Attention module-based spatial temporal graph convolutional networks for skeleton-based action recognition

Yinghui Kong,[a] Li Li,[a] Ke Zhang,[a] Qiang Ni,[b] Jungong Han[b]
[a]North China Electric Power University, Department of Electronic and Communication Engineering, North Yonghua Road #619, Baoding, China, 071000
[b]Lancaster University, School of Computing and Communications, Bailrigg, Lancaster, United Kingdom, LA1 4YW

Abstract. Skeleton-based action recognition is a significant direction of human action recognition, because the skeleton contains important information for recognizing action. The spatial temporal graph convolutional networks (ST-GCN) automatically learn both the temporal and spatial features from the skeleton data, and achieve remarkable performance for skeleton-based action recognition. However, ST-GCN just learn local information on a certain neighborhood, but does not capture the correlation information between all joints (i.e., global information). Therefore, we need to introduce global information into the spatial temporal graph convolutional networks. In this work, we propose a model of dynamic skeletons called attention module-based Spatial Temporal Graph Convolutional Networks (AM-STGCN), which solves these problems by adding attention module. The attention module can capture some global information, which brings stronger expressive power and generalization capability. Experimental results on two large-scale datasets, Kinetics and NTU-RGB+D, demonstrate that our model achieves significant improvements over previous representative methods.

Keywords: action recognition, spatial temporal graph convolution network, non-local neural network, attention module.

***Yinghui Kong,** E-mail: kongyhbd2015@ncepu.edu.cn

1 Introduction

Action recognition technology plays an increasingly important role in many fields such as intelligent monitoring, human-computer interaction, video sequence understanding, and medical health. Video action recognition technology is challenged by factors such as occlusion, dynamic background, mobile camera, angle of view and illumination change.

Before the advent of deep learning, the best algorithm for human action recognition in video was iDT[1,2], and the subsequent works were basically improved based on the iDT method. Human action recognition uses multiple modalities of data such as appearance, depth, optical flows, and body skeletons.[3] With the continuous development of deep learning and its excellent performance in image understanding tasks, more and more researchers are beginning to use deep learning methods to solve the problem of video analysis. Action recognition methods based on RGB video or optical flows, such as Two-Stream[4,5], C3D[6], I3D[7], RNN[8] methods, are greatly affected by illumination, scene and camera lens movement, so it is difficult to describe the motion of the human body in the sequence, the recognition performance in some complex datasets needs to be improved. In recent years, due to the cost reduction of depth sensors (such as Kinect) and the emergence of real-time human pose estimation algorithms, skeleton-based action recognition has become more and more popular.

Skeleton-based action recognition methods have been widely studied and paid attention due to its strong adaptability to dynamic environments and complex backgrounds. Traditional methods[9,10] require hand-crafted features and traversal rules, which are less efficient. Ordinary deep learning-based methods[11-20] manually structure the skeleton into joint coordinate vectors or pseudo-images, which are then sent to the RNN or CNN network for prediction of the action categories. The human skeleton is naturally constructed as a graph in a non-Euclidean space, in which the joint acts as a node, and the edge is constructed according to the natural connection relationship of the human body. Recently, the Graph Convolutional Networks (GCN) have extended convolution operations from images to graph structures, and have been successfully applied to many applications. For skeleton-based action recognition, GCN-based methods contain ST-GCN[3], STGC[21], SR-TSL[22], AGCN[23], PB-GCN[24], GR-GCN[25] and DPRL+GCNN[26]. ST-GCN applied GCN for skeleton-based action recognition task and directly model the original skeleton data, it extended graph neural networks to a spatial-temporal graph model, and obtained better action representations. Compared to ordinary deep learning-based methods, GCN-based methods can better express the dependencies between joints. However, the convolution operation in the ST-GCN method is performed on the 1-neighbor of the root node and cannot capture global information. For the action categories in which the interaction joints are not in the same neighborhood, such as brushing, clapping, but there are relations between these nonadjacent joints, attention mechanism can learn these relations. Paying more attention to those joints may improve recognition performance. Attention modules that work well include non-local neural networks[27], Interaction-aware attention[28], CBAM[29], SENet[30] etc.

In order to solve this problem, we propose an improved method based on ST-GCN, which is attention module-based Spatial Temporal Graph Convolutional Networks (AM-STGCN). Attention module helps the model focus on all positions and learn different weights for each position. In AM-STGCN, we add the non-local neural network as an attention module after the convolution operation of the baseline model ST-GCN to learn the feature representation with long-range dependencies. In addition, we discussed the effects of adding attention blocks to different layers, as well as the effects of adding multiple attention blocks. We did a lot of experimentation and analysis, and finally got the best strategy. The experimental results on two large-scale action recognition datasets Kinetics[31] and NTU-RGB+D[32] show that AM-STGCN can significantly outperform ST-GCN in action recognition.

In the remainder of the paper, we first provide some related work in Sec. 2, and then introduce the original ST-GCN model and our AM-STGCN model in Sec. 3. We summarize and analyze the experimental results in Sec. 4. Finally, we draw conclusions and point out future research direction in Sec. 5.

2 Related Work

2.1 Action Recognition Based on RGB Video or Optical Flows

Most previous studies were based on RGB video or optical flows. Traditional action recognition methods are mostly based on optical flows, and the representative algorithm is iDT[1,2]. DT algorithm utilize optical flow field to obtain some trajectories in the video sequence, then extract the HOF, HOG, MBH and trajectory characteristics along the trajectory. IDT improves dense trajectories by explicitly estimating camera motion. Then, some methods based on deep learning gradually appeared, and their performance was much better than traditional methods. Two-stream method was originally proposed by Simonyan et al.[4], and Feichtenhofer et al.[5] improved the model. Two-stream method utilizes both appearance and optical flows information: in spatial stream, in the form of appearance on a single frame, the scene and target information depicted by video are carried; in temporal stream, the motion of the observer (camera) and the target are expressed in the form of multi-frame optical flows. Tran et al.[6] adopted 3D convolution and 3D pooling to construct a network, which can directly process video, and its efficiency is much higher than other methods. Carreira et al.[7] proposed a model named "I3D" based on Inceptionv1, which inflates Inceptionv1's filters and pooling kernels into 3D, leading to very deep, naturally spatiotemporal classifiers. Du et al.[8] introduced a novel pose-attention mechanism to adaptively learn pose-related features at every time-step action prediction of RNNs.

Although action recognition methods based on RGB video or optical flows perform high performance, there are still some problems. For example, it is susceptible to background, illumination and appearance changes, and extract optical flow information requires high computational cost.

2.2 Skeleton-based Action Recognition

The human skeleton can provide a very good representation of the human body motions, which is beneficial to the analysis of human actions. On the one hand, skeleton data is inherently robust in background noise, and provides abstract and high-level features of human motion. On the other hand, the size of the skeleton data is very small compared to RGB data, which allows us to design a lightweight and hardware-friendly model.

Skeleton-based action recognition approaches can be categorized into traditional methods and deep learning methods. Deep learning methods contain RNN based methods, CNN based methods and graph convolutional network (GCN) based methods.

Some traditional methods shown in Refs. 9 and 10 require hand-crafted features and traversal rules to achieve skeleton action recognition. With the development of deep learning, RNN based methods appears gradually. Du et al.[11] divided the human skeleton into five parts according to human physical structure, and then separately feeded them to five bidirectionally recurrently connected subnets. Song et al.[12] proposed an end-to-end spatial and temporal attention model, which learns to selectively focus on discriminative joints of skeleton within each frame of the inputs and pays different levels of attention to the outputs of different frames. Zhang et al.[13] designed a view adaptive recurrent neural network (RNN) with LSTM architecture, which enables the network itself to adapt to the most suitable observation viewpoints from end to end. In recent years, a number of CNN based approaches have also emerged. Kim et al.[14] re-designed the original TCN by factoring out the deeper layers into additive residual terms which yields both interpretable hidden representations and model parameters. Liu et al.[15] proposed an enhanced skeleton visualization method to represent a skeleton sequence as a series of visual and motion enhanced color images, which implicitly describe spatio-temporal skeleton joints in a compact yet distinctive manner. Li et al.[16] designed a novel skeleton transformer module to rearrange and select important skeleton joints automatically. Li et al.[17] proposed an end-to-end convolutional co-occurrence feature learning framework to aggregate different levels of contextual information. Liu et al.[18] proposed a recurrent attention mechanism for their GCA-LSTM network, which is able to selectively focus on the informative joints in the action sequence with the assistance of global contextual information. Xie et al.[19] designed a temporal-then-spatial recalibration scheme, resulting in an end-to-end Memory Attention Networks (MANs) which consist of a Temporal Attention Recalibration Module (TARM) and a Spatio-Temporal Convolution Module (STCM). Zheng et al.[20] designed an adaptive attentional module to focus attention on the most discriminative parts in the single skeleton. Although RNN based methods has a strong ability to model sequence data, and CNN based methods has good parallelism and easier training process, however, neither CNN nor RNN fully represent the structure of the skeleton.

Recently, some methods based on graph convolution have appeared, and the effect has been improved obviously. Yan et al.[3] directly simulated the original skeleton using the graph convolution, which eliminates the need for manual part assignment, and it is easier to design and potent to learn better action representations. Li et al.[21] designed multi-scale convolutional filters to encode the graph structure data, and proposed a recursive graph convolution model. Si et al.[22] utilized a spatial reasoning network to capture the high-level spatial structural features within each frame, and utilized a composition of multiple skip-clip LSTMs to model the detailed temporal dynamics of skeleton sequences. In order to design individual graphs for different samples, Shi et al.[23] introduced non-local neural networks into graph convolution operation to model the multi-level semantic information, which brings more flexibility and generality.

Thakkar et al.[24] divided the skeleton graph into four subgraphs, and used relative coordinates and temporal displacements as features at each node instead of 3D joint coordinates which improves action recognition performance. Gao et al.[25] constructed a generalized graph via spectral graph theory to capture the space-time variation. Tang et al.[26] proposed a deep progressive reinforcement learning (DPRL) method to extract key frames, and employed the graph-based convolutional neural network to capture the dependency between the joints for action recognition.

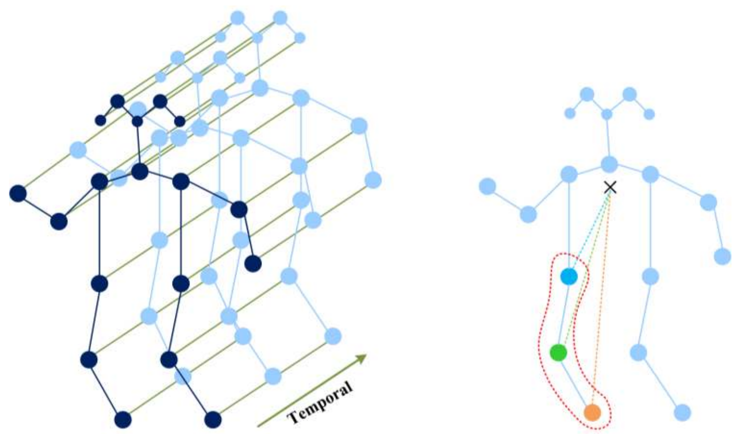
3 Methodology

We briefly describe the original spatial temporal graph convolutional networks (ST-GCN) in Sec. 3.1. And in Sec. 3.2, we give a briefly description about the methods of utilizing the attention module to boost the performance, and propose the improved model -- attention module-based spatial temporal graph convolution network (AM-STGCN).

3.1 Spatial-Temporal Graph Convolutional Networks (ST-GCN)

As shown in Ref. 3, the authors take joints as nodes and the connections between nodes as edges to construct the skeleton graph. Fig. 1 (a) shows an example of a spatial-temporal skeleton graph. In one frame, the natural connections between the joints (i.e., the human bones) act as spatial edges; in adjacent frames, the same joints are joined as temporal edges. The property of each node is the coordinate vector of the joint. Multi-layers spatial-temporal graph convolution operation is applied to the spatial-temporal skeleton graph to obtain advanced feature map, and then use the SoftMax classifier to predict the action category.

ST-GCN applies the spatial configuration partitioning strategy shown in Fig. 1(b) in frame. The spatial configuration partitioning strategy divides the node's 1-neighbor into three subsets: 1) the root node (green dot); 2) the centripetal subset (blue dots): the neighboring nodes closer to the gravity center of the skeleton (black cross); 3) the centrifugation subset (yellow dots): the neighboring nodes that are further to the gravity center of the skeleton. Each color in the Fig. 1(b) corresponds to a specific learnable weight vector. The authors of ST-GCN propose three partitioning strategy, and it has been proved that the spatial configuration partitioning strategy shown in Fig. 1(b) is the best, so this work directly adopts this strategy.



(a) (b)

Fig. 1 (a) Spatial temporal graph of the skeleton. (b) Partitioning strategy, different colors represent different subsets.

Spatial graph convolution is formulated as:

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(v_{tj}) \cdot w(l_{ti}(v_{tj})), \tag{1}$$

where f is the feature map. v_{ti} is the node of the graph. $\boldsymbol{B}(v_{ti})$ is the sampling area, which is defined as the 1-neighbor set of joint nodes. The neighbor set $\boldsymbol{B}(v_{ti})$ of a joint node v_{ti} is partitioned into a fixed number of K subsets, where each subset has a numeric label.[3] The mapping function l_{ti} maps a node in the neighborhood to its subset label. The weight function w gives different weights according to different l_{ti} values. The normalizing term $Z_i(v_j)$ equals the cardinality of the corresponding subset.

To model the spatial temporal dynamics within skeleton sequence, since the number of neighbors per node is fixed at 2 (the corresponding joint in the previous and subsequent frames), it is directly to perform the graph convolution similar to the classical convolution operation, concretely, we perform a $K_t \times 1$ convolution on the output feature map computed above.[23]

In the single frame case, ST-GCN with the spatial configuration partitioning strategy can be implemented with the following formula:

$$f_{out} = \sum_j (\Lambda_j^{-\frac{1}{2}} \mathrm{A}_j \Lambda_j^{-\frac{1}{2}}) \otimes \mathrm{M}_j f_{in} W_j \,. \tag{2}$$

In formula 2, f is the $C_{in} \times T \times V$ feature map where V denotes the number of nodes, T denotes the temporal length and C_{in} denotes the number of input channels. A is the $18\times18\times3$ adjacency matrix, whose element A_{ij} indicates whether the node v_i is in the subset of node v_j. $\mathrm{A}_0 = \mathrm{I}$ denotes the self-connections of vertexes, A_1 denotes the connections of centripetal subset and A_2 denotes the centrifugal subset. $\Lambda_j^{ii} = \sum_k (A_j^{ki}) + \alpha$ is the normalized diagonal matrix, α is set to 0.001 to avoid the empty rows in A. W_j is the $C_{out} \times C_{in} \times 1 \times 1$ weight vector of the 1×1 convolution operation. M is a $V \times V$ learnable attention map which indicates the importance of each node. $\otimes$ denotes the element-wise product between two matrixes. This means that if one of the elements in A is 0, then whatever the value of M is, it will always be 0. So M just operates in the 1-neighbor of the root node.

3.2 Attention Module-based Spatial Temporal Graph Convolution Network

In the spatial temporal graph convolution model, the receptive field of the convolution operation is the 1-neighbor of the root node, so it only captures local features. However, in different sample of different action classes, the relationship between the joints is not limited to the 1-

neighbor of the joint. For example, for many actions such as combing hair, brushing teeth, the relationship between the hand and the head may be important. In order to solve this problem, we introduce the idea of non-local neural network[27], make some improvements to the ST-GCN model, and then propose AM-STGCN skeleton-based action recognition method based on the non-local attention mechanism, which directly focuses on the features of all joints, and get more efficient features by attention operations.

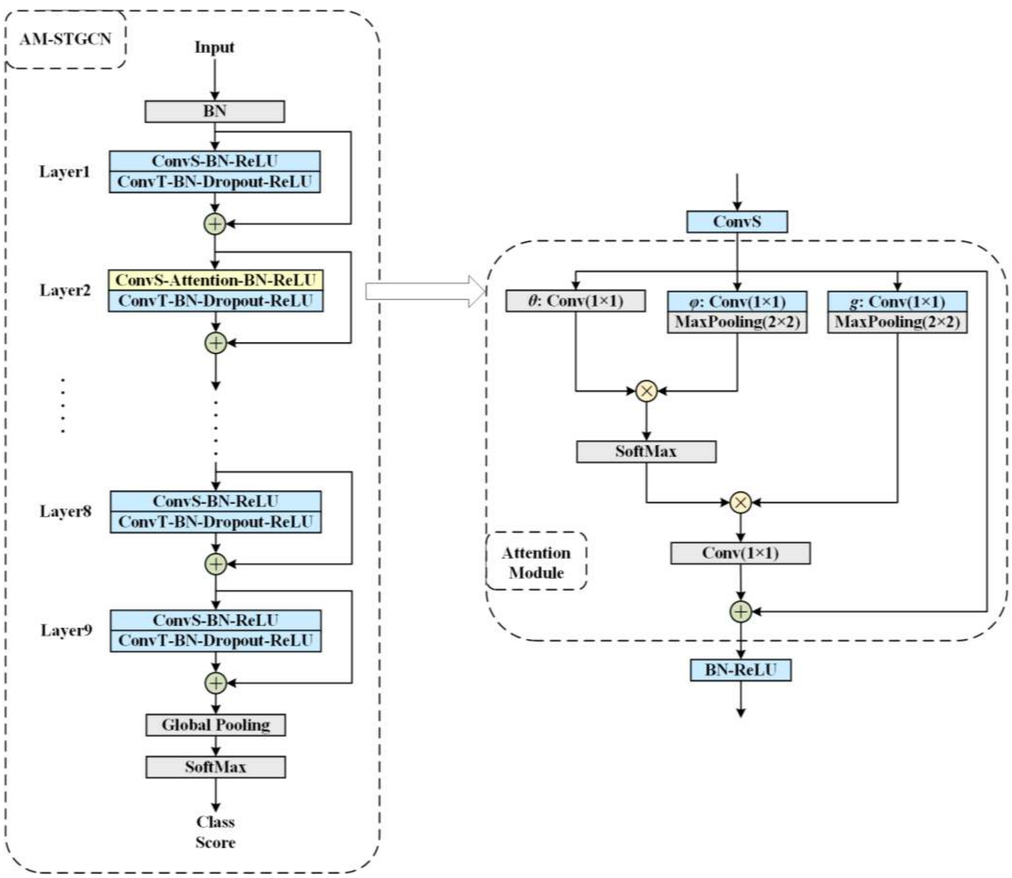


Fig. 2 The structure of AM-STGCN.

Fig. 2 shows the network structure of AM-STGCN, where we add the attention module after the spatial convolution operation (ConvS) of Layer2. The model consists of nine layers of spatial temporal graph convolution operators. The first three layers have 64 output channels, the middle three layers have 128 output channels, and the last three layers have 256 output channels. Each layer of AM-STGCN includes the spatial convolution operation (ConvS) and the temporal convolution operation (ConvT). The residual connection[33] is added on each layer.

Non-local neural network is a versatile, flexible building block, it can be easily embedded into existing 2D and 3D convolutional networks to improve or visualize related CV tasks. This allows us to combine global and local information to build richer hierarchy. In Fig. 2, the right side is our attention module, which is used to capture the correlation between all joints. We construct the attention module mainly following the idea of non-local neural network: first, linear mapping is conducted on the feature map of ConvS , which is implemented as 1×1 convolution, and then get the θ, ϕ, g features; second, we perform a matrix point multiplication operation on θ and ϕ to calculate the autocorrelation in the feature, and then carry out Softmax operation to obtain the self-attention coefficient; third, the attention coefficient is multiplied back into the feature matrix g; at last, residual connection is established with the original input feature map, and then we get a new set of features. Specifically, we add 2×2 MaxPooling operation after θ, ϕ features to reduce computational cost. Such an attention module is called one attention block, and multiple attention blocks will be used in the work. How many attention blocks are added to the model and where they are added will be analyzed in detail in Sec. 4, and the experimental results are given at the same time.

4 Experiments and Analysis

In this section, we evaluate the performance of the AM-STGCN model. In order to compare with the baseline model ST-GCN, our experiments are performed on the same two large-scale action recognition datasets: the human action dataset Kinetics[31] is the largest unconstrained action recognition dataset up to now, and NTU-RGB+D[32] is the largest constrained indoor captured action recognition dataset. First, we conduct a detailed ablation study of the Kinetics dataset to analyze the contribution of the proposed model to recognition performance. Then, the corresponding experiments are carried out on the NTU-RGB+D dataset to verify whether the proposed model has certain generalization ability. Finally, we compare AM-STGCN with ST-GCN and some state-of-the-art results of skeleton-based action recognition on Kinetics and NTU-RGB+D. All experiments were performed on PyTorch deep learning framework using two 1080Ti GPUs.

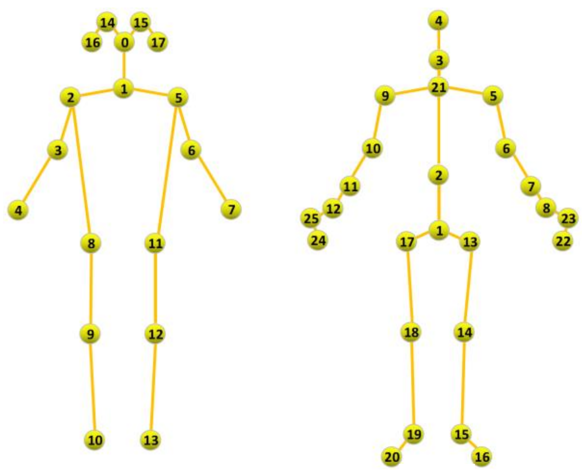
4.1 Datasets

Kinetics[31]: Kinetics is a large human action dataset that contains 400 action classes taken from different YouTube video, each class with at least 400 video clips, each clip lasts about 10 seconds[31]. These actions include the interaction between people and objects, such as playing an instrument, and the interaction between people, such as shaking hands.

The Kinetics dataset only provides raw video clips and does not provide skeleton joint data. As shown in Ref. 3, they use the public available OpenPose[34] toolbox to estimate the location of 18 joints on every frame of the clips. In this work, we use the Kinetics-skeleton dataset provided by the author of ST-GCN, which marks the position of 18 joints in each frame. The dataset provides a training set of 240,000 clips and a validation set of 20,000 clips. In accordance with the recommendations in Ref. 31, in this work, we train the model on the training set and report the top-1 and top-5 recognition accuracies on the validation set.

Fig. 3(a) shows the joint label of the Kinetics-skeleton dataset. The joint labels are: 0 nose, 1 neck, 2 right shoulder, 3 right elbow, 4 right wrist, 5 left shoulder, 6 left elbow, 7 left wrist, 8 right hip, 9 right knee, 10 right ankle, 11 left hip, 12 left knee, 13 left ankle, 14 right eye, 15 left eye, 16 right ear, 17 left ear.

NTU-RGB+D[32]: NTU-RGB+D is the largest dataset with 3D joint annotations currently used for human action recognition tasks. The dataset contains 60 action classes with a total of 56,000 action clips. All of these clips are performed by 40 volunteers in a constrained lab environment, and captured by 3 cameras of the same height but from different horizontal angles: -45°, 0°, 45°[32]. The dataset provides the 3D joint location of each frame detected by the Kinect depth sensor. There are 25 joints per subject in the skeleton sequence. Each clip is guaranteed to have a maximum of 2 subjects.



(a) (b)

Fig. 3 The joint label of Kinetics-skeleton and NTU-RGB+D datasets.

The original paper of the NTU-RGB+D dataset recommended two benchmarks: 1) cross-subject (X-Sub) benchmark: The dataset in this benchmark is divided into a training set (40,320 clips) and a validation set (16,560 clips). The subjects in these two subsets are different; 2) cross-view (X-View) benchmark: The training set in this benchmark contains 37,920 clips captured by cameras 2 and 3, and the validation set contains 18,960 clips captured by camera 1[32]. We follow this convention and report the top-1 recognition accuracy of the two benchmarks.

Fig. 3(b) shows the joint label of the NTU-RGB+D dataset. The joint labels are: 1 base of the spine, 2 middle of the spine, 3 neck, 4 head, 5 left shoulder, 6 left elbow, 7 left wrist, 8 left hand, 9 right shoulder, 10 right elbow , 11 right wrist, 12 right hand, 13 left hip, 14 left knee, 15 left ankle, 16 left foot, 17 right hip, 18 right knee, 19 right ankle, 20 right foot, 21 spine, 22 left hand tip, 23 left hand Thumb, 24 right hand tip, 25 right thumb.

4.2 Effectiveness Analysis of AM-STGCN

In this section, we first conduct a lot of ablation experiments on the Kinetics-skeleton dataset: 1) Adding attention block after the ConvS (spatial convolution) of different layers of the ST-GCN; 2) Adding multiple attention blocks after the ConvS of different layers; 3) Adding attention blosks after ConvT (temporal convolution) of the layer; 4) Adding two other attention mechanisms with different structures, CBAM[29] and SENet[30], to ST-GCN. Experiments are then performed on NTU-RGB+D dataset to verify the generalization capabilities of the proposed model AM-STGCN.

4.2.1 Baseline

In order to evaluate the recognition performance of our improved model, we used baseline for comparison experiments. Since our model is improved on the basis of the ST-GCN model, we use the ST-GCN model as a baseline to analyze the advantages of AM-STGCN. We reproduced the ST-GCN model on the Kinetics dataset based on the Ref. 3, and obtained very close results to the original paper (see Table 1).

Table 1 Baseline.

Method	Top-1(%)	Top-5(%)
ST-GCN[3]	30.7	52.8
Our ST-GCN Baseline	30.7	53.7

4.2.2 Ablation experiment

Table 2 The results of adding one attention block to the different layers of the ST-GCN. ST-GCN1's ConvS + 1 represents adding one attention block after the ConvS (spatial convolution) of the first layer of the ST-GCN. Thereafter, Tables 3, 4, 5, and 6 have the same representation rules.

Method	Top-1(%)	Top-5(%)
Our ST-GCN Baseline	30.7	53.7
ST-GCN$_1$'s ConvS + 1	31.6	54.3
ST-GCN$_2$'s ConvS + 1	**31.9**	**54.7**
ST-GCN$_3$'s ConvS + 1	**31.9**	**54.7**
ST-GCN$_4$'s ConvS + 1	31.3	53.8
ST-GCN$_9$'s ConvS + 1	31.0	53.7

Table 2 shows the experimental results of adding one attention block after the ConvS (spatial convolution) of different layers of the ST-GCN model. The results demonstrate that no matter which layer we add an attention block to, the recognition accuracy always higher than the baseline. The improvement of adding one attention block in the second and third layers is similar, which can lead to ~1.2% (on Top1) improvement over the baseline. The results of the remaining layers are slightly lower.

Table 3 The results of adding multiple attention blocks to different layers.

Method	Top-1(%)	Top-5(%)
Our ST-GCN Baseline	30.7	53.7
ST-GCN$_1$'s ConvS + 2	32.0	54.5
ST-GCN$_2$'s ConvS + 2	32.1	54.4
ST-GCN$_3$'s ConvS + 2	31.4	54.4
ST-GCN$_1$'s ConvS + 3	30.6	53.1

ST-GCN$_2$'s ConvS + 3	31.1	53.5
ST-GCN$_3$'s ConvS + 3	**32.2**	**55.1**
ST-GCN$_4$'s ConvS + 3	31.1	53.1

Table 3 shows the results of adding multiple attention blocks to different layers of the ST-GCN. It can be seen from Table 2 that adding one attention block to the first few layers of the model is better than adding to the lower layer, so in the experiment of Table 3, we add two and three attention blocks after the ConvS (spatial convolution) of the first few layers of ST-GCN. Obviously, the results of adding multiple attention blocks after ConvS of a layer outperform adding a single attention block, especially on ST-GCN$_3$'s ConvS + 3, which can lead to 1.5% (on Top1) and 1.4% (on Top5) improvement over the baseline. It demonstrates that more attention blocks usually lead to better performance. We argue that multiple attention blocks can reinforce the correlation information learned in the previous attention block, thus assigning each node a more appropriate weight.

Table 4 The results of adding multiple attention blocks to multi-layers.

Method	Top-1(%)	Top-5(%)
Our ST-GCN Baseline	30.7	53.7
ST-GCN$_2$'s ConvS + 1 ST-GCN$_3$'s ConvS + 1	31.4	54.1
ST-GCN$_1$'s ConvS + 2 ST-GCN$_2$'s ConvS + 2	30.9	53.3
ST-GCN$_2$'s ConvS + 2 ST-GCN$_3$'s ConvS + 2	**32.3**	**55.1**
ST-GCN$_1$'s ConvS + 2 ST-GCN$_3$'s ConvS + 2	31.5	54.2

Table 4 shows the results of adding multiple attention blocks to multi-layers of the ST-GCN model. As shown in Tables 2, 3 and 4, we can find that only the third combination ($ST\text{-}GCN_2$'s ConvS + 2 & $ST\text{-}GCN_3$'s ConvS + 2) improves accuracy compared to adding attention blocks to single layer. The rest of the combinations do not improve accuracy compared to the individual structure in the combination.

Table 5 The results of adding attention blocks after ConvT (temporal convolution) of one layer.

Method	Top-1(%)	Top-5(%)
Our ST-GCN Baseline	30.7	53.7
$ST\text{-}GCN_2$'s ConvT + 2	32.0	54.9
$ST\text{-}GCN_3$'s ConvT + 3	**32.9**	**55.4**
$ST\text{-}GCN_5$'s ConvT + 3	31.7	54.3

Table 5 shows the results of adding attention blocks after ConvT (temporal convolution) of different layers of the ST-GCN model. Comparing the results of Table 3 and Table 5, we can find that adding attention blocks after ConvT perform better than after ConvS. $ST\text{-}GCN_3$'s ConvT + 3 obtain the best improvement of adding attention blocks after ConvT, which outperforms Our ST-GCN Baseline by 2.2% and 1.7% on Top1 and Top5 recognition accuracies; $ST\text{-}GCN_3$'s ConvS + 3 obtain the best improvement of adding attention blocks after ConvS, which outperforms Our ST-GCN Baseline by 1.5% and 1.4% on Top1 and Top5 recognition accuracies. One possible explanation is that ConvT has a bigger kernel size (9×1) and ConvS has a small kernel size (1×1), thus ConvS is insufficient to capture precise spatial information. Adding attention blocks after ConvT can learn the correlation of all nodes in all frames, while adding attention blocks after ConvS can only learn the correlation of all nodes in one frame, thus adding attention blocks after ConvT perform better than after ConvS.

Table 6 The results of adding attention blocks after ConvT and ConvS of multi-layers.

Method	Top-1(%)	Top-5(%)
Our ST-GCN Baseline	30.7	53.7
ST-GCN$_2$'s ConvT + 2 ST-GCN$_3$'s ConvT + 3	**32.3**	**54.4**
ST-GCN$_2$'s ConvS + 1 ST-GCN$_2$'s ConvT + 2	31.5	53.8
ST-GCN$_2$'s ConvS + 2 ST-GCN$_3$'s ConvT + 3	31.8	54.0

Table 6 shows the results of adding attention blocks after ConvT and ConvS of multi-layers. As shown in Tables 2, 3, 5 and 6, we can see that none of the combinations in Table 6 improves accuracy compared to adding attention blocks to single layer. The results of Table 4 and 6 prove that adding attention blocks to multiple layers does not further improve accuracy.

From Tables 2, 3, 4, 5 and 6, we find that adding attention blocks to the second and third layer of ST-GCN can result in better performance. The possible reason is that the features learned in these two layers are more consistent with the semantic representation of human motion.

Table 7 The results of adding CBAM and SENet to ST-GCN.

Method	Top-1(%)	Top-5(%)
ST-GCN+CBAM	31.9	54.3
ST-GCN+SENet	31.6	54.2
Our AM-STGCN	**32.9**	**55.4**

We selected two other attention mechanisms with different structures, CBAM[29] and SENet[30], to be added to ST-GCN. CBAM contains spatial attention and channel attention, while SENet is just channel attention. Table 7 shows the results of adding CBAM and SENet. As shown in

Table7, the results of our method are clearly better than those of the other two attention structures, which prove that our attention mechanism is more suitable for ST-GCN.

4.2.3 Further analysis on"Kinetics-Motion"

The authors of ST-GCN select a subset of 30 classes strongly related with body motions, named as "Kinetics-Motion[3]". For a detailed comparison, we further investigate the per-class differences in accuracy on this subset. In Fig. 4, the horizontal axis is the action category of "Kinetics-Motion", and the vertical axis is the accuracy of per-class. The dark blue represents Our ST-GCN Baseline and the light blue represents AM-STGCN, here AM-STGCN is the optimal structure (i.e., ST-GCN$_3$'s ConvT + 3) obtained after the analysis in the previous section. It can be observed obviously that the accuracy of most actions get improved. Some classes even get more than 10% improvement, such as hitting baseball, hopscotch, salsa dancing and squat. These results also verify the superiority of our model for skeleton-based action recognition, in particular on those classes strongly related with body motions.

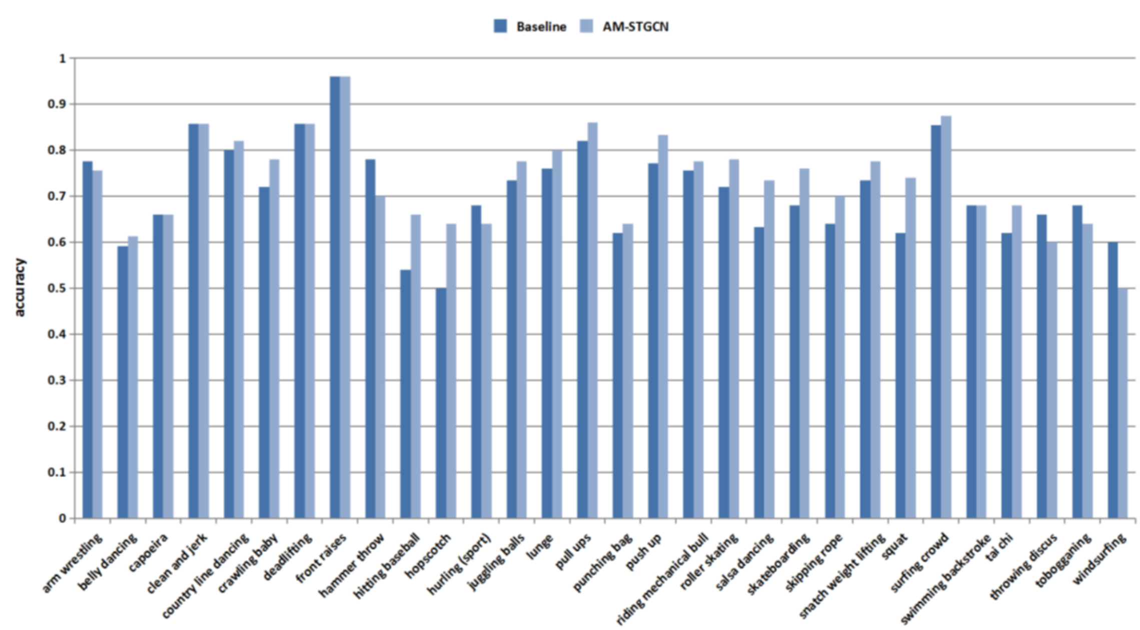


Fig. 4 Category accuracies on the "Kinetics Motion" subset of the Kinetics dataset.

4.2.4 Time comparison on Kinetics

The Kinetics dataset provides a training set of 240,000 video clips, each clip contain 300 frames. Every frame of the video clips is converted into a sequence of human skeletons represented by coordinates through OpenPose[34] toolbox. We compared the training time of one epoch of AM-STGCN model and our ST-GCN baseline on Kinetics dataset, and the results are shown in Table 8. ST-GCN_3's ConvS + 3 and ST-GCN_3's ConvT + 3, which performed better in the above experiments, are selected to be compared with our ST-GCN baseline. The training time of ST-GCN_3's ConvS + 3 and our ST-GCN baseline are similar, and ST-GCN_3's ConvT adds the calculation in temporal dimension, so the training time is a little longer. These results demonstrate that our AM-STGCN model do not add much time cost than ST-GCN model.

Table 8 The training time of AM-STGCN and ST-GCN methods.

Method	The number of skeleton sequence.	Training time of one epoch. (h)
Our ST-GCN Baseline	240,000	0.58
ST-GCN_3's ConvS + 3	240,000	0.61
ST-GCN_3's ConvT + 3	240,000	0.70

4.2.5 Comparison with state-of-the-art methods

On Kinetics dataset, we compare AM-STGCN with "Feature Encoding"[10], Deep LSTM[32], Temporal ConvNet[14] and ST-GCN[3] methods. Their recognition performance in terms of Top-1 and Top-5 accuracies are listed in Table 9. Obviously, our AM-STGCN with using attention module outperforms ST-GCN by 2.2% and 2.6% on Top1 and Top5 recognition accuracies respectively. It can be seen from Table 9 that our AM-STGCN is able to outperform previous representative methods.

Table 9 Comparison with the state-of-the-art on Kinetics dataset.

Method	Date	Top-1(%)	Top-5(%)
Feature Encoding.[10]	2015	14.9	25.8
Deep LSTM[32]	2016	16.4	35.3
Temporal ConvNet[14]	2017	20.3	40.0
ST-GCN[3]	2018	30.7	52.8
Our ST-GCN Baseline	-	30.7	53.7
Our AM-STGCN	-	**32.9**	**55.4**

We found that most of the current skeleton-based action recognition studies are conducted on NTU-RGB+D dataset, so we compare our method with state-of-the-art methods on NTU-RGB+D dataset.

On NTU-RGB+D dataset, we compare AM-STGCN with Lie Group[9], H-RNN[11], Deep LSTM[32], VA-LSTM[13], Temporal ConvNet[14], Two-stream CNN[16], HCN[17], STA-LSTM[12], GCA-LSTM[18], ARRN-LSTM[20], MANs[19], ST-GCN[3], DPRL+GCNN[26], SR-TSL[22], PB-GCN[24] and AGCN[23] methods. The results are shown in Table 10.

Comparisons with hand-craft feature based methods, CNN based methods and RNN based methods. Table 10 shows that the performance of graph convolution based methods is generally better than hand-craft feature based methods, CNN based methods and RNN based methods. In particular, our AM-STGCN obtains very close results to HCN method on cross-view (X-View) benchmark, which performs best among CNN based methods. At the same time, multi-person feature fusion is added in HCN, thus resulting in better performance on cross-subject (X-Sub) benchmark, but it also leads to the increase of computation.

Comparisons with other methods based on attention. We compare AM-STGCN with other methods based on attention including STA-LSTM[12], GCA-LSTM[18], ARRN-LSTM[20] and

MANs[19]. From Table 10, we can see that our AM-STGCN is better than any other result except for MANs under the X-View benchmark. MANs consists of Temporal Attention Recalibration Module (TARM) and DenseNet-161, we can find that their baseline is higher than ST-GCN, which may be due to DenseNet-161, because DenseNet-161 is much deeper and more complex than ST-GCN. On X-View benchmark, our AM-STGCN outperforms ST-GCN by 3.1% and MANs outperforms MANs (no attention) by 1.07%, which prove that our method can improve the performance of the model more.

Comparisons with graph convolution based methods. 1) Single stream network. In Table 10，we can see clearly that our AM-STGCN with using attention module outperforms ST-GCN by 1.9% and 3.1% on cross-view (X-View) benchmark and cross-subject (X-Sub) benchmark respectively, which prove that our AM-STGCN model is equally effectiveness on NTU-RGB+D dataset. Our AM-STGCN performs very close results to DPRL+GCNN on cross-subject (X-Sub) benchmark and outperforms DPRL+GCNN by 1.6% on cross-view (X-View) benchmark in Table 10. 2) Two-stream networks. The joint locations is the only input data of our AM-STGCN. SR-TSL, PB-GCN and AGCN all have another form of input data as input to different streams, thus forming a two-stream networks. SR-TSL(Position), PB-GCN(*Jloc*) and Js-AGCN are the same as ST-GCN with only joint locations as input data. Among these methods, it can be seen obviously form Table 10 that our AM-STGCN is superior to SR-TSL(Position) and PB-GCN(*Jloc*) on both cross-subject (X-Sub) and cross-view (X-View) benchmark. In the paper of AGCN, we find AGCN's baseline is 92.7% on cross-view (X-View) benchmark, outperforms ST-GCN by 4.4%, but Js-AGCN outperforms their baseline by only 1%. We think it may be that different experimental environments cause different baselines. So in terms of relative increase in accuracy, our method has achieved a good performance improvement. In addition, we have added our attention module to Js-AGCN. In Table 10, the results of Js-AGCN+our attention outperforms Our Js-AGCN Baseline by 0.5% and 0.4% on cross-view (X-View) benchmark and cross-subject (X-Sub) benchmark respectively, which shows that our attention mechanism is also effective on AGCN method, and proves that our method has certain robustness.

These results show our AM-STGCN model achieves a significant performance improvement.

Table 10 Comparison with the state-of-the-art on NTU-RGB+D dataset.

Method	Date	X-Sub(%)	X-View(%)
Lie Group[9]	2014	50.1	52.8
H-RNN[11]	2015	59.1	64.0
Deep LSTM[32]	2016	60.7	67.3
Temporal ConvNet[14]	2017	74.3	83.1
VA-LSTM[13]	2017	79.4	87.6
Two-stream CNN[16]	2017	83.2	89.3
HCN[17]	2018	86.5	91.1
STA-LSTM[12]	2017	73.4	81.2
GCA-LSTM[18]	2017	74.4	82.8
ARRN-LSTM[20]	2019.04	81.8	89.6
MANs (no attention)[19]	2018	81.41	92.15
MANs[19]		83.01	93.22
ST-GCN[3]	2018	81.5	88.3
DPRL+GCNN[26]	2018	83.5	89.8
SR-TSL(Position)[22]	2018	78.8	88.2
SR-TSL(Velocity)[22]		82.2	90.6
SR-TSL[22]		84.8	92.4

PB-GCN(J_{loc})[24]	2018	82.8	90.3
PB-GCN($D_R \| D_T$)[24]		87.5	93.2
Js-AGCN[23]	2019.05	-	93.7
Bs-AGCN[23]		-	93.2
2s-AGCN[23]		88.5	95.1
Our Js-AGCN Baseline	-	85.9	93.7
Js-AGCN + our attention	-	86.4	94.1
Our AM-STGCN	-	**83.4**	**91.4**

5 Conclusion

In this paper, we propose a new skeleton-based action recognition method called attention module-based Spatial Temporal Graph Convolutional Networks(AM-STGCN), which can overcome the weakness of ST-GCN model. In order to capture global information of skeleton sequences, attention modules are added to learn the correlation information between all joints of both spatial and temporal dimension. So AM-STGCN can extract long-range relationships from input skeleton sequences, which improve the ability to model the dynamic change of human body motions. Experiments on two large-scale action recognition datasets Kinetics and NTU-RGB+D achieve the better results, which indicate that AM-STGCN can effectively improve the recognition accuracy. In future, we will improve our AM-STGCN in many possible directions, such as improving attention modules or merging RGB modality.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 61871182, 61302163), Hebei Province Natural Science Foundation (Grant Nos.F2015502062), Hebei province science and technology support (Grant Nos.13210905).

References

1. H. Wang and C. Schmid, "Action recognition with improved trajectories," in *2013 IEEE International Conference on Computer Vision*, pp. 3551-3558, IEEE, Sydney, NSW (2013) [doi:10.1109/ICCV.2013.441].
2. H. Wang et al., "Dense trajectories and motion boundary descriptors for action recognition," in *International journal of computer vision*, **103**(1), 60-79 (2013).
3. S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 7444-7452, AAAI Press, New Orleans, Louisiana, USA (2018).
4. K. Simonyan, A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, pp. 568-576 (2014).
5. C. Feichtenhofer, A. Pinz and A. Zisserman, "Convolutional two-Stream network fusion for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1933-1941, IEEE, Las Vegas, NV (2016) [doi:10.1109/CVPR.2016.213].
6. D. Tran et al., "Learning spatiotemporal features with 3D convolutional networks,"in *2015 IEEE International Conference on Computer Vision*, pp. 4489-4497, IEEE, Santiago (2015) [doi:10.1109/ICCV.2015.510].

7. J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724-4733, IEEE, Honolulu, HI (2017) [doi:10.1109/CVPR.2017.502].
8. W. Du, Y. Wang and Y. Qiao, “RPAN: An end-to-end recurrent pose-attention network for action recognition in videos,” in *2017 IEEE International Conference on Computer Vision*, pp. 3745-3754, IEEE, Venice (2017) [doi:10.1109/ICCV.2017.402].
9. R. Vemulapalli, F. Arrate, and R. Chellappa, “Human action recognition by representing 3d skeletons as points in a lie group,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 588-595, IEEE, Columbus, OH (2014) [doi:10.1109/CVPR.2014.82].
10. B. Fernando et al., “Modeling video evolution for action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5378-5387, IEEE, Boston, MA (2015) [doi:10.1109/CVPR.2015.7299176].
11. Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1110-1118, IEEE, Boston, MA (2015) [doi:10.1109/CVPR.2015.7298714].
12. S. Song et al., “An end-to-end spatio-temporal attention model for human action recognition from skeleton data,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 4263-4270, AAAI Press, San Francisco, California, USA (2017).
13. P. Zhang et al., “View adaptive recurrent neural networks for high performance human action recognition from skeleton data,” in *2017 IEEE International Conference on Computer Vision*, pp. 2136-2145, IEEE, Venice (2017) [doi:10.1109/ICCV.2017.233].
14. T. S. Kim and A. Reiter, “Interpretable 3d human action analysis with temporal convolutional networks,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1623-1631, IEEE, Honolulu, HI (2017) [doi:10.1109/CVPRW.2017.207].
15. M. Liu, H. Liu, and C. Chen, “Enhanced skeleton visualization for view invariant human action recognition,” in *Pattern Recognition*, **68**, pp. 346-362, Elsevier (2017).

16. C. Li et al., “Skeleton-based action recognition with convolutional neural networks,” in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 597-600, IEEE, Hong Kong (2017) [doi:10.1109/ICMEW.2017.8026285].

17. C. Li et al., “Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation,” in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 786-792, AAAI Press, Stockholm, Sweden (2018).

18. J. Liu et al., “Global Context-Aware Attention LSTM Networks for 3D Action Recognition,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3671-3680, IEEE, Honolulu, HI (2017) [doi: 10.1109/CVPR.2017.391].

19. C. Xie et al., “Memory attention networks for skeleton-based action recognition,” in *Proceedings of the 27th International Joint Conference on Artificial Intelligence,* pp. 1639-1645, AAAI Press, Stockholm, Sweden (2018).

20. W. Zheng et al., “Relational Network for Skeleton-Based Action Recognition,” arXiv preprint arXiv:1805.02556v4, 2019.

21. C. Li et al., “ Spatio-temporal graph convolution for skeleton based action recognition,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 3482-3489, AAAI Press, New Orleans, Louisiana, USA (2018).

22. C. Si et al., “Skeleton-based action recognition with spatial reasoning and temporal stack learning,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, vol **11205**, pp. 106-121, Springer, Cham (2018) [https://doi.org/10.1007/978-3-030-01246-5_7].

23. L. Shi et al., “Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp.12026-12035 (2019).

24. K. Thakkar, P J. Narayanan, “Part-based Graph Convolutional Network for Action Recognition,” arXiv preprint arXiv:1809.04983, 2018.

25. X. Gao et al., "Optimized Skeleton-based Action Recognition via Sparsified Graph Regression," arXiv preprint arXiv:1811.12013v2, 2019.

26. Y. Tang et al., "Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5323-5332, IEEE, Salt Lake City, UT (2018) [doi: 10.1109/CVPR.2018.00558].

27. X. Wang et al., "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794-7803, IEEE, Salt Lake City, UT, USA (2018) [doi:10.1109/CVPR.2018.00813].

28. Y. Du et al., "Interaction-Aware Spatio-Temporal Pyramid Attention Networks for Action Classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, vol **11220**, pp. 388-404, Springer, Cham (2018) [https://doi.org/10.1007/978-3-030-01270-0_23].

29. S. Woo et al., "CBAM: Convolutional Block Attention Module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, vol **11211**, pp. 3-19, Springer, Cham (2018) [https://doi.org/10.1007/978-3-030-01234-2_1].

30. J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132-7141, IEEE, Salt Lake City, UT (2018) [doi:10.1109/CVPR.2018.00745].

31. W. Kay et al., "The kinetics human action video dataset," arXiv preprint arXiv:1705.06950, 2017.

32. A. Shahroudy et al., "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1010-1019, IEEE, Las Vegas, NV (2016) [doi:10.1109/CVPR.2016.115].

33. K. He et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, IEEE, Las Vegas, NV (2016) [doi:10.1109/CVPR.2016.90].

34. Z. Cao et al., "Realtime multi-person 2D pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1302-1310, IEEE, Honolulu, HI (2017) [doi:10.1109/CVPR.2017.143].

Yinghui Kong is a professor at North China Electric Power University, Baoding, China. She received BS degree in Communication Engineering from Xidian University, Xian, China, in 1987, and MS degree in Power System & Its Automation from North China Electric Power University, Beijing, China, in 1993, and PhD degree in Electrical Theory and New Techniques from North China Electric Power University in 2009. Her research interests include deep learning, computer vision, behavior recognition, expression recognition.

Caption List

Fig. 1 (a) Spatial-temporal graph of the skeleton and (b) Partitioning strategy, different colors represent different subsets.

Fig. 2 The structure of AM-STGCN.

Fig. 3 The joint label of Kinetics-skeleton and NTU-RGB+D datasets.

Fig. 4 Category accuracies on the "Kinetics Motion" subset of the Kinetics dataset.

Table 1 Baseline.

Table 2 The results of adding a attention block to the different layers of the ST-GCN. ST-GCN1's ConvS + 1 represents adding one attention block after the ConvS of the first layer of the ST-GCN. Thereafter, Tables 3, 4, 5, and 6 have the same representation rules.

Table 3 The results of adding multiple attention blocks to different layers.

Table 4 The results of adding multiple attention blocks to multi-layer.

Table 5 The results of adding attention blocks after ConvT of one layer.

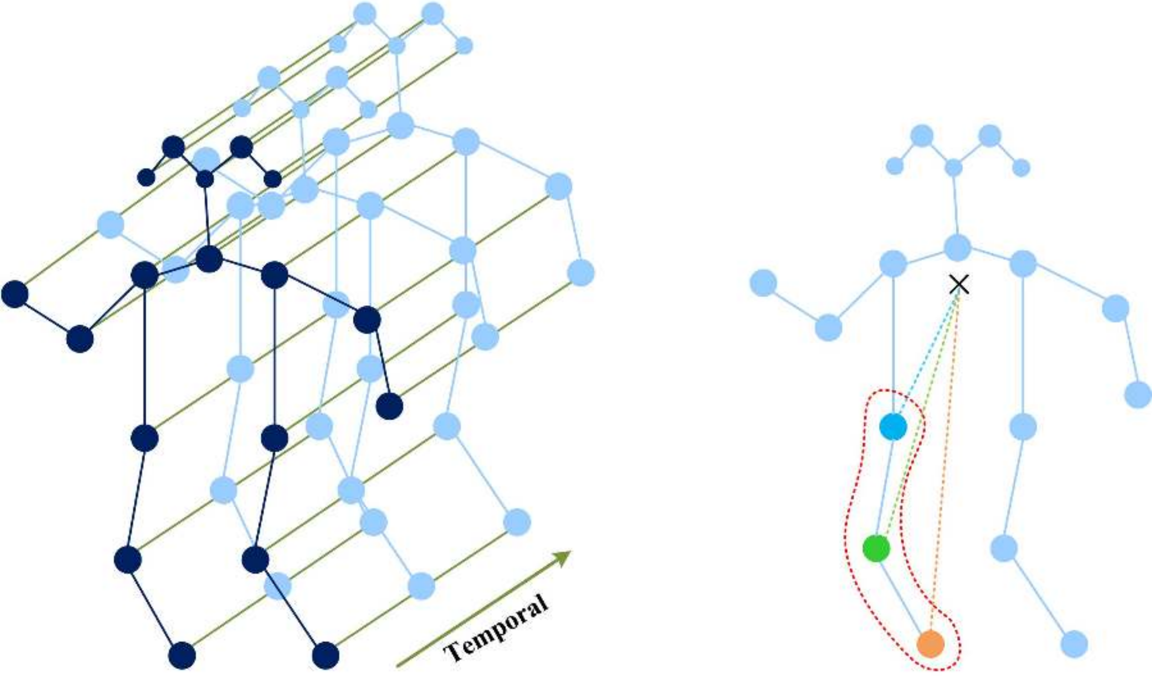
Table 6 The results of adding attention blocks after ConvT and ConvS of multi-layer.

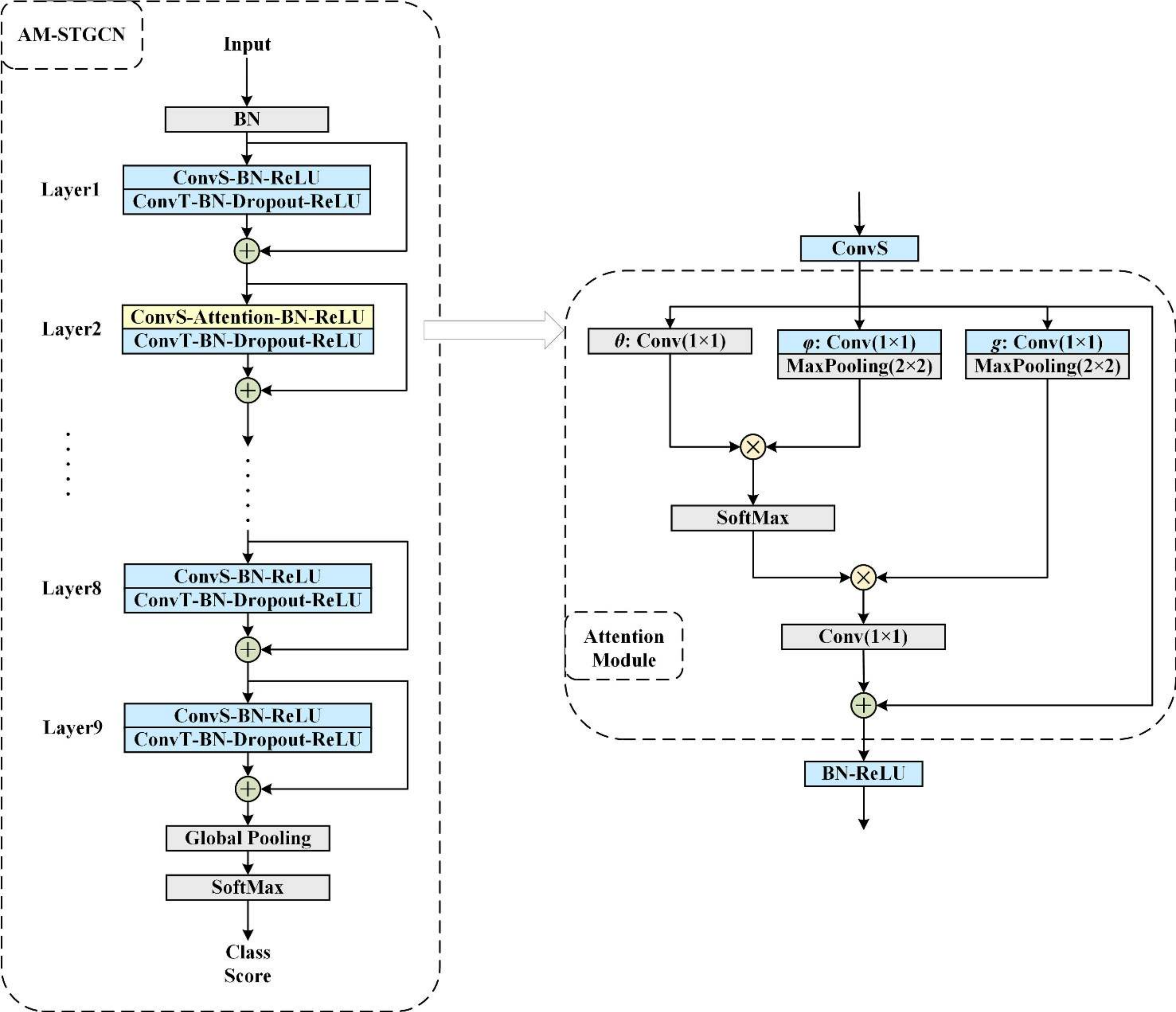
Table 7 The results of adding CBAM and SENet to ST-GCN.

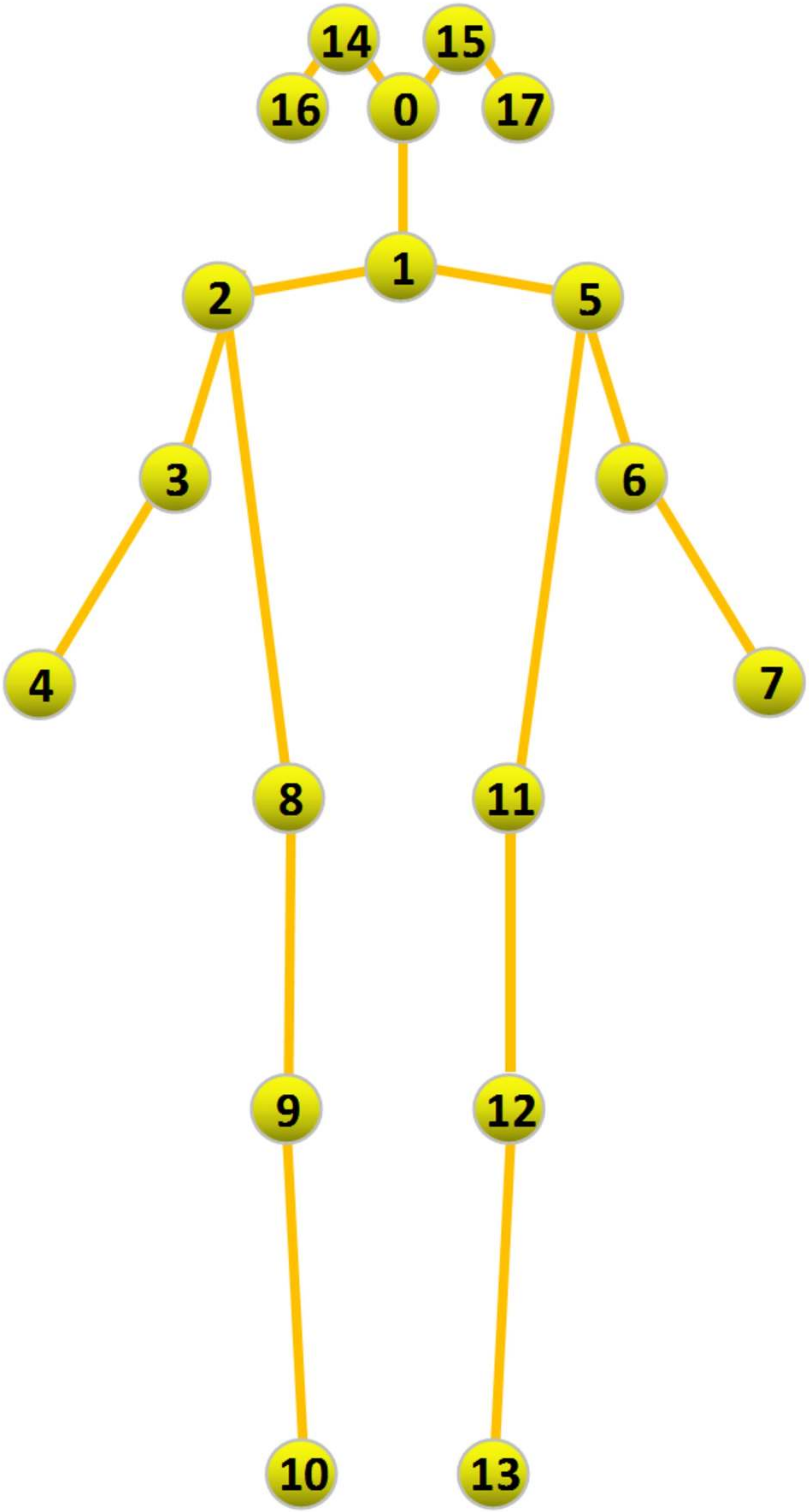
Table 8 The training time of AM-STGCN and STGCN methods.

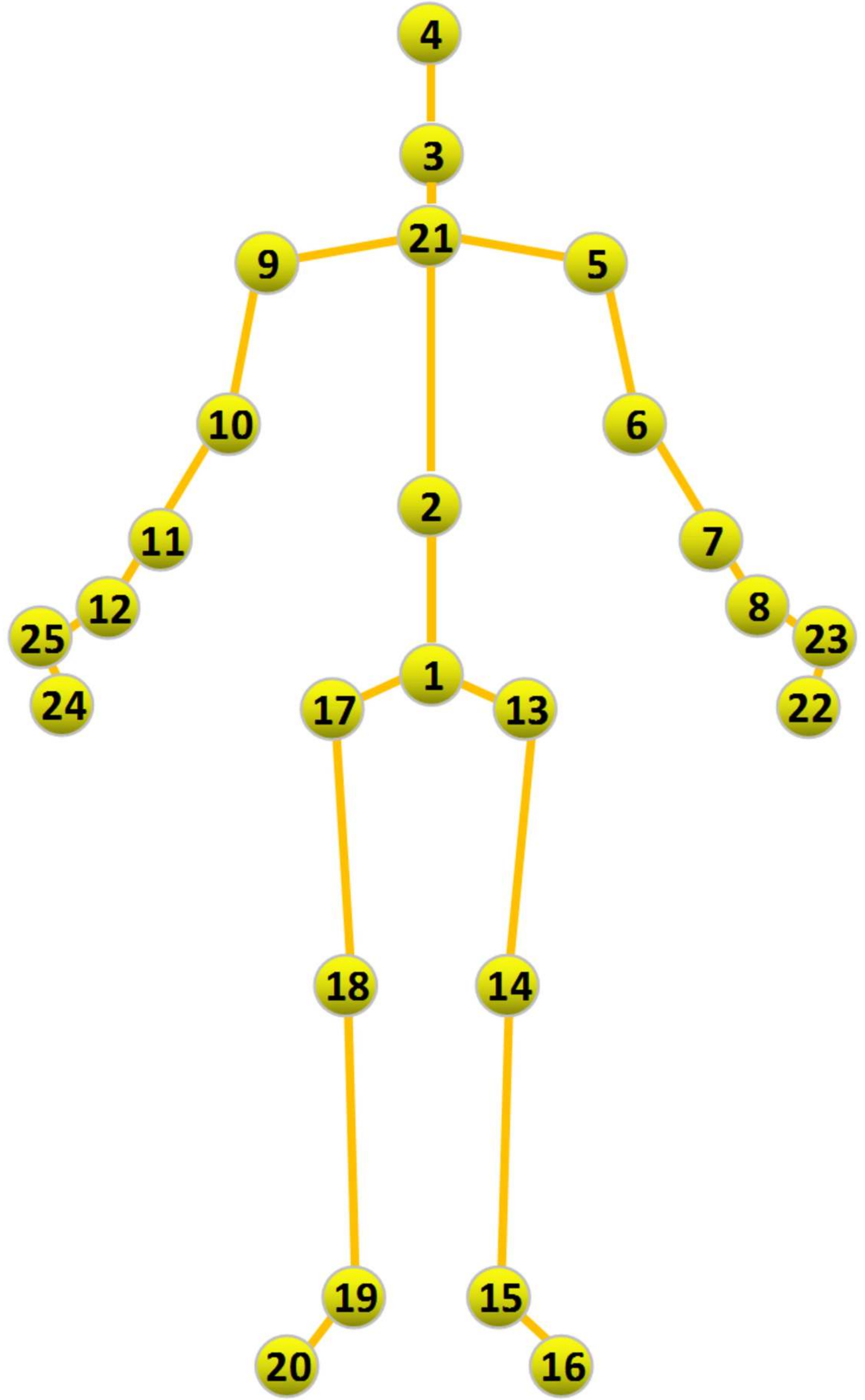
Table 9 Comparison with the state-of-the-art on Kinetics dataset.

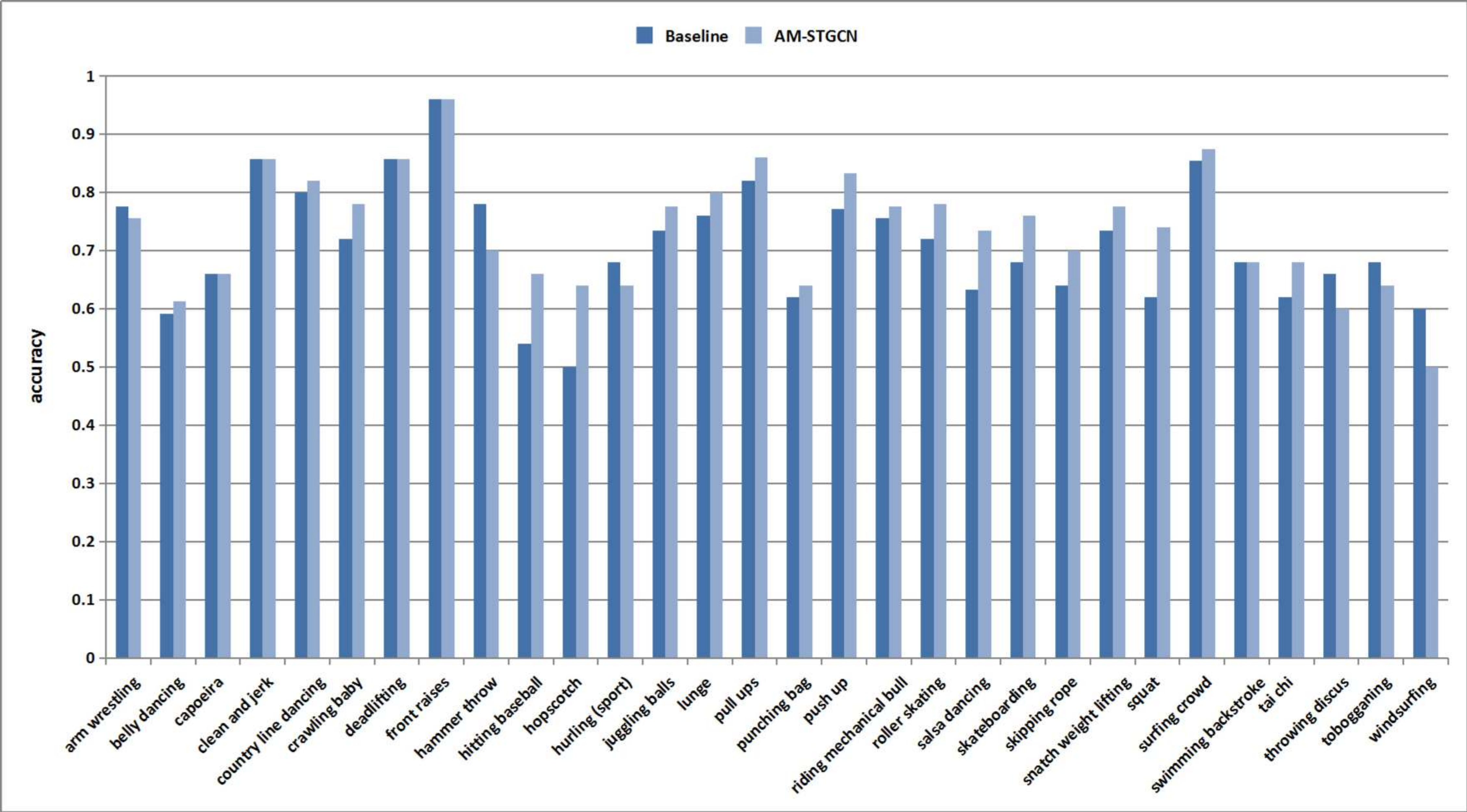
Table 10 Comparison with the state-of-the-art on NTU-RGB+D dataset.


Temporal


AM-STGCN
Input
BN
Layer1
ConvS-BN-ReLU
ConvT-BN-Dropout-ReLU
Layer2
ConvS-Attention-BN-ReLU
ConvT-BN-Dropout-ReLU
Layer8
ConvS-BN-ReLU
ConvT-BN-Dropout-ReLU
Layer9
ConvS-BN-ReLU
ConvT-BN-Dropout-ReLU
Global Pooling
SoftMax
Class Score
ConvS
θ: Conv(1×1)
φ: Conv(1×1)
MaxPooling(2×2)
g: Conv(1×1)
MaxPooling(2×2)
SoftMax
Conv(1×1)
Attention Module
BN-ReLU


14
15
16
0
17
1
2
5
3
6
4
7
8
11
9
12
10
13


4
3
21
9
5
10
6
2
11
7
12
8
25
23
24
1
17
13
22
18
14
19
15
20
16



Baseline
AM-STGCN
accuracy
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
arm wrestling
belly dancing
capoeira
clean and jerk
country line dancing
crawling baby
deadlifting
front raises
hammer throw
hitting baseball
hopscotch
hurling (sport)
juggling balls
lunge
pull ups
punching bag
push up
riding mechanical bull
roller skating
salsa dancing
skateboarding
skipping rope
snatch weight lifting
squat
surfing crowd
swimming backstroke
tai chi
throwing discus
tobogganing
windsurfing