Spatio-Temporal Attention-Based LSTM Networks for 3D Action Recognition and Detection

Sijie Song, *Student Member, IEEE*, Cuiling Lan, *Member, IEEE*, Junliang Xing, *Senior Member, IEEE*, Wenjun Zeng, *Fellow, IEEE*, and Jiaying Liu, *Senior Member, IEEE*

***Abstract*— Human action analytics has attracted a lot of attention for decades in computer vision. It is important to extract discriminative spatio-temporal features to model the spatial and temporal evolutions of different actions. In this paper, we propose a spatial and temporal attention model to explore the spatial and temporal discriminative features for human action recognition and detection from skeleton data. We build our networks based on the recurrent neural networks with long short-term memory units. The learned model is capable of selectively focusing on discriminative joints of skeletons within each input frame and paying different levels of attention to the outputs of different frames. To ensure effective training of the network for action recognition, we propose a regularized cross-entropy loss to drive the learning process and develop a joint training strategy accordingly. Moreover, based on temporal attention, we develop a method to generate the action temporal proposals for action detection. We evaluate the proposed method on the SBU Kinect Interaction data set, the NTU RGB + D data set, and the PKU-MMD data set, respectively. Experiment results demonstrate the effectiveness of our proposed model on both action recognition and action detection.**

***Index Terms*— Spatio attention, temporal attention, action recognition, action detection, skeleton data.**

I. Introduction

HUMAN action analytics, including action recognition and detection, is a fundamental yet challenging task in computer vision. It has revealed rapid development due to its wide applications such as intelligent video surveillance, human-computer interaction, video summary and understanding. One of the main challenges in this research field is the modeling of spatial and temporal evolutions for different actions.

Manuscript received September 6, 2017; revised February 11, 2018; accepted March 14, 2018. Date of publication March 22, 2018; date of current version April 12, 2018. This work was supported in part by the National Natural Science Foundation of China under Contract 61772043 and Contract 61672519, in part by the Microsoft Research Asia Fund under Project ID FY17-RES-THEME-013, and the CCF-Tencent Open Research Fund. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xudong Jiang. *(Corresponding authors: Cuiling Lan; Jiaying Liu.)*

S. Song and J. Liu are with the Institute of Computer Science and Technology, Peking University, Beijing 100080, China (e-mail: ssj940920@pku.edu.cn; liujiaying@pku.edu.cn).

C. Lan and W. Zeng are with the Microsoft Research Asia, Beijing 100080, China (e-mail: culan@microsoft.com; wezeng@microsoft.com).

J. Xing is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China (e-mail: jlxing@nlpr.ia.ac.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.2018.2818328

Taking differences in inputs into account, human action analytics can be categorized into RGB video-based and skeleton-based methods. For RGB videos [1]–[5], since each frame is a capture of a highly articulated human in a 2D space, it loses some information of the 3D space and then damnifies the flexibility to achieve human location and scale invariance. The other general approach leverages high level information from skeleton data, which represents a person by 3D coordinate positions of key joints (head, neck, *etc.*). Without combining RGB information, there is a lack of appearance information. Fortunately, biological observations [6] suggest that the positions of a small number of joints can effectively represent human behavior even without appearance information. Skeleton-based human representation has attracted increasing attention for recognizing human actions thanks to its high level representation and robustness to variations of locations and appearances [7]. The prevalence of cost-effective depth cameras such as Microsoft Kinect [8] and the advance of a powerful human pose estimation technique from depth [9] make 3D skeleton data easily accessible. It boosts research on skeleton-based human action analytics. In this work, we focus on action recognition and detection from skeleton data.

The human body can be represented by several key joints in terms of coordinate positions in the 3D space. The articulated configurations of joints constitute various postures and human actions can be identified by the trajectories of skeletal joints. With skeletons as explicit high level representations of human posture, many works design algorithms that use the positions of joints as inputs. Some works design and mine discriminative features from the skeleton, such as the histograms of 3D joint locations (HOJ3D) [10], pairwise relative position features [11], relative 3D geometry features [12], and co-occurrence feature learning [13]. Some works learn and model the temporal dynamics, such as Hidden Markov Model [10], Conditional Random Fields [14], and Recurrent Neural Networks [15]–[18]. In this work, we present a spatio-temporal attention model to effectively incorporate the two components.

For the spatial joints of skeletons, we propose a spatial attention module that conducts automatic mining of discriminative joints. A certain type of action is usually only associated with and characterized by the combinations of a subset of kinematic joints. As the action proceeds, the associated joint set may change accordingly. For example, the joints *hand*, *elbow*, and *head* are discriminative for the action of *drinking* while the leg

joints can be considered noise. Different from actionlet [11], the attention to joints is allowed to vary over time, being content-dependent.

Furthermore, for a sequence of frames, we propose a temporal attention module, which explicitly learns and allocates content-dependent attention to the output of each frame. For a sequence of some action, the flow of the action may experience different stages, *e.g.* the preparation, climax, and end. Taking the action of *punching* as an example, the two persons approach each other, stretch out the hands, and kick out the legs. The frames for identifying stretching out the hands and kicking out the legs are part of the key sub-stages. Different sub-stages/frames have different degrees of importance. In this paper, in contrast to the ideas of extracting key frames [19], [20], our proposed scheme pays different attention to different frames instead of simply skipping frames.

We leverage the attention modules for efficient action recognition and detection. With spatial and temporal attention, discriminative joints and key frames can be automatically determined and allocated with different levels of importance, being content dependent. We further show how to take advantage of the temporal attention in long sequences to localize temporal segments for action detection.

Compared with our previous work [21], we propose the spatial and temporal attention model for both action recognition and detection. We enrich the experiment analysis to give more insights on our attention model, with visualizations of the learned patterns. The influence of parameter choice is also explored. Furthermore, we develop a temporal action proposal generation method for the purpose of action detection. To evaluate the efficiency of our proposed model, we conduct experiments on the largest action detection dataset for 3D data, PKU-MMD [22]. The main contributions of our work are summarized as follows:

- We develop an LSTM network with two types of attention modules for action recognition and detection. A spatial attention module with joint-selection gates is designed to adaptively allocate different levels of attention to different skeleton joints within each frame. A temporal attention module with a frame-selection gate is designed to allocate different attention to different frames.
- Spatio-temporal regularizations are proposed to enable better learning of the networks. The spatial regularization encourages the exploration of all joints rather than overemphasizing only some joints. The temporal regularization prevents temporal attention from increasing unboundedly and gradients from vanishing.
- A joint training strategy is designed to efficiently train the entire network. To reduce the mutual influence of the main network and subnetworks, we develop an iterative training scheme to approach the optimized solution.
- We introduce a method for generating action temporal proposals based on temporal attention. The start and end points of actions in an untrimmed sequence can be accurately localized and our method achieves state-of-the-art performance on action detection.

The remainder of this paper is organized as follows. In Section II, we discuss the related works on action analytics and attention models. In Section III, we introduce our proposed spatio-temporal attention model for action recognition and detection. We evaluate the effectiveness of our proposed method through experiments and analysis in Section IV. And we conclude this paper in Section V.

II. Related Work

A. Action Recognition

The key to the success of action recognition is how to extract discriminative features to effectively model the spatial and temporal evolutions of different actions. Many algorithms have been designed to explore spatial co-occurrence and temporal dynamics. An action is usually associated with and characterized by the interactions and combinations of a subset of skeleton joints. An actionlet ensemble model [11] is proposed to mine such discriminative joints, where an actionlet is a particular conjunction of the features for a subset of the joints and an action is represented as a linear combination of the actionlets. For example, for the action of *drinking*, the subset of joints including *hand*, *elbow*, and *head* composes an actionlet. Orderlet [23] makes an extension of the actionlet by including the feature of pairwise joint distance and allowing various sizes of a subset. Actionlets or orderlets are mined from training samples for robust performance. With RNN, a group sparsity constraint [13] is introduced to the connection matrix to encourage the network to explore the co-occurrence of joints.

For identifying an action, not all frames in a sequence have the same importance. Some frames capture less meaningful information, or even carry misleading information associated with other types of actions, while some other frames carry more discriminative information [24]. A number of approaches have proposed using key frames as a representation for action recognition. One is to utilize the conditional entropy of visual words to measure the discriminative power of a given frame, and the classification results from the top 25% most discriminative frames are employed to make a majority vote for recognition [20]. Another one [24] employs AdaBoost to select the most discriminative key frames for action recognition. The learning of key frames can also be cast in a max-margin discriminative framework by treating them as latent variables [25].

B. Action Detection

To localize and recognize actions in untrimmed video sequences, many detection methods utilize a sliding window approach. In [26]–[28], they slide the observation window along temporal frames and conduct classification within each window using multiple features, *i.e.* dense trajectories, CNN features, combined with action classifiers. Inspired by recent works on object detection from still images [29], [30], the idea of generating object proposals has been borrowed to perform action detection from video sequences [31]–[39]. Some of these methods [33]–[35] produce spatio-temporal object volumes to perform spatio-temporal detection of simple or cyclic actions. To generate high quality spatial proposals, Peng and Schmid [38] took full advantage of region

proposal networks in a two-stream model, which outperform other action detection methods. A more recent work in [39] performs real-time spatio-temporal action localization and early prediction, and achieves new state-of-the-art results, by constructing action tubes with the help of Single Shot MultiBox Detector. Besides, many works [36], [37] focus on temporal action proposals which are likely to contain human actions. Based on techniques for learning sparse dictionaries, Heilbron *et al.* [36] introduced a learning framework to represent and retrieve high quality activity proposals from sampled proposal candidates. An efficient multi-stage CNN is proposed to obtain better localization accuracies [37]. Li *et al.* [16] and Liu *et al.* [40] developed an online action detection method to regress the start and end points of actions with an LSTM network elegantly due to frame level prediction. Different from all those methods, we propose to use an attention model for action proposal generation.

C. Attention-Based Models

When observing the real-world, a human usually focuses on some fixation points at the first glance of a scene, *i.e.* paying different attentions to different regions [41]. Many applications leverage predicted saliency maps for performance enhancement [42], [43], which explicitly learn the saliency maps guided by human labeled groundtruths.

The human labeled groundtruths for explicit attention are generally unavailable and might not be consistent with real attention related to specific tasks. Recently, the exploitation of an attention model which implicitly learns attention has attracted increasing interest in various fields. Bahdanau *et al.* [44] are the first to employ the attention mechanism on machine translation, bringing about a new phase of state-of-the-art implementation. Xu *et al.* [45] incorporate soft and hard attention for image caption generation. Trained with reinforcement learning, the model in [46] is able to pay attention to the most relevant regions of the input image for multiple object recognition. For action recognition and detection, selective focus on different spatial regions is proposed on RGB videos [47]. Ramanathan *et al.* [48] propose an attention model to detect events in RGB videos while attending to the people responsible for the event. The fusion of neighboring frames within a sliding window with learned attention weights is proposed to enhance the performance of dense labeling of actions in RGB videos [49]. However, all the attention models above for action analytics are based on RGB videos. There is a lack of investigation for skeleton sequences, which exhibit different characteristics from RGB videos.

III. Deep LSTM With Spatio-Temporal Attention

We propose a multi-layered LSTM network with spatial and temporal attention mechanisms for action analytics, including action recognition and detection. The network is designed to automatically determine dominant joints within each frame through the spatial attention module, and assign different degrees of importance to different frames through the temporal attention module. Fig. 1 shows the overall architecture of our spatio-temporal attention LSTM network (STA-LSTM),

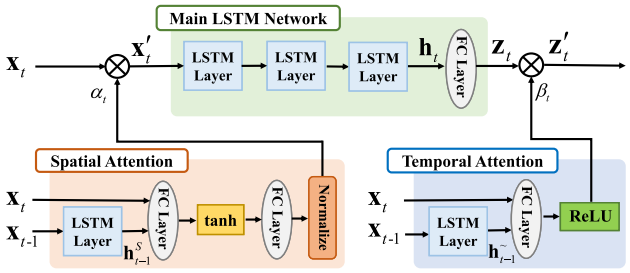

Fig. 1. Overall architecture of our STA-LSTM.

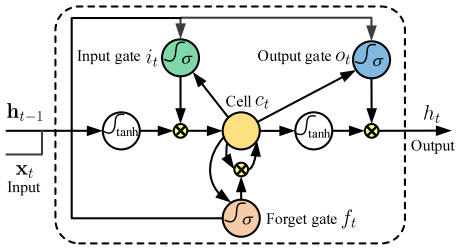

Fig. 2. LSTM neuron structure.

which consists of a main LSTM network, a spatial attention subnetwork, and a temporal attention subnetwork.

In the following, we first briefly review the Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) to make the paper self-contained. Then we discuss the proposed spatial attention module and temporal attention module, respectively. To enable better classification, we introduce a regularized learning objective to our module and a joint training strategy to help overcome the difficulty of model learning for highly coupled networks. Moreover, we exploit temporal attention to generate action proposals for action detection.

A. Overview of RNN and LSTM

RNN is a popular model for sequential data modeling and feature extraction [50]. The output response $\mathbf{h}_t$ at time step t is determined by the input $\mathbf{x}_t$ and the hidden outputs $\mathbf{h}_{t-1}$ from RNN themselves at the last time step

$$\mathbf{h}_t = \theta\left(\mathbf{w}_{xh}^{\mathrm{T}}\mathbf{x}_t + \mathbf{w}_{hh}^{\mathrm{T}}\mathbf{h}_{t-1} + b_h\right), \tag{1}$$

where θ represents a non-linear activation function, $\mathbf{w}_{xh}$ and $\mathbf{w}_{hh}$ denote the learnable connection vectors, and b_h is the bias value. The recurrent structure and the internal memory of RNN facilitate its modeling of the long-term temporal dynamics of sequential data.

LSTM is an advanced RNN architecture which mitigates the vanishing gradient effect of RNN [50]–[52]. As illustrated in Fig. 2, an LSTM neuron contains a memory cell c_t which has a self-connected recurrent edge of weight 1. At each time step t, the neuron can choose to write, reset, and read the memory cell governed by the input gate i_t, and the forget gate f_t.

B. Spatial Attention With Joint-Selection Gates

The action of persons can be described by the evolution of a series of human poses represented by the 3D coordinates of

joints. In general, different actions involve different subsets of joints as discussed in Section II-A.

We propose a spatial attention model to automatically explore and exploit the different degrees of importance of joints. With a soft attention mechanism, each joint within a frame is assigned a spatial attention weight based on the joint-selection gates. This enables our model to adaptively focus more on those discriminative joints.

At each time step t, given the full set of K joints $\mathbf{x}_t = \left(\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,K}\right)^{\mathrm{T}}$, with $\mathbf{x}_{t,k} \in \mathbb{R}^3$, the scores $\mathbf{s}_t = (s_{t,1}, \cdots, s_{t,K})^{\mathrm{T}}$ for indicating the importance of the K joints are jointly obtained as

$$\mathbf{s}_t = U_s \tanh(W_{xs}\mathbf{x}_t + W_{hs}\mathbf{h}^s_{t-1} + \mathbf{b}_s) + \mathbf{b}_{us}, \tag{2}$$

where U_s, W_{xs}, and W_{hs} are the learnable parameter matrixes, and $\mathbf{b}_s$, $\mathbf{b}_{us}$ are the bias vectors. $\mathbf{h}^s_{t-1}$ is the hidden variable from an LSTM layer as illustrated in Fig. 1. For the k^{th} joint, the activation as the joint-selection gate is computed as

$$\alpha_{t,k} = \frac{\exp(s_{t,k})}{\sum_{i=1}^{K}\exp(s_{t,i})}, \tag{3}$$

which is a normalization of scores. The set of gates controls the amount of information from each joint to flow to the main LSTM network. Among the joints, the larger the activation, the more important the joint is for determining the type of action. We also refer to the activation values as attention weights. Instead of assigning equal degrees of importance to all the joints $\mathbf{x}_t$, as illustrated in Fig. 3, the input to the main LSTM network is modulated to $\mathbf{x'}_t = \left(\mathbf{x'}_{t,1}, \ldots, \mathbf{x'}_{t,K}\right)^{\mathrm{T}}$, with $\mathbf{x'}_{t,k} = \alpha_{t,k} \cdot \mathbf{x}_{t,k}$.

Note that the proposed spatial attention model determines the importance of joints based on all the joints of the current time step and the hidden variables from an LSTM layer. On one hand, the hidden variables $\mathbf{h}_{t-1}$ contain past information, benefiting from the merit of LSTM which is capable of exploring temporal long range dynamics. Our spatial attention subnetwork is composed of an LSTM layer, two fully connected layers and a normalization unit as illustrated in Fig. 1. On the other hand, leveraging all joints within the current frame provides a necessary ingredient for determining their importance. Bridged by the joint-selection gate, the main LSTM network and the spatial attention subnetwork can be jointly trained to implicitly learn the spatial attention model.

C. Temporal Attention With Frame-Selection Gate

For a sequence, the amount of valuable information provided by different frames is generally not equal. Only some of the frames (key frames) contain the most discriminative information while the other frames provide contextual information. For example, for the action of *shaking hands*, the sub-stage of *approaching* should have lower importance than the sub-stage of *hands together*. Based on this observation, we design a temporal attention module to automatically pay different levels of attention β to different frames.

For sequence level classification, based on the output $\mathbf{z}_t$ of the main LSTM network and the temporal attention value β_t at each time step t, the scores for C classes are the weighted summation of the scores at all time steps

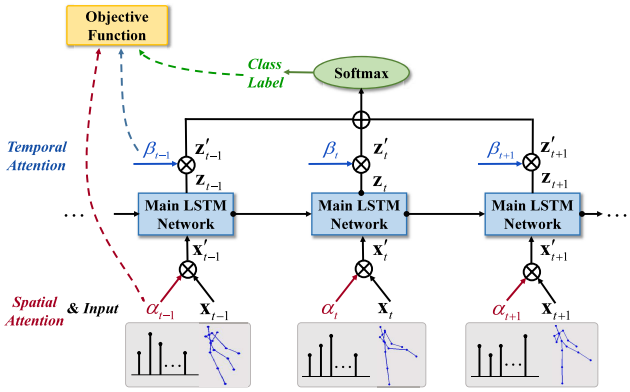

Fig. 3. Illustration of how spatial attention output α and temporal attention output β influence the LSTM network.

$$\mathbf{g} = \sum_{t=1}^{T}\beta_t \cdot \mathbf{z}_t, \tag{4}$$

where $\mathbf{g} = (g_1, g_2, \cdots, g_C)^{\mathrm{T}}$, T denotes the length of the sequence. Fig. 3 illustrates how the temporal attention output β is incorporated into the main LSTM network. The predicted probability of being the i^{th} class given a sequence X is

$$p\left(C_i|X\right) = \frac{e^{g_i}}{\sum_{j=1}^{C} e^{g_j}}, \quad i = 1, \ldots, C. \tag{5}$$

As illustrated in Fig. 1, the attention module is composed of an LSTM layer, a fully connected layer, and a ReLU non-linear unit. It serves as the soft frame selection. The activation as the frame-selection gate can be computed as

$$\beta_t = \mathrm{ReLU}(\mathbf{w}_{x\sim}\mathbf{x}_t + \mathbf{w}_{h\sim}\mathbf{h}^{\sim}_{t-1} + b_{\sim}), \tag{6}$$

which depends on the current input $\mathbf{x}_t$, and the hidden variables $\mathbf{h}^{\sim}_{t-1}$ of time step $t-1$ from an LSTM layer. We use the non-linear function ReLU due to its good convergence performance. The gate controls the amount of information of each frame to be used for making the final classification decision. The works [13] and [15] are our special cases where the attention weights on each frame are equal. Bridged by the frame-selection gate, the main LSTM network and the temporal attention subnetwork can be jointly trained to implicitly learn the temporal attention model.

D. Action Recognition With Joint Spatio-Temporal Attention

For action recognition, it is important to extract discriminative spatio-temporal features to model spatial and temporal evolutions of different actions. To enable the network to pay different levels of attention to different joints and assign different degrees of importance to different frames as an action proceeds, we integrate spatial and temporal attention with LSTM in the same network as illustrated in Fig. 1. How the spatial attention model acts on the input and how the temporal attention model acts on the output of the main LSTM network are illustrated in Fig. 3.

1) Regularized Objective Function: We formulate the final objective function of the spatio-temporal attention network with a regularized cross-entropy loss for a sequence as

$$L = -\sum_{i=1}^{C} y_i \log \hat{y}_i + \lambda_1 \sum_{k=1}^{K} \left(1 - \frac{\sum_{t=1}^{T} \alpha_{t,k}}{T}\right)^2 + \frac{\lambda_2}{T} \sum_{t=1}^{T} \|\beta_t\|_2 + \lambda_3 \|W_{uv}\|_1, \quad (7)$$

where $\mathbf{y} = (y_1, \cdots, y_C)^{\mathrm{T}}$ denotes the groundtruth label. If it belongs to the i^{th} class, then $y_i = 1$ and $y_j = 0$ for $j \neq i$. $\hat{y}_i$ indicates the probability that the sequence is predicted as the i^{th} class, where $\hat{y}_i = p(C_i|X)$. The scalars λ_1, λ_2, and λ_3 balance the contribution of the three regularization terms. We discuss the regularization designs in the following.

The first regularization item aims to encourage the spatial attention model to dynamically focus on more spatial joints in a sequence. We find the spatial attention model is prone to consistently ignoring many joints even though these joints are also valuable for determining the type of action, *i.e.* trapped to a local optimum. We introduce this regularization item to avoid such ill-posed solutions. For clarity, we re-describe it as $\sum_{t=1}^{T} \alpha_{t,k} \approx T$, with $k = 1, \cdots, K$. This encourages equal attention to be paid to different joints.

The second regularization item with l_2 norm is to regularize the learned temporal attention values under control rather than to increase them unboundedly. This alleviates gradient vanishing in back propagation, where the back-propagated gradient is proportional to $1/\beta_t$.

The third regularization item with l_1 norm is to reduce overfitting of the networks. W_{uv} denotes all the connection matrixes in the networks.

2) Joint Training of the Networks: Due to the mutual influence of the three networks, optimization is rather difficult. We propose a joint training strategy to efficiently train the spatial and temporal attention modules, as well as the main LSTM network. The separate pre-training of the attention modules ensures the convergence of the networks. The training procedure is described in Algorithm 1.

Algorithm 1 Joint Training of the LSTM Network With Spatio-Temporal Attention Model

Input: model training parameters N_1, N_2 (*e.g.* $N_1 = 1000$, $N_2 = 500$).

1: Initialize the network parameters using Gaussian.
// Pre-train Temporal Attention Model.
2: With spatial attention weights being fixed as ones, jointly train the main LSTM network with only one LSTM layer and the temporal attention subnetwork to obtain the temporal attention model.
3: Fix this learned temporal attention subnetwork. Train the main LSTM network after increasing its number of LSTM layers to three by N_1 iterations.
4: Fine-tune this temporal attention subnetwork and the main LSTM network by N_2 iteration.
// Pre-train Spatial Attention Model.
5: With temporal attention weights being fixed as ones, jointly train the main LSTM network with only one LSTM layer and the spatial attention subnetwork to obtain the spatial attention model.
6: Fix this learned spatial attention subnetwork. Train the main LSTM network after increasing its number of LSTM layers to three by N_1 iterations.
7: Fine-tune this spatial attention subnetwork and the main LSTM network for N_2 iterations.
// Train the Main LSTM Network.
8: Fix both temporal and spatial attention subnetworks learned in Step-4 and Step-7. Fine-tune the main LSTM network by N_1 iterations.
// Jointly Train the Whole Network.
9: Jointly fine-tune the whole network (main LSTM network, the spatial attention subnetwork, and the temporal attention subnetwork) by N_2 iterations.

Output: the final converged whole model.

E. Action Detection With Temporal Proposals

Motivated by object detection [29], [30], [53], our action detection method consists of two stages, action proposal generation and action classification, as shown in Fig. 4. Instead of using sliding window strategy [28], [31], [54], we leverage a temporal attention network to generate temporal region proposals. Guided by attention responses, the proposals can be localized tightly and accurately. Then each proposal is recognized by an action classifier to identify the action type.

1) Action Proposal Generation: We leverage a temporal attention proposal subnetwork (TAP-LSTM) to produce a temporal attention curve and then generate temporal action proposals based on the attention curve. As shown in Fig. 4, we train the temporal attention proposal using the training data composed of trimmed valid action clips (with non-action clips excluded) collected from the detection training set. The training procedure is supervised by classification loss,

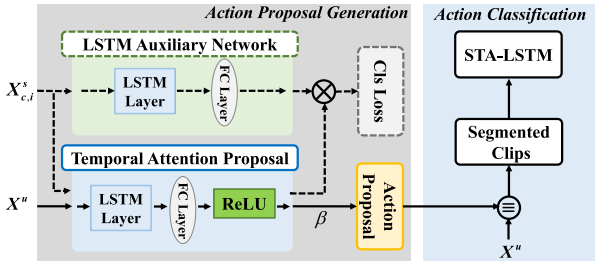

Fig. 4. Our action detection framework. The action proposal generation network is used for proposing action clips. The module with dashed lines is for training and removed in testing.

which can be found in (7). We denote the training set for temporal attention network as $\mathcal{D}$, where $\mathcal{D} = \{D_c\}_{c=1}^{C}$ with C valid action classes, and $D_c = \{X_{c,i}^s\}_{i=1}^{n_c}$ is the training data corresponding to class c, $X_{c,i}^s$ denotes the i^{th} trimmed clip of the action class c. The training procedure in action proposal generation is shown in Fig. 4 by the Action Proposal Generation module.

With the well-trained temporal attention network, we forward the untrimmed video X^u to get the attention response

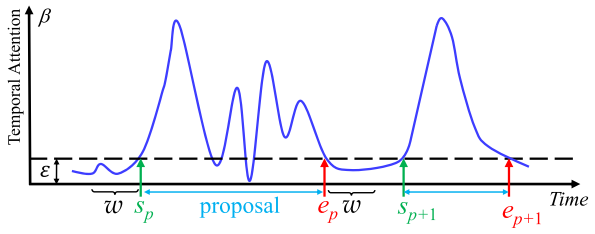

Fig. 5. Action proposal generations by temporal attention.

from the attention model. Given the response $\beta = [\beta_1, \ldots, \beta_T]$ on X^u, we now introduce how to locate action proposals. An interval (a, b) is roughly defined as an action proposal if it satisfies (1) $a = \varepsilon$, (2) $b = \varepsilon$, (3) the response of each point β_i within (a, b) is larger than the threshold ε, that is, $\beta_i > \varepsilon, \forall i \in (a, b)$. With the rules above, the interval (s_{p+1}, e_{p+1}) in Fig. 5 is located as an action proposal. However, the temporal attention response might be disturbed and lower than ε for a short time due to skeleton noises. As in Fig. 5, with the aforementioned rules, the interval (s_p, e_p) includes three short but close proposals, which actually belong to one action. To address this issue, we merge adjacent proposals whose distance in between is less than w. Thus, (s_p, e_p) is another action proposal in Fig. 5. To formulate the action proposal generation, we locate the ordered start points of proposals as

$$S = \{s_p | \beta_{s_p} = \varepsilon, \beta_{s_p - t} < \varepsilon, \quad \forall t \in (0, \min(s_p, w))]\}, \quad (8)$$

where s_p is the start point position of the p^{th} proposal and is subject to $s_{p-1} < s_p < s_{p+1}$, ε denotes the threshold for the response. Since the temporal attention response could be disturbed due to skeleton noises as illustrated above, we use w to ignore such influence to be robust to noises. Similarly, the ordered end points are defined as

$$E = \{e_p | \beta_{e_p} = \varepsilon, \beta_{e_p + t} < \varepsilon, \quad \forall t \in (0, \max(T - e_p, w))]\}, \quad (9)$$

where e_p is the end point position of the p^{th} proposal and is subject to $e_{p-1} < e_p < e_{p+1}$. Therefore, the generated temporal proposals are $\{(s_p, e_p) | p = 1, \ldots, P\}$.

2) Multiscale Scheme: Inspired by the design of multiple scale anchor boxes in object detection [29], at each proposal location, we simultaneously generate several temporal proposals with different centers and lengths. By default, given the original proposal centered at l, where $l = \lfloor (s_p + e_p)/2 \rfloor$, we consider additional proposals centered at $\{l - \delta, l, l + \delta\}$, where $\delta = \lfloor (e_p - s_p + 1)/3 \rfloor$. At each center, proposals with length $\delta, 3\delta, 4\delta$, are generated. That is, for each proposal location, 9 proposal candidates are produced.

3) Action Classification: Each proposal goes through an action classifier which is able to distinguish $C + 1$ classes (including an extra class corresponding to non-action clips). In practice, we leverage the proposed spatio-temporal attention network (see Fig. 1) as the action classifier trained with the data $\mathcal{D}^*$, where $\mathcal{D}^* = \{D_c\}_{c=1}^{C+1}$. Afterwards, to eliminate the proposals highly overlapped with others, non-maximum suppression (NMS) is adopted on the proposal regions based on their classification scores. Finally, we merge the adjacent proposals that share the same label and the distance is less than 20 frames to reduce fragments.

We summarize the training procedure for action detection in Algorithm 2.

Algorithm 2 Training for Action Detection

Input: untrimmed training sequences, action interval labels.
//Generate training data.
1: With the groundtruth of action detection, generate training sets $\mathcal{D}$, $\mathcal{D}^*$ with trimmed clips, where $\mathcal{D} = \{D_c\}_{c=1}^{C}$ and $\mathcal{D}^* = \{D_c\}_{c=1}^{C+1}$.
//Train Proposal Generation Network.
2: Train the temporal attention proposal subnetwork as Fig. 4 using the Action Proposal Generation module with $\mathcal{D}$. Note that only classification loss is back-propagated. The LSTM auxiliary network will be removed during the test.
//Train Action Classification Network.
3: Train the proposed spatio-temporal attention LSTM network STA-LSTM (with detailed structure shown in Fig. 1) with $\mathcal{D}^*$.
Output: Well-trained model for proposal generation and action classification.

IV. Experimental Results

A. Datasets and Settings

We perform our experiments on the following datasets: the SBU Kinect interaction dataset [55], the largest RGB+D dataset of the NTU [56], and the newly collected action detection dataset, PKU-MMD [22]. Note that we utilize PKU-MMD for both action recognition and detection.

1) SBU Kinect Interaction Dataset (SBU): The SBU dataset is an interaction action recognition dataset with two subjects for each action. It contains 230 sequences of 8 classes (6614 frames) with subject independent 5-fold cross validation. Each person has 15 joints and the dimension of the input vector is $15 \times 3 \times 2 = 90$. Note that we smooth each joint's position of the skeleton in the temporal domain to reduce the influence of noise [13], [15].

2) NTU RGB + D Dataset (NTU): The NTU dataset is currently the largest action recognition dataset with high quality skeleton [56]. It contains 56880 sequences (with 4 million frames) of 60 classes, including Cross-Subject (CS) and Cross-View (CV) settings. Each person has 25 joints. We implement a preprocessing method similar to [56] to have position and view invariance. To avoid destroying the continuity of a sequence, no temporal down-sampling is performed.

3) PKU Multi-Modality Dataset (PKU-MMD): The PKU-MMD is a newly captured large-scale dataset for human action understanding with well annotated action positions and types. PKU-MMD [22] consists of 1076 video sequences of 51 action categories, performed by 66 subjects in three camera views. It contains more than 5.4 million frames and 20000 action clips in over 3000 minutes. Each long sequence

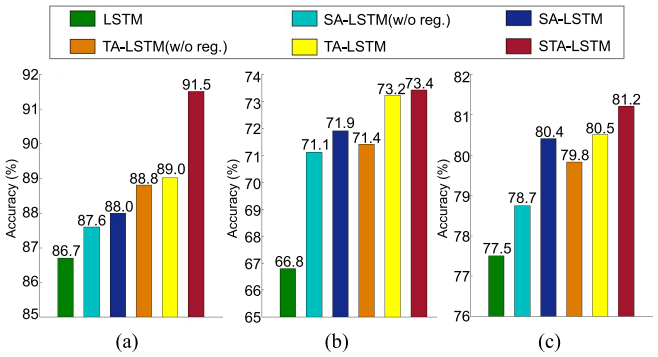

Fig. 6. Performance evaluation of our attention models and the regularization items on two datasets in terms of accuracy (%).

contains about 20 action shots in $3-4$ minutes. We also follow the Cross-Subject and Cross-View protocols in [22].

Since the action detection dataset is well annotated and is large with flexible actions and actors, we cut the annotated valid action clips under the same Cross-Subject and Cross-View participation defined in action detection. We take them as a new action recognition dataset. We use this dataset for both action recognition and action detection evaluation. The detailed partition protocol can be found in [22].

4) Implementation Details: For the action recognition network, we use three LSTM layers for the main LSTM network, and one LSTM layer for each attention network. Each LSTM layer is composed of 100 LSTM neurons. For the action proposal generation module, we use bidirectional LSTM in each layer to leverage both the previous and the future information for proposal generation. Adam [57] is adopted to automatically adjust the learning rate during training. The batch sizes for the SBU, NTU and PKU-MMD datasets are 8, 256, and 256 respectively. We set λ_1, λ_2, and λ_3 to 0.001, 0.0001, and 0.0005 for the SBU dataset, and 0.01, 0.001 and 0.00005 for the NTU and PKU-MMD datasets experimentally. Dropout with a probability of 0.5 is utilized to mitigate overfitting [58].

B. Evaluation on Action Recognition

In this subsection, we validate the effectiveness of our spatio-temporal attention model for action recognition. The ablation study, overall performance comparison, and visualization analysis will be given respectively.

1) Ablation Study of Attention Models: To validate the effectiveness of our attention designs in action recognition, we conduct experiments with different configurations on the SBU and NTU datasets as follows.

- **LSTM**: main LSTM network without attention designs.
- **SA-LSTM(w/o reg.)**: LSTM + spatial attention without regularization (only includes 1^{st} and 4^{th} items in (7)).
- **SA-LSTM**: LSTM + spatial attention network.
- **TA-LSTM(w/o reg.)**: LSTM+temporal attention without regularization (only includes 1^{st} and 4^{th} items in (7)).
- **TA-LSTM**: LSTM + temporal attention network.
- **STA-LSTM**: LSTM + spatio-temporal attention network.

Fig. 6 shows the performance comparisons on the SBU, NTU (Cross-Subject), NTU (Cross-View) datasets.

TABLE I
COMPARISONS ON THE SBU DATASET IN ACCURACY (%)

Methods	Acc. (%)
Raw skeleton [54]	49.7
Joint feature [54]	80.3
Raw skeleton [58]	79.4
Joint feature [58]	86.9
Hierarchical RNN [15]	80.4
Co-occurrence RNN [13]	90.4
ST-LSTM + Trust Gate [59]	93.3
Two-Stream RNN [60]	94.8
VA-LSTM [61]	**97.2**
LSTM	86.7
STA-LSTM	91.5

TABLE II
COMPARISONS ON THE NTU DATASET WITH CROSS-SUBJECT AND CROSS-VIEW SETTINGS IN ACCURACY (%)

Methods	Cross-Subject	Cross-View
Lie Group [62]	50.1	52.8
Skeleton Quads [63]	38.6	41.4
Dynamic Skeletons [64]	60.2	65.2
HBRNN [15]	59.1	64.0
Deep LSTM [55]	60.7	67.3
Part-aware LSTM [55]	62.9	70.3
ST-LSTM + Trust Gate [59]	69.2	77.7
Two-Stream RNN [60]	71.3	79.5
VA-LSTM [61]	**79.4**	**87.6**
LSTM	66.8	77.5
STA-LSTM	73.4	81.2

In comparison with the baseline scheme LSTM, the introduction of the spatial attention module (SA-LSTM) and the temporal attention module (TA-LSTM) generates up to 5.1% and 6.4% accuracy improvement, respectively. The best performance is achieved by combining both modules (STA-LSTM). In the objective function as defined in (7), the second and the third items for regularizations are designed for the spatial and temporal attention model, respectively. We see that they improve the performance of both the spatial attention model and temporal attention model.

2) Overall Performance Comparisons: We show the overall performance comparisons of our final scheme in Table I, Table II and Table III for the SBU, NTU and PKU-MMD datasets, respectively. Thanks to the introduction of the spatio-temporal attention models with efficient regularizations and the training strategy, our model is capable of extracting discriminative spatio-temporal features. For the SBU and NTU datasets, we achieve comparable results with more recent works [59], [60]. On the PKU-MMD dataset, the introduction of attention model brings performance improvement of 3.2% and 1.6% for CS and CV settings, respectively.

3) Influence of Parameters: We explore the effects of λ_1 and λ_2 as in (7), which control the contribution of spatial and temporal regularization terms and influence the learned spatial and temporal patterns.

TABLE III

COMPARISONS ON THE PKU-MMD WITH CROSS-SUBJECT AND CROSS-VIEW SETTINGS IN ACCURACY (%)

Methods	Cross-Subject	Cross-View
LSTM	83.7	91.0
SA-LSTM	86.3	91.4
TA-LSTM	86.6	92.3
STA-LSTM	**86.9**	**92.6**

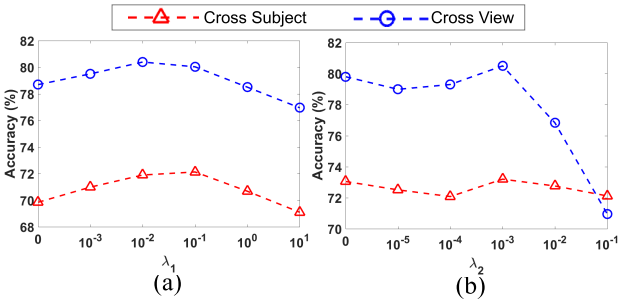

Fig. 7. Influence on performance of parameters of (a) λ_1 for spatial attention regularization term and (b) λ_2 for temporal attention regularization term.

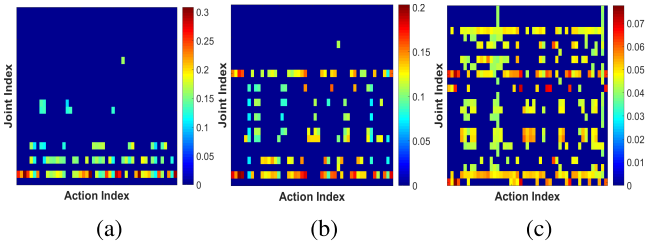

Fig. 8. Distribution of most engaged joints to different actions under various parameter settings (NTU Cross-Subject). Note that the joints whose average attention responses are less than 80% of the maximum response value are suppressed. (a) $\lambda_1 = 0$. (b) $\lambda_1 = 0.01$. (c) $\lambda_1 = 1$.

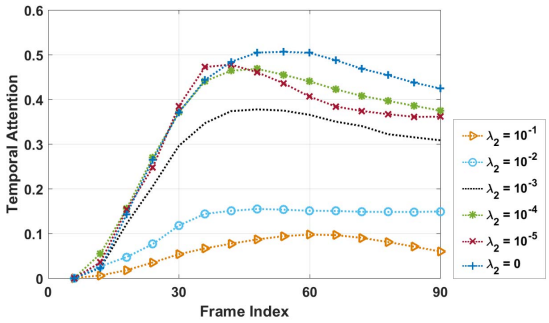

Fig. 9. Temporal attention curves on action *drinking water* under different λ_2 settings.

Fig. 7 shows the performance of SA-LSTM and TA-LSTM with different parameter settings on the NTU dataset, respectively. We can see that the optimized performance is achieved when λ_1 and λ_2 is around 0.01 and 0.001 respectively.

To analyze how the parameters affect the response of spatial attention, we count the most engaged joints for various actions under different values for λ_1. As shown in Fig. 8, a larger λ_1 (Fig. 8(c)) leads to spatial attention distribution on many more joints, making it hard for the network to extract discriminative joints. Whereas a smaller λ_1 (Fig. 8(a)) makes the network focus on too few joints, resulting in information loss.

On the other hand, λ_2 is utilized to avoid unbounded increases of temporal attention. The temporal attention curves of the action *drinking water* learned from different λ_2 settings is shown in Fig 9. A smaller λ_2 results in larger amplitude of the attention response and the network faces with gradient vanishing, considering that the gradient is proportional to $1/\beta$, where β is the attention response. However, with a larger λ_2, temporal attention is constrained to be small or even close to zero. The attention difference among different frames also decreases, resulting in the failure of attending informative frames. As a result, the performance degrades significantly as shown in Fig. 7(b). We find the influence of λ_3 is much less significant.

4) Visualization of the Learned Attentions: We analyze the learned spatial and temporal attention by visualizing the attention responses of our model for the test sequences. We have observed the attention patterns on a variety of actions with the response of each action obtained statistically from many sequences of that action type.

Fig. 10 shows the statistical visualization of spatial attention and temporal attention for four actions from the NTU dataset. We first calculate the average sequence length in the dataset, which is about 85 frames per sequence. Then, the spatial attention map for each sequence is resized to $N \times 85$ by bicubic interpolation, where $N = 25 \times n$, n represents the number of actors within one frame and 25 is the number of joints for each actor. The average spatial attention map obtained from many sequences of the same type can be generated. The statistical temporal attention visualization is generated in a similar manner.

For spatial attention, it is observed that the characteristics of spatial attention differ for different action types. In Fig. 10, for *sitting down* and *kicking*, we observe the ankles, knees, and feet have high degrees of spatial attention. Interestingly, these joints are the most discriminative joints for determining these actions for human sense. For actions mainly involving arms, such as *making a phone call*, we see that the left elbow and left finger have high degrees of spatial attention. In the NTU dataset, we find that most actors make a phone call with their left arms. For the action dominated by both arms like *crossing hands*, the model concentrates on the joints on left and right elbows and fingers. The learned spatial model is capable of paying more attention to the discriminative joints, being content adaptive.

For temporal attention, the attention value increases as the action proceeds. For instance, the attention response for *sitting down* goes to its climax when the actors bend their knees and are about to sit down. The response curve for *kicking* reaches its peak when the actor lift their legs. For *making a phone call*, the latter frames are paid more attention than the earlier frames. It is also interesting that the spatial and temporal attention are synchronous in temporal evolution. Taking *crossing hands* as an example (Fig. 10(c)), the joints

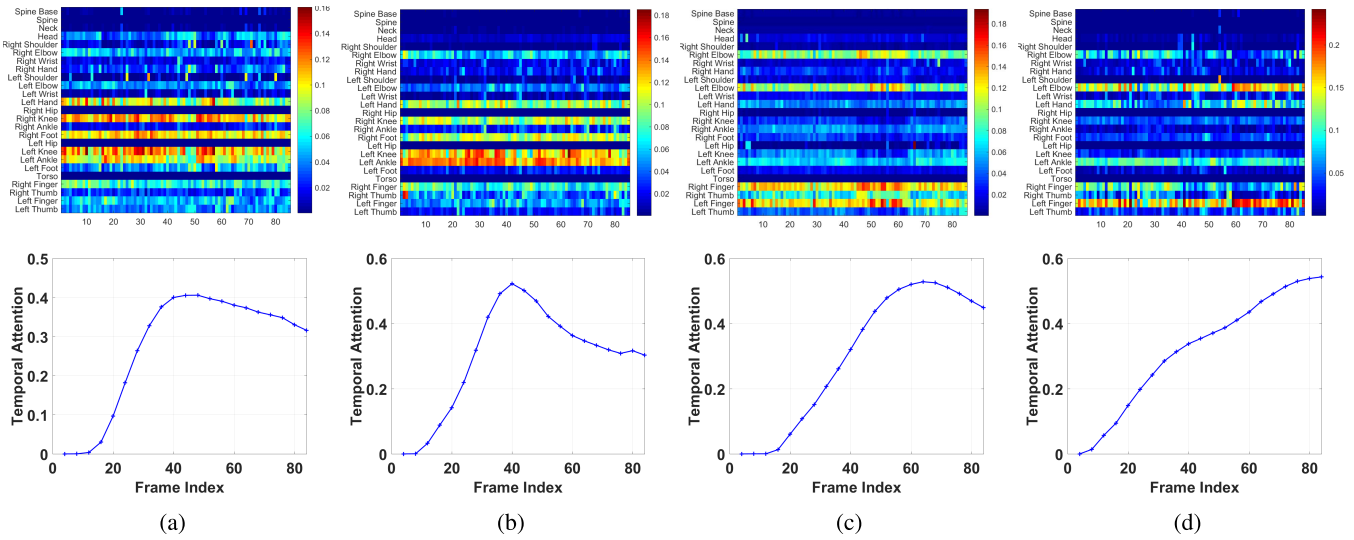

Fig. 10. Visualization of spatial attention (top subfigures) and temporal attention (bottom subfigures) on actions of *sitting down*, *kicking*, *crossing hands*, and *making a phone call* (NTU Cross-Subject). Horizontal axis denotes the frame indexes (time). (a) *Sitting down*. (b) *Kicking*. (c) *Crossing hands*. (d) *Making a phone call*.

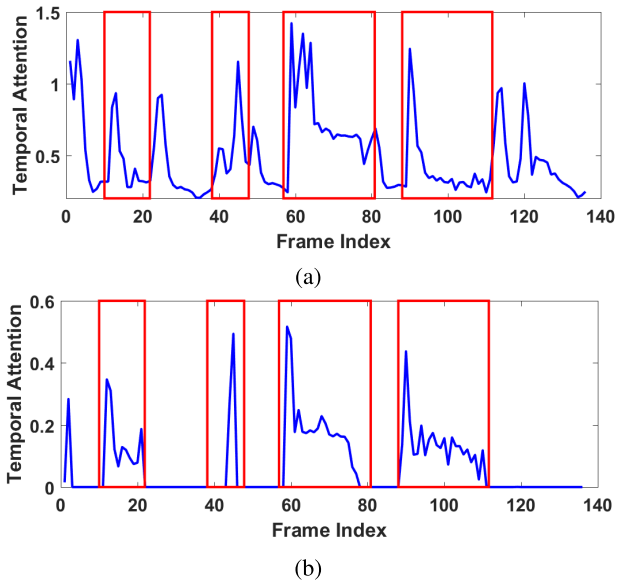

Fig. 11. The temporal attention obtained from bidirectional LSTM is more effective and the action can be localized more accurately. The groundtruth of action proposals are shown in red boxes. (a) Temporal attention response from unidirectional LSTM. (b) Temporal attention response from bidirectional LSTM.

with larger spatial attention (the reg region) concentrate in the duration from the 50^{th} frame to 60^{th}, while the temporal attention also has stronger response in the same duration. More examples can be found in Fig. 10.

C. Evaluation on Action Detection

In this subsection, we validate the effectiveness of our temporal attention method for action detection. The implementation details on proposal generation, classification, evaluation criteria, and experiment results will be discussed.

1) Proposal Generation: As described in Section III-E, temporal attention is utilized to generate temporal segments and thus action proposals. The interval is selected as a proposal if the attention response is higher than a threshold. The TAP-LSTM is trained with trimmed videos. The temporal attention response curve can be obtained for a long untrimmed sequence after going through TAP-LSTM and thus the action proposals.

We find that the temporal attention learned from bidirectional LSTM for TAP-LSTM is more effective and the action can be localized more accurately, as shown in Fig. 11. Both history frames and future frames are utilized for learning in the bidirectional LSTM network, leading to cleaner action boundaries and suppressing the wrong responses over non-action intervals in Fig. 11(b).

In practice, we smooth the temporal attention curve with an average filter of window size 30. To increase the robustness of our proposal generation approach, multi-scale schemes are adopted and each proposal location generates 9 proposal candidates. We explore the influence of parameters ε, ω and δ in Fig. 12 by average recall analysis (IoU $\in \{0.1, 0.3, 0.5\}$), respectively. We first assess the impact of ε and ω when generating proposals with bidirectional TAP-LSTM. With fixed $\omega = 30$, we evaluate with different $\varepsilon \in \{0, 0.1, 0.2, 0.3\}$ in Fig. 12(a), which controls the suppression of lower attention response. The results suggest that ε is a crucial hyper-parameter for achieving higher recall. A lower ε results in proposal fragments, while a higher ε could easily filter some actions as well. Based on the analysis, ε is set to 0 and 0.2 for the Cross-Subject and Cross-View settings, respectively. As shown in Fig. 12(b), our model is not very sensitive to ω, which limits the distance of adjacent proposals. Though more proposals are generated with a smaller ω, the recall rate is not satisfactory. In the multi-scale scheme in Fig. 12(c), which generates more proposals compared with single scale and achieves higher recall, there is little impact for different δ with fixed ω and ε. We choose a value of $\delta = \lfloor (e_p - s_p + 1)/3 \rfloor$ for our multi-scale scheme.

2) Proposal Classification: The generated proposals are fed into an action classifier. Here we take our STA-LSTM as the

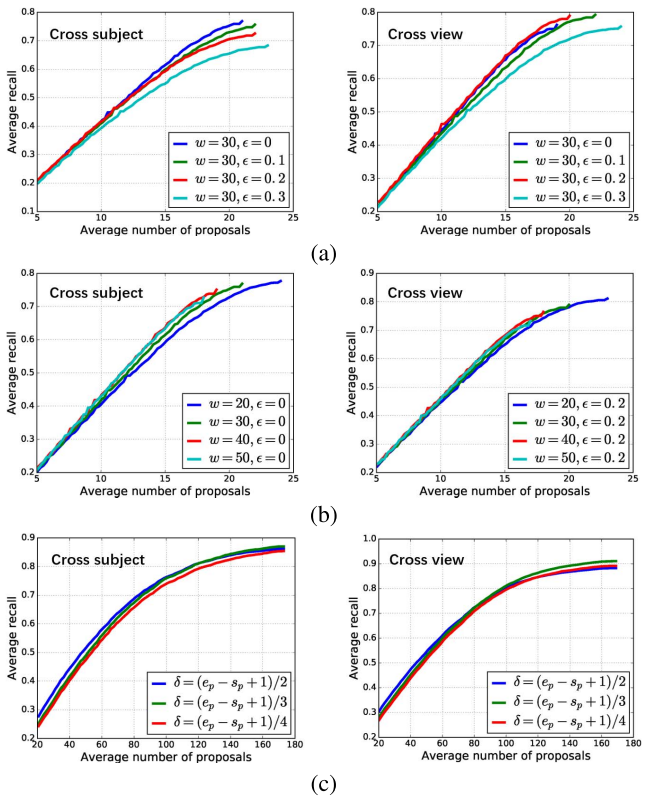

Fig. 12. We explore the influence of parameters ε, w and δ in the process of proposal generation using bidirectional LSTM. (a) The influence of ε. (b) The influence of w. (c) The influence of δ ($w = 30, \varepsilon = 0$ for Cross-Subject and $w = 30, \varepsilon = 0.2$ for Cross-View).

classifier. Since the PKU-MMD is well annotated, we cut the long sequences to action clips and we can train our STA-LSTM on these clips. Note, we add non-action as an extra action type (*background*) to exclude false proposals. Table III shows the performance with STA-LSTM, which also illustrates the effectiveness of our spatial and temporal attention model. With the well-trained model for action recognition, action detection is achieved by conducting classification on those proposals and NMS on the classification results (the confidence threshold is set as 0.1).

3) Evaluation Criteria: Inspired by the objection detection in images [65], the action proposal is determined as correct when the overlapping ratio between the proposed interval I and the groundtruth interval I^* exceeds a threshold θ, which is given as

$$\frac{|I \cap I^*|}{|I \cup I^*|} > \theta, \tag{10}$$

where $I \cap I^*$ denotes the intersection of the predicted and ground-truth intervals and $I \cup I^*$ denotes their union. With the above criterion to determine a correct detection, we use F1 score and mean average precision (mAP) to measure the detection performance.

- $F1$-Score. With a threshold θ, the precision p and recall r can be calculated. Therefore, the $F1$-score is

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}. \tag{11}$$

- Mean Average Precision (mAP). As [65], we use the interpolated precision p_{int}, which is able to remove jiggles on the precision-recall curve, to calculate mean average precision. At a given recall level r, the interpolated precision p_{int} is defined as

$$p_{int}(r, \theta) = \max_{r' \geq r} p(r', \theta), \tag{12}$$

where $p(r, \theta)$ is the precision-recall function under threshold θ. Then the mean average precision is formulated by

$$\text{mAP}(\theta) = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{m_c} \sum_{k=1}^{m_c} p_{int}(r_{ck}, \theta), \tag{13}$$

where C is the total number of action classes, for each class with type id of c, there are m_c action occurrences and r_{ck} is the recall result of the k^{th} ranked detections.

4) Performance Comparisons: We evaluate our action detection performance scheme with the following configurations:

- **TAP-U**: The detection scheme with our temporal action proposal generation network built by **U**nidirectional LSTM layers.
- **TAP-U-M**: The detection scheme with our temporal action proposal generation network with **M**ulti-scale proposals built by **U**nidirectional LSTM layers.
- **TAP-B**: The detection scheme with our temporal action proposal generation network built by **B**idirectional LSTM layers.
- **TAP-B-M**: The detection scheme with our temporal action proposal generation network with **M**ulti-scale proposals built by **B**idirectional LSTM layers.

To further verify the effectiveness of the proposed action detection method, we introduce several approaches for comparison. (a) SVM-SW. An SVM detector is trained to detect the action with a sliding window (SW) strategy. (b) STA-LSTM-SW. Detection is performed based on sliding window design with each window recognized by STA-LSTM. The window size is set to 10 with a step of 5 for both SVM-SW and STA-LSTM-SW. Other different window sizes are evaluated experimentally and the window size of 10 gives relatively good average results. (c) JCR-RNN [16]. The Joint Classification-Regression RNN regresses the start and end points in an end-to-end manner with frame level action detection. (d) STA-LSTM-JCR. We use the proposals generated by JCR-RNN [16], and each proposal is classified by STA-LSTM.

We show the mAP and $F1$-score results in Table IV and Table V, respectively. For each method, we calculate the $F1$-score using the precision and recall value which take all the generated proposals into account. Due to cleaner action boundaries, the quality of temporal proposals built from bidirectional LSTM is much better and thus obtain superior performance compared with unidirectional LSTM. The multiple scale proposal scheme results in even better performance, since it generates more proposal candidates around the center candidate to refine the action boundary. Fig. 13 compares precision-recall curves for different methods. The performance

TABLE IV
COMPARISON OF DIFFERENT METHODS IN MAP ON THE PKU-MMD

Method	Cross-Subject				Cross-View			
θ	0.1	0.3	0.5	0.7	0.1	0.3	0.5	0.7
SVM-SW	0.157	0.075	0.018	0.002	0.209	0.111	0.033	0.004
STA-LSTM-SW	0.416	0.298	0.129	0.015	0.425	0.306	0.148	0.027
JCR-RNN [16]	0.499	0.440	0.345	0.169	0.612	0.525	0.417	0.222
STA-LSTM-JCR	0.504	0.465	0.364	0.177	0.584	0.540	0.425	0.176
TAP-U	0.398	0.348	0.241	0.074	0.419	0.361	0.228	0.052
TAP-U-M	0.466	0.410	0.216	0.047	0.522	0.470	0.292	0.079
TAP-B	0.483	0.461	**0.395**	**0.222**	0.572	0.530	0.450	0.255
TAP-B-M	**0.513**	**0.480**	0.352	0.155	**0.632**	**0.585**	**0.486**	**0.290**

TABLE V
COMPARISON OF DIFFERENT METHODS IN $F1$-SCORE ON THE PKU-MMD

Method	Cross-Subject				Cross-View			
θ	0.1	0.3	0.5	0.7	0.1	0.3	0.5	0.7
SVM-SW	0.266	0.168	0.074	0.010	0.322	0.220	0.108	0.026
STA-LSTM-SW	0.470	0.373	0.231	0.067	0.492	0.397	0.258	0.100
JCR-RNN [16]	0.354	0.305	0.243	0.142	0.378	0.323	0.262	0.167
STA-LSTM-JCR	0.384	0.354	0.290	0.176	0.421	0.391	0.323	0.206
TAP-U	0.502	0.450	0.352	0.169	0.467	0.404	0.280	0.196
TAP-U-M	0.551	0.505	0.345	0.128	0.551	0.505	0.361	0.160
TAP-B	0.544	0.514	**0.461**	**0.327**	0.614	0.578	0.511	0.356
TAP-B-M	**0.557**	**0.530**	0.431	0.242	**0.651**	**0.608**	**0.531**	**0.385**

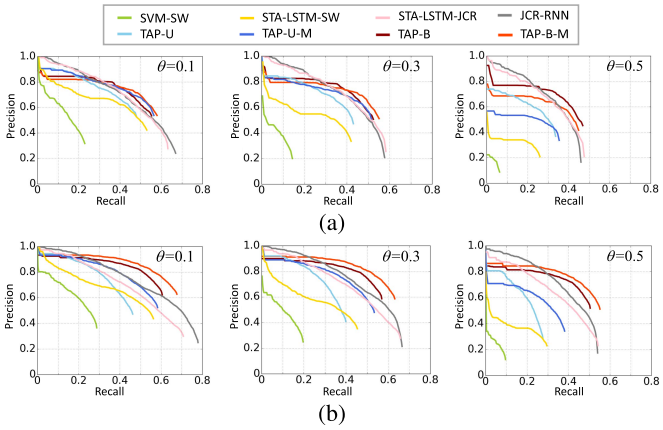

Fig. 13. Precision-recall curves for action detection on the PKU-MMD under different IoU θ. (a) Cross Subject. (b) Cross View.

of other methods is significantly inferior to our proposed action detection framework. Sliding-window based methods lack flexibility on window sizes while adaptive window sizes require high computation complexity. JCR-RNN and STA-LSTM-JCR detect actions in the frame level, resulting in many action fragments. Relying on the temporal attention, our proposal generation method is more flexible. Thanks to accurate proposal localizations and remarkable classification performance, our method outperforms other state-of-the-arts.

V. CONCLUSION

We present an attention based LSTM network for action analysis from skeleton data. To select discriminative joints automatically and adaptively, we propose a spatial attention module with joint-selection gates to assign different levels of importance to different joints. To automatically exploit the different levels of importance for different frames, we propose a temporal attention module to allocate different levels of attention to each frame within a sequence. We design a joint training procedure to efficiently combine spatial and temporal attention with a regularized cross-entropy loss. Our temporal attention is capable of locating the action intervals. We leverage the attention response to generate multiple scale action proposals. Action detection is then achieved by recognizing the action types of each proposal. Experiment results demonstrate the effectiveness of our proposed scheme, which achieves remarkable performance in comparison with other state-of-the-art methods for both action recognition and detection.

ACKNOWLEDGMENT

The authors would like thank the support of NVIDIA Corporation with the GPU for this research. The majority of this work was done at Microsoft Research Asia.

REFERENCES

[1] R. Poppe, "A survey on vision-based human action recognition," *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, 2010.
[2] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Comput. Vis. Image Understand.*, vol. 115, no. 2, pp. 224–241, Feb. 2011.
[3] Y.-G. Jiang, Q. Dai, W. Liu, X. Xue, and C.-W. Ngo, "Human action recognition in unconstrained videos by explicit motion modeling," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3781–3795, Nov. 2015.
[4] F. S. Khan, J. van de Weijer, R. M. Anwer, M. Felsberg, and C. Gatta, "Semantic pyramids for gender and action recognition," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3633–3645, Aug. 2014.
[5] J. Miao, X. Xu, S. Qiu, C. Qing, and D. Tao, "Temporal variance analysis for action recognition," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5904–5915, Dec. 2015.

[6] G. Johansson, "Visual perception of biological motion and a model for it is analysis," *Perception Psychophys.*, vol. 14, no. 2, pp. 201–211, 1973.

[7] F. Han, B. Reily, W. Hoff, and H. Zhang, "Space-time representation of people based on 3D skeletal data: A review," *Comput. Vis. Image Understand.*, vol. 158, pp. 85–105, May 2017.

[8] Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE MultiMedia*, vol. 19, no. 2, pp. 4–10, Feb. 2012.

[9] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1297–1304.

[10] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 20–27.

[11] J. Wang, Z. Liu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1–3.

[12] R. Vemulapalli, F. Arrate, and R. Chellappa, "R3DG features: Relative 3D geometry-based skeletal representations for human action recognition," *Comput. Vis. Image Understand.*, vol. 152, pp. 155–166, Nov. 2016.

[13] W. Zhu *et al.*, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2016, p. 8.

[14] C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Conditional models for contextual human motion recognition," *Comput. Vis. Image Understand.*, vol. 104, nos. 2–3, pp. 210–220, 2006.

[15] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1110–1118.

[16] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 203–220.

[17] Y. Du, Y. Fu, and L. Wang, "Representation learning of temporal dynamics for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3010–3022, Jul. 2016.

[18] Y. Hu, C. Liu, Y. Li, S. Song, and J. Liu, "Temporal perceptive network for skeleton-based action recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–2.

[19] S. Carlsson and J. Sullivan, "Action recognition by shape matching to key frames," in *Proc. Workshop Models Versus Exemplars Comput. Vis.*, 2001, vol. 1. no. 18, pp. 1–8.

[20] Z. Zhao and A. Elgammal, "Information theoretic key frame selection for action recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2008, pp. 1–10.

[21] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proc. AAAI Conf. Artif. Intell.*, 2017, p. 7.

[22] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu. (2017). "PKU-MMD: A large scale benchmark for continuous multi-modal human action understanding." [Online]. Available: https://arxiv.org/abs/1703.07475

[23] G. Yu, Z. Liu, and J. Yuan, "Discriminative orderlet mining for real-time recognition of human-object interaction," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 50–65.

[24] L. Liu, L. Shao, and P. Rockett, "Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition," *Pattern Recognit.*, vol. 46, no. 7, pp. 1810–1818, 2013.

[25] M. Raptis and L. Sigal, "Poselet key-framing: A model for human activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2650–2657.

[26] S. Karaman, L. Seidenari, and A. Del Bimbo, "Fast saliency based pooling of fisher encoded dense trajectories," in *Proc. Eur. Conf. Comput. Vis. THUMOS Workshop*, 2014, vol. 1. no. 2, p. 5.

[27] D. Oneata, J. Verbeek, and C. Schmid, "Action and event recognition with fisher vectors on a compact feature set," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1817–1824.

[28] L. Wang, Y. Qiao, and X. Tang, "Action recognition and detection by combining motion and appearance features," *THUMOS Action Recognit. Challenge*, vol. 1, no. 2, p. 2, 2014.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[30] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[31] P. Siva and T. Xiang, "Weakly supervised action detection," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 65.1–65.0.

[32] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "DAPs: Deep action proposals for action understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 768–784.

[33] M. Jain, J. van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek, "Action localization with tubelets from motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 740–747.

[34] G. Gkioxari and J. Malik, "Finding action tubes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 759–768.

[35] W. Chen, C. Xiong, R. Xu, and J. J. Corso, "Actionness ranking with lattice conditional ordinal random fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 748–755.

[36] F. Caba Heilbron, J. Carlos Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1914–1923.

[37] Z. Shou, D. Wang, and S. Chang, "Action temporal localization in untrimmed videos via multi-stage CNNS," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1049–1058.

[38] X. Peng and C. Schmid, "Multi-region two-stream R-CNN for action detection," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 744–759.

[39] 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.*, Oct. 2017, pp. 3637–3646.

[40] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, and W. Zeng, "Multi-modality multi-task recurrent neural network for online action detection," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[41] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1915–1926, Oct. 2012.

[42] Y. Yu, G. K. I. Mann, and R. G. Gosine, "An object-based visual attention model for robotic applications," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 5, pp. 1398–1412, Oct. 2010.

[43] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–2.

[44] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[45] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[46] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–2.

[47] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," in *Proc. Int. Conf. Learn. Represent. Workshop*, 2016, pp. 1–11.

[48] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and F.-F. Li, "Detecting events and key actors in multi-person videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3043–3053.

[49] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, "Every moment counts: Dense detailed labeling of actions in complex videos," *Int. J. Comput. Vis.*, vol. 126, nos. 2–4, pp. 1–15, 2017.

[50] A. Graves, *Supervised Sequence Labelling With Recurrent Neural Networks*, vol. 385. Berlin, Germany: Springer, 2012.

[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[52] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Neural Networks*. Piscataway, NJ, USA: IEEE Press, 2001.

[53] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[54] M. Hoai and F. De la Torre, "Max-margin early event detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2863–2870.

[55] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 28–35.

[56] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1010–1019.

[57] D. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[58] W. Zaremba, I. Sutskever, and O. Vinyals. (2014). "Recurrent neural network regularization." [Online]. Available: https://arxiv.org/abs/1409.2329
[59] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 816–833.
[60] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3633–3642.
[61] Y. Ji, G. Ye, and H. Cheng, "Interactive body part contrast mining for human interaction recognition," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, Jul. 2014, pp. 1–6.
[62] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 588–595.
[63] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2014, pp. 4513–4518.
[64] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2186–2200.
[65] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

Sijie Song (S'17) received the B.S. degree in computer science from Peking University, Beijing, China, in 2016, where she is currently pursuing the Ph.D. degree with the Institute of Computer Science and Technology. Her research interests include computer vision and image processing.

Cuiling Lan received the B.Sc. degree in electrical engineering and the Ph.D. degree in intelligent information processing from Xidian University, China, in 2008 and 2014, respectively. In 2014, she joined the Microsoft Research Asia. Her research interests include computer vision, image and video compression, and transmission.

Junliang Xing (S'09–M'12–SM'18) received the dual B.S. degrees in computer science and mathematics from Xi'an Jiaotong University, Shaanxi, China, in 2007, and the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2012. He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing. His research interests include computer vision problems related to faces and humans.

Wenjun (Kevin) Zeng (M'97–SM'03–F'12) received the B.E. degree from Tsinghua University, the M.S. degree from the University of Notre Dame, and the Ph.D. degree from Princeton University, respectively. He was with the PacketVideo Corporation, Sharp Labs of America, Bell Labs, and Panasonic Technology. He was with the University of Missouri from 2003 to 2016, most recently as a Full Professor. Since 2014, he has been leading the video analytics research empowering the Microsoft Cognitive Services and Azure Media Analytics Services. He is currently a Principal Research Manager and a member of the Senior Leadership Team with the Microsoft Research Asia. He has contributed significantly to the development of international standards (ISO MPEG, JPEG2000, and OMA). His current research interests include mobile cloud media computing, computer vision, social network/media analysis, and multimedia communications and security.

Dr. Zeng was the recipient of several best paper awards, e.g., the IEEE VCIP'2016, the IEEE ComSoC MMTC 2015 Best Journal Paper, and the ACM ICMCS'2012. He served as the Steering Committee Chair of the IEEE ICME in 2010 and 2011. He is serving or has served as the General Chair or a TPC Chair for several IEEE conferences, e.g., ICME'2018 and ICIP'2017. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and the IEEE TRANSACTIONS ON MULTIMEDIA (TMM). He is an Associate Editor-in-Chief of the IEEE *Multimedia Magazine*. He was a Special Issue Guest Editor for the Proceedings of the IEEE TMM and TCSVT, ACM TOMCCAP, and the IEEE *Communications Magazine*. He was on the Steering Committee of the IEEE TRANSACTIONS ON MOBILE COMPUTING and IEEE TMM.

Jiaying Liu (S'08–M'10–SM'17) received the B.E. degree in computer science from Northwestern Polytechnic University, Xi'an, China, and the Ph.D. degree (Hons.) in computer science from Peking University, Beijing, China, in 2005 and 2010, respectively.

She is currently an Associate Professor with the Institute of Computer Science and Technology, Peking University. She has authored over 100 technical articles in refereed journals and proceedings. She holds 24 granted patents. Her current research interests include image/video processing, compression, and computer vision.

Dr. Liu is a CCF Senior Member. She was a Visiting Scholar with the University of Southern California, Los Angeles, CA, USA, from 2007 to 2008. She was a Visiting Researcher with the Microsoft Research Asia in 2015 supported by the Star Track for Young Faculties. She has also served as a TC member in the IEEE CAS MSA and APSIPA IVM. She was the APSIPA Distinguished Lecture from 2016 to 2017.