewpoints and sub-lenses. The comparatively complete motion in each division also shows unsteady and complex dynamics. The dynamics within divisions convey important discriminative information of the action and hence should not be eliminated.

TABLE II

COMPARISON OF MAPS USING THE TWO POOLING METHODS ON THE OLYMPIC SPORTS DATASET WITH THE BOW-BASED FRAME-WISE FEATURES

Pooling Method	$L = 5$	$L = 10$	$L = 15$	$L = 20$	$L = 25$	$L = 30$	$L = 35$	$L = 40$	$L = 45$	$L = 50$
M-HDPE	**88.20**	**87.82**	**88.87**	**89.12**	**89.18**	**88.07**	**88.74**	**88.30**	**88.58**	**88.63**
R-HDPE	87.45	87.64	87.26	85.35	85.60	85.51	83.89	83.62	85.54	83.11



(a)

(b)

Fig. 4. The ending frames of the parsed divisions on the example video from (a) the Olympic Sports dataset and (b) the Hollywood2 dataset. The trajectory length for the IDT features is set to 15, and hence the ending frame of the last division is a few frames ahead of the last frame of the video. Temporal clustering is applied (Top) without and (Bottom) with the LDS modeling.

TABLE III

COMPARISON OF MAPS USING THE TWO POOLING METHODS ON THE OLYMPIC SPORTS DATASET WITH THE FV-BASED FRAME-WISE FEATURES

Pooling Method	$L = 10$	$L = 20$	$L = 30$	$L = 40$	$L = 50$
M-HDPE	89.69	**90.37**	**89.75**	**90.37**	**89.44**
R-HDPE	**90.28**	88.47	88.49	86.47	84.28

TABLE IV

COMPARISON OF MAPS USING THE TWO POOLING METHODS ON THE HOLLYWOOD2 DATASET WITH THE BOW-BASED FRAME-WISE FEATURES

Pooling Method	$L = 10$	$L = 20$	$L = 30$	$L = 40$	$L = 50$
M-HDPE	59.07	60.99	61.68	61.36	61.72
R-HDPE	**64.93**	**66.54**	**66.16**	**64.25**	**61.75**

TABLE V

COMPARISON OF MAPS USING THE TWO POOLING METHODS ON THE HOLLYWOOD2 DATASET WITH THE FV-BASED FRAME-WISE FEATURES

Pooling Method	$L = 10$	$L = 20$	$L = 30$	$L = 40$	$L = 50$
M-HDPE	66.92	68.30	**69.12**	**68.75**	**68.94**
R-HDPE	**67.43**	**69.22**	67.29	66.07	64.27

It can also be observed that for video-based generic actions, especially when the FV-based frame-wise features are used, the rank pooling outperforms the mean pooling when the number of divisions is small, while the mean pooling works better when more divisions are parsed. The more the parsed divisions, the finer each division is, and the more stable the dynamic within each division. Conversely, if few divisions are parsed from an action, each division is longer and the dynamic within it should be more complex. Rank pooling should then be used to capture such dynamic within each division.

TABLE VI

COMPARISON OF ACCURACIES USING THE TWO POOLING METHODS WITH AND WITHOUT LDS ON THE UCF101 DATASET WITH THE C3D FRAME-WISE FEATURES

Pooling Method	$L = 5$	$L = 10$	$L = 15$	$L = 20$	$L = 25$
M-HDPE	83.44	**83.53**	**83.58**	83.34	83.20
M-HDPE + LDS	**83.46**	**83.53**	**83.58**	**83.36**	**83.23**
R-HDPE	**81.98**	**82.17**	82.52	82.58	**83.02**
R-HDPE + LDS	81.97	82.12	**82.56**	**82.75**	82.98

In the following experiments, we use M-HDPE on the Chalearn gesture dataset, the Olympic Sports dataset and the UCF101 dataset, and adopt R-HDPE on the Hollywood2 dataset, unless otherwise specified.

D. Effects of LDS

We visualize the parsing results with and without LDS modeling of two sample videos from the Olympic Sports dataset and the Hollywood2 dataset in Fig. 4. The number of division L is set to 10, and the dimension of the LDS states is set to 30. In Fig. 4(a) and Fig. 4(b), the top line shows the ending frames of the parsed divisions by temporal clustering on the original sequence of the IDT-based frame-wise features, where MBH descriptors are aggregated by BoW encoding. The bottom line shows the results on the LDS state sequence. In the two examples, we can see that the parsed divisions are nearly the same whether or not the LDS modeling is used. However, the times for performing temporal clustering directly on the observation sequence and applying LDS modeling followed by temporal clustering on the state sequence are 3.4694 and 1.2207 for the sample from the Olympic Sports

TABLE VII

COMPARISON OF PERFORMANCES WITH UNIFORM PARSING AND DYNAMIC PARSING ON THE CHALEARN DATASET

Alignment Method	Precision	Recall	F-score
M-HDPE + uniform parsing	77.85	77.68	77.60
M-HDPE	**78.34**	**78.18**	**78.15**
R-HDPE + uniform parsing	67.49	67.48	67.33
R-HDPE	75.95	75.83	75.79

TABLE VIII

COMPARISON OF MAPS WITH UNIFORM PARSING AND DYNAMIC PARSING ON THE OLYMPIC SPORTS AND HOLLYWOOD2 DATASETS

Alignment Method	Olympic	hollywood2
HDPE + uniform parsing	88.41	64.69
HDPE	**89.12**	**66.16**

dataset, respectively, and 8.8531 and 1.8244 for the sample from the Hollywood2 dataset, respectively. The LDS modeling accelerates the dynamic parsing significantly. Moreover, from the performance comparisons of using the HDPE representation with and without the LDS modeling as shown in Tab. VI, it can be observed that adding the LDS layer even slightly improves the performances of M-HDPE with different values of L on the UCF101 dataset.

E. Effects of Dynamic Parsing

To evaluate the effects of the dynamic parsing by temporal clustering algorithm 1, we compare the HDPE using the temporal clustering with HDPE using uniform parsing on the three datasets. "HDPE+uniform parsing" denotes that the action sequence is first uniformly parsed into divisions, and the two-layer hierarchical dynamic encoding is applied. The uniform parsing can be viewed as the initialization of the temporal clustering. The numbers of divisions for both uniform parsing and temporal clustering are set to 7, 20 and 30 on the Chalearn gesture, Olympic Sports and Hollywood2 datasets, respectively. The band factor f is set to 2 for all the datasets. The BoW-based frame-wise features are adopted for the Olympic Sports and Hollywood2 datasets. Both M-HDPE and R-HDPE are evaluated on the Chalearn dataset. The comparisons are shown in Tab. VII and Tab. VIII. The proposed temporal clustering outperforms the uniform parsing on all the datasets. This indicates that temporal clustering is able to parse the action into dynamic-coherent divisions and the dynamic parsing did benefit the final performance.

The improvements are more significant for R-HDPE, as reflected in the results on the Chalearn dataset and the Hollywood2 dataset. This suggests that more reliable parsings are required to use rank pooling in the first layer. For mean pooling, the output representation is not sensitive to the outliers in the division. Even for rough parsing, although the frames in some divisions may contain inconsistency outliers, the means of different divisions generally still reflect the evolution of the action. However, the non-linear rank pooling considers the relative ordering of all the frames in each division. If the parsing in the first layer leads to divisions with inconsistent dynamic, the resulting encodings of these divisions encode such non-smooth evolution and hence their ordering relationships are interrupted. It will be difficult to extract discriminative representation from these noisy encodings by rank pooling in the second layer.

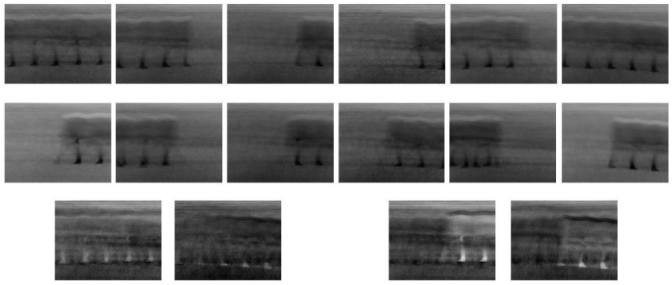
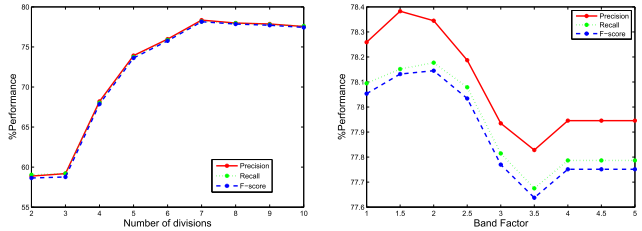


Fig. 5. The essential sequence (Top) before and (Middle) after the dynamic parsing. Bottom: the forward and reverse HDPE representations.

Fig. 4 visualizes the ending frames of the parsed segments of two example videos. The above simple action is segmented by the turning frames of poses such as take-off, start to fall and flip. The bottom complex action is divided by the starting and ending frames of different scenes or shots. Fig. 5 visualizes the essential sequence $\mathbf{U}$ before and after the dynamic parsing, as well as the learned HDPE representation. For simplicity, we conduct this visualization experiment on a sample video from the KTH action dataset [76], because the action was taken over homogeneous backgrounds by a static camera and hence the visualizations are less intrusive. We transform each RGB frame image to a gray-scale image and concatenate directly the columns into a vector as the frame-wise features of this frame. We set the number of divisions L to 6 and initialize the partition path $\mathbf{P}$ with a uniform division. We average the frame-wise features within each division, and the initial essential sequence is the sequence of the averaged features. For visualization, we reshape the averaged features to the size of the original frame image. We rescale the pixel values of the averaged frame such that the minimum value is projected to 0, the maximum value is projected to 255, and other values are interpolated linearly. The initial $\mathbf{U}$ is visualized in the first row of Fig. 5. The dynamics within some divisions are quite crowded, and there is no obvious regularity among divisions.

We then update $\mathbf{P}$ by the proposed temporal clustering, and update $\mathbf{U}$ accordingly. The visualization of the final learned $\mathbf{U}$ is shown in the second row of Fig. 5. We can observe that the learned $\mathbf{U}$ exhibits obvious gradual, phased and periodic characteristics. In this video, a person is walking back and forth around the camera lens. Taking the midline as the boundary, each division captures a clear stage of the moving positions. The stages are periodic and continuous along with the walking cycle progresses back and forth.

To obtain the HDPE representation, we apply forward and reverse rank pooling on the essential sequence $\mathbf{U}$ consisting of the averaged pixel-value-based frame-wise features. For visualization, we again reshape the learned representations to the size of frame images, and rescale the values by min-max interpolation. The forward and reverse HDPE representations based on the initial $\mathbf{U}$ are visualized in the first two images of the last row in Fig. 5, respectively, and those based on the final



(a) (b)

Fig. 6. The performances with the increase of (a) the number of divisions and (b) the value of band factor on the Chalearn Gesture dataset.

learned $\mathbf{U}$ are shown in the last two images of the last row in Fig. 5, respectively. We can observe that without temporal clustering, the dynamics reflected in the representations are more uniform; while the learned HDPE representations with temporal clustering reveal phased and contrasted dynamics, and emphasize representative local motions.

F. Influence of Parameters

There are mainly two parameters of the proposed HDPE: the number of divisions L for parsing the action sequence by temporal clustering and the band factor f for aligning the sequence to the essential sequence by dynamic programming. We evaluate the influences of the two parameters on the final performance.

We first evaluate the influence of L. For the Chalearn Gesture recognition dataset, the average number of frames is 39.7. We fix f to 2, and vary L from 2 to 10. The performances (the precision, recall and F-score) are shown in Fig. 6(a). For the Olympic Sports dataset and the Hollywood2 dataset, we fix f to 2, and vary L from 10 to 50. The performances (mAP) of HDPE using BoW-based and FV-based frame-wise features with the increasing values of L are shown in Fig. 7(a), Fig. 7(b), Fig. 8(a) and Fig. 8(b), respectively. We find that on the Chalearn dataset and the Hollywood2 dataset, at first all performance measures improve with the increase of the number of divisions, because more temporal structures information can be captured. When L is larger than 7 on the Chalearn dataset and 20 on the Hollywood2 dataset, the performances stop increasing. This may be because redundant divisions exist, which break the intrinsic temporal structures and slightly interfere the rank pooling of the second layer. The performances on the Olympic Sports dataset are quite oscillatory with the increasing of L, but generally inflection points exist and the amplitudes decay after these points. We set L to be the value of the inflection point for the corresponding dataset and frame-wise features in the subsequent experiments.

HDPE also supports to set different L for different sequences. For example, we can set L as N/r, r is a factor measuring on an average how many frames a state should contain and can be estimated according to prior knowledge of the data. We set L to be the same for all sequences, because as long as L is large enough, the evolution of L key stages should contain the information for discriminating different classes. Although the states of a more dynamic action are more complex, the local dynamics within these states are captured by the 1st layer modeling.

TABLE IX

COMPARISON OF PRECISIONS USING THE REPRESENTATIONS OF DIFFERENT LAYERS ON THE CHALEARN DATASET

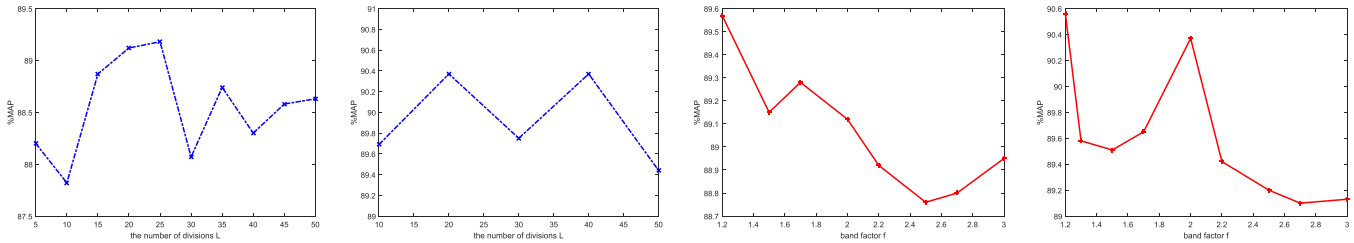
L	8	10	12	14	16	18	20
1st layer	77.70	77.53	77.42	76.54	76.24	76.36	75.52
2nd layer	68.92	73.84	76.92	77.79	76.60	76.49	76.27
3rd layer	57.01	59.30	62.10	68.90	68.74	70.34	72.63
1st+2nd	**78.76**	**79.09**	**78.44**	**78.10**	77.24	76.76	**76.53**
All layers	78.13	78.78	77.95	77.36	**78.20**	**77.36**	76.30

We then evaluate the influence of f. For the Chalearn dataset, we fix the number of divisions L to 7, and vary the band factor f from 1 to 5 with an interval of 0.5. When $f = 1$, it means that the alignment is strictly restricted to the uniform alignment. When $f > 4$, the allowed maximal capacity of a division is larger than the length of the sequence, and it is equivalent to perform unconstrained dynamic time warping, which may mistake outliers as individual divisions and lead to extremely unbalanced alignment. The results are shown in fig. 6(b). For the Olympic Sports dataset and the Hollywood2 dataset, we fix L to 20, and vary f from 1.2 to 3. The performances (mAP) of HDPE using BoW-based and FV-based frame-wise features with the increasing values of f are shown in Fig. 7(c), Fig. 7(d), Fig. 8(c) and Fig. 8(d), respectively. For the Olympic Sports dataset with the BoW-based frame-wise features, relatively balanced parsings with a small amount of wrappings lead to better results. For the other two datasets, sufficient wrappings are required and applying appropriate constraints on the capacity of each division benefits the performances. We set f in the range of 1.5 to 2 in the subsequent experiments. $f = 2$ means that the maximal number of elements within one division should not be larger than twice the average number of elements by uniform alignment.

G. Effects of Multi-Layers

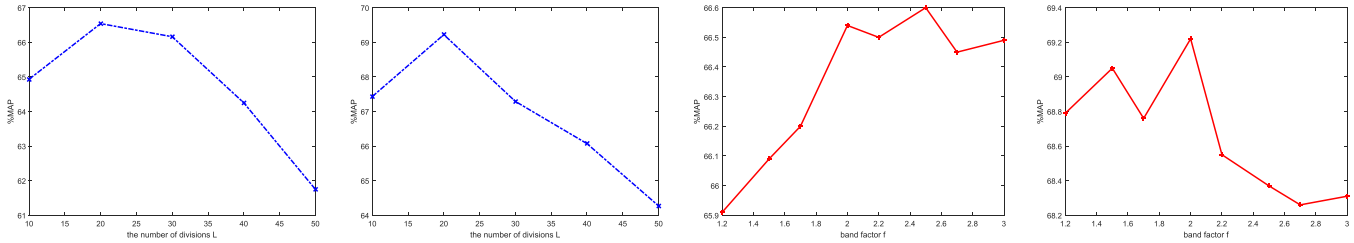
As introduced in Sec. III-E, HDPE can be generalized to multiple layers. To evaluate the effects of "deeper" hierarchy, we build HDPE model with three layers and extract the representations from all the layers. The number of divisions parsed in a higher layer is set to the half of the number of divisions in its lower layer. For the Chalearn dataset, the band factor is set to 2, and the number of divisions in the first ground layer is set from 8 to 20 with an interval of 2. The performances by using the representations of different layers and their combinations are shown in Tab. IX, Tab. X and Tab. XI. For the Olympic Sports dataset, the band factor is set to 1.5, and the number of divisions in the first layer is fixed to 40. "1st+2nd" means that the presentations of the first layer and the second layer are concatenated. "All layers" means that the representations of all the three layers are combined by concatenation.

We can find that in both datasets, the performances of the third layer are worse than the first two layers. This may indicate that two layers w.r.t. key poses and samples of the



(a) (b) (c) (d)

Fig. 7. The performances of HDPE with the increase of (a) the number of divisions using the BoW-based frame-wise features (b) the number of divisions using the FV-based frame-wise features (c) the value of band factor using the BoW-based frame-wise features and (d) the value of band factor using the FV-based frame-wise features on the Olympic Sports dataset.



(a) (b) (c) (d)

Fig. 8. The performances of HDPE with the increase of (a) the number of divisions using the BoW-based frame-wise features (b) the number of divisions using the FV-based frame-wise features (c) the value of band factor using the BoW-based frame-wise features and (d) the value of band factor using the FV-based frame-wise features on the Hollywood2 dataset.

TABLE X

COMPARISON OF RECALLS USING THE REPRESENTATIONS OF DIFFERENT LAYERS ON THE CHALEARN DATASET

L	8	10	12	14	16	18	20
1st layer	77.77	77.59	77.30	76.61	76.33	76.27	75.43
2nd layer	68.88	73.60	76.71	77.48	76.60	76.45	76.16
3rd layer	57.30	59.25	62.14	68.63	68.65	70.26	72.59
1st+2nd	**78.85**	**78.99**	**78.47**	**78.08**	77.20	76.65	**76.54**
All layers	78.21	78.75	78.03	77.27	**78.19**	**77.28**	76.28

TABLE XI

COMPARISON OF F-SCORES USING THE REPRESENTATIONS OF DIFFERENT LAYERS ON THE CHALEARN DATASET

L	8	10	12	14	16	18	20
1st layer	77.64	77.45	77.19	76.45	76.18	76.16	75.29
2nd layer	68.75	73.53	76.67	77.51	76.49	76.37	76.08
3rd layer	56.90	58.94	61.84	68.54	68.49	70.14	72.44
1st+2nd	**78.69**	**78.91**	**78.33**	**77.98**	77.11	76.54	**76.39**
All layers	78.04	78.64	77.89	77.21	**78.09**	**77.16**	76.13

pose are enough for simple actions such as gestures and single sport actions. However, when the number of divisions in the first layer increases, the performances of the third layer are improved on the Chalearn dataset. This means that finer parsing in the first layer is necessary to build more layers for HDPE, because more layers aim to parse the dynamics into finer levels, and the more the number of layers, the more meticulous the first layer parsing should be. The combinations of the three layers outperform the combinations of only the first two layers when enough divisions are parsed in the first layer. This may imply that high layers encoding spectrum of dynamics contain discriminative information that benefits the final performance. For more complex activities, additional layers w.r.t. high-level semantic can be more beneficial.

TABLE XII

COMPARISON OF MAPS USING THE REPRESENTATIONS OF DIFFERENT LAYERS ON THE OLYMPIC SPORTS DATASET

Layer	1st	2nd	3rd	1st+2nd	1st+2nd+3rd
MAP	90.52	89.90	88.89	**90.67**	90.64

H. Combinations

A potential advantage of the proposed method is the representations produced from different numbers of partitions in the first layer encode the temporal structures in different scales. If the number of divisions is set to 1, the temporal information is totally discarded and the proposed HDPE method boils down to the "IDT" method [8]. If the number of divisions is set to be the length of the sequence, no local appearances are smoothed and the proposed HDPE method boils down to the "rank pooling" method [6]. The more divisions are parsed from the action, the finer the scale of the captured temporal information is. The representations generated in different scales provide complementary information to each other. Combining them together incorporates multi-scale temporal information together. We evaluate the combinations of HDPE where 20 divisions are parsed in the first layer with either "local" or "rank pooling" or both on the Olympic Sports dataset and the Hollywood2 dataset. FV-based frame-wise features are used for both datasets. The comparisons are shown in Tab. XIII. We find that the combinations of HDPE with "rank pooling" achieve the best results on both datasets. The combinations are achieved by simple concatenation. More advanced fusion methods and combinations with more middle temporal scale parsings in the first layer may further improve the performance.

TABLE XIII

COMPARISON OF MAPS USING DIFFERENT COMBINATIONS ON THE OLYMPIC SPORTS AND HOLLYWOOD2 DATASETS

Alignment Method	Olympic	hollywood2
HDPE+IDT	90.85	69.26
HDPE+rank pooling	**91.18**	**70.40**
HDPE+both	90.94	**70.40**

TABLE XIV

COMPARISON OF THE PROPOSED HDPE WITH STATE-OF-THE-ART RESULTS ON THE CHALEARN GESTURE DATASET

Method	Precision	Recall	F-score
Wu et al. [77]	59.9	59.3	59.6
Yao et al. [72]	-	-	56.0
Pfister et al. [73]	61.2	62.3	61.7
Fernando et al. [6]	75.3	75.1	75.2
Rank pooling [6]	74.0	73.8	73.9
HDPE	**78.34**	**78.18**	**78.15**
1st+2nd HDPE	**79.09**	**78.99**	**78.91**

I. Comparison With State-of-the-Art

It may be difficult to perform a fair comparison with state-of-the-art results because different methods use different components such as types of features and sample argument methods. The state-of-the-art results are usually achieved by fusing different types of features and adopting data augmentation techniques. We compare the proposed HDPE with the improved dense trajectory features (denoted by "IDT") encoded by Bag-of-Words (BoW) or Fisher Vector (FV) encoding [8] and learning to rank based temporal encoding (denoted by "rank pooling") [6] of the whole video as well as the several other state-of-the-art results on the ChaLearn Gesture dataset, the Olympic Sports dataset and the Hollywood2 dataset, as shown in Tab. XIV, Tab. XV Tab. XVII, respectively. For HDPE, the number of divisions for each video is set to be 7, 20 and 20 for the three datasets, respectively. The band factor is set to be 2 for all these datasets.

From Tab. XIV, it can be observed that HDPE outperforms the state-of-the-art method [6] on the ChaLearn dataset. In [6], the results are achieved by combining the rank pooling representation with a local method, and the results by rank pooling along are also reported, as denoted by "Rank pooling". Since we use the same frame-wise features provided by [6], the superior performance comes from the hierarchical parsing and modeling. The combination with the first layer pooling further improves the performances of HDPE.

Tab. XV shows that the result of the single HDPE with FV-based frame-wise features is slightly worse than the best result reported in [8]. There is a certain randomness when extracting IDT descriptors and FV encoding. Our reproduction of IDT with FV is about 89%. Based on strictly the same features, HDPE outperforms IDT. When combined with rank pooling, HDPE outperforms the reported result. The authors in [8] also report their results with the Bag-of-words encoding, as denoted by "IDT(BoW)" in Tab. XV. Our method outperforms the IDT method that encoding descriptors in all frames into a single representation without considering the temporal information by a margin of 4%.

We also evaluate the multi-class accuracy on the Olympic Sports dataset in Tab. XVI. The BoW-based frame-wise features are used. The proposed HDPE representation itself significantly outperforms the reported results by a margin of 7.5%, and the combination of IDT, rank pooling and HDPE representations further extends the margin to about 10%.

As shown in Tab. XVII, on the Hollywood2 dataset, with the BoW encoding, our method significantly outperforms the "IDT(BoW)" method reported in [8] and our reproduction of rank pooling by a margin of 4%. With the FV encoding, the combination of HDPE with rank pooling outperforms the combination of rank pooling and IDT reported in [6]. Fernando *et al.* [6] and Hoai and Zisserman [80] achieve higher mAPs, where the performances of rank pooling and IDT were improved by 5% and 2%, respectively, using the data augmentation (DA) technique [80]. DA doubles the training data by flipping each video and averages the classification

TABLE XV

COMPARISON OF THE PROPOSED HDPE WITH STATE-OF-THE-ART RESULTS ON THE OLYMPIC SPORTS DATASET. MAP IS USED AS THE PERFORMANCE MEASURE

Method	Olympic Sports
Brendel et al. [48]	77.3
Gaidon et al. [78]	82.7
Jain et al. [22]	83.2
Wang et al. [8] (IDT+FV)	91.1
Wang et al. [14]	90.4
Wang et al. [35] (MoFAP)	**92.6**
IDT(BoW) [8]	83.3
HDPE(BoW)	**89.12**
HDPE(FV)	90.37
HDPE+Rank pooling(FV)	**91.18**

TABLE XVI

COMPARISON OF THE PROPOSED HDPE WITH STATE-OF-THE-ART RESULTS ON THE OLYMPIC SPORTS DATASET. ACCURACY IS USED AS THE PERFORMANCE MEASURE

Method	Accuracy
Laptev et al. [9]	62.0
Niebles et al. [43]	72.1
Tang et al. [79]	66.8
Wang et al. [42]	73.8
HDPE	**81.34**
HDPE+Rank Pooling+IDT	**83.58**

TABLE XVII

COMPARISON OF THE PROPOSED HDPE WITH STATE-OF-THE-ART RESULTS ON THE HOLLYWOOD2 DATASET. MAP IS USED AS THE PERFORMANCE MEASURE. * DENOTES THAT THE RESULT IS REPORTED BY OUR REPRODUCTION WITH THE BOW REPRESENTATION

Method	Hollywood2
Jain et al. [22]	62.5
Wang et al. [8]	64.3
Wang et al. [14]	66.8
Hoai et al. [80]	73.6
Fernando et al. [6]	**73.7**
IDT(BoW) [8]	62.2
Rank pooling(BoW) [6]*	62.19
HDPE(BoW)	**66.54**
IDT(FV) [8]	64.3
Rank pooling+IDT(FV) [6]	70.0
HDPE(FV)	69.22
HDPE(FV+DA)	69.41
HDPE+Rank pooling(FV)	70.40
HDPE+Rank pooling(FV+DA)	**70.80**

TABLE XVIII

COMPARISON OF THE PROPOSED HDPE WITH STATE-OF-THE-ART RESULTS ON THE UCF101 DATASET. ACCURACY IS USED AS THE PERFORMANCE MEASURE. TOP: METHODS WITH A SINGLE DEEP NETWORK TAKING ONLY RGB FRAMES AS INPUTS; BOTTOM: METHODS USING MULTIPLE NETWORK FUSIONS OR MULTIPLE FEATURE COMBINATIONS

Method	Accuracy
CNN [37]	65.4
Spatial stream network [51]	72.6
LRCN [81]	71.1
LSTM composite model [40]	75.8
C3D [41]	82.3
C3D + Rank pooling	77.12
HDPE	**83.58**
Two-stream networks [51]	88.0
LRCN [81]	82.9
LSTM composite model [40]	84.3
TDD + IDT [38]	91.5
$F_{ST}CN$ [53]	88.1
C3D (3 nets) + IDT [41]	90.4
TSN (3 modalities) [54]	**94.2**

scores of a test video and its mirrored version. We also augment the data by mirroring, but we train two SVM classifiers for the original videos and the mirrored videos, respectively, because this obtains better results in our experiments. We can observe that DA only brings a very small improvement on HDPE. This could be due to both the local temporal structures and their dynamic evolutions of the mirrored videos generally vary greatly from the original videos, and augmenting them into the training set will increase the intra-class variances; while training two separate classifiers for the original and mirrored videos cannot provide much complementary information.

Tab. XVIII shows the comparisons of HDPE with mean pooling, rank pooling, and deep-learning-based state-of-the-art methods on the UCF101 dataset. For HDPE, L and f are set to be 15 and 2, respectively. It has been shown in [38] and [41] that the deep-learning-based representations and the hand-crafted IDT representations are complementary. The state-of-the-art results shown in the bottom part of Tab. XVIII are obtained by combining different types of representations and fusing multiple deep networks built on both RGB frames and optimal flows. Since we only employ the features extracted by a single C3D network, we mainly compare with the results obtained by a single deep architecture taking only RGB frames as inputs in the top part of Tab. XVIII. In [41], mean pooling is applied to the frame-wise C3D features to form the final representation. We can see that based on the same C3D frame-wise features, HDPE outperforms both mean pooling and rank pooling; based on a single deep network, HDPE also outperforms other deep-learning-based representations or methods.

V. CONCLUSIONS

In this paper, we have presented a hierarchical dynamic parsing and encoding method for action recognition, which learns higher-level representation from a single action sequence by exploring the temporal structures and building the hierarchical architecture in an unsupervised way. The hierarchy disentangles the local appearances and the global dynamic into different layers. In the lower layer, the sequence is parsed into different divisions, and local appearance information within each uniformly-evolved division is captured via local mean or rank pooling. In the higher layer, the global dynamic of the appearances among the divisions is encoded. The learned representation is robust because outliers or noisy frames cannot directly impact on the global dynamic since they must be assigned to a corresponding division, while their influence within a division is greatly diminished by pooling. Experimental results on several action datasets have demonstrated the potential of the proposed method.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.

[2] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.

[3] K. Li, J. Hu, and Y. Fu, "Modeling complex temporal composition of actionlets for activity prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 286–299.

[4] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Conditional models for contextual human motion recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2. Oct. 2005, pp. 1808–1815.

[5] B. Yao and S.-C. Zhu, "Learning deformable action templates from cluttered videos," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1507–1514.

[6] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5378–5387.

[7] B. Su, J. Zhou, X. Ding, H. Wang, and Y. Wu, "Hierarchical dynamic parsing and encoding for action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 202–217.

[8] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2013, pp. 3551–3558.

[9] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[10] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. Brit. Mach. Vis. Conf.*, 2008, pp. 275–285.

[11] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1932–1939.

[12] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3169–3176.

[13] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[14] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 219–238, 2016.

[15] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2486–2493.

[16] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1794–1801.

[17] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 3360–3367.

[18] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 141–154.

[19] Z. Cai, L. Wang, X. Peng, and Y. Qiao, "Multi-view super vector for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 596–603.

[20] X. Yang and Y. Tian, "Action recognition using super sparse coding vector with spatio-temporal awareness," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 727–741.

[21] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 3304–3311.

[22] M. Jain, H. Jégou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2555–2562.

[23] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2004–2011.

[24] J. Wang, Z. Chen, and Y. Wu, "Action recognition with multiscale spatio-temporal contexts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3185–3192.

[25] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2046–2053.

[26] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 581–595.

[27] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1290–1297.

[28] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3D human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 914–927, May 2014.

[29] M. Sapienza, F. Cuzzolin, and P. H. S. Torr, "Learning discriminative space-time actions from weakly labelled videos," in *Proc. BMVC*, vol. 2. 2012, p. 3.

[30] M. Sapienza, F. Cuzzolin, and P. H. S. Torr, "Learning discriminative space–time action parts from weakly labelled videos," *Int. J. Comput. Vis.*, vol. 110, no. 1, pp. 30–47, 2014.

[31] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1242–1249.

[32] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu, "Action recognition with actons," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2013, pp. 3559–3566.

[33] L. Wang, Y. Qiao, and X. Tang, "Motionlets: Mid-level 3D parts for human motion recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2674–2681.

[34] A. Jain, A. Gupta, M. Rodriguez, and L. S. Davis, "Representing videos using mid-level discriminative patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2571–2578.

[35] L. Wang, Y. Qiao, and X. Tang, "MoFAP: A multi-level representation for action recognition," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 254–271, 2016.

[36] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[37] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[38] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4305–4314.

[39] G. Chéron, I. Laptev, and C. Schmid, "P-CNN: Pose-based CNN features for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2015, pp. 3218–3226.

[40] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 843–852.

[41] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4489–4497.

[42] J. Wang and Y. Wu, "Learning maximum margin temporal warping for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2688–2695.

[43] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *Proc. Eur. Conf. Comput. Vis.*, Dec. 2010, pp. 392–405.

[44] L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 810–822, Feb. 2014.

[45] H. Pirsiavash and D. Ramanan, "Parsing videos of actions with segmental grammars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 612–619.

[46] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom sequence models for efficient action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Jun. 2011, pp. 3201–3208.

[47] A. Gaidon, Z. Harchaoui, and C. Schmid, "Temporal localization of actions with actoms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2782–2795, Nov. 2013.

[48] W. Brendel and S. Todorovic, "Learning spatiotemporal graphs of human activities," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 778–785.

[49] B. Wu, C. Yuan, and W. Hu, "Human action recognition based on context-dependent graph kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2609–2616.

[50] B. Su and G. Hua, "Order-preserving wasserstein distance for sequence matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1049–1057.

[51] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[52] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4041–4049.

[53] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4597–4605.

[54] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 20–36.

[55] B. Su and X. Ding, "Linear sequence discriminant analysis: A model-based dimensionality reduction method for vector sequences," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 889–896.

[56] B. Su, X. Ding, H. Wang, and Y. Wu, "Discriminative dimensionality reduction for multi-dimensional sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2017.2665545.

[57] B. Su, X. Ding, C. Liu, H. Wang, and Y. Wu, "Discriminative transformation for multi-dimensional temporal sequences," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3579–3593, Jul. 2017.

[58] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 994–999.

[59] S. Hongeng and R. Nevatia, "Large-scale event detection using semi-hidden Markov models," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1455–1462.

[60] Y. Wang and G. Mori, "Hidden part models for human action recognition: Probabilistic versus max margin," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1310–1323, Jul. 2010.

[61] T. Xiang and S. Gong, "Beyond tracking: Modelling activity and understanding behaviour," *Int. J. Comput. Vis.*, vol. 67, no. 1, pp. 21–51, 2006.

[62] M. Pei, Y. Jia, and S.-C. Zhu, "Parsing video events with goal inference and intent prediction," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 487–494.

[63] G. Luo, S. Yang, G. Tian, C. Yuan, W. Hu, and S. J. Maybank, "Learning human actions by combining global dynamics and local appearance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2466–2482, Dec. 2014.

[64] Y. Song, L.-P. Morency, and R. Davis, "Action recognition by hierarchical sequence summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3562–3569.

[65] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning, and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2649–2656.

[66] F. Zhou, F. de la Torre, and J. F. Cohn, "Unsupervised discovery of facial events," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2574–2581.

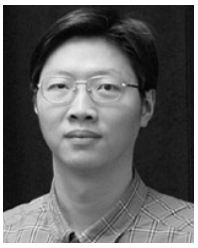
[67] M. Hoai and F. de la Torre, "Maximum margin temporal clustering," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2012, pp. 1–9.

[68] B. Su, X. Ding, C. Liu, and Y. Wu, "Heteroscedastic max-min distance analysis," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4539–4547.
[69] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Int. J. Comput. Vis.*, vol. 51, no. 2, pp. 91–109, 2003.
[70] S. Escalera *et al.*, "Multi-modal gesture recognition challenge 2013: Dataset and results," in *Proc. 15th ACM Int. Conf. Multimodal Interact.*, 2013, pp. 445–452.
[71] S. Escalera *et al.*, "Chalearn multi-modal gesture recognition 2013: Grand challenge and workshop summary," in *Proc. 15th ACM Int. Conf. Multimodal Interact.*, 2013, pp. 365–368.
[72] A. Yao, L. Van Gool, and P. Kohli, "Gesture recognition portfolios for personalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1915–1922.
[73] T. Pfister, J. Charles, and A. Zisserman, "Domain-adaptive discriminative one-shot learning of gestures," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 814–829.
[74] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," Center Res. Comput. Vis., Univ. Central Florida, Orlando, FL, USA, Tech. Rep. CRCV-TR-12-01, Nov. 2012.
[75] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.
[76] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 3. Aug. 2004, pp. 32–36.
[77] J. Wu, J. Cheng, C. Zhao, and H. Lu, "Fusing multi-modal features for gesture recognition," in *Proc. 15th ACM Int. Conf. Multimodal Interact.*, 2013, pp. 453–460.
[78] A. Gaidon, Z. Harchaoui, and C. Schmid, "Recognizing activities with cluster-trees of tracklets," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–30.
[79] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1250–1257.
[80] M. Hoai and A. Zisserman, "Improving human action recognition using score distribution and ranking," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 3–20.
[81] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2625–2634.

Bing Su received the B.S. degree in information engineering from the Beijing Institute of Technology, Beijing, in 2010, and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, in 2016. He is currently an Assistant Professor with the Institute of Software Chinese Academy of Sciences, Beijing. His research interests include pattern recognition, computer vision, and machine learning.

Jiahuan Zhou received the B.E. degree from Tsinghua University in 2013. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Northwestern University. His current research interests include computer vision, pattern recognition and machine learning.

Xiaoqing Ding (LF'15) received the degree from Tsinghua University, China, in 1962. She is currently a Professor and a Ph.D. Supervisor with the Department of Electronic Engineering, Tsinghua University. She has published more than 600 papers, co-authored three books and holds 27 Invention Patents. Her research interests include pattern recognition, image processing, characters recognition, biometrics, computer vision, and video surveillance. She is an IAPR Fellow. She received the graduate Golden Medal. For information theoretic-based approaches, she has got successful achievements on Chinese character recognition, multi-language recognition, and also on biometric recognition, such as face recognition. She has received Best Overall Performing Face Verification Algorithm in the 2004 Face Authentication Test in 17th ICPR2004. Her character recognition and face recognition systems have been licensed to a number of companies, including MS Office 2000, and are being marketed and deployed worldwide. She has received four top prestigious national scientific and technical progress awards in China for multi-language character and document recognitions and for face and writer recognition, in 1992, 1998, 2003, and 2008, respectively.



Ying Wu received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, the M.S. degree from Tsinghua University, Beijing, China, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), Urbana, IL, USA, in 1994, 1997, and 2001, respectively. From 1997 to 2001, he was a Research Assistant with the Beckman Institute for Advanced Science and Technology, UIUC. From 1999 to 2000, he was a Research Intern with Microsoft Research, Redmond, WA, USA. In 2001, he joined the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, USA, as an Assistant Professor. He was promoted as an Associate Professor in 2007 and a Full Professor in 2012. He is currently a Full Professor of Electrical Engineering and Computer Science with Northwestern University. His current research interests include computer vision, image and video analysis, pattern recognition, machine learning, multimedia data mining, and human-computer interaction. He received the Robert T. Chien Award by UIUC in 2001 and the NSF CAREER Award in 2003. He serves as an Associate Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the *SPIE Journal of Electronic Imaging*, and the *IAPR Journal of Machine Vision and Applications*.