Spatio-Temporal Attention Networks for Action Recognition and Detection

Jun Li, Xianglong Liu, Wenxuan Zhang, Mingyuan Zhang, Jingkuan Song, Nicu Sebe

***Abstract*—Recently, 3D Convolutional Neural Network (3D CNN) models have been widely studied for video sequences and achieved satisfying performance in action recognition and detection tasks. However, most of the existing 3D CNNs treat all input video frames equally, thus ignoring the spatial and temporal differences across the video frames. To address the problem, we propose a spatio-temporal attention (STA) network that is able to learn the discriminative feature representation for actions, by respectively characterizing the beneficial information at both the frame level and the channel level. By simultaneously exploiting the differences in spatial and temporal dimensions, our STA module enhances the learning capability of the 3D convolutions when handling the complex videos. The proposed STA method can be wrapped as a generic module easily plugged into the state-of-the-art 3D CNN architectures for video action detection and recognition. We extensively evaluate our method on action recognition and detection tasks over three popular datasets (UCF-101, HMDB-51 and THUMOS 2014), and the experimental results demonstrate that adding our STA network module can obtain the state-of-the-art performance on UCF-101 and HMDB-51, which has the top-1 accuracies of 98.4% and 81.4% respectively, and achieve significant improvement on THUMOS 2014 dataset compared against original models.**

***Index Terms*—3D CNN, spatio-temporal attention, temporal attention, spatial attention, action recognition, action detection**

I. Introduction

In recent years, the explosion of the availability of video data has brought challenges to efficient video analysis and understanding. Video action recognition and detection has become one of the most widely studied tasks in computer vision [1]–[5], and encouraging progress has been achieved following the success of various architectures based on deep convolutional neural networks (CNNs) [6], [7]. However, common CNNs methods have been primarily developed for 2D images, and can hardly capture the temporal information contained in the video, which plays an important role in action detection and recognition.

Modelling the temporal variations in the video is a challenging problem that is made even harder if the detection is to be performed in an online setting and in real-time. The traditional solutions such as the improved Dense Trajectories (iDT) [8] highly depend on the optical flow-based hand-crafted features to extract temporal trajectories. There are also deep end-to-end solutions [9], [10]. For example, Simonyan and Zisserman [9] introduced a two-stream CNN architecture that uses two 2D networks corresponding to the optical flow and the appearance, respectively. Despite the good performance, extracting the optical flow is usually computationally intensive and becomes intractable on large-scale datasets [11].

To avoid the expensive computation while still capturing the spatio-temporal correlations, a number of 3D CNNs methods have been proposed in the past few years [1], [6], [7], [12]–[17]. In [13], Ji *et al.* were among the first to introduce the concept of the 3D CNN and applied 3D convolution to extract spatio-temporal features from videos. Tran *et al.* proposed a simple, yet effective 3D CNN approach (C3D) that can deal with input videos of varying temporal lengths well [1]. Recently, various types of 3D CNNs with different structures were proposed, e.g., pseudo-3D CNNs [18], two-stream I3D [19], mixed 3D/2D convolutional tube [20], etc. The 3D CNNs can directly extract spatial and temporal features from raw videos, and thus their performance in the field of action recognition and detection has been significantly improved recently.

In practice 3D CNNs can largely improve the video processing speed with a satisfying performance, e.g., at least an order of magnitude faster than real-time [21]. Many studies further modified and improved 3D CNNs to increase the capacity of representing finer temporal relations in a local spatio-temporal window [22]. Despite the promising progress, without explicitly extracting the most informative information in spatial and temporal dimensions of the videos, the existing 3D CNN solutions still lack learning capacity of discriminative spatio-temporal feature representation for actions. For example, traditional 3D CNNs usually learn the temporal feature by equally treating the consecutive frames, while in practice, different frames might convey quite different contributions to the action recognition, e.g., the frames with motion blur may provide fewer cues for activity recognition. Similarly, along the spatial dimension, the differences between the visual information from different channels are usually undistinguished in the existing 3D CNN solutions.

It has been noted in visual cognition literature that

J. Li, X. Liu, W. Zhang and M. Zhang are with the State Key Lab of Software Development Environment, Beihang University, Beijing, China. X. Liu is also with Beijing Advanced Innovation Center for Big Data-Based Precision Medicine, Beihang University, Beijing, China. (Corresponding author: Xianglong Liu, xlliu@nlsde.buaa.edu.cn)

J. Song is with the Innovation Center, University of Electronic Science and Technology of China, Chengdu, China.

N. Sebe is with the Department of Information Engineering and Computer Science, University of Trento, Trento, Italy.

This work was supported by National Natural Science Foundation of China (61690202, 61872021), Fundamental Research Funds for Central Universities (YWF-19-BJ-J-271), Beijing Municipal Science and Technology Commission (Z171100000117022), and State Key Lab of Software Development Environment (SKLSDE-2018ZX-04).

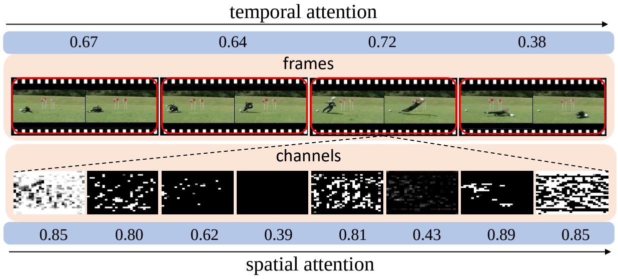


Fig. 1. Spatio-temporal attention of STA-ResNeXt-101(64f) on "Jump" action from HMDB-51 dataset. The top row shows the successive frames in the video and the corresponding temporal attention weights learnt by our STA module, where the frames with the most discriminative features for "Jump" action can be identified with the larger weights. The bottom row shows the feature maps (channels) for the specified frame and their spatial attention weights, where the more informative feature maps play more important roles in the action recognition and detection.

humans do not focus their attention on an entire scene of each frame at once [23]. Instead, they focus sequentially on different parts in different frames to extract relevant information [24]. Thus, to improve the power of the 3D CNNs for extracting the informative features from the video, in this paper we propose a novel spatio-temporal attention (STA) network for action recognition and detection. The proposed network can exploit the discriminative information at two levels (i.e., frame level and channel level), and thus improve the capability of the 3D CNNs with a more powerful spatio-temporal feature learning. In the temporal dimension, while the traditional convolution is localized at the local receptive field, our approach can capture the temporal attention by learning varying frame-wise weights with the global visual information. Besides, in the spatial dimension, we further consider the differences among the various channels in 3D CNNs for activity representation, and develop a spatial attention at the channel level, which together with frame level attention can avoid the computational bottleneck and also enhance the feature representing power. Our spatio-temporal attention network is flexible and efficient. It can be easily plugged into the state-of-the-art 3D CNN frameworks and thus improve their performance significantly.

The main contribution of this paper is that we devise a novel and efficient spatio-temporal attention mechanism in the 3D CNNs for action recognition and detection tasks. Instead of directly employing two similar temporal or spatial attention separately, our mechanism captures temporal and spatial attention simultaneously in a single module, and behaves as a kind of soft attention which learns the attention weights adaptively. The proposed module serves as a simple, yet generic module for many 3D CNNs, and in practice it only needs to be appended to the later convolutional layers without increasing too much computational cost. Figure 1 shows a typical example using our STA module plugged into ResNeXt-101(64f) on "Jump" action from HMDB-51 dataset. We extensively evaluate our method on action recognition and detection tasks over three popular datasets (UCF-101, HMDB-51 and THUMOS 2014), and the experimental results demonstrate that our STA module can achieve the state-of-the-art performance for action recognition task and obtain considerably improved performance compared against original models in action detection task.

II. Related Work

A. Action Recognition and Action Detection

Action recognition [1], [7], [9], [18] and action detection [25]–[29] have been an important research field for visual understanding [30], [31]. In the literature, more and more researchers have attempted to improve the accuracy of the action recognition on the common action recognition datasets, including HMDB-51, UCF-101, Sports-1M. However, thus far the results achieved by the action detection methods, using traditional hand-crafted features [32], [33], CNN features [34]–[36] and fusing two types of features [37], have not been satisfactory. In recent years, as the video data explosively increased, the expensive computation and insufficient annotations have become the key challenges for video action recognition and detection. Therefore, more and more researchers dedicated effort to the problems of fast recognition or detection with limited supervised information [21], [38] and even unsupervised methods [39], [40], which have been proven effective to deal with the challenge in the large-scale recognition and detection tasks.

For video action recognition and detection, the temporal feature learning serves as a critical part to achieve satisfying performance. In [9], Simonyan *et al.* proposed a two-stream CNN architecture which incorporates a spatial network and a temporal one based on the multi-frame dense optical flow. Peng *et al.* developed a multi-region two-stream R-CNN model by stacking optical flow over several frames [41]. Feichtenhofer *et al.* directly exploited the power of the CNN in videos and studied a number of ways of fusing CNNs both spatially and temporally [42]. Most of the state-of-the-art action recognition approaches rely on traditional local optical flow estimation and the computationally expensive two-stage prediction. In [43] the motion information is implicitly captured by the CNN architecture, achieving the goal of fast computation and end-to-end training.

B. 3D CNNs

In the literature, Kim *et al.* first introduced the 3D filter field descriptor [12], and later Ji *et al.* further proposed the 3D CNNs descriptor [13]. Following the basic idea, and based on the practical work [1] and its released source code, extensive studies have been proposed and focusing mainly on the 3D CNNs [6], [14], [15], [44]. Recently, the focus has been on how to improve the performance of features learning for 3D CNNs considering the temporal dimension [22]. In [7], Hara *et al.* presented an explicit analysis on various 3D CNNs, and concluded that the deep 3D CNNs together with Kinetics [19] can retrace the successful results of 2D CNNs and ImageNet. Since the 3D CNNs heavily rely on expensive computation and memory cost, Qiu *et al.* introduced a surrogate architecture named Pseudo-3D

CNNs, which demonstrated superior performances over several state-of-the-art techniques at the same time being more computationally efficient and using less memory [18]. In [20], the authors employed the mixed 3D/2D convolutional tube for human action recognition. This approach integrates 2D CNNs with the 3D convolution module to generate deeper and more informative feature maps, while reducing the training complexity in each round of spatio-temporal fusion, and taking advantage of well-established 2D CNNs for visual recognition in static images.

C. Attention

Attention mechanism has been widely used in various fields [45]–[50]. For example, Bahdanau *et al.* introduced attention into the neural machine translation [45], Rush *et al.* proposed a neural attention model for abstractive sentence summarization [51], Chorowski *et al.* employed attention-based models for speech recognition [46], and Xu *et al.* proposed an attention based framework for image caption generation [52]. In the literature, the attention mechanism plays an important role in the common image classification tasks. Recently, Wang *et al.* developed a residual attention network built by stacking attention modules, capturing the attention in the image that softly weighted output features [53]. Hu *et al.* proposed an SE-block [54], which is a lightweight gating mechanism, to model channel-wise relationships. In [55] the authors devised the Convolutional Block Attention Module (CBAM), a simple yet effective attention module for feed-forward convolutional neural networks, which is a channel and spatial attention for image classification and image object detection task. Similarly, for video data, Sharma *et al.* proposed a soft attention based recurrent model for action recognition and demonstrated that the model tends to recognize important elements in video frames based on the activities it detects [24]. [47] designed a fully differentiable temporal attention filters for human activity recognition from videos. [56] proposed an attentional pooling for Action Recognition. As to the 3D CNNs, Xie *et al.* replaced many of the 3D convolutions by low-cost 2D convolutions to seek a balance between speed and accuracy, and further improved the accuracy of their model by using feature gating based attention mechanism [15]. Long *et al.* proposed a local feature integration framework based on attention clusters, and introduced a shifting operation to capture more diverse signals [57]. [58] presented the non-local operations to capturing long-range dependencies for video classification. [59] devised a factorized action-scene network (FASNet) to generate content-attention representation, which can encode and fuse the most relevant motion information and scene information for action recognition. [50] introduced a hierarchical self-attention network for video action localization.

D. Discussions

We have to point out that our STA is the lightweight and generic module capturing both the temporal and spatial attentions for 3D CNN models. To guarantee the learning power of the module, we devise that the squeezing and expanding operations can perform simultaneously along both the spatial and temporal dimensions. In [54], Hu *et al.* proposed a similar network structure named Squeeze-and-Excitation (SE) block, which adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channels. However, our STA fundamentally differs from SE in terms of both the receptive field and the network structure. First, STA module fits the 3D CNN models by considering both the frame level and the channel level, while SE only focuses on 2D CNNs at the channel level. Besides, STA possesses a much stronger learning power than SE, owing to its paired convolution/deconvolution network structure, which can significantly improve the network capacity for better feature representation.

Very recently, in [16], Diba *et al.* extended the SE block to a Spatio-Temporal Channel Correlation (STC) block for the 3D networks, mainly attempting to model the correlations between channels with respect to temporal and spatial features, rather than capturing the spatial and temporal attentions in a simultaneous way. Therefore, compared to STC, our STA module couples spatial and temporal dimensions, and thus can better find the informative characteristics of the features, based on the global contextual information from multiple dimensions.

III. Spatio-temporal Attention Networks

Attention is the behavioral and cognitive process of selectively concentrating on a discrete aspect of information, whether deemed subjective or objective, while ignoring other perceivable information. Visual attention is one of the key mechanisms of perception that enables humans to efficiently select the visual data of most potential interest [60]. For video action recognition, a proper attention model can help answer the question of where and when it needs to look at the image evidence to draw a classification decision [48]. The attention model for action recognition/detection can potentially help infer the action happening in videos by focusing only on the relevant places in certain frames along the temporal domain.

In this paper, we introduce the spatio-temporal attention that learns different focusing weights for different frames in the temporal dimension and different focusing weights for different channels in the spatial dimension. Before elaborating our spatio-temporal attention (STA) network model, we first introduce the basic notations.

Suppose we have an input frame sequence $\mathrm{X} = \{x_{i,j} \in \mathbb{R}^{h \times w}, i = 1, \ldots, l, j = 1, \ldots, c\}$, where $x_{i,j}$ is the feature map at time i and channel j, l is the number of frames in each input sliding temporal window, and c denotes the number of channels. Our STA module attempts to learn the attention $\mathrm{W} = \{w_{i,j}, i = 1, \ldots, l, j = 1, \ldots, c\}$ weighting the frames and channels respectively in temporal and spatial dimension. We define an attention function $\mathcal{T}(\cdot)$ for the STA module, which learns the weights W from the input features X. Based on $\mathcal{T}$, we can define the output

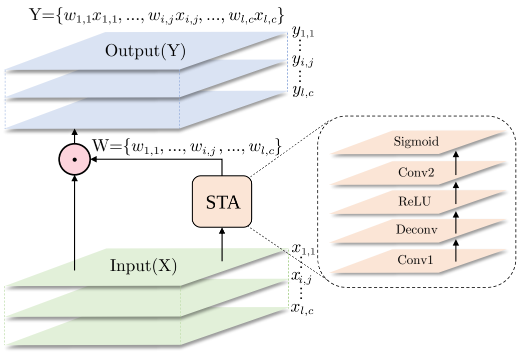

$Y=\{w_{1,1}x_{1,1}, ..., w_{i,j}x_{i,j}, ..., w_{l,c}x_{l,c}\}$
Output(Y)
$y_{1,1}$
$y_{i,j}$
$y_{l,c}$
$W=\{w_{1,1}, ..., w_{i,j}, ..., w_{l,c}\}$
STA
Sigmoid
Conv2
ReLU
Deconv
Conv1
Input(X)
$x_{1,1}$
$x_{i,j}$
$x_{l,c}$

Fig. 2. The architecture of our STA network module embedded in the 3D CNN model. It consists of two convolutional layers (Conv1 and Conv2), one deconvolutional layer (Deconv), and two active functions (ReLU and Sigmoid). Our STA module outputs W that weights the input feature maps ($\odot$ denotes the element-wise multiplication between the input feature map and the weights).

sequence $\mathrm{Y} = \{y_{i,j} \in \mathbb{R}^{h\times w}, i = 1, \ldots, l, j = 1, \ldots, c\}$ generated by passing X through the STA module. In this paper, there will be two types of attention function, i.e., the temporal attention function $\mathcal{T}_t(\cdot)$ at frame level and the spatial attention function $\mathcal{T}_s(\cdot)$ at the channel level. Figure 2 shows the whole spatio-temporal attention network module.

The design of the STA module follows the following principles: (1) the module should be simple and efficient, relying on simple operations like convolution and deconvolution; (2) the module should enable the robust and nonlinear learning capacity of the spatio-temporal feature representation in 3D CNNs.

A. Temporal Attention Module

In practice a robust action recognition model can generate correct predictions by only focusing on a small but informative part of action video, instead of the whole one. This indicates that the attention mechanism plays quite an important role in learning discriminative feature representation for actions. The existing 3D CNN models, simultaneously utilizing both the spatial and temporal information, have achieved promising performance. However, they usually indiscriminately handle the temporal frames and thus can hardly guarantee a stable recognition performance. Next we first consider integrating the temporal attention into the 3D convolutions, so that the enhanced model can distinguish the most informative frames and thus extract the characteristics of the actions.

1) Temporal Attention: We introduce a temporal attention mechanism at the frame level. Specifically, for the input feature maps X_t (here we use X_t to denote the input into each layer without distinguishing the original "frame sequence" for the first layer and "feature maps" for the hidden layers), the temporal attention mechanism learns a frame-wise weights matrix W_t via the transform function $\mathcal{T}_t$. In practice the input sliding window is usually too short for extracting the variance information among the consecutive frames, especially when a convolutional layer is applied to further reduce the length, and causes the loss of the valuable information for action recognition. To avoid this problem, we design our temporal attention function $\mathcal{T}_t$, which first expands the temporal dimension by a deconvolutional (Deconv) layer to preserve more temporal information, and then squeezes it by a convolutional layer (Conv2) to maintain the original length for further processing.

As we know, down-sampling will lose the information and up-sampling will increase the model capacity for capturing more information. For example, the traditional Deconv based on the bilinear interpolation can preserve the original information and increase more additional interpolated information [61]. We adopt the Deconv operation to up-sample the feature maps via learning the corresponding weights. Owing to the temporal space enlarged via Deconv operation, the information in the temporal space is expanded.

Specifically, the processing flow of $\mathcal{T}_t$, whose output can assign the attention weights $\mathrm{W}_t \in \mathbb{R}^{l\times 1}$ to each individual frames, can be defined as a composite function:

$$\mathcal{T}_t = \delta_t \circ \mathcal{S}_t \circ \psi_t \circ \mathcal{E}_t, \tag{1}$$

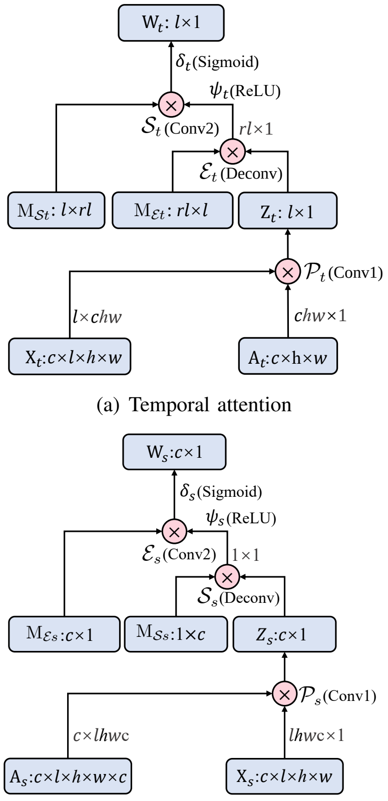
where $\mathcal{S}_t$ and $\mathcal{E}_t$ are the squeezing and expanding operations, with the corresponding filter parameters $\mathrm{M}_{\mathcal{S}t}$ and $\mathrm{M}_{\mathcal{E}t}$, respectively. $\mathcal{E}_t$ corresponds to the operation expanding the l-length input to the rl-length one, and $\mathcal{S}_t$ keeps the input length unchanged. $\circ$ denotes the composition operation over multiple functions, generating the compound function. We apply the nonlinear activation functions ψ_t and δ_t to further amplifying the differences among the temporal sequences. In our experiments, we choose ReLU and Sigmoid as the nonlinear activation functions ψ_t and δ_t, respectively.

2) Dimension Squeezing: The above temporal attention network module can differentiate the temporal variation and activate the informative part in the input sequence. However, since the input feature maps X_t consists of multiple channels, learning the attention weights will be computationally expensive. Besides, since the above module processes each channel independently, it is unable to exploit the global temporal attention based on the contextual information among multiple channels. To alleviate this problem, we further employ a squeezing function $\mathcal{P}_t$ to reduce the dimensions and leave only the temporal one, namely, transforming the input X_t to $\mathrm{Z}_t \in \mathbb{R}^{l\times 1}$ that only holds the temporal dimension.

The function $\mathcal{P}_t$ not only condenses the channel information, but also largely reduces the computation. In practice, it can be easily implemented by a convolutional layer (Conv1), which has $c \times h \times w$ learnable parameters $\mathrm{A}_t = \{a_j \in \mathbb{R}^{h\times w}, j = 1, \ldots, c\}$:

$$\mathrm{Z}_t(i) = \sum_{j=1}^{c}\sum_{m=1}^{h}\sum_{n=1}^{w} a_j(m,n)\cdot x_{i,j}(m,n). \tag{2}$$

3) Network Module: Figure 3(a) shows the details of the temporal attention network module. First the Conv1 layer combines information from multiple channels and



(a) Temporal attention

(b) Spatial attention

Fig. 3. The design details of the temporal attention and spatial attention network modules. Both the temporal attention module and the spatial attention module consist of two convolutional layers (Conv1 and Conv2) and one deconvolutional layer (Deconv), which respectively weight the input feature maps at the frame level and the channel level (⊗ denotes the matrix multiplication).

condenses them into the input frame-wise features in the temporal dimension. It behaves like the bottleneck structure [62] and significantly reduces the computational cost. After the dimension squeezing, the remaining operations in Equation (1) only work in the temporal dimension.

Specifically, $\mathcal{E}_t$ and $\mathcal{S}_t$ in Equation (1) respectively accomplish the expanding and squeezing operations, which can be implemented by the matrix multiplication in Deconv layer and Conv2 layer. The Deconv becomes quite useful and indispensable to enhance the temporal variance, when we place the temporal attention module at the higher layers of the deep 3D CNNs, because the input frame windows will be largely squeezed after a number of convolutions before the layer. The Conv2 layer keeps the temporal dimension the same as the input, and thus guarantees that our temporal attention module can be inserted into the 3D CNN architectures without network modification.

B. Spatial Attention Module

The various channels in 3D CNN models contain the correlated spatial information, and discovering the spatial attention across them also benefits the learning of discriminative features for actions. Inspired by [54], we attempt to exploit the differences between different channels in 3D CNNs, expecting the success in 2D CNN field can be transferred into 3D CNN field.

1) Spatial Attention: Specifically, for the input sequence X_s, our spatial attention network module applies the function $\mathcal{T}_s$ to discriminate the meaningful channels and acquire the corresponding score for each channel. The attention extracting process includes one channel squeezing operation and one expanding operation. Specifically, we attempt to borrow the bottleneck structure, where the squeezing function $\mathcal{S}_s$ compresses multiple channels into a correlated one by a deconvolution (Deconv), concentrating on the spatial information for the global view. Besides, to resume the channel dimension for module integration in the 3D CNNs, an expanding function $\mathcal{E}_s$ based on convolution (Conv2) is further applied at the channel level. Such a bottleneck structure not only purifies the spatial information, but also reduces the computational complexity. Now we have the formal definition of the spatial attention function $\mathcal{T}_s$:

$$\mathcal{T}_s = \delta_s \circ \mathcal{E}_s \circ \psi_s \circ \mathcal{S}_s, \quad (3)$$

where the nonlinear functions δ_s and ψ_s are also necessary in a channel-wise attention mechanism for enhancing the representational power. We choose Sigmoid and ReLU as δ_s and ψ_s, respectively. Similar to the temporal attention module, $\mathcal{E}_s$ and $\mathcal{S}_s$ are the expanding and squeezing operations, with the corresponding filter parameters $M_{\mathcal{E}s}$ and $M_{\mathcal{S}s}$, respectively. By inputting a number of action video sequence data X_s, the attention module $\mathcal{T}_s$ will learn the attention weight W_s for all channels.

2) Dimension Squeezing: Similar to temporal attention, we prefer to first reducing the dimensions involved in the 3D convolutions and squeeze both the temporal dimension and the feature map in each channel, remaining the channel dimension only. This operation makes the network module mainly focus on the spatial content among different channels, without the side effect of the other dimensions. Similarly, we have the dimension squeezing function $\mathcal{P}_s$ for spatial attention, transforming the input X_s into one dimensional sequence $Z_s \in \mathbb{R}^{c\times 1}$ for $\mathcal{T}_s$ to discriminate the most informative channels. This can be implemented by a convolution with $c \times l \times h \times w \times c$ learnable parameters $A_s = \{a_k \in \mathbb{R}^{l\times c\times h\times w}, k = 1, \ldots, c\}$:

$$Z_s(k) = \sum_{i=1}^{l}\sum_{j=1}^{c}\sum_{m=1}^{h}\sum_{n=1}^{w} a_k(i,j,m,n) \cdot x_{i,j}(m,n). \quad (4)$$

3) Network Module: Figure 3(b) shows the whole structure of the spatial module. Conv1 and Deconv layers together accomplish the compression of the feature map, temporal dimension and the channels. Namely, we squeeze all the other dimensions using $\mathcal{P}_s$ except the channel dimension in Conv1 layer, and exploit the bottleneck structure to reduce the computational complexity using $\mathcal{S}_s$ in Deconv layer. Conv2 layer further helps completing the channel resumption with more discriminative information at the channel level (i.e., $\mathcal{E}_s$). In practice, to reduce both the memory cost of GPU and the cost of multiplication computation, we can simply merge the squeezing channel operation $\mathcal{S}_s$ of Deconv into Conv1.

C. Spatio-Temporal Attention

We can put together the temporal and the spatial attention modules to constitute the spatio-temporal attention (STA) module for 3D CNN models. The STA module can simultaneously pursue the corresponding frame-wise and channel-wise attention weights, from the input temporal sliding window at a specified layer of the 3D CNNs.

1) Spatio-Temporal Composition: Traditional 3D CNN models apply the 3D convolutional kernel on the local receptive field, and thus lack the capability of acquiring the contextual information in the feature map.

To address this problem, in the STA module we first mix the squeezing operations along temporal and channel dimensions. This process can fully utilize the whole feature maps to capture the temporal and spatial attention, which at the same time can largely reduce the computational cost. We define a dimension squeezing function $\mathcal{P}$ as the Cartesian product of the squeezing functions in the spatial and temporal attention modules:

$$\mathcal{P} = \mathcal{P}_s \times \mathcal{P}_t, \tag{5}$$

transforming the input X into a two dimensional matrix. Later we will show that this actually corresponds to a convolution operation.

Based on the input squeezing, now we can have the transformation $\mathcal{T}$ for our STA module. The transformation consists of both the linear compression/expansion operations and the nonlinear activation to increase the model capacity:

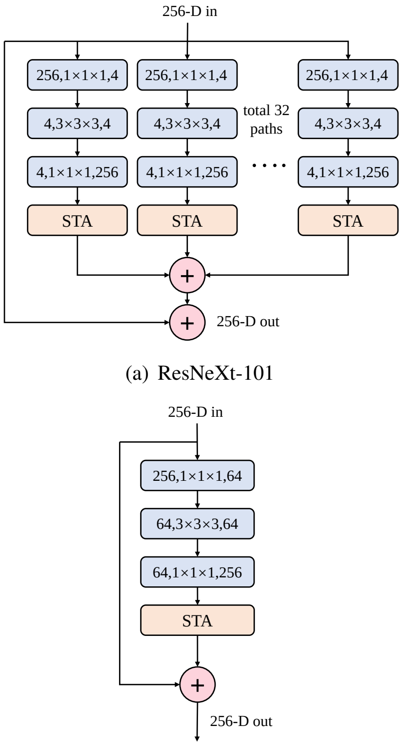
$$\mathcal{T} = \delta \circ (\mathcal{S}_t \circ \mathcal{E}_s) \circ \psi \circ (\mathcal{E}_t \circ \mathcal{S}_s) \circ \mathcal{P}, \tag{6}$$

where δ and ψ are the nonlinear activation functions, and we adopt Sigmoid and ReLU respectively. In our experiments, to reduce the number of multiplication computation and learnable parameters, we move the operation $\mathcal{S}_s$, squeezing channel dimension, from Deconv to Conv1 in Figure 2.

Finally, for the input temporal sliding window features X, our STA module learns the spatio-temporal attention weight matrix W to represent X by Y as follows:

$$y_{i,j} = w_{i,j} x_{i,j}. \tag{7}$$

2) Network Module: Corresponding to Equation (6), there are two convolutional layers (Conv1 and Conv2), one deconvolutional layer (Deconv) and two activation functions. Figure 2 shows the components of our STA integrating both the frame-wise and channel-wise attention. Usually we can simply place the module into the 3D CNN architecture, and feed it with the frame-wise features. Then Conv1 accomplishes $\mathcal{P}$ operation, i.e., the compression and isolation of the spatial information from each channel, only leaving the temporal dimension and channel dimension with the global spatial information. Then corresponding to $\mathcal{E}_t \circ \mathcal{S}_s$, the Deconv layer is appended to increase the learning capacity of the module in the inserted layer. Finally, we add another convolutional layer Conv2 to resume both the temporal length and the channel number as $\mathcal{S}_t \circ \mathcal{E}_s$ does.



(a) ResNeXt-101

(b) ResNet-152

Fig. 4. STA examples in ResNeXt-101 and ResNet-152: (a) a block of STA-ResNeXt-101 with cardinality = 32, (b) a block of STA-ResNet-152. We add our STA module at the bottom of the blocks (⊕ denotes the element-wise sum).

In Figure 4, we also give the examples of our STA in two state-of-the-art 3D CNN architectures: ResNeXt-101 and ResNet-152.

IV. Experiments

In this section, we will extensively evaluate our spatio-temporal attention (STA) network model in the tasks of video recognition and detection over several widely-used video datasets. Besides, we will compare our method with a number of state-of-the-art video recognition and detection methods including the two-stream CNN and 3D CNN methods. Furthermore, we conduct some extensive experiments to analyze and discuss our STA module.

A. Datasets

We adopt the three challenging datasets UCF-101 [72], HMDB-51 [73] and THUMOS 2014 [74] for video action recognition and detection. UCF-101 dataset consists of 13,320 videos from 101 realistic action categories, such as “Diving” and “Walking with a dog”, partitioned into three splits for training and testing. HMDB-51 dataset contains 7,000 video clips distributed in 51 action classes, which also has three splits as UCF-101. The above two datasets are widely used to evaluate the performance of action recognition algorithms. For action detection task, as in [34], [75], [76], we choose the THUMOS 2014 dataset consisting of 13,320 videos for training, 1,010 videos for

TABLE I
Top-1 accuracy performance on UCF-101 and HMDB-51 compared with state-of-the-art methods.

Type	Method	UCF-101	HMDB-51
RGB Only + Others	Girdhar *et al.* [56]	-	52.2
	ST Multiplier Net [63]	94.2	68.9
	Meng *et al.* [48]	87.1	53.1
	TSM+Offline [64]	95.9	73.5
	STM [65]	96.2	72.2
RGB Only + 3D CNNs	C3D [1]	77.4	46.7
	P3D-199 [18]	89.2	62.9
	STC-ResNet-101(64f) [16]	93.7	66.8
	Nonlocal-ResNeXt-101(64f) [58]	95.0	72.4
	I3D RGB + DMC-Net (I3D) [66]	96.5	77.8
RGB Only + 3D CNNs	ResNeXt-101(16f) [7]	90.7	63.8
	ResNeXt-101(64f) [7]	94.5	70.2
	STA-ResNeXt-101(64f) (Ours)	95.5	74.1
RGB + Optical flow	Two-stream CNN [9]	88.0	59.4
	TDD [67]	90.3	63.2
	Two-Stream Fusion [42]	92.5	65.4
	TSN [68]	94.2	69.4
	Piergiovanni *et al.* [47]	-	68.4
	Feichtenhofer *et al.* [63]	94.9	72.2
	I3D [19]	98.0	80.7
	PAN [69]	96.2	74.8
	Piergiovanni *et al.* [70]	-	81.1
	Zhao *et al.* [71]	98.2	81.3
RGB + Optical flow	MARS [5]	97.7	79.1
	STA-MARS (Ours)	97.7	80.2
	MARS+Flow [5]	98.0	79.5
	STA-MARS+Flow (Ours)	98.1	81.0
	MARS+RGB+Flow [5]	98.2	79.7
	STA-MARS+RGB+Flow (Ours)	**98.4**	**81.4**

validation, and 1,574 videos for testing from 20 action categories. Unlike UCF-101 and HMDB-51, the videos in THUMOS 2014 are untrimmed. The irrelevant frames in these untrimmed videos make the detection more difficult.

B. Implementation Details

Our STA module can be easily integrated into the existing 3D CNNs frameworks. In our experiments, we will plug it into a number of popular architectures for action recognition and detection, including ResNeXt-101(16f), ResNeXt-101(64f), ResNet-18(16f) and ResNet-152(16f) with different frame lengths such as 16 frames (16f) and 64 frames (64f). Since the receptive field in their later convolutional layers is relatively wider and more semantically meaningful for these powerful deep models, we simply prefer to place our STA module at their later convolutional layers so that the model can well capture the spatial and temporal correlations. In all the experiments, our model will be fine-tuned based on the well-trained raw 3D CNN models.

C. Action Recognition and Detection Performance

In our experiments, we first investigate the performance of our model in the tasks of action recognition and detection.

1) Action Recognition: We evaluate the action recognition performance on UCF-101 and HMDB-51 datasets. Table I reports the top-1 recognition accuracy. We compare our model with a number of state-of-the-art action recognition methods, including the two-stream models [5], [9], [19], [42], [70], [71], the 3D CNN models [1], [18], [66] and other very recent models [48], [65] extracting the temporal features. The results on each dataset are averaged over the three splits. For our method, we simply adopt and embed our module into ResNeXt-101 architecture, which is fine-tuned based on the published well-trained models in [5], [7].

TABLE II
Top-1 accuracy performance on different splits of UCF-101 and HMDB-51.

Split	Method	UCF-101	HMDB-51
1	ResNeXt-101	94	71.2
	STA-ResNeXt-101	95.0	75.0
	MARS+RGB+Flow	97.8	81.5
	STA-MARS+RGB+Flow	**98.1**	**82.4**
2	ResNeXt-101	94.8	68.9
	STA-ResNeXt-101	96.1	73.3
	MARS+RGB+Flow	98.5	80.1
	STA-MARS+RGB+Flow	**98.6**	**81.2**
3	ResNeXt-101	94.8	70.5
	STA-ResNeXt-101	95.4	74.1
	MARS+RGB+Flow	98.4	77.5
	STA-MARS+RGB+Flow	**98.5**	**80.5**

TABLE III
Complexity comparison on split1 of HMDB-51 using ResNeXt-101(64f) and STA-ResNeXt-101(64f) respectively, including the total number of parameters (# param), the size of the model and GFLOPs.

Method	# Param	Size (MB)	GFLOPs
ResNeXt-101(64f)	47.5×10^6	365	38.5014
STA-ResNeXt-101(64f)	47.6×10^6	384	38.5019

From Table I we can easily observe that the basic 3D CNN models (C3D [1] and P3D-199 [18]), only relying

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.2965434, IEEE Transactions on Multimedia

TABLE IV

Complexity comparison between STC and STA, including the total number of layers (# layers), parameters (# param) and multiplications(# mult).

Method	# layers	# param	# mult
STC [16]	10	$c^2 \times (l^2 + l + 2)/r \approx \mathcal{O}(c^2)$	$c \times (l + 3) + c^2 \times (l^2 + l + 2)/r \approx \mathcal{O}(c^2)$
STA	5	$c \times (h \times w + 4) + 5 \approx \mathcal{O}(c)$	$c \times l \times (h \times w + 4) + 3 \times l \approx \mathcal{O}(c)$

TABLE V

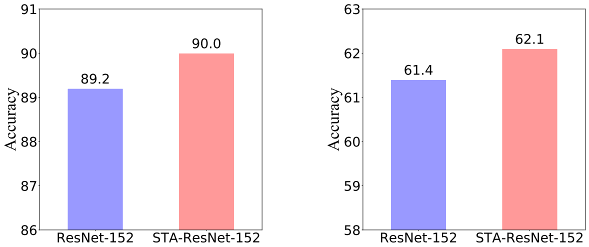
Top-1 accuracy performance on split1 of UCF-101 and HMDB-51 for C3D and P3D models.

Method	UCF-101	HMDB-51
C3D [1]	77.4	46.7
STA-C3D	**78.2**	**48.1**
P3D-199 [18]	89.2	62.9
STA-P3D-199	**89.8**	**64.3**

on the 3D convolutions without considering the spatio-temporal attentions, can hardly beat the others. Among all the methods, ResNeXt-101(64f), i.e., using 3D convolution with 64 frames per temporal sliding window, can obtain a very satisfying performance. But with the help of our STA module, STA-ResNeXt-101(64f) can faithfully obtain almost the best performance among RGB only methods, with encouraging performance gains based on the discriminating information at both frame level and channel level. The performance gain over HMDB-51 dataset is more obvious, increasing from the original 70.2% obtained by ResNeXt-101(64f) to 74.1%. Moreover, we insert our STA module into MARS [5] model, which is the state-of-the-art model using optical flow for action recognition with published source codes[1]. As shown in Table I, MARS embedding our STA module, obtains improved performance, especially, "STA-MARS+RGB+Flow" obtains the state-of-the-art performance, 98.4 and 81.4 on UCF-101 and HMDB-51 datasets respectively.

In Table II, we further investigate the performance over the three splits of UCF-101 and HMDB-51 using the ResNeXt-101 and MARS architectures. It is obvious that equipped with our STA module, STA-ResNeXt-101 can obtain improved performance, compared with the original ResNeXt-101 network on each split, without too much additional computational cost. Furthermore, "STA-MARS+RGB+Flow" obtains the best performance for every split. Table III reports the model complexity with or without STA module in ResNeXt-101 network, where the STA module only brings a slight increase (5.2%) in model size compared to the base one.

Additionally, we compare the complexity between STA and STC modules in Table IV. The state-of-the-art method STC needs 10 layers but our STA only needs 5 layers to model the spatio-temporal attention mechanism for video action analysis. Since c is usually larger (e.g., ResNeXt layer4: c=1024) than l, h and w (e.g., ResNeXt-101(64f) layer4: l=8, h=7, and w=7, respectively), the parameters and multiplication complexity in our STA module are $\mathcal{O}(c)$, while those of STC are $\mathcal{O}(c^2)$. We can see that our STA module is much more computational efficient than STC.



(a) UCF-101 split1

(b) HMDB-51 split1

Fig. 5. Accuracy performance of STA-ResNet-152 and ResNet-152 on split1 of UCF-101 and HMDB-51.

To further demonstrate the performance improvement when applying our STA module, we also adopt another popular network architecture C3D, P3D-199 and ResNet-152, and compare the performance on the split1 of UCF-101 and HMDB-51 respectively. For C3D and P3D-199, we adopted the pretrained models and code (C3D[2] is pretrained on Sports-1M; P3D-199[3] is pretrained on Kinetics600) and then fine-tuned our STA based on the original models. As Table V shows, both STA-C3D and STA-P3D-199 achieve better performance than the base networks C3D and P3D-199. Besides, we can also find the performance improvements on HMDB-51 dataset are larger than those on UCF-101. This is mainly due to the fact that the average video length in UCF-101 dataset is longer than that in HMDB-51 dataset.

Figure 5 shows the accuracy performance of STA-ResNet-152 and ResNet-152. Again, we observe that our STA module embedded in the 3D CNNs can help improve the performance, as it simultaneously considers both the spatial and the temporal attention in learning good features for video action. The experimental results over different 3D CNN architectures including ResNeXt-101 and ResNet-152 also show that STA is a generic and flexible module in practice.

2) Action Detection: Besides action recognition, we also study the performance of our STA module in the action detection task. In this experiment, we compare our model with the state-of-the-art method S-CNN [14]. The evaluation is conducted in the provided well-trained proposal network and localization network. We insert our module in the localization network based on ResNet-18 architecture. To fine-tune our STA 3D CNN model, we decompress UCF-101 and THUMOS 2014 videos into frames at 25 frames-per-second (fps). For UCF-101, we select temporal

[1] https://github.com/craston/MARS

[2] https://github.com/jfzhang95/pytorch-video-recognition

[3] https://github.com/qijiezhao/pseudo-3d-pytorch

TABLE VI
mAP PERFORMANCE USING OUR STA MODULE OVER 3D CNNS INCLUDING S-CNN AND R-C3D ON THUMOS 2014.

METHODS	IOU				
	0.1	0.2	0.3	0.4	0.5
S-CNN [14]	45.7	42.9	37.8	27.5	18.4
STA-S-CNN	**46.8**	**43.9**	**38.2**	**28.2**	**18.7**
R-C3D [30]	55.2	55.1	52.8	45.9	34.8
STA-R-C3D	**56.6**	**56.4**	**53.4**	**47.5**	**36.8**

TABLE VII
TOP-1 ACCURACIES OF OUR SA-RESNEXT-101, TA-RESNEXT-101, STA-RESNEXT-101 AND SENET-RESNEXT-101 MODELS ON UCF-101 AND HMDB-51 SPLIT1.

METHOD	UCF-101	HMDB-51
ResNeXt-101(64f)	94.0	71.2
SENet-ResNeXt-101(64f)	94.2	74.0
SA-ResNeXt-101(64f)	94.3	74.2
TA-ResNeXt-101(64f)	**95.0**	73.9
STA-ResNeXt-101(64f)	**95.0**	**75.0**

TABLE VIII
TOP-1 ACCURACY PERFORMANCE WHEN EMBEDDING STA MODULE INTO DIFFERENT LAYERS, I.E., THE EARLY LAYERS (EL) AND THE LATER LAYERS(LL), OF RESNEXT-101.

METHOD	UCF-101	HMDB-51
ResNeXt-101(64f)	94.0	71.2
STA-ResNeXt-101(64f)-EL	94.5	73.0
STA-ResNeXt-101(64f)-LL	**95.0**	**75.0**

sliding windows, whose Intersection-over-Union (IoU) with groundtruth instance is larger than 0.5, as our training samples, and set its IoU as the measurement of overlap in the experiment.

The mean average precision (mAP) results are shown in Table VI. From the table, we can see that our STA-S-CNN consistently outperforms the original S-CNN [14] with respect to different IoU settings. Especially when IoU is lower, the performance gain becomes more significant. This phenomenon indicates that our STA module, capable of distinguishing the frames in the temporal dimension, can explore the most informative ones, even when they fall in the overlap region between the predicted sliding window and the groundtruth. Subsequently, it can robustly boost the detection accuracy under extreme scenarios.

Figure 6 demonstrates this observation by two action detection examples on THUMOS 2014. In the two examples, both STA-S-CNN and S-CNN models attempt to detect the action "Javalin Throw" and "Basketball Dunk" respectively, and the overlaps are set between 0.1 and 0.2 during the evaluation. For action "Javalin Throw", we can see that S-CNN fails to correctly detect the action, but STA-S-CNN can find the most important frames (i.e., from frame #14480 to #14527) with strong temporal attention for the action, and thus is able to generate the correct prediction. A similar observation can be made in the example of action "Basketball Dunk" detection.

To further demonstrate the effectiveness of our STA module, we also insert our STA module into the R-C3D [30] model (named STA-R-C3D). For both R-C3D and STA-R-C3D, we choose the same pretrained model and code for training (R-C3D[4] is pretrained on Sports-1M). As Table VI shows, compared to original R-C3D model, our STA-R-C3D can get higher mAP performance, which is similar to the performance when using S-CNN.

D. Model Analysis and Discussion

In this section, we will extensively study the proposed STA module, including the effect of attention mechanism, the parameter setting, and the place to plug it in.

1) Ablation Analysis: First we check the effect of the spatial attention (SA), temporal attention (TA) and spatio-temporal attention (STA) respectively. In this experiment, we adopt the base model ResNext-101(64f), and form three 3D CNN models with different attention parts, namely SA-ResNext-101(64f), TA-ResNext-101(64f) and STA-ResNext-101(64f). We evaluate their performance on split1 of UCF-101 and HMDB-51 datasets, and report the best accuracy they obtained in Table VII. From the table, we can see that both our spatial attention and temporal attention can help improve the recognition accuracy, owing to the fact that the mechanisms can exploit the most discriminative information at frame level or channel level respectively. Besides, the two attention modules combined (i.e., our STA) can obtain further performance gains with more spatio-temporal feature learning power. Additionally, we also show the performance for SENet because its behavior similar to our SA module. However, our SA-ResNeXt-101(64f) model can obtain better performance compared against SENet-ResNeXt-101(64f) on both datasets.

2) Embedded Layer: As mentioned previously, we usually prefer to placing the STA module at the later convolutional layers of the 3D CNN models, since STA can capture temporal and spatial attentions at the high level. To validate this, Table VIII shows the difference when placing the STA module at the early layers (stage conv2) and the later layers (stage conv5) of ResNext-101. On both UCF-101 and HMDB-51, STA embedded into the later layers can obtain the best performance. This is consistent with our intuition that later layers of the deep networks provide more semantic and meaningful information in temporal and spatial dimensions. This means that in practice it is more necessary to capture the attentions and then determine the beneficial frames and channels at the later layers.

3) Temporal Deconvolution & Step Size: In standard 3D CNN models, the convolution operation along the temporal dimension will compress the video sequence and inevitably lose the valuable information for action recognition. This is more critical at the later layers where the temporal dimension has been largely transformed and squeezed. To alleviate this issue, a deconvolution (Deconv) operation is introduced in our STA module, which will stretch the temporal length and enhance the feature representing capacity.

We evaluate the performance of the STA-ResNeXt-101(64f) network with or without the Deconv layer. Table IX lists the accuracies of the two settings on the split1 of

[4] https://github.com/sunnyxiaohu/R-C3D.pytorch

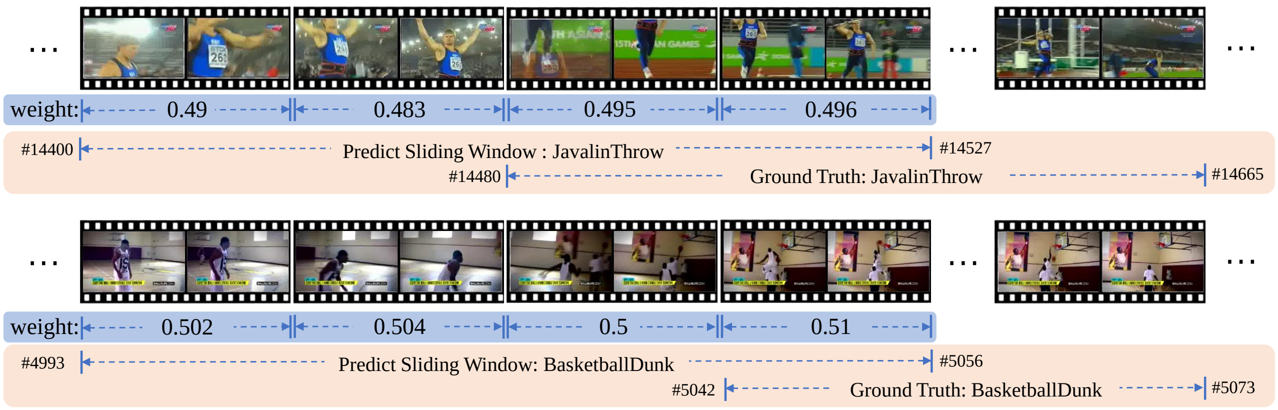
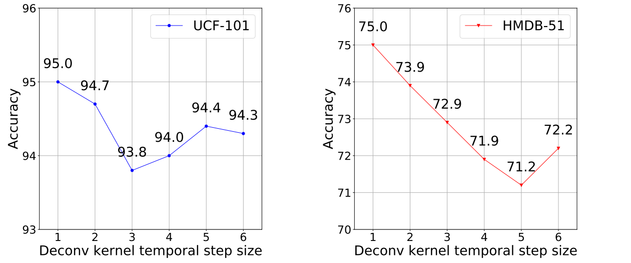

weight:
0.49
0.483
0.495
0.496
#14400
Predict Sliding Window : JavalinThrow
#14527
#14480
Ground Truth: JavalinThrow
#14665
weight:
0.502
0.504
0.5
0.51
#4993
Predict Sliding Window: BasketballDunk
#5056
#5042
Ground Truth: BasketballDunk
#5073

Fig. 6. Illustrative examples showing the action detection result when the overlap threshold is lower. The overlap threshold is set between 0.1 and 0.2 during evaluation. Detection results for two action instances ("Javalin Throw" and "Basketball Dunk") from THUMOS 2014 dataset: two correct predictions are obtained by STA-S-CNN model, while neither is correct by the original S-CNN model.

TABLE IX
THE EFFECT OF DECONV LAYER USING STA-RESNEXT-101(64F) ON UCF-101 AND HMDB-51.

SETTINGS	UCF-101	HMDB-51
w/o Deconv	94.5	74.4
w/ Deconv	**95.0**	**75.0**



(a) UCF-101 (b) HMDB-51

Fig. 7. The effect of different temporal step sizes in Deconv layer. (a) Top-1 accuracy performance for different step sizes of STA-ResNeXt-101 on UCF-101 split1. (b) Top-1 accuracy performance for different step sizes of STA-ResNeXt-101 on HMDB-51 split1.

UCF-101 and HMDB-51. It is easy to see that using the temporal deconvolution can enlarge the learning capacity of our STA module, and thus increase the recognition performance.

In Figure 7 we also investigate the effect of the temporal step size in the Deconv operation. On both datasets, we can see that the small step size (i.e., 1) gives the best performance. This is because the larger step size indicates an undesirable stretching transformation along the temporal dimension, which might bring more distortions among the frames for recognition. Therefore, in practice a proper small step size for deconvolution is preferred for a satisfying performance without too much computations. In our experiments setting, we choose 1 and 3 for step size and kernel size respectively to accomplish up-sampling operation.

4) Sliding Window Size: The STA module works over an input window to capture the temporal attentions. Therefore, it is important to see the connections between the window size and the performance of STA module. In Table X we compare the performance of ResNeXt-101 model using different window sizes including 16 and 64 respectively. It is obvious that with a large input window, our STA module can provide much better performance on both UCF-101 and HMDB-51 datasets. This is because the long input frame sequence conveys more temporal information. Subsequently, it is easier for STA to capture the frame-wise attention, and find the most meaningful frames characterizing the action.

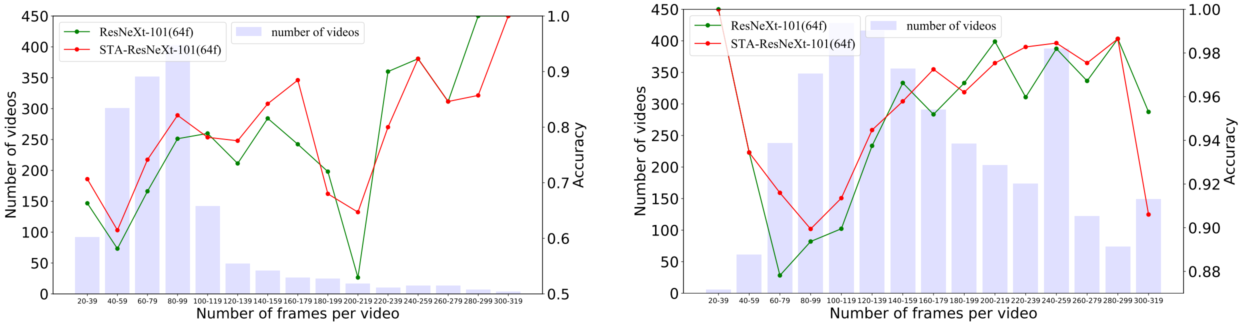
TABLE X
THE EFFECT OF THE SLIDING WINDOW SIZE ON SPLIT1 OF UCF-101 AND HMDB-51.

METHOD	UCF-101	HMDB-51
ResNeXt-101(16f)	90.1	64.1
STA-ResNeXt-101(16f)	90.7	64.6
STA-ResNeXt-101(64f)	**95.0**	**75.0**

5) Video Length: In Figure 8 (a) and (b) we show the statistics of the video length (i.e., the number of frames decompressed with 30 fps) in HMDB-51 and UCF-101 datasets. It is easy to see that the video length of HMDB-51 varying mainly between 20 and 120, is very different from that of UCF-101 ranging from 60 to 320. This means that on the average the video in UCF-101 is longer than that in HMDB-51. Furthermore, we also investigate the performance using our STA model on videos of different length from both datasets. Figure 8 also shows the accuracy curves, where the input sliding window length of our STA model is set to 64 frames. On both datasets, we can conclude that the proposed STA model can work better on short videos than on longer ones. This is mainly because by setting a proper sliding window length, STA can find the most meaningful frames characterizing the action in the input window. This helps explain why in Table I in most cases the performance improvement on HMDB-51 is larger than that on UCF-101.

V. CONCLUSIONS

In this paper we proposed a novel spatio-temporal attention (STA) module to improve 3D CNNs for action recog-



(a) HMDB-51

(b) UCF-101

Fig. 8. Performance comparison between ResNeXt with STA and without STA on HMDB-51 and UCF-101 datasets.

nition and detection, which attempts to exploit the discriminative information at both frame level and channel level. Comprised of a number of convolution and deconvolution operations, our STA module simultaneously distinguishes the characteristics in temporal and spatial dimensions, and further improves the capability of the 3D CNNs with more powerful feature learning. The proposed module serves as a generic module for many 3D CNNs without increasing too much computational cost. Our experiments on several state-of-the-art 3D CNNs architectures and three different datasets have demonstrated that the STA module can obtain the state-of-the-art performance in action recognition task (98.4% and 81.4% on UCF-101 and HMDB-51 datasets respectively), and achieve improved performance for action detection task.

REFERENCES

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in *IEEE International Conference on Computer Vision*, 2015.

[2] J. Zhang, K. Mei, Y. Zheng, and J. Fan, “Exploiting mid-level semantics for large-scale complex video classification,” *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2518–2530, 2019.

[3] Z. Gao, L. Wang, Q. Zhang, Z. Niu, N. Zheng, and G. Hua, “Video imprint segmentation for temporal action detection in untrimmed videos,” in *the Association for the Advance of Artificial Intelligence*, 2019.

[4] H. Song, X. Wu, B. Zhu, Y. Wu, M. Chen, and Y. Jia, “Temporal action localization in untrimmed videos using action pattern trees,” *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 717–730, 2019.

[5] Nieves Crasto, Philippe Weinzaepfel, Karteek Alahari, and Cordelia Schmid, “MARS: Motion-augmented rgb stream for action recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[6] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” *arXiv preprint arXiv:1711.11248*, 2017.

[7] K. Hara, H. Kataoka, and Y. Satoh, “Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and imagenet,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[8] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *IEEE International Conference on Computer Vision*, 2014.

[9] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in Neural Information Processing Systems*, 2014.

[10] Y. Shi, Y. Tian, Y. Wang, and T. Huang, “Sequential deep trajectory descriptor for action recognition with three-stream CNN,” *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1510–1520, 2017.

[11] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang, “End-to-end learning of motion representation for video understanding,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[12] H. Kim, J. Lee, and H. Yang, “Human action recognition using a modified convolutional neural network,” in *International Symposium on Neural Networks*, 2007.

[13] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[14] Z. Shou, D. Wang, and S. Chang, “Temporal action localization in untrimmed videos via multi-stage CNNs,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[15] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, “Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification,” in *European Conference on Computer Vision*, 2018.

[16] A. Diba, M. Fayyaz, V. Sharma, M. Arzani, R. Yousefzadeh, J. Gall, and L. Gool, “Spatio-temporal channel correlation networks for action classification,” in *European Conference on Computer Vision*, 2018.

[17] X. Wang, L. Gao, P. Wang, X. Sun, and X. Liu, “Two-stream 3-D convnet fusion for action recognition in videos with arbitrary size and length,” *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 634–644, 2018.

[18] Z. Qiu, T. Yao, and T. Mei, “Learning spatio-temporal representation with pseudo-3D residual networks,” in *IEEE International Conference on Computer Vision*, 2017.

[19] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[20] Y. Zhou, X. Sun, Z. Zha, and W. Zeng, “MiCT: Mixed 3D/2D convolutional tube for human action recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[21] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, “Real-time action recognition with enhanced motion vector CNNs,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[22] L. Wang, W. Li, W. Li, and L. Van Gool, “Appearance-and-relation networks for video classification,” *arXiv preprint arXiv:1711.09125*, 2017.

[23] Ronald A. Rensink, “The dynamic representation of scenes,” *Visual Cognition*, vol. 7, no. 1-3, pp. 17–42, 2000.

[24] S. Sharma, R. Kiros, and R. Salakhutdinov, “Action recognition using visual attention,” *arXiv preprint arXiv:1511.04119*, 2015.

[25] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng, “Temporal action localization by structured maximal sums,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[26] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, “Turn tap: Temporal unit regression network for temporal action proposals,” in *IEEE International Conference on Computer Vision*, 2017.

[27] J. Gao, K. Chen, and R. Nevatia, “Ctap: Complementary temporal action proposal generation,” in *European Conference on Computer Vision*, 2018.

[28] Y. Chao, S. Vijayanarasimhan, B. Seybold, D. Ross, J. Deng, and R. Sukthankar, “Rethinking the faster R-CNN architecture for temporal action localization,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[29] A. Piergiovanni and M. Ryoo, "Temporal gaussian mixture layer for videos," *arXiv preprint arXiv:1803.06316*, 2018.

[30] H. Xu, A. Das, and K. Saenko, "R-C3D: Region convolutional 3D network for temporal activity detection," in *IEEE International Conference on Computer Vision*, 2017.

[31] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "RGB-D-based human motion recognition with deep learning: A survey," *Computer Vision and Image Understanding*, vol. 171, pp. 118–139, 2018.

[32] D. Oneata, J. Verbeek, and C. Schmid, "The lear submission at thumos 2014," 2013.

[33] S. Karaman, L. Seidenari, and A. D. Bimbo, "Fast saliency based pooling of fisher encoded dense trajectories," in *European Conference on Computer Vision THUMOS Workshop*, 2014.

[34] S. Yeung, O. Russakovsky, G. Mori, and F. Li, "End-to-end learning of action detection from frame glimpses in videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[35] R. Hou, R. Sukthankar, and M. Shah, "Real-time temporal action localization in untrimmed videos by sub-action discovery," in *British Machine Vision Conference*, 2017.

[36] J. Gleason, R. Ranjan, S. Schwarcz, C. Castillo, J. Chen, and R. Chellappa, "A proposal-based solution to spatio-temporal action detection in untrimmed videos," in *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[37] L. Wang, Y. Qiao, and X. Tang, "Action recognition and detection by combining motion and appearance features," *THUMOS14 Action Recognition Challenge*, vol. 1, no. 2, pp. 2, 2014.

[38] A. Richard, H. Kuehne, and J. Gall, "Action sets: Weakly supervised action segmentation without ordering constraints," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[39] Z. Luo, B. Peng, D. Huang, A. Alahi, and F. Li, "Unsupervised learning of long-term motion dynamics for videos," *arXiv preprint arXiv:1701.01821*, vol. 2, 2017.

[40] K. Soomro and M. Shah, "Unsupervised action discovery and localization in videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[41] X. Peng and C. Schmid, "Multi-region two-stream R-CNN for action detection," in *European Conference on Computer Vision*, 2016.

[42] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[43] Y. Zhu, Z. Lan, S. Newsam, and A. Hauptmann, "Hidden two-stream convolutional networks for action recognition," *arXiv preprint arXiv:1704.00389*, 2017.

[44] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S. Chang, "CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[45] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[46] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015.

[47] A. Piergiovanni, C. Fan, and M. Ryoo, "Learning latent subevents in activity videos using temporal attention filters," in *the Association for the Advance of Artificial Intelligence*, 2017.

[48] L. Meng, B. Zhao, B. Chang, G. Huang, W. Sun, F. Tung, and L. Sigal, "Interpretable spatio-temporal attention for video action recognition," in *IEEE International Conference on Computer Vision Workshops*, 2019.

[49] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. Hoi, and H. Ling, "Learning unsupervised video object segmentation through visual attention," in *IEEE Conference on Computer Vision and Pattern Recognition*.

[50] R. Pramono, Y. Chen, and W. Fang, "Hierarchical self-attention network for action localization in videos," in *IEEE International Conference on Computer Vision*, 2019.

[51] A. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv:1509.00685*, 2015.

[52] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015.

[53] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," *arXiv preprint arXiv:1704.06904*, 2017.

[54] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, 2017.

[55] S. Woo, J. Park, J. Lee, and I. Kweon, "CBAM: Convolutional block attention module," in *European Conference on Computer Vision*, 2018.

[56] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," in *Advances In Neural Information Processing Systems*, 2017.

[57] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[58] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

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

[60] S. Frintrop, "Computational visual attention," *Computer Analysis of Human Behavior*, pp. 69–101, 2011.

[61] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE International Conference on Computer Vision*, 2015.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[63] C. Feichtenhofer, A. Pinz, and R. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[64] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *IEEE International Conference on Computer Vision*, 2019.

[65] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, "STM: Spatiotemporal and motion encoding for action recognition," in *IEEE International Conference on Computer Vision*, 2019.

[66] Z. Shou, X. Lin, Y. Kalantidis, L. Sevilla-Lara, M. Rohrbach, S. Chang, and Z. Yan, "DMC-Net: Generating discriminative motion cues for fast compressed video action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[67] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[68] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European Conference on Computer Vision*, 2016.

[69] C. Zhang, Y. Zou, G. Chen, and L. Gan, "PAN: Persistent appearance network with an efficient motion cue for fast action recognition," in *ACM Conference on Multimedia*, 2019.

[70] A. Piergiovanni and M. Ryoo, "Representation flow for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[71] H. Zhao, A. Torralba, L. Torresani, and Z. Yan, "HACS: Human action clips and segments dataset for recognition and temporal localization," in *IEEE International Conference on Computer Vision*, 2019.

[72] K. Soomro, A. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[73] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *IEEE International Conference on Computer Vision*, 2012.

[74] Y. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "THUMOS challenge: Action recognition with a large number of classes," http://crcv.ucf.edu/THUMOS14/, 2014.

[75] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *IEEE International Conference on Computer Vision*, 2017.

[76] J. Huang, N. Li, T. Zhang, and G. Li, "A self-adaptive proposal model for temporal action detection based on reinforcement learning," *arXiv preprint arXiv:1706.07251*, 2017.