Confidence-Guided Self Refinement for Action Prediction in Untrimmed Videos

Jingyi Hou, Xinxiao Wu, *Member, IEEE*, Ruiqi Wang, Jiebo Luo, *Fellow, IEEE*, and Yunde Jia, *Member, IEEE*

***Abstract*— Many existing methods formulate the action prediction task as recognizing early parts of actions in trimmed videos. In this paper, we focus on predicting actions from ongoing untrimmed videos where actions might not happen at the very beginning of videos. It is extremely challenging to predict actions in such untrimmed videos due to ambiguous or even no information of actions in the early parts of videos. To address this problem, we propose a prediction confidence that assesses the decision quality of a prediction model. Guided by the confidence, the model continuously refines the prediction results by itself with the increasing observed video frames. Specifically, we build a Self Prediction Refining Network (SPR-Net) which incrementally learns the confidence for action prediction. SPR-Net consists of three modules: a temporal hybrid network, an incremental confidence learner, and a self-refining Gumbel softmax sampler. The temporal hybrid network generates the action category distributions by integrating static scene and dynamic motion information. The incremental confidence learner calculates the confidence in an incremental manner, judging the extent to which the temporal hybrid network should believe its prediction result. The self-refining Gumbel softmax sampler models the mutual relationship between the prediction confidence and the category distribution, which enables them to be jointly learned in an end-to-end fashion. We also present a sparse self-attention mechanism to encode local spatio-temporal features into the frame-level motion representation to further improve the prediction performance. Extensive experiments on five datasets (i.e., UT-Interaction, BIT-Interaction, UCF101, THUMOS14, and ActivityNet) validate the effectiveness of the proposed method.**

***Index Terms*— Action prediction, decision confidence, hybrid networks, attention mechanism.**

I. Introduction

Action prediction in videos refers to inferring the action label in an ongoing video with high categorization accuracy and low observational latency. It has a variety of applications such as intelligent video surveillance and autonomous navigation. Most existing methods [1]–[10] predict actions in trimmed videos. They mainly focus on discovering discriminative clues from onsets of actions to make predictions. However, most of the realistic videos are untrimmed with plenty of various backgrounds or idle streams at the beginning, and it is unrealistic to trim an ongoing video before the prediction.

In this paper, we deal with a more practicable and general task of predicting actions in both untrimmed and trimmed videos. The problem is challenging due to the ambiguous or even no information of actions in the early stage of videos. Fortunately, some informative and action-related clues from the early video frames before the happening of an action can help predict category label to some extent. For example, one can tell that someone is going to dive at the sight of the springboard and swimming pool. The information of "springboard" and "swimming pool" is important for predicting the action of diving. However, in the process of action prediction, whether the observed information is sufficient to infer correct category labels remains agnostic to the predictor, which is quite different from typical cognitive tasks such as action recognition [11]–[16] and multimedia event detection [17]–[19]. In practice, the sufficiency of the information for decision-making is not absolutely agnostic for human, because human brain can generate a self-assessment, namely confidence, to adjust the decision. The process is referred to as metacognition in the literatures of psychology [20], [21]. So it is more suitable to regard action prediction as a dynamic process involving metacognition, rather than just a cognitive task of recognizing an action in a partly available video.

Therefore, we formulate the action prediction in untrimmed videos as a metacognitive process. Accordingly, three issues need to be addressed: (1) **When** the observed information is enough for making a prediction? It is essential to indicate when the decision is reliably made for prediction, because in the already observed video stream, the information might be insufficient to lead to misclassification. In this paper, we introduce the concept of confidence to judge the quality of decisions. (2) **What** information should be used for action prediction? Since there is ambiguous information in videos, such as cluttered backgrounds and irrelevant actions, which will degrade the prediction performance, learning discriminative feature representations from the observed video stream becomes an important issue for accurately predicting actions. (3) **How** to fully exploit the relationship between the aforementioned two issues? The issues of "**when**" and "**what**" are obviously related to each other. On one hand, the confidence serves as a judgment on the discriminative power of the currently available information, so as to guide the prediction model to learn robust and discriminative representations. On the other

Manuscript received September 27, 2018; revised March 4, 2019 and February 3, 2020; accepted March 29, 2020. Date of publication April 17, 2020; date of current version April 29, 2020. This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant 61673062. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jing-Ming Guo. *(Corresponding author: Xinxiao Wu.)*

Jingyi Hou, Xinxiao Wu, Ruiqi Wang, and Yunde Jia are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 10081, China (e-mail: houjingyi@bit.edu.cn; wuxinxiao@bit.edu.cn; wangruiqi@bit.edu.cn; jiayunde@bit.edu.cn).

Jiebo Luo is with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA (e-mail: jluo@cs.rochester.edu).

Digital Object Identifier 10.1109/TIP.2020.2987425

hand, the learning procedure of feature representations can affect the value of confidence to encourage the prediction model to make correct decisions in a timely manner. So the joint learning of "when" and "what" can boost the prediction performance.

In order to address the above three issues, we propose a Self Prediction Refining Network, called SPR-Net, which incrementally learns the confidence values to automatically refine the results of action prediction. We factorize the prediction task into estimating the confidence of prediction via an incremental confidence learner and learning the discriminative representation for classification via a temporal hybrid network, addressing the issues of "**when**" and "**what**", respectively. To jointly learn these two sub-tasks to make them benefit each other, a self-refining Gumbel softmax sampler is proposed, which handles the "**how**" issue. Fig. 1 illustrates the procedure of action prediction by SPR-Net.

To be specific, the prediction confidence is the embodiment of information and becomes higher as the amount of information increases. Intuitively, the increment of the confidence value at each time step depends on the criticality and freshness of the information currently. Therefore, we propose the incremental confidence learner to dynamically update the confidence.

In order to classify action categories, it is necessary to represent the detailed action-related patterns from the observed videos. The temporal hybrid network, which consists of an RNN layer with GRUs and a 1D convolutional layer (Conv1D), is introduced to exploit action-related information for generating the distributions of action categories. The RNN with GRUs [22] represents contextual scene information in the observed video stream, and Conv1D describes the salient patterns of videos via the convolutional operations.

The self-refining Gumbel softmax sampler models the mutual relationship of the prediction confidence and the category distribution. This sampler adjusts the smoothness of the category distribution according to the prediction confidence via a re-parametrization procedure. In the first few frames containing ambiguous action patterns, the sampler generates a relative low confidence value as well as Gumbel random variables to make the output distribute evenly for exploring more possible action categories. When it comes to frames containing key information, the increased confidence will enforce the distributions to approximate to one-hot vectors, which contributes to easier convergence of training. Through the learning of the self-refining Gumbel softmax sampler, the adjusted confidence helps discover more discriminative information for the correct category distribution, and the refined category distribution facilitates timely predictions by forcing the model to generate high confidence early.

We also integrate a Sparse Self-Attention (SSA) module into SPR-Net to represent motion information at each time step. The SSA module performs multiple self-attention operations to fuse the local spatio-temporal motion features into a unified frame-level feature. The self-attention mechanism captures different aspects with regard to the motion of short video clip ahead of the current time step. Instead of the softmax activation, we use the sparsemax [23] to force the attention weights to be sparse, since the sparsemax truncates some noisy information to zero and reduces the strong constraint of the penalty term to guarantee the variety of attention operations.

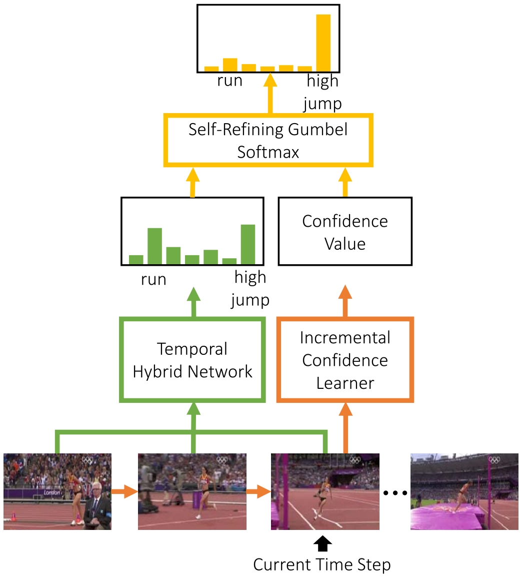


Fig. 1. Procedures of action prediction by SPR-Net. The category distribution is generated from the observed video stream via the temporal hybrid network. The prediction confidence is calculated by the mutual relationship between the prediction confidence and the category distribution is modeled by the self-refining Gumbel softmax sampler. Through back propagation of SPR-Net, the confidence value can be learned to refine the final prediction.

Experiments on five publicly available video datasets demonstarte that our method outperforms the state-of-the-art methods. Overall, the main contributions are:

- We propose the prediction confidence to predict actions with low observation latency in untrimmed videos by formulating the action prediction as a metacognitive process.
- We build a novel deep neural network called SPR-Net for confidence-based action prediction, involving an incremental confidence learner, a temporal hybrid network and a self-refining Gumbel softmax sampler, which can be trained in an end-to-end manner.
- We develop a new sparse self-attention module to effectively represent motion information of actions, which can also be easily integrated into other networks for action analysis.

II. Related Work

A. Action Prediction

Traditional action prediction methods focus on inferring *ongoing actions* from the onsets of action videos, which is first defined by Ryoo [1]. Early approaches are based on conventional machine learning methodologies. Ryoo [1] generated integral and dynamic bag-of-words for action prediction. Yu *et al.* [2] managed to match the training and test videos by a random forest structure to predict future actions. The methods of [2], [10] predict the locations and labels of actions via multiple-level features. In [3], [7], [24], action is predicted by SVMs. Lan *et al.* [4] proposed a structured hierarchical model to predict actions before they are actually executed.

Singh *et al.* [25] detected spatial-temporal action location and leveraged the localization result to predict actions in videos. Lai *et al.* [26] exploited a global-local temporal model that applies temporal saliency to adapt the contribution of each local-temporal distance for prediction.

With the rapid development of deep learning, recurrent neural network (RNN) has been widely explored, and long short-term memory (LSTM) methods have attracted increasing attentions in the field of action prediction. The methods in [6], [8] penalize classification mistakes over time by designing new losses for anticipating actions. Kong *et al.* [9] utilized a bi-directional LSTM by adding residual connections for action prediction. Hu *et al.* [27] introduced several soft regression-based models that consider both subsequence and discrepancy of soft labels for both RGB and RGB-D action prediction. Zhao and Wildes [28] learned to propagate residuals of each frame to learn compact spatiotemporal representation of partial videos. Other deep learning based methods resort to transferring knowledge from full videos to represent partial videos for action prediction. Kong *et al.* [29] adopt adversarial learning to learn discriminative representation of partial videos from full videos. Cai *et al.* [30] learned feature embeddings and classifier from full videos to improve the prediction performances of partial videos. Wang *et al.* [31] proposed a teacher-student learning method that distills knowledge from action recognition network for action prediction. The latent global network [32] provides complementary information of full videos to local temporal information for improving skeleton-based action prediction.

All the aforementioned methods predict action labels only when the actors start to perform, which refers to anticipating actions in trimmed videos. Our method can cope with untrimmed videos of which the onsets might not contain any related actions by exploiting the underlying key information of actions. Liu *et al.* [33], [34] dealt with 3D streaming sequence containing multiple action instances. Different from them, we focus on predicting actions in untrimmed videos containing background before action starts.

Several other approaches aim at predicting future actions that are different from the current observed one by using temporal detection annotations [35]–[39]. The system in [40] predicts atomic actions to construct the overall activity in the video. Although these methods claim that they can deal with untrimmed videos, our method is different from them. They focus on forecasting future actions with the current observed actions as a prior knowledge. In contrast, our goal is to predict the category of the incoming or ongoing action in a video accurately and timely. There are some works [41]–[43] of predicting human motion, and our work differs from them in dealing with predicting action category labels.

B. Decision Confidence

Confidence refers to a measure of human decision quality, which has been widely studied in the field of psychology [44]–[47]. As a type of metacognitive judgment, confidence plays a crucial role in human metacognition, such as serving as a guidance for making further decision [48] or integrating different options of decision [49]. Several researches [50], [51] in experimental psychology suggest that confidence and decision should be modeled separately and in parallel in an architecture. There are also evidences [21] in neuroscience suggesting that the parts of the confidence coding and the decision-making are in separate areas. Fleming and Dow [52] proved that confidence operating as a second-order computation of the decision is suitable in a general framework for metacognitive computation. The framework represents the decision and confidence variables as two segregated but correlated hidden states. The decision is derived from the hidden state of decision variable, and the calculation of confidence is conditioned on the confidence variables and the decision. Encouraged by these findings, we build SPR-Net to separately calculate the prediction decision and confidence, and make them affect each other during training.

The concept of confidence has also been applied to several visual tasks, such as object detection and tracking. Elliethy and Sharma [53] presented a framework of stochastic progressive association across multiple frames and used the confidence to decide whether to maintain the tracking windows or not. Guo *et al.* [54] encoded prior information in frames as the confidence of training samples, and proposed a max-confidence boosting algorithm for visual tracking. Bae and Yoon [55] proposed an online tracking method based on the tracklet confidence. Michael *et al.* [56], [57] built frameworks with continuous confidence of detectors for multi-person tracking. Other detection or tracking methods [58]–[60] treat the confidence as a selection metric. Online action detection methods [61], [62] calculate the confidence score for judging whether the current time step is an action or not. Shou *et al.* [63] used the confidence to measure the action start time in videos. These above-mentioned works learn confidence as a threshold to make a decision, and assume that the decision probability and confidence are identical. Different from them, our method separately learns the decision and confidence in parallel, and exploits the relationship between them for action prediction.

C. Attention Mechanism for Video Representation

Recently, attention mechanism has been employed to represent videos [15], [64]–[66] by aggregating local features. For one kind of local features, these methods apply attention operation only once. Our sparse self-attention module can capture multiple aspects of action videos using repetitive attention operations. It is more reasonable because a video clip might contain multiple features worthy of attention, and only one attention operation is not adequate for capturing all the information. Du *et al.* [67] proposed a spatial-temporal attention module with LSTM to capture global action representations, and developed an actor-attention regularization to focus on local salient regions of the frames. Long *et al.* [68] generated action clusters by applying multiple attention operations, and replaced the penalty by a shifting operation to make learning easy. The shifting operation will add more parameters to ensure the diversity of features after different attention operations. Our method can weaken the strong restrictions of

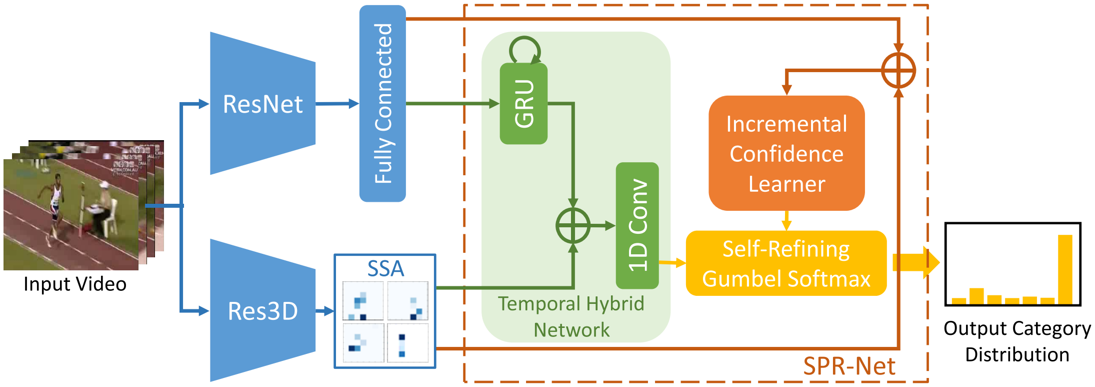

Input Video
ResNet
Res3D
Fully Connected
SSA
GRU
1D Conv
Temporal Hybrid Network
Incremental Confidence Learner
Self-Refining Gumbel Softmax
SPR-Net
Output Category Distribution

Fig. 2. Architecture overview of the proposed SPR-Net. The frame-level scene features are is extracted by ResNet. The motion features are obtained by Res3D with the Self Sparse Attention (SSA). SPR-Net consists of the temporal hybrid network (shown in green block), incremental confidence learner (shown in orange block) and Self-refining Gumbel softmax (shown in yellow block).

the penalty without introducing any extra parameters. Also, the sparsity of our method can remove the redundancy via the repetitive attention operations.

III. Self Prediction Refining Network (SPR-Net)

The architecture overview of SPR-Net is illustrated in Fig. 2. Each input action video at current time step is represented by the frame-level scene feature using ResNet-152 [69] and the local spatio-temporal features using Res3D [70]. The local spatio-temporal features are then spatially aggregated into the frame-level motion features via the Sparse Self-Attention (SSA) module, which will be elaborated on in Sec. IV. After that, these frame-level features are fed into SPR-Net. In the following subsections, we will introduce the three components of SPR-Net, i.e., the temporal hybrid network, the incremental confidence learner, and the self-refining Gumbel softmax.

A. Temporal Hybrid Network

To effectively utilize the frame-level scene and motion information to represent the observed video information for generating the category distribution of actions, we build a new temporal hybrid network which contains a single layer RNN with GRUs and a single layer Conv1D. The RNN with GRUs is applied to record the historical frame-level scene information. We use GRUs in the interest of computational speed. Since multiple convolution operations have a good property of representing a variety of discriminative patterns for classification, we use Conv1D after GRUs to generate the action category distributions.

Specifically, the frame-level scene features of the observed videos are integrated into a vector that carries contextual scene information via RNNs with GRUs. The vector is then combined with the frame-level motion feature via the concatenation operation. Before the concatenation, the two features, i.e., the contextual scene feature and the frame-level motion feature, are aligned by the layer normalization operation [71], because these two kinds of features are inconsistent in scale and only using concatenation can cause the loss function to be trapped in NaN in the training procedure. Layer normalization is applied independently to the elements within a feature vector, so it is of great convenience to handle the feature vectors with varied temporal dimensions. Taking a D-dimensional contextual scene feature $\boldsymbol{s}^t = [s_1^t, \ldots, s_D^t] \in \mathbf{R}^D$ at time step t as an example, the layer normalization is given by

$$\begin{aligned}\hat{\boldsymbol{s}}^t &= \frac{\boldsymbol{\alpha} \odot (\boldsymbol{s}^t - \mu \cdot \mathbf{1})}{\sigma} + \boldsymbol{\beta},\\ \mu &= \frac{1}{D}\sum_{i=1}^{D} s_i^t,\\ \sigma &= \left[\frac{1}{D}\sum_{i=1}^{D}(s_i^t - \mu)^2\right]^{1/2},\end{aligned} \tag{1}$$

where μ and σ are the mean and the standard deviation of $\{s_i^t\}_{i=1}^D$, respectively. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the learnable parameters of this layer. $\odot$ denotes the element-wise product. As can be easily inferred from Eq. (1), features after layer normalization are invariant to their former scale because the minus operation of μ and the division operation of σ decorrelate features with the scale.

After the concatenation operation, Conv1D is used to exploit action-related information for generating the category distribution at the current time step, as shown in Fig. 3. In this work, the 1D convolutions operate on the temporal dimension with the kernel size of p. Note that the outputs of the current time step are derived from the "past" input data, so the very first of the input features should be padded with $p-1$ zero vectors. The number of input channels of the 1D convolutional kernels is the dimension of the input feature vector.

B. Incremental Confidence Learner

The incremental confidence learner is developed with the hypothesis that the confidence value depends on whether the currently observed information has occurred before (freshness) and whether the currently observed information is helpful to predict the action category (criticality). The confidence $\hat{c}^t$ at the time step t is calculated as

$$\hat{c}^t = \begin{cases} 0, & t = 0,\\ \hat{c}^{t-1} + \Delta^t, & t \in \mathbf{N}^+, \end{cases} \tag{2}$$

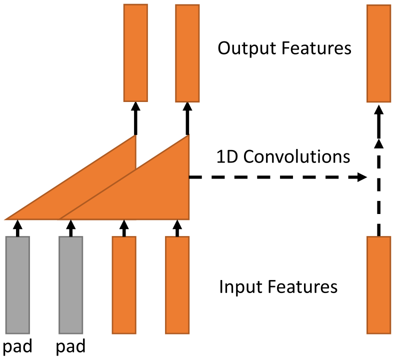


Fig. 3. Expanding 1D convolutions through time.

where Δ^t is the increment of the confidence value depending on the frame-level motion feature $\boldsymbol{v}^t$ and the frame-level scene feature $\boldsymbol{r}^t$, and $\mathbf{N}^+$ is a set of all the positive integers. For clarity, we define the feature $\boldsymbol{f}^t$ for generating the confidence value as

$$\boldsymbol{f}^t = f(\boldsymbol{v}^t, \boldsymbol{r}^t), \tag{3}$$

where $f(\cdot)$ is a nonlinear function implemented by a neural network that takes the concatenation of $\boldsymbol{v}^t$ and $\boldsymbol{r}^t$ as input. Using $\boldsymbol{f}^t$, the confidence increment Δ^t is given by

$$\Delta^t = \omega^t \cdot g(\boldsymbol{f}^t), \tag{4}$$

where $g(\cdot)$ maps $\boldsymbol{f}^t$ from high dimensional vector space to the positive real space $\mathbf{R}^+$ to guarantee that the confidence value is monotonically non-decreasing over time. ω^t is a scale factor that measures the dissimilarity of frames between the current and previous time steps:

$$\omega^t = \begin{cases} 1, & t = 1, \\ \min_{i<t}(d(\boldsymbol{f}^t, \boldsymbol{f}^i)), & t \in \mathbf{N}^+ \text{ and } t > 1. \end{cases} \tag{5}$$

$d(\cdot,\cdot)$ calculates the distance as

$$d(\boldsymbol{f}^t, \boldsymbol{f}^i) = \frac{1- < \boldsymbol{f}^t, \boldsymbol{f}^i >}{2}, \tag{6}$$

where $< \cdot, \cdot >$ denotes the cosine similarity. To hold the range of the confidence value, we limit the confidence value as

$$c^t = \gamma \cdot \text{sigmoid}(\eta \hat{c}^t), \tag{7}$$

where sigmoid$(\cdot)$ is a sigmoid activation function, and γ and η are the hyper parameters controlling the range and steepness of the sigmoid function, respectively.

C. Self-Refining Gumbel Softmax

The action category probability distribution at each time step can be obtained through a fully connected operation and a softmax activation to the final feature. Let $\{\pi_i^t | i = 1, 2, \cdots, L\}$ be the obtained action category probability distribution at time step t, where L is the number of action categories, and the action category is predicted from the distribution $\{\pi_i^t\}$. To fully exploit the action distribution information, we predict the action category by sampling from the distribution with the Gumbel max trick which has been successfully used in training generative models and reinforcement learning [72], [73]. The Gumbel max is formulated as

$$\begin{aligned} \tilde{\boldsymbol{y}}^t &= \text{onehot}(\arg\max_j \acute{y}_j^t, L), \\ \acute{y}_j^t &= \log(\pi_j^t) + g_j^t, \end{aligned} \tag{8}$$

where onehot(i, L) is a function returns a one-hot vector with the length of L and the i-th element value of 1, and g_i^t is a random variable drawn from the Gumbel$(0, 1)$ distribution and independent with each other.

Eq. (8) cannot be used as the final action category distribution for three considerations. First, $\tilde{\boldsymbol{y}}^t$ is non-differentiable due to the arg max function, which results in that the network cannot be trained via the back-propagation algorithm. Second, there should be other action category distribution vectors between different one-hot vectors since the ambiguous information at the early stages of typical realistic videos makes it difficult to give correct predictions. For example, the sub-action "run" would appear at the early stage of either the action "long jump" or the action "high jump", which indicates that the two actions have a common sub-action, and the transition between the two one-hot distribution vectors of the actions should be smooth. Third, the confidence at the time step t in Eq. (7) should be taken into account to ensure that a high confidence brings a sharp distribution and vice versa. In other words, the category distribution in this work should be converged from continuous to discrete along with the growing prediction confidence from the observer instead of the increasing training epochs.

Based on the above considerations, we propose a self-refining Gumbel softmax sampler for jointly learning the category distribution and the prediction confidence. At the time step t, the output distribution of the i-th category is refined by the confidence c^t, expressed as

$$y^t = \frac{\exp(c^t(\log(\pi_i^t) + g_i^t))}{\sum_{j=1}^{L} \exp(c^t(\log(\pi_j^t) + g_j^t))}. \tag{9}$$

It can be observed that $\boldsymbol{y}^t$ equals to taking softmax transformation on $c^t \cdot \acute{\boldsymbol{y}}^t$, which is differentiable and used to approximate the operator in Eq. (8). The term g_i^t makes it possible to choose the action label with non-maximum probability when the output probability is close to a uniform distribution. Once the prediction is right, the confidence value will become larger, which means that the potential information is discovered. The output distribution will become sharper and approximate to the one-hot distribution when c^t grows larger, since that

$$\tilde{\boldsymbol{y}}^t = \lim_{c^t \to +\infty} \boldsymbol{y}^t. \tag{10}$$

The prediction confidence and the category distribution are jointly learned by optimizing SPR-Net with self-refining Gumbel softmax. During training, the refinement of category distribution by the confidence enables the model to learn more discriminative features for action prediction, since the learned features gain more attentions when the self-refining Gumbel sampler generates sharper prediction distribution with higher confidence. Moreover, the prediction of the category distribution also has a positive effect on the confidence to enable the

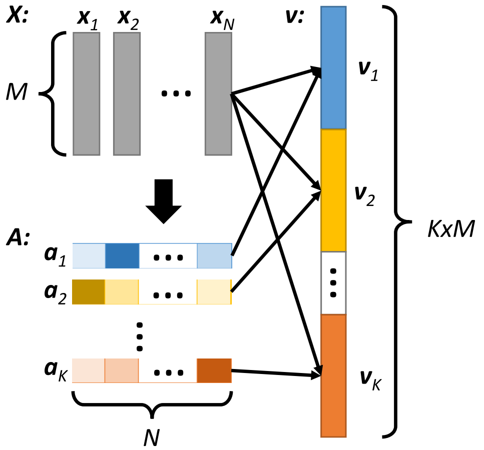


Fig. 4. Illustration of the proposed sparse self-attention module. The attention vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K$ are generated by different non-linear functions with the local spatio-temporal features $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ as input. After K times weighted addition of the local spatio-temporal features by the attention vectors, we get $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K$ and concatenate them into a vector $\boldsymbol{v}$ for the final motion representation.

proposed model to make timely correct decisions. In particular, the predictions guide the adjustment of the confidence during training, which helps the model locate the earliest frame in a video that can produce a large confidence increment in a video in the test phase. Consequently, the processes of predicting the category distribution and learning the prediction confidence are mutually beneficial to each other for high predicting accuracy and low observational latency.

IV. Sparse Self-Attention (SSA) of Motions

In order to effectively exploit the motion information with close relevance to the action prediction task, we propose a sparse self-attention model to learn the frame-level motion features, as shown in Fig. 4. This mechanism allows the local spatio-temporal features extracted by Res3D to be aggregated into the frame-level motion feature at each time step. Since only the prediction-relevant action information is needed, the proposed SSA module is integrated into SPR-Net for learning to highlight most useful local features and get rid of noisy ones in an end-to-end manner. Considering that there are multiple salient local features in a video clip, SSA applies attention operations more than once for each group of local features at each time step and concatenates the output attended features into the frame-level motion feature. A sparse mechanism and a penalty term are introduced to generate attention weights via sparsemax for reducing the redundancy. The penalty term keeps the diversity of attention operations at each time step.

To be specific, the local spatio-temporal features of the i-th frame in a video are obtained by using the $(i-15)$-th to the i-th frames as input of Res3D. When $i \leq 15$, the input frames are padded with the first frame of the video. The local spatio-temporal features of each frame are represented by a matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbf{R}^{M\times N}$, where $\boldsymbol{x}_n \in \mathbf{R}^{M\times 1}$ represents the feature vector of the n-th spatial grid, N is the number of spatial grids, and M is the number of feature maps. Given $\boldsymbol{X}$ as input, we apply K times attention operations to obtain features $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_K$, where $\boldsymbol{v}_k$ represents the feature vector after the k-th attention operation given by

$$\boldsymbol{v}_k = \boldsymbol{X}\boldsymbol{a}_k^{\mathsf{T}}. \tag{11}$$

$\boldsymbol{a}_k = [a_k^1, \cdots, a_k^N] \in \mathbf{R}^{1\times N}$ stands for the k-th attention weights. The frame-level motion feature $\boldsymbol{v}$ at each time step is described by the concatenation of $\boldsymbol{v}_k$, $k = 1, 2, \ldots, K$. Note that there is no need to align them before concatenation, because all the weighted features lie in a space spanned by the local spatio-temporal features $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. The space of the k-th weighted feature is formulated by $\{\sum_{i=1}^{N} a_k^i \boldsymbol{x}_i \mid \sum_{i=1}^{N} a_k^i = 1, a_k^i \geq 0\}$.

The K groups of attention weights are calculated by using K non-linear mapping functions. It is equivalent to calculating a matrix of attention weights $\boldsymbol{A} = [\boldsymbol{a}_1^{\mathsf{T}}, \boldsymbol{a}_2^{\mathsf{T}}, \ldots, \boldsymbol{a}_K^{\mathsf{T}}]^{\mathsf{T}}$:

$$\begin{aligned} \boldsymbol{A} &= \mathrm{sparsemax}(\hat{\boldsymbol{A}}), \\ \hat{\boldsymbol{A}} &= \boldsymbol{W}_1 \mathrm{ReLU}(\boldsymbol{W}_2 \boldsymbol{X} + \boldsymbol{b}_2) + \boldsymbol{b}_1, \end{aligned} \tag{12}$$

where $\boldsymbol{W}_1 \in \mathbf{R}^{K\times d}$ and $\boldsymbol{W}_2 \in \mathbf{R}^{d\times M}$ are the weight matrices. $\boldsymbol{b}_1 \in \mathbf{R}^{K\times N}$ and $\boldsymbol{b}_2 \in \mathbf{R}^{d\times N}$ are the biases matrices that are derived from two vectors of K and d dimensions tiled by N times. sparsemax$(\cdot)$ denotes the sparsemax operation which is performed along the second dimension of the input matrix.

Since the sparsemax is operated in rows, we just take the k-th row $\hat{\boldsymbol{a}}_k = [\hat{a}_k^1, \ldots, \hat{a}_k^N]$ in $\hat{\boldsymbol{A}}$ for explanation. For $\hat{\boldsymbol{a}}_k$, the sparsemax aims at optimizing

$$\mathrm{sparsemax}(\hat{\boldsymbol{a}}_k) = \arg\min_{\boldsymbol{a}_k \in \mathbf{B}^N} ||\boldsymbol{a}_k - \hat{\boldsymbol{a}}_k||_2^2, \tag{13}$$

where $\mathbf{B}^N = \{\boldsymbol{b} \in \mathbf{R}^N | \mathbf{1}^{\mathsf{T}}\boldsymbol{b} = 1,\ \boldsymbol{b} \geq 0\}$ is the N-dimensional simplex, and $\mathbf{1}$ is a N-dimensional vector with all the elements equal to 1. The optimization problem in Eq. (13) can be solved by projecting the point $\hat{\boldsymbol{a}}_k$ onto the simplex $\mathbf{B}^N$.

Concretely, firstly the elements of the vector $\hat{\boldsymbol{a}}_k$ are sorted in a descending order, given by $\hat{a}_k^{(1)} \geq \hat{a}_k^{(2)} \geq \ldots \geq \hat{a}_k^{(N)}$, where $\hat{a}_k^{(i)} \in \{\hat{a}_k^1, \ldots, \hat{a}_k^N\}$, $\forall i \in \{1, \ldots, N\}$. The subscript i^* is then selected to satisfy

$$i^* = \max\{i \in \mathbf{N}^+ | 1 + i\hat{a}_k^{(i)} > \sum_{j\leq i} \hat{a}_k^{(j)},\ i \leq N\}. \tag{14}$$

Then, the function $\delta(\cdot) \in \mathbf{R}$ is defined as

$$\delta(\hat{\boldsymbol{a}}_k) = \frac{1}{i^*}(\sum_{j\leq i^*} \hat{a}_k^{(j)} - 1). \tag{15}$$

Finally, the solution of the sparsemax $\boldsymbol{a}_k$ is described as

$$\boldsymbol{a}_k = \max(\hat{\boldsymbol{a}}_k - \delta(\hat{\boldsymbol{a}}_k) \cdot \mathbf{1}, 0). \tag{16}$$

To ensure the diversity of the K groups of attention weights, a penalty term P is needed as follows:

$$P = ||\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}} - \boldsymbol{A}\boldsymbol{A}^{\mathsf{T}} \odot \boldsymbol{I}||_{\mathrm{F}}^2, \tag{17}$$

where $||\cdot||_{\mathrm{F}}$ is Frobenius normalization, and $\boldsymbol{I}$ is an identity matrix. The penalty term P restricts the value of non-diagonal elements in $\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}$ to approximate to 0, where each element of $\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}$ is the summation of the corresponding elements of two attention weight vectors: $\boldsymbol{a}_i\boldsymbol{a}_j^{\mathsf{T}}$, $\forall i \neq j$. Thus there will be few overlaps between $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$, which means that the i-th and the j-th attention operations have selected different salient features. Compared with [74] that uses softmax activation for generating attention distributions, the proposed penalty term need not add the L2 regularization to the diagonal elements for sparsity. Moreover, it is easy to optimize the term P because the attention weight vectors are sparse via the sparsemax activation.

V. Experiments and Discussion

A. Datasets

Extensive experiments are conducted on the UT-Interaction [75], BIT-Interaction [76], UCF101 [77], THUMOS14 [78], and ActivityNet [79] datasets to evaluate the proposed method by reporting the mean accuracy of predictions.

1) UT-Interaction: The UT-Interaction dataset consists of videos covering six human-human interaction categories, i.e., shake-hands, point, hug, push, kick and punch. Not only actors are contained in the videos, but also some irrelevant pedestrians are present. The dataset contains 120 videos and is divided into two sets. We evaluate the action prediction performances on this dataset using 10-fold leave-one-out cross validation per set by following [75].

2) BIT-Interaction: The BIT-Interaction dataset contains eight human interaction categories in realistic scenes, i.e., bow, boxing, handshake, high-five, hug, kick, pat and push. The videos are captured with cluttered backgrounds, moving objects and partial occluded body parts. It totally consists of 400 videos, with 50 videos per class. Each video is annotated by one category label. 272 videos are randomly sampled for training and the rest is for testing, which is the same as the setting in [76].

3) UCF101: The UCF101 dataset comprises of 101 action categories with 13,320 realistic video clips which are collected from YouTube. The dataset can be divided into five types: Human-Object Interaction, Body-Motion only, Human-Human Interaction, Playing Musical Instruments and Sports. The large variations in camera motion, object scale, viewpoint and illumination conditions make the UCF101 dataset extremely challenging. This dataset has three standard training/test splits for evaluation and we follow the setting in [77].

4) THUMOS14: The THUMOS14 dataset has the same action categories with the UCF101 dataset. It contains $2,584$ untrimmed videos, which are split into a validation set with $1,010$ videos and a testing set with $1,574$ videos. Practically, there are only 20 action categories used for evaluating the methods of temporal action localization [80], [81]: BaseballPitch, BasketballDunk, Billiards, CleanAndJerk, CliffDiving, CricketBowling, CricketShot, Diving, FrisbeeCatch, GolfSwing, HammerThrow, HighJump, JavelinThrow, LongJump, PoleVault, Shotput, SoccerPenalty, TennisSwing, ThrowDiscus, and VolleyballSpiking. The videos in the THUMOS14 dataset are all untrimmed videos, each of which contains no less than one action of the same category. We choose videos from the commonly used 20 categories to evaluate the action prediction performances on untrimmed videos. During training, we cut the selected videos into short clips according to the following standard. Only the first action with non-action related frames before the starting of the action in a video is clipped as the experimental videos. The action-irrelevant frames ahead of the actions are randomly selected ranging from 0% to 30% of the length of their corresponding actions, which produces the length variation of videos to make the dataset more challenging. As a result, in total we collect 413 videos of 20 action categories, where 200 videos collected from the validation set are regarded as the training data, and 213 videos collected from the testing set serve as the testing data. Videos of the actions of "CliffDiving" and "Diving" are similar with different annotations, so we merge the two categories and the total number of action categories on the THUMOS14 dataset is 19 in our experiment. During testing, although each video in the THUMOS14 dataset contains multiple action segments of the same category, for easily observable results, we report the performances on the first action of each video which is operated as the training data.

5) ActivityNet: The ActivityNet dataset contains 4,819 training and 2,383 validation videos, which totally provides 7,202 untrimmed videos of 100 actions. Each video contains about 1.7 action instance in average. On this dataset, we conduct the same pre-processing operation as THUMOS14.

Note that we use the THUMOS14 and the ActivityNet datasets instead of other small-scale video datasets for the evaluation on untrimmed videos because they are the most challenging video datasets that contain more realistic and complex action videos.

B. Implementation Details

1) Feature Representation: The frame-level scene and local spatio-temporal features are extracted from ResNet-152 [69] and Res3D [70], respectively. For the scene feature, each frame is extracted from the last convolutional layer (pool5) of ResNet-152 which is pre-trained on the large scale ImageNet dataset. For the local spatio-temporal feature, the local spatio-temporal features are derived from the spatial grids of the last convolutional layer of Res3D before the pooling operation (res5b). Res3D is pre-trained on the Sports-1M dataset. We fine-tune Res3D using the training data of the BIT-Interaction, UT-Interaction and the UCF101 datasets for their respective experiments. For the THUMOS14 and ActivityNet datasets, we do not fine-tune Res3D for the reason that the background parts of untrimmed videos might mislead the model on representing motion features while fine-tuning. The local spatio-temporal features are then aggregated spatially by the sparse self-attention module into the frame-level motion feature.

2) Data Augmentation: To avoid the over-fitting problem, several data augmentation strategies are employed. We make geometric transformations following [8]. The color transformation is performed by shifting the HSV channels. The values of S and V are adjusted by factors randomly selected in the range of 0.9 to 1.1. Note that, we also randomly cut off the first few frames of videos to augment videos, and the numbers

TABLE I

ACCURACIES (%) OF DIFFERENT METHODS ON THE UNTRIMMED VIDEO DATASETS

Method	THUMOS14			ActivityNet		
	0.1	0.5	1.0	0.1	0.5	1.0
MTSSVM [3]	50.9	55.4	59.4	31.4	43.9	49.0
DeepSCN [7]	53.0	64.8	67.8	37.7	48.6	55.3
MS-LSTM [8]	55.2	66.1	69.9	41.9	54.7	56.8
Mem-LSTM [9]	59.4	70.2	72.8	39.0	55.3	59.4
MSRNN [27]	58.9	**70.5**	71.5	39.5	56.7	60.0
Ours	**64.3**	**70.5**	**73.7**	**44.2**	**57.3**	**61.2**

of these frames are limited to less than one-third of the total frame numbers of their corresponding videos.

3) Parameter Setting: For the temporal hybrid network, the units of the GRU layers are empirically set to 1,024. The kernel size p of Conv1D is set to 5, and the filter size is set to 128. For the incremental confidence learner, we use two fully connected layers with 128 and 1 units, and a ReLU activation to learn the confidence value. For the self-refining Gumbel softmax, the parameters γ and η in Eq. 7 are set to 2 and 0.01, respectively. For SSA, the number of attention operations K is set to 3. We use the cross-entropy loss to optimize the proposed model. The coefficients of the cross-entropy loss function and the penalty term are set to 1 and 50, respectively, for balancing their order of magnitude. We use dropout with the rate of 0.5 and the Adam optimization algorithm for training. Our model is implemented using TensorFlow [82] tool on a Titan X GPU with 12G memory.

C. Results on Untrimmed Video Datasets

1) Comparison With State-of-the-Art Methods: To evaluate the effectiveness of our proposed method for predicting actions in untrimmed videos, we compare our method with several state-of-the-art methods: MTSSVM [3], DeepSCN [7], MS-LSTM [8], mem-LSTM [9], and MSRNN [27]. For fair comparison, all these methods use both ResNet and Res3D features as video representations which is the same as our method. Specifically, for MS-LSTM, mem-LSTM and MSRNN, we connect the SSA module with these networks to integrate the local spatio-temporal Res3D features into global motion features. Since MTSSVM and DeepSCN are not end-to-end trainable frameworks, SSA cannot be applied to them. So we aggregate the local spatio-temporal features of the res5b layer of Res3D using average pooling and concatenate this feature with the ResNet-152 feature after $l2$ normalization as the input features of MTSSVM and DeepSCN (we also tried other feature fusion strategies, and the best results are derived from using this strategy). We report the action prediction accuracies at observation ratios {0.1, 0.5, 1.0} on untrimmed THUMOS14 and ActivityNet datasets in Table I, which shows that SPR-Net outperforms other models on both datasets. Particularly, it is clearly observed that our model achieves much higher accuracies of 64.3% and 43.6% at the observation ratio of 0.1 than other models, respectively, which further validates the effectiveness of our method on predicting actions in early parts of videos.

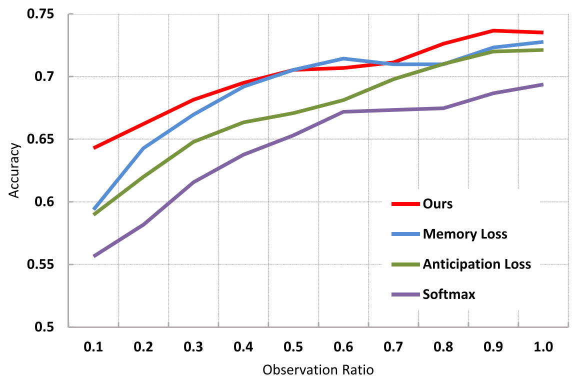


Fig. 5. Performances of our model with different losses for action prediction on the THUMOS14 dataset.

2) Evaluation on the Self-Refining Mechanism With Confidence: The self-refining mechanism is implemented by the union of the self-refining Gumbel softmax sampler and the incremental confidence learner. In this subsection, we first separately evaluate the Gumbel softmax sampler and the confidence learner, and then show and analyze some qualitative results of the self-refining mechanism.

We compare our self-refining Gumbel softmax loss with other three losses: softmax loss, anticipation loss [8] and memory loss [9]. For fair comparison, same input features and a same structure of temporal hybrid model are used for these losses. Fig. 5 illustrates the prediction accuracies of different losses at different observation ratios on the THUMOS14 dataset. The accuracy gaps between our loss and other losses are obviously large at the early stage (observation ratio $= 0.1, 0.2, 0.3$), which verifies that our method is effective on coping with the irrelevant information in untrimmed videos for action prediction, and our network is able to yield correct prediction as soon as the discriminative information appears. The accuracy gaps among these methods become small as the observation ratio close to 1.0 (i.e. the entire video is observed), which indicates that SSA and temporal hybrid network can stably represent the global action information in videos with a relatively reasonable loss function as guidance.

To validate the rationale of setting the confidence value to be monotonically non-decreasing over time, we remove the constraint of non-decreasing involved in the incremental confidence learner of our model referred to as "w/o Increment". As a contrast, we call our full model "Increment". Table II shows the comparisons of the accuracies and average confidence values between the two models. As can be observed from the table, the confidence value of "w/o Increment" is still growing over time, and the action prediction performance declines slightly. We show the changes of average confidence values of the test samples at each time step during training in Fig. 6. An interesting phenomenon is that the confidence value grows larger with the increasing training epochs, which further confirms that the confidence is the assessment of the decision quality of a prediction model and the level of confidence grows with the gain of information. We also note that curves of "Increment" is relatively smooth.

TABLE II

ACCURACIES (%) OF DIFFERENT CONFIDENCE SETTINGS ON THE THUMOS14 DATASET

Observation Ratio	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Confidence Value										
w/o Increment	1.631	1.681	1.727	1.771	1.810	1.849	1.881	1.927	1.960	1.982
Increment	1.631	1.678	1.725	1.766	1.804	1.844	1.888	1.915	1.946	1.975
Prediction Accuracy										
w/o Increment	62.9	65.6	66.1	67.4	68.3	**70.9**	70.7	71.4	73.2	72.8
Increment	**64.3**	**66.2**	**68.2**	**69.5**	**70.5**	70.7	**71.1**	**72.6**	**73.7**	**73.5**

TABLE III

ACCURACIES (%) OF DIFFERENT TEMPORAL NETWORKS ON THE THUMOS14 DATASET

Structure	Observation Ratio									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Conv1D + GRU	63.3	64.4	65.5	67.3	68.4	69.9	70.0	70.7	71.0	71.1
Conv1D + Conv1D	64.1	64.5	65.3	66.4	68.1	68.8	69.4	69.6	69.9	69.6
GRU + GRU	63.7	**67.2**	68.1	68.6	68.8	70.4	**71.5**	71.3	71.4	71.5
GRU + Conv1D (ours)	**64.3**	66.2	**68.2**	**69.5**	**70.5**	**70.7**	71.1	**72.6**	**73.7**	**73.5**

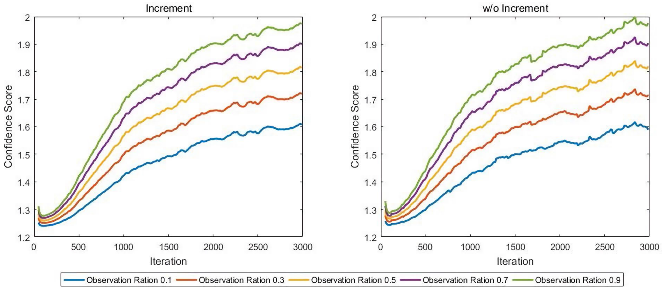


Fig. 6. Trends of average confidence values at each time step during training of different confidence settings.

The confidence value without drastic changes could guarantee a stable model during training. It might explain the reason why "Increment" performs slightly better than "w/o Increment".

Some qualitative experiments of the self-refining mechanism are conducted, and the results are shown in Fig. 7. It is interesting to notice that a significant performance improvement is always accompanied with an large increase of the confidence value. The increasing rate of the confidence and the correctness of the prediction are positively related, which is consistent with our statement that the confidence and prediction decision affect each other. An obvious performance improvement of our method over the method with the anticipation loss clearly validates the effectiveness of the proposed self-refining mechanism. Different from the anticipation loss using temporal information as the penalty for prediction, our method refines the prediction results according to the learned confidence value.

Since the confidence value changes with the variation of video content and quickly increases when the prediction turns out to be right, our method is more reasonable and close to the intuitive sense than using the temporal information. Especially in untrimmed action videos, our method is able to distinguish the action related and unrelated information via the confidence learning. Compared with the memory loss which mainly focuses on remembering video segments that are "hard to predict", our method aims at exploiting rather than remembering the discriminative information from these kinds of segments using the confidence mechanism. Thus, our method is more suitable for predicting untrimmed videos that contain various background clips before action starts.

3) Evaluation on the Temporal Hybrid Network: To evaluate the effectiveness of the temporal hybrid structure, several comparative experiments have been conducted by randomly selecting one module from GRU and Conv1D twice to generate four temporal networks, i.e., GRU+GRU, Conv1D+GRU, Conv1D+Conv1D and GRU+Conv1D (ours). Table III compares the results of the four temporal networks. It is interesting to observe that:

(i) At the large observation ratios, the networks containing at least one GRU perform better. It is because GRU can record video information from the beginning of the video, while Conv1D can only capture information from a few time steps before the current time limited by its kernel size.

(ii) At the early stage, the networks with Conv1D as the second layer outperforms others, which shows the better performance of the 1D convolutional operation than GRU on characterizing video clips with the same length for classification.

(iii) The networks with Conv1D as the first layer show overall relatively bad performance. The main reason is that the cascade of ResNet and the first Conv1D are not so complementary to the Res3D to capture enough useful clues for classification. In other words, they both operate convolutions along the space and time dimensions, which leads to that the scene and motion features share similar patterns.

4) Evaluation on SSA: In order to evaluate the contribution of SSA to the prediction performance, we perform an ablation study on the amount of the attention operations K, the sparsemax, and the importance of the penalty term in Eq. 17, as shown in Table IV. The 1-st row in Table IV indicates the method using average pooling. The 2-nd row

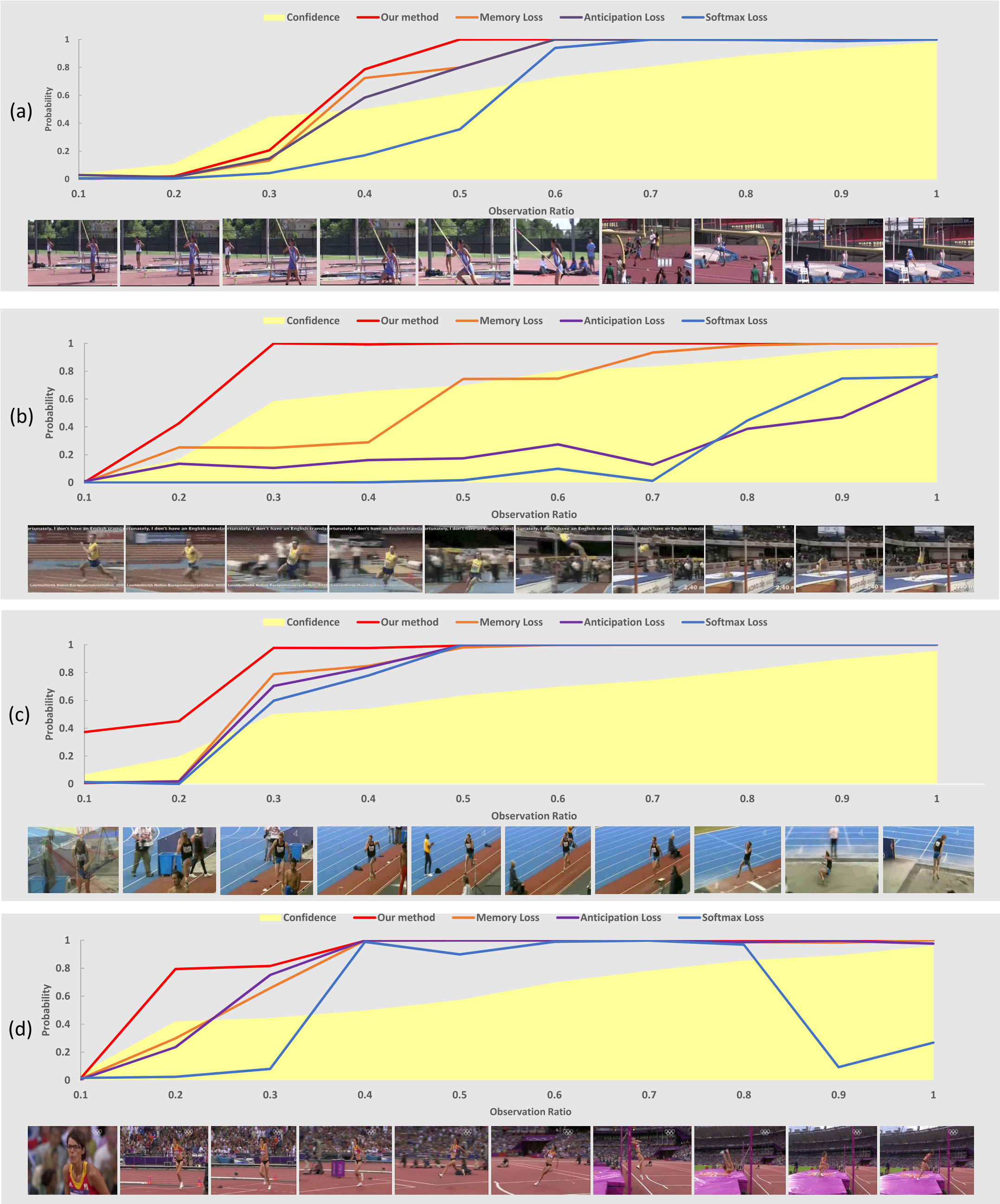
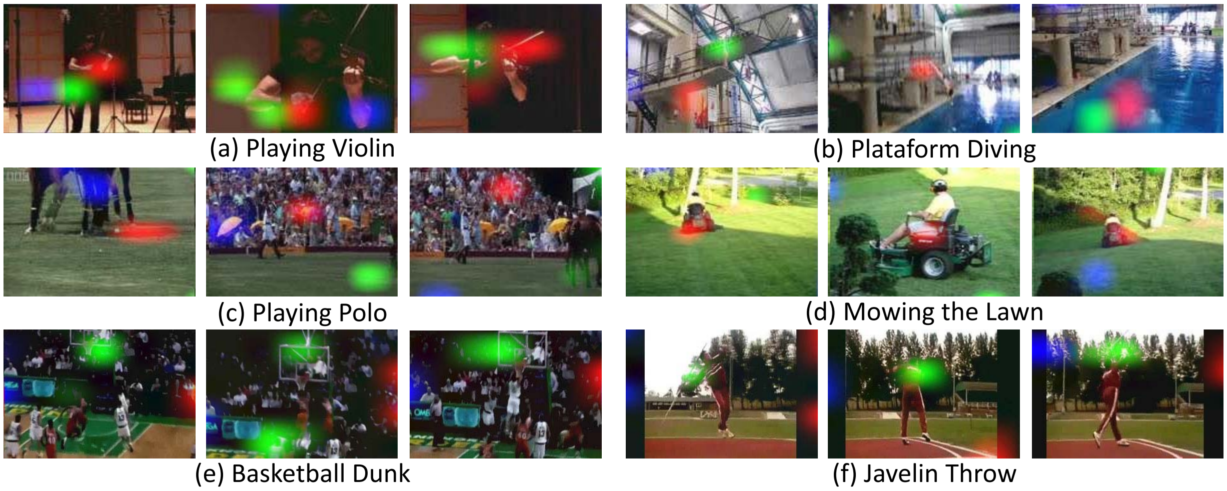


Fig. 7. Examples of qualitative results of the self-refining mechanism on the THUMOS14 dataset. There are four groups of illustrations in (a), (b), (c) and (d). The vertical axis of each chart indicates the output probabilities of the ground truth action category, and the horizontal axis indicates the observation ratio. For each group, the curves in the chart records the output probabilities of different methods, and the yellow shadow represents the trend of the confidence values over time.

indicates a vanilla attention method with sparsemax. The 3-rd, 6-th, 7-th and 8-th rows represent the proposed sparse self-attention mechanism with different numbers of attention operations, with or without the sparsemax, and with or without the penalty term.

From Table IV, we can observe that the methods of the 3-rd to the 8-th rows perform much better than the method of the 1-st row, which verifies that the self-attention mechanism is useful for representing videos. Comparing the 3-rd, 6-th and 7-th rows, we can qualitatively conclude that performing more attention operations can learn more discriminative representation to improve the prediction performance. When K grows to 4 (the 7-th row), the performance slightly decreases. The probable reasons are: (i) There may not be many useful regions in a video to represent actions, and a large K might cause the information redundancy. (ii) Too many model parameters caused by a large K may cause the overfitting problem while training. The results of the 4-th and the 6-th rows suggest that the proposed attention mechanism benefits from the sparsity of the attention weights. The results


(a) Playing Violin

(b) Plataform Diving

(c) Playing Polo

(d) Mowing the Lawn

(e) Basketball Dunk

(f) Javelin Throw

Fig. 8. Visualization of attention maps on the THUMOS14 and the ActivityNet datasets. Attended regions of different attention operations are shown by different colors. Since each figure herein is the last frame of the video subsegment for extracting local spatio-temporal features, the attention regions in the frame might be slightly drifted.

TABLE IV
ABLATION STUDIES OF THE SELF-ATTENTION MECHANISM ON THE THUMOS14 DATASET

No.	K	Attention	Sparse	Penalty Term	Observation Ratio		
					0.1	0.5	1.0
1	-	×	×	×	55.3	59.2	61.0
2	1	√	√	×	61.5	68.4	68.7
3	2	√	√	√	64.2	69.2	70.9
4	3	√	×	√	61.6	66.1	69.2
5	3	√	√	×	62.9	66.8	71.0
6	3	√	√	√	64.3	**70.5**	**73.5**
7	4	√	√	√	**66.5**	68.3	68.3
8	5	√	√	√	63.8	62.9	68.3

TABLE V
ACCURACIES (%) OF DIFFERENT METHODS ON THE UT-INTERACTION DATASET

Method	Observation Ratio		
	0.1	0.5	1.0
D-BoW [1]	17.4	70.0	85.0
I-BoW [1]	15.2	65.0	81.7
CuboidBayes [1]	–	25.0	71.7
Hierarchical Movemes[4]	35.4	76.1	85.8
MTSSVM [3]	34.7	76.7	90.2
DP-SVM [10]	12.5	13.0	14.6
S-SVM [10]	11.0	11.0	13.4
MMAPM [24]	41.8	76.7	90.8
MS-LSTM [8]	–	84.0	90.0
Structural Context Model [83]	50.8	83.3	**92.5**
Ours (cropped)	**79.2**	**85.3**	90.5
Ours (uncropped)	55.8	85.0	91.7

of the 5-th and the 6-th rows show the importance of the penalty term.

We also visualize the learned attention weights of different attention operations of several examples from the THUMOS14 and ActivityNet datasets. We upsample the attention weights calculated by each attention operation to the size of the corresponding video frame by cubic interpolation. From Fig. 8, it can be seen that the attentions are usually focus on the most relevant regions, such as the violin in (a), water in (b), and audience in (c) and (e). We also observe that the attended regions are always located in the black areas of the frames with relative simple scene such the figure in (f). It is probably due to that the proposed model prefers black information rather than too much noisy information for classification.

D. Results on Trimmed Video Datasets

We also apply our method to trimmed videos to further evaluate its performance.

1) Comparison With State-of-the-Art Methods: We compare our method with several state-of-the-art methods on three trimmed video datasets, i.e., the UT-Interaction, the BIT-Interaction and the UCF101 datasets. According to the evaluation standards of some off-the-shelf methods of action prediction [7]–[9] on the three datasets, we report the prediction accuracies at different observation ratios of $\{0.1, 0.5, 1.0\}$.

On the UT-Interaction dataset, we conduct experiments using two versions of the dataset provided by the website.[1] The difference between the two versions is whether video frames are cropped based on the ground truth bounding box. Table V shows that our method performs better than most of the state-of-the-art methods on the UT-Interaction dataset. It is worth noting that at the observation ratio of 0.1, the proposed model outperforms all the other methods, which demonstrates the effectiveness of our method on the action prediction task. We also observe that at the observation ratio of 1.0, the advantage of our method is not apparent compared with some methods [3], [24] which use hand-crafted features and SVMs to predict actions. The probable reason is that it is much easier to run into the overfitting problem when training our deep network on the small dataset, which degrades the prediction performance. [83] performs better

[1] http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html

TABLE VI

ACCURACIES (%) OF DIFFERENT METHODS ON THE BIT-INTERACTION DATASET

Method	Observation Ratio		
	0.1	0.5	1.0
MTSSVM [3]	28.9	60.0	78.7
MMAPM [24]	32.8	68.0	79.7
DeepSCN [7]	37.5	78.1	90.6
Global-Local Model [26]	27.6	79.4	85.3
Structural Context Model [83]	41.4	79.7	86.0
Ours	**84.1**	**100**	**99.2**

TABLE VII

ACCURACIES (%) OF DIFFERENT METHODS ON THE UCF101 DATASET

Method	Observation Ratio		
	0.1	0.5	1.0
D-BoW [1]	36.3	53.8	75.8
I-BoW [1]	36.3	73.9	75.8
MTSSVM [3]	40.1	62.0	82.7
DeepSCN [7]	45.0	85.8	88.5
Mem-LSTM [9]	51.0	88.4	90.5
MSRNN [27]	68.0	89.3	90.9
Ours	**88.7**	**91.6**	**91.4**

that our method at the observation ratio of 1.0. Probably due to that [83] is particularly designed for interaction video prediction and they train a detector using extra bounding box annotations to crop the body parts of each actor in the video for more precise action representation. The results on the BIT-Interaction dataset are reported in Table VI. It is obvious that our method achieves a significant improvement of performance on the dataset and even gets 100% accuracy when the observed part is just half of the full video.

We notice that our method achieves impressive performances at the observation ratio of 0.1 on the aforementioned two datasets. It indicates that our model is able to discover discriminative information from the video frames for early action prediction. Besides, our model can be trained in an end-to-end manner which has the merit of learning finer and relevant features for classification, while the compared frameworks except [8], [83] on the two dataset are not end-to-end trainable. It is obvious that informative clues at the early stage of videos are subtle and hard to discover. And especially, our method contains the mechanisms that are dedicated to processing discriminative information, such as Conv1D for learning discriminative representation, SSA for discovering salient information, and the confidence for judging the quality of decisions derived from the video.

Experiments are also conducted on the more challenging UCF101 dataset. The results shown in Table VII demonstrate the superiority of our method. Especially at the early stage, our method performs much better than prior works, which proves that SPR-Net is able to predict actions with high accuracy and low latency of classification by effectively removing noisy information. When the videos are fully observed, our method achieves comparable results with [7], [9]. The reason might come from the different video features. Using diverse features for representing videos could contribute to the high performance, such as the combination of either optical flow or dense trajectories [9] and the RGB based features. In this paper, we just use RGB based features to guarantee the efficiency of our method. Because calculating the optical flow and trajectories is time consuming, and if the model cannot output predictions promptly, it will lose its practical meaning. What's more, this paper focuses on the action prediction task, i.e., to make accurate prediction with low observational latency, rather than the action recognition problem which classifies the actions after the video is fully observed.

TABLE VIII

ACCURACIES (%) OF DIFFERENT METHODS USING DIFFERENT INPUT FEATURES ON THE UCF101 DATASET

Feature	Method	Observation Ratio		
		0.1	0.5	1.0
2D CNN	MS-LSTM [8]	66.7	79.5	82.4
	Ours	**68.1**	**83.7**	**85.1**
Two-stream CNN	Mem-LSTM [9]	51.0	88.4	90.5
	MSRNN [27]	68.0	89.3	**90.9**
	Ours	**73.4**	**89.5**	90.7
3D CNN	DeepSCN [7]	45.0	85.8	88.5
	Ours	**76.3**	**89.7**	**91.2**

On both BIT-Interaction and UCF101 datasets, we find that the accuracies drop slightly when the observation ratio rise from 0.5 to 1.0. There are two possible reasons for this phenomenon: (1) the gradient vanishing problem may exist in the GRU model of SPR-Net, and the temporal information may be lost during the back propagation of gradient which degrades the accuracy; (2) the settings of kernel size of Conv1D is and the number of Conv1D layer lead to the receptive field less than the temporal length, which might cause some important information of the early stage being missing.

2) Comparison of Different Features: We report experimental results of our method using different features on the UCF101 dataset to validate the generalization of our model and to provide fair comparisons between our method and other state-of-the-art methods on this dataset. Three kinds of features used in [7]–[9], [27] are employed for comparison:

- 2D CNN. We follow the settings in [8] and extract the context-aware and action-aware features from the VGG-16 [84] as the input of SPR-Net.
- Two-stream CNN. Following [9], VGG-19 [84] and ResNet-18 [69] are pre-trained for the optical flow and RGB images on the UCF101 dataset, respectively. And we replace the Res3D and ResNet-152 features with the extracted two-stream features from VGG-19 and ResNet-18, respectively.
- 3D CNN. Following [7], the features are extracted from C3D [85] pre-trained on the Sports-1M dataset. To fed the features into SPR-Net, we leverage the 4096-d feature of the fc7 layer as the input to the GRU module and leverage the features of the pool5 layer as the local spatio-temporal features.

Table VIII shows the comparison results. It is evident that our method outperforms others when using the same input features.

VI. Conclusion

We have presented that the core of action prediction is to generate self-assessment (prediction confidence) to judge whether the information is enough for inferring the correct action label, and to adjust the decisions according to the self-assessment like metacognitive processes of human beings. A Self Prediction Refining Network (SPR-Net) is thus proposed for action prediction in untrimmed videos. In SPR-Net, an incremental confidence learner is built, which can dynamically learn the confidence value for making predictions. In parallel, a temporal hybrid network learns the action representation for generating the action category distribution. The self-prediction refining mechanism can be accomplished by a self-refining Gumbel softmax sampler that models the relationship between the prediction confidence and the category distribution. In addition, we have built a Sparse Self-Attention (SSA) module which is readily pluggable into other networks to encode local spatio-temporal features into the frame-level motion representation. With the help of the SSA module, the action prediction performance can be further improved. Comprehensive experiments on five public action datasets show the superior performances of the proposed method for early action prediction. In the future, we will focus on a more difficult and realistic task, i.e., predicting multiple actions in one video by learning a multi-label action classifier.

References

[1] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1036–1043.

[2] G. Yu, J. Yuan, and Z. Liu, "Predicting human activities using spatio-temporal structure of interest points," in *Proc. 20th ACM Int. Conf. Multimedia (MM)*, 2012, pp. 1049–1052.

[3] Y. Kong, D. Kit, and Y. Fu, "A discriminative model with multiple temporal scales for action prediction," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2014, pp. 596–611.

[4] T. Lan, T.-C. Chen, and S. Savarese, "A hierarchical representation for future action prediction," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2014, pp. 689–704.

[5] K. Soomro, H. Idrees, and M. Shah, "Predicting the where and what of actors and actions through online action localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2648–2657.

[6] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 3118–3125.

[7] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1473–1481.

[8] M. S. Aliakbarian, F. S. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson, "Encouraging LSTMs to anticipate actions very early," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 280–289.

[9] Y. Kong, S. Gao, B. Sun, and Y. Fu, "Action prediction from videos via memorizing hard-to-predict samples," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7000–7007.

[10] K. Soomro, H. Idrees, and M. Shah, "Online localization and prediction of actions and interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 459–472, Feb. 2019.

[11] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, May 2013.

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

[13] 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.

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

[15] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[16] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang, "End-to-End learning of motion representation for video understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6016–6025.

[17] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann, "DevNet: A deep event network for multimedia event detection and evidence recounting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2568–2577.

[18] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative CNN video representation for event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1798–1807.

[19] C. Gan, T. Yao, K. Yang, Y. Yang, and T. Mei, "You lead, we exceed: labor-free video concept learning by jointly exploiting Web videos and images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 923–932.

[20] A. Kepecs, N. Uchida, H. A. Zariwala, and Z. F. Mainen, "Neural correlates, computation and behavioural impact of decision confidence," *Nature*, vol. 455, no. 7210, pp. 227–231, Sep. 2008.

[21] P. Grimaldi, H. Lau, and M. A. Basso, "There are things that we know that we know, and there are things that we do not know we do not know: Confidence in decision-making," *Neurosci. Biobehav. Rev.*, vol. 55, pp. 88–97, Aug. 2015.

[22] K. Cho *et al.*, "Learning phrase representations using RNN Encoder–Decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[23] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2016, pp. 1614–1623.

[24] Y. Kong and Y. Fu, "Max-margin action prediction machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1844–1858, Sep. 2016.

[25] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin, "Online real-time multiple spatiotemporal action localisation and prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3657–3666.

[26] S. Lai, W.-S. Zheng, J.-F. Hu, and J. Zhang, "Global-local temporal saliency action prediction," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2272–2285, May 2018.

[27] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, J. Lai, and J. Zhang, "Early action prediction by soft regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2568–2583, Nov. 2019.

[28] H. Zhao and R. Wildes, "Spatiotemporal feature residual propagation for action prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7003–7012.

[29] Y. Kong, Z. Tao, and Y. Fu, "Adversarial action prediction networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 539–553, Mar. 2020.

[30] Y. Cai, H. Li, J.-F. Hu, and W.-S. Zheng, "Action knowledge transfer for action prediction with partial videos," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 8118–8125, Jul. 2019.

[31] X. Wang, J.-F. Hu, J.-H. Lai, J. Zhang, and W.-S. Zheng, "Progressive teacher-student learning for early action prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3556–3565.

[32] Q. Ke, M. Bennamoun, H. Rahmani, S. An, F. Sohel, and F. Boussaid, "Learning latent global network for skeleton-based action prediction," *IEEE Trans. Image Process.*, vol. 29, pp. 959–970, 2020.

[33] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot, "SSNet: Scale selection network for online 3D action prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8349–8358.

[34] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. Kot Chichung, "Skeleton-based online action prediction using scale selection network," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 12, 2019, doi: 10.1109/TPAMI.2019.2898954.

[35] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 201–214.

[36] A. Chakraborty and A. K. Roy-Chowdhury, "Context-aware activity forecasting," in *Proc. Asian Conf. Comput. Vis.* Berlin, Germany: Springer, 2014, pp. 21–36.

[37] F. Rezazadegan, S. Shirazi, M. Baktashmotlagh, and L. S. Davis, "On encoding temporal evolution for real-time action prediction," 2017, *arXiv:1709.07894*. [Online]. Available: http://arxiv.org/abs/1709.07894

[38] T. Mahmud, M. Hasan, and A. K. Roy-Chowdhury, "Joint prediction of activity labels and starting times in untrimmed videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5773–5782.

[39] K. Li and Y. Fu, "Prediction of human activity by discovering temporal sequence patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1644–1657, Aug. 2014.

[40] C. Liu, X. Wu, and Y. Jia, "A hierarchical video description for complex activity understanding," *Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 240–255, Jun. 2016.

[41] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1591–1599.

[42] T. Yao, M. Wang, B. Ni, H. Wei, and X. Yang, "Multiple granularity group interaction prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2246–2254.

[43] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M. Yang, "Flow-grounded spatial-temporal video prediction from still images," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 609–625.

[44] T. J. Pleskac and J. R. Busemeyer, "'Two-stage dynamic signal detection: A theory of choice, decision time, and confidence': Erratum," *Psychol. Rev.*, vol. 118, no. 1, p. 56, 2010.

[45] A. Boldt, V. De Gardelle, and N. Yeung, "The impact of evidence reliability on sensitivity and bias in decision confidence," *J. Experim. Psychol., Hum. Perception Perform.*, vol. 43, no. 8, pp. 1520–1531, 2017.

[46] C. Voskuilen, R. Ratcliff, and G. McKoon, "Aging and confidence judgments in item recognition.," *J. Experim. Psychol., Learn., Memory, Cognition*, vol. 44, no. 1, pp. 1–23, 2018.

[47] K. Desender, A. Boldt, and N. Yeung, "Subjective confidence predicts information seeking in decision making," *Psychol. Sci.*, vol. 29, no. 5, pp. 761–778, May 2018.

[48] S. Bonaccio and R. S. Dalal, "Advice taking and decision-making: An integrative literature review, and implications for the organizational sciences," *Organizational Behav. Hum. Decis. Processes*, vol. 101, no. 2, pp. 127–151, Nov. 2006.

[49] B. Bahrami, K. Olsen, P. E. Latham, A. Roepstorff, G. Rees, and C. D. Frith, "Optimally interacting minds," *Science*, vol. 329, no. 5995, pp. 1081–1085, Aug. 2010.

[50] L. Charles, J.-R. King, and S. Dehaene, "Decoding the dynamics of action, intention, and error detection for conscious and subliminal stimuli," *J. Neurosci.*, vol. 34, no. 4, pp. 1158–1170, Jan. 2014.

[51] M. C. Schmid *et al.*, "Blindsight depends on the lateral geniculate nucleus," *Nature*, vol. 466, no. 7304, pp. 373–377, Jul. 2010.

[52] S. M. Fleming and N. D. Daw, "Self-evaluation of decision-making: A general Bayesian framework for metacognitive computation," *Psychol. Rev.*, vol. 124, no. 1, pp. 91–114, Jan. 2017.

[53] A. Elliethy and G. Sharma, "Vehicle tracking in wide area motion imagery via stochastic progressive association across multiple frames," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3644–3656, Jul. 2018.

[54] W. Guo, L. Cao, T. X. Han, S. Yan, and C. Xu, "Max-confidence boosting with uncertainty for visual tracking," *IEEE Trans. Image Process.*, vol. 24, no. 5, pp. 1650–1659, May 2015.

[55] S.-H. Bae and K.-J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1218–1225.

[56] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1515–1522.

[57] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson Tracking-by-Detection from a single, uncalibrated camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1820–1833, Sep. 2011.

[58] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *Proc. 7th IEEE Workshops Appl. Comput. Vis. (WACVMOTION)*, Jan. 2005, pp. 29–36.

[59] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 631–640, May 2010.

[60] S. Stalder, H. Grabner, and L. Van Gool, "Cascaded confidence filtering for improved tracking-by-detection," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 369–382.

[61] R. D. Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, "Online action detection," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 269–284.

[62] J. Gao, Z. Yang, and R. Nevatia, "RED: Reinforced encoder-decoder networks for action anticipation," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–11.

[63] Z. Shou *et al.*, "Online detection of action start in untrimmed, streaming videos," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 534–551.

[64] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. M. Snoek, "VideoLSTM convolves, attends and flows for action recognition," *Comput. Vis. Image Understand.*, vol. 166, pp. 41–50, Jan. 2018.

[65] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 33–44.

[66] J. Hou, X. Wu, Y. Sun, and Y. Jia, "Content-attention representation by factorized action-scene network for action recognition," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1537–1547, Jun. 2018.

[67] W. Du, Y. Wang, and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1347–1360, Mar. 2018.

[68] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen, "Attention clusters: Purely attention based local feature integration for video classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7834–7843.

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[70] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "ConvNet architecture search for spatiotemporal feature learning," 2017, *arXiv:1708.05038*. [Online]. Available: http://arxiv.org/abs/1708.05038

[71] L. J. Ba, R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: https://arxiv.org/abs/1607.06450

[72] R. Shetty, M. Rohrbach, L. A. Hendricks, M. Fritz, and B. Schiele, "Speaking the same language: Matching machine to human captions by adversarial training," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4155–4164.

[73] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, "Improved variational autoencoders for text modeling using dilated convolutions," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3881–3890.

[74] Z. Lin *et al.*, "A structured self-attentive sentence embedding," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–15.

[75] M. S. Ryoo and J. K. Aggarwal. (2010). *UT-Interaction Dataset, ICPR Contest on Semantic Description of Human Activities (SDHA)*. [Online]. Available: http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html

[76] Y. Kong, Y. Jia, and Y. Fu, "Interactive phrases: Semantic descriptionsfor human interaction recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1775–1788, Sep. 2014.

[77] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*. [Online]. Available: http://arxiv.org/abs/1212.0402

[78] Y. Jiang *et al.*, "Thumos challenge: Action recognition with a large number of classes," Springer, Berlin, Germany, Tech. Rep. 1, 2014.

[79] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 961–970.

[80] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," *Int. J. Comput. Vis.*, vol. 128, no. 1, pp. 74–95, Jan. 2020.

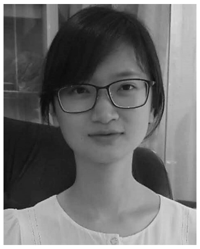
[81] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, "UntrimmedNets for weakly supervised action recognition and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6402–6411.

[82] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. OSDI*, vol. 16, 2016, pp. 265–283.

[83] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Leveraging structural context models and ranking score fusion for human interaction prediction," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1712–1723, Jul. 2018.

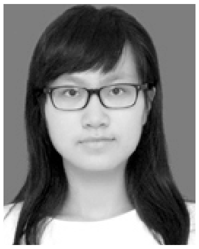
[84] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[85] 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. (ICCV)*, Dec. 2015, pp. 4489–4497.



Jingyi Hou received the B.S. degree in electrical engineering and automation from the China University of Mining and Technology, Beijing, in 2014. She is currently pursuing the Ph.D. degree with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology. Her research interests include computer vision, pattern recognition, and video content analysis.

Xinxiao Wu (Member, IEEE) received the B.S. degree in computer science from the Nanjing University of Information Science and Technology, Nanjing, China, in 2005, and the Ph.D. degree in computer science from the Beijing Institute of Technology (BIT), Beijing, China, in 2010. From 2010 to 2011, she was a Postdoctoral Research Fellow with Nanyang Technological University, Singapore. She is currently an Associate Professor with the School of Computer Science, BIT. Her research interests include machine learning, computer vision, and video analysis, and understanding.



Ruiqi Wang received the B.S. degree in computer science and technology from the Minzu University of China, in 2017. She is currently pursuing the M.S. degree at the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology. Her research interest is video understanding.

Jiebo Luo (Fellow, IEEE) is currently a Professor of computer science with the University of Rochester which he joined in 2011 after a prolific career of fifteen years at Kodak Research Laboratories. He has authored over 400 technical articles. He holds over 90 U.S. patents. His research interests include computer vision, NLP, machine learning, data mining, computational social science, and digital health. He is also a Fellow of ACM, AAAI, SPIE, and IAPR. He has been involved in numerous technical conferences, including serving as a Program Co-Chair of *ACM Multimedia* in 2010, the IEEE CVPR 2012, ACM ICMR 2016, and the IEEE ICIP 2017, as well as a General Co-Chair of *ACM Multimedia* in 2018. He has served on the editorial boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the IEEE TRANSACTIONS ON MULTIMEDIA (TMM), the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), the IEEE TRANSACTIONS ON BIG DATA (TBD), *ACM Transactions on Intelligent Systems and Technology* (TIST), *Pattern Recognition, Knowledge, and Information Systems* (KAIS), *Machine Vision and Applications*, and *Journal of Electronic Imaging*. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON MULTIMEDIA.

Yunde Jia (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Beijing Institute of Technology (BIT) in 1983, 1986, and 2000, respectively. He was a Visiting Scientist with Robot Institute, Carnegie Mellon University (CMU), from 1995 to 1997. He is currently a Professor with the School of Computer Science, BIT, and the Team Head of BIT innovation on vision and media computing. He serves as the Director of the Beijing Lab of Intelligent Information Technology. His interests include computer vision, vision-based HCI and HRI, and intelligent robotics.