IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 11, 2021, accepted April 10, 2021, date of publication April 14, 2021, date of current version April 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073107

Multi-Scale Spatial Temporal Graph Neural Network for Skeleton-Based Action Recognition

DONG FENG[1,2,3], ZHONGCHENG WU[1,2], JUN ZHANG[1,3], AND TINGTING REN[1,3]

[1] High Magnetic Field Laboratory, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China
[2] School of Hefei Institutes of Physical Science, University of Science and Technology of China, Hefei 230031, China
[3] High Magnetic Field Laboratory of Anhui Province, Hefei 230031, China

Corresponding author: Tingting Ren (ttren@hmfl.ac.cn)

This work was supported in part by the Key Research and Development Program of Anhui Province, Research on Data Sharing Standardization of New Energy Vehicle under Grant 202004h07020031, in part by the Key Research and Development Project of Hefei Large Science Center, Chinese Academy of Sciences, Research on Management Methods and Key Technologies of Scientific Data for Large Scientific Equipment under Grant 2019HSC-KPRD003, and in part by the High Magnetic Field Laboratory of Anhui Province.

ABSTRACT Graph convolutional networks (GCNs) have achieved remarkable performance on skeleton-based action recognition. Existing GCN-based methods usually apply the fixed graph topology and one fixed temporal convolution kernel to extract the spatial features of joints and temporal features, which is from a single-scale perspective. Actually, human actions are coordinated by various body parts in the spatial domain, and exhibit different characteristics in the temporal domain. Therefore, it is appropriate to model the multi-scale information that can enhance both the explainability and stability, which is ignored in current literatures. To address this issue, we propose a multi-scale spatial-temporal graph neural network (MSTGNN) to discover multi-scale discriminative features from spatial and temporal aspects simultaneously. Our contributions are three-folds: 1) For the spatial domain, inspired by the kinematics of the human action, we develop a three-scale graph data structures in a fine-to-coarse way. A novel hybrid spatial pooling module is then proposed to dynamically exploit the global and comprehensive information step-by-step. 2) For the temporal domain, we design a multi-scale temporal convolution module adaptively fusing the temporal features extracted by different scale convolution kernels. 3) As utilizing one-stream architecture instead of multi-stream architecture, the proposed model can be trained in an end-to-end manner. MSTGNN achieves state-of-the-art performance with less computation complexity. Experimental results conducted on two large datasets (NTU-RGB+D and NTU-RGB+D-120) demonstrate the superiority of MSTGNN.

INDEX TERMS Skeleton-based action recognition, multi-scale, spatial-temporal network, graph convolutional network, adaptive fusion.

I. INTRODUCTION

Human action recognition attracts considerable attention due to their potential advantages for many applications in intelligent video surveillance, human-machine interaction and virtual reality [1]–[3]. With the continuous development of depth sensor technology [4] and pose estimation algorithms (e.g., Openpose [5], Alphapose [6]), the skeleton based action recognition methods have been widely studied in recent years. Compared with traditional RGB-based video action recognition, skeleton-based human action recognition can provide more detail position and movement information which is essential for action understanding. Moreover, human skeleton and joint trajectory are robust to backgrounds interference and sense changes.

Previous action recognition methods [7]–[9] usually rely on hand-crafted features which cannot effectively capture the discriminative spatial and temporal information from skeletons. Due to the rapid progress of deep learning, models based on convolutional neural networks (CNNs) [10]–[15] and recurrent neural networks (RNNs) [16]–[22] have become the mainstream, which normally regard the coordinates of human joints as pseudo-images or vector sequences. Although these methods have made great

The associate editor coordinating the review of this manuscript and approving it for publication was Khin Wee Lai.

progress, they are only suitable for dealing with the regular data in Euclidean space, and are not suitable for exploring the crucial spatial correlations among joints for action recognition. The human skeleton is naturally structured as a graph with the characteristic that the joints as vertexes and the bones in the human body as edges. Recently, graph convolution networks (GCNs), with their superior capability in dealing with graph structural data, have been introduced to skeleton-based action recognition. Yan *et al.* [23] propose the Spatial Temporal Graph Convolutional Network(ST-GCN) for skeleton-based action recognition. The ST-GCN include two important components, the GCN and 1D temporal convolution. The former is used to extract the spatial features of human skeleton and the latter is applied to the temporal edges between the corresponding joints in consecutive frmmes. Shi *et al.* [24] propose the two-stream adaptive graph convolution network (2s-AGCN) to adaptively learn the co-relation between non-local joints for various tasks. Moreover, the 2s-AGCN further boost the performance through modeling both the joint and the bone data simultaneously. Afterward, many methods [25]–[31] based STGCN and AGCN have been proposed to gain unprecedented performance for skeleton based action recognition.

Although GCN-based methods make great improvements in accuracy of action recognition, from a careful review, there are two drawbacks in the methods mentioned above: (1) Most existing methods only consider the predefined relationships among individual joints but ignore the body-parts corrections which include a lot of fine-grained information. Every movement of the human body is completed by the interaction and coordination of various parts of the body. For example, the action of "walking" tends to be understand based on the collaborative movements of abstract arms and legs, rather than the detailed locations of fingers and toes [32]. Therefore, such a single-scale graph is still insufficient to reflect the high-level representations for different action sequences. (2) Most existing methods utilize only one fixed convolution kernel 9×1 to extract temporal context, resulting in the fact of duration differences of different actions is considered insufficient in existing researches. To be specific, some actions such as "writing" can be recognized in a very short time, some actions such as "wear a shoe" should take a relatively long time to judge. Thus, only apply one fixed convolution temporal kernel is inadequate for feature extraction of diverse human actions.

In light of preceding analysis, for different actions, different parts of the human body have different degrees of correlations and their movement speed is also very different. Therefore, we argue that taking multi-scale spatial-temporal correlations into account could enhance the explainability and stability of the classification results. Consequently, in this paper we propose a network named multi-scale spatial temporal graph neural networks (MSTGNN) to obtain the multi-scale discriminative spatial-temporal features for skeleton-based action recognition. For the spatial domain, we firstly develop a multi-scale graph data structure to establish a more comprehensive body-part relationship model hierarchically. A novel hybrid spatial pooling module combining graph convolution operation and attention mechanism is proposed to exploit the global and comprehensive information step-by-step. The graph convolution operation can merge body-parts features adaptively and the attention mechanism can enhance the expressiveness of features. For the temporal domain, we design a multi-scale temporal convolution module. In this module, we employ multiple convolution kernels with different sizes to capture temporal features for actions with different durations. Inspired by the success of SKNet [33], we utilize it to aggregate all the temporal features adaptively to get more plentiful features. Moreover, our proposed model fuse two complementary data branches including position branch and motion branch at the early stage, which lead to one-stream architecture and can be trained in an end-to-end manner. The bottleneck structure is introduced to alleviate the amount of parameters tuning costs. Overall, our model has fewer parameters and less computation complexity. We evaluate the proposed method on two large-scale datasets, NTU-RGB+D and NTU-RGB+D 120. The proposed model achieves excellent performance with the state-of-the-art on both datasets. The main contributions of our work are summarized as follows:

(1) In the spatial domain, inspired by the kinematics of the human action, we develop a three-scale graph data structures in a fine-to-coarse way. A novel hybrid spatial pooling module is then proposed to dynamically exploit the global and comprehensive information step-by-step.
(2) In the temporal domain, we design a multi-scale temporal convolution module adaptively fusing the temporal features extracted by different scale convolution kernels.
(3) The proposed MSTGNN belonging to one-stream architecture can be trained in an end-to-end manner, which achieves state-of-the-art performance with less computation complexity on two real-world large scale datasets, NTU-RGB+D and NTU-RGB+D 120.

II. RELATED WORK

In this section, we provide a brief overview of the previous methods. Earlier handcrafted-feature based methods [7]–[9] mainly employ shallow architecture for learning the features designed on the basis of human knowledge, which cannot effectively capture the discriminative spatial and temporal information from skeletons. Recent advances in deep learning make it possible to model more complicated spatial-temporal dependency and achieve best results at present. Considering whether or not the spatial features are captured by graph convolution network (GCN), we can classify the deep learning based methods into two categories: Non-GCN-based methods and GCN-based methods.

A. NON-GCN-BASED METHODS

The Non-GCN-based methods mainly applied Convolutional neural Networks (CNNs) and Recurrent Neural Network (RNNs) for human action recognition. The RNN-based methods [16]–[22] treat the 3D coordinates of all joints of human body in time sequence as a vector sequence and then use RNN to extract temporal information. For example, Wang and Wang [22] propose a novel two-stream RNN architecture to model both temporal dynamics and spatial configurations for skeleton data. Song *et al.* [19] combined an LSTM model with a spatial-temporal attention mechanism to automatically select highly discriminative joints and learn the particular attention for each frame on the timeline. However, affected by gradient explosion and disappearances problems, RNN-based methods are difficult to train, computationally heavy and less effective to learn long-term periodic temporal dependencies. Compared with the RNN, the CNN is easier to train and parallelize. The CNN-based methods [10]–[15] manually convert skeleton sequences into pseudo-images according to the fixed designed transformation rules, which encoder temporal dynamics and skeleton joints into rows and columns respectively, and then input the pseudo-images into CNN for classification. Although the schemes mentioned above have achieve high accuracies, joint vector sequences and pseudo-images are inadequacy to express the correction of the human body structure. Therefore, they fail to make an effective use of the skeleton structure of human body to capture complex spatial correlations, resulting in no further performance improvement.

B. GCN-BASED METHODS

In recent years, methods which represent skeletons as a graph and apply GCN to capture structural feature provide a good solution for human action recognition task. Yan *et al.* [23] first propose a general GCN method ST-GCN to model skeleton data and construct a predefined graph according to the physical connections of human body. But as the fixed topology constraint, the dependencies of non-physically connected joins which be crucial for action recognition are not well exploited. Subsequently, substantial research on the basis of ST-GCN are generated gradually. Shi *et al.* [24] propose the two-stream adaptive graph convolution network (2s-AGCN) which employe an adaptive adjacency matrix to exploit the co-relation between global joints. Furthermore, the 2s-AGCN adopts two-stream architecture to promote the recognition accuracy. Some works such as [28], [30] further add more complicated spatial and temporal attention mechanisms to capture the dynamic spatial and temporal correlations. In the above GCN-based methods, they take the skeleton graph as a whole, and neglect an important aspect that most of human actions are performed by the co-movement of various parts of the body. The relations or constraints between different body-parts are not well exploited in these methods. Afterward, a recent work [34] propose a part-based skeleton model (PL-GCN) which apply graph pooling to explore the features of body parts and unpooling operation to reconstruct the joint-level graph. Another work [31] propose a part-wise attention module (PartAtt) to focus on discovering the importance of different body parts and restore the body-parts features to joint-level features. Essentially, only a single-scale graph is used in the two methods, which is still insufficient to reflect the abundant information for different action sequences. Different from the previous work, our model develop a multi-scale skeleton graph in a fine-to-coarse way, and extract global and high-level semantic discriminative information hierarchically.

III. METHODS

In this section, we illustrate the proposed model MSTGNN. Firstly, we formulate the problems to be resolved in this paper. Next, we illustrate the overall architecture of the framework. And then we introduce important components of the framework in detail.

A. PROBLEM FORMULATION OF SKELETON BASED ACTION RECOGNITION

Since the skeleton is abstracted as graphs, skeleton-based action recognition can be formulated as a graph modeling problem. Following previous studies, we defined the N skeleton joints as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V}$ is the set of N joints, $\mathcal{E}$ is a set of several bones, and $A \in \{0, 1\}^{N\times N}$ is predefined adjacency matrix representing the joins' connections. Given a skeleton sequence $S = [s_1, s_2, \ldots, s_T]$, s_t is the group of 3D coordinates of all the N joints at time t. T is the total number of frames in sequence S. The join data can be directly obtained from the original skeleton coordinates, which is represented as:

$$J_{i,t} = (x_{i,t}, y_{i,t}, z_{i,t}), \quad \forall i \in N, \ t \in T \tag{1}$$

where $i = 1, 2, \ldots, N$, $t = 1, 2, \ldots, T$. Our goal is to propose efficient model $\mathcal{M}$ to exploit both the spatial structural information and temporal dynamics embedded in the skeleton sequence. Then map the sequence to a certain action class l:

$$l = \mathcal{M}([s_1, s_2, \ldots, s_T], \mathcal{G}) \tag{2}$$

B. MODEL OVERVIEW

An overview of our proposed MSTGCN is illustrated in Figure 1. The whole model employ a backbone named Spatial-Temporal Bottleneck Block (STBB) including a graph convolution module and a multi-scale temporal convolution module. The graph convolution module is used to explore spatial features between joints or parts of body, following by a multi-scale temporal convolution convolution module which is used to aggregate the contextual features embedded in adjacent frames. Both modules apply bottleneck structure to alleviate the amount of parameters tuning costs. We fuse two branches features including the position branch and motion branch at the early stage of the model. The position branch considers the locations information while the motion branch is designed to model the action dynamic cues. The two branch are complementary to each other. Each branch contains three layer STBB which the number of output channels are all 64. The extracted features of each branch are concatenated and feed into a main stream which contains a six-layer STBB to extract discriminative features. In the

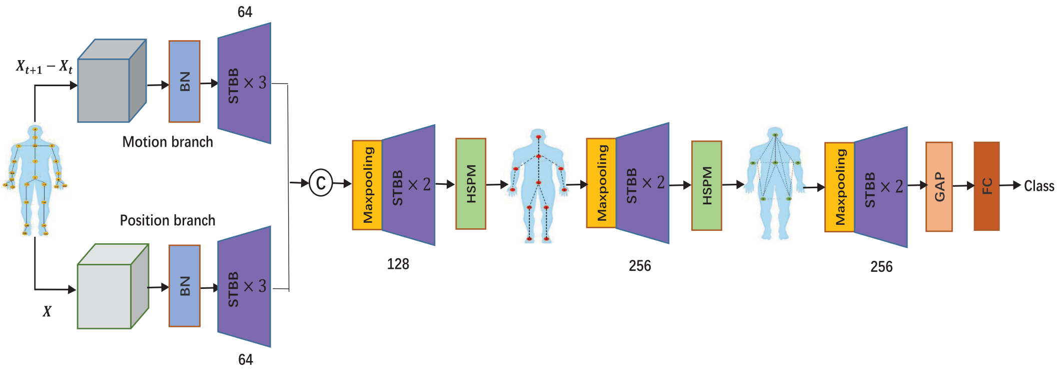

FIGURE 1. The illustration of our proposed MSTGNN architecture. STBB is the spatial-temporal bottleneck block, which contains a sequential execution of one graph convolution module and one multi-scale temporal convolution module. HSPM denotes the hybrid spatial pooling module. The numbers inside and at the bottom of STBB represent the number of module and output channels, respectively. BN means Batchnorm layer, GAP means global average pooling. Maxpooling is the max pooling layer which both the kernel and stride are 2 × 1.

main stream, two hybrid spatial pooling modules (HSPM) which can hierarchically capture high-level and comprehensive information are embedded behind the 2-th and the 4-th STBB. The output channels of the six-layer STBB in the main stream are 128, 128, 256, 256, 256 and 256, respectively. The data batch normalization (BN) layer is added at the beginning to normalize the input data. The maxpooling layer is used to reduce the feature map' temporal size by half. Finally, the extracted features are processed by global average pooling (GAP) and full connected layer (FC) successively to obtain the softmax score of each action. In this way, the whole model can be trained in an end-to-end manner.

C. HYBRID SPATIAL POOLING MODULE

Most of the current existing methods take the skeleton graph as a whole, and ignore the fact that the human body is coordinated by several parts in the process of movement. Therefore, existing methods using a single-scale graph miss high-level semantic features which are essential to discriminate action representations. To solve the problem, we first develop a multi-scale graph structure to establish a more comprehensive body-part relationship model. Then, we propose a hybrid spatial pooling module to capture richer fine-grained information. The hybrid spatial pooling module can reduce the spatial size of inputs, summarize all the representations and reduce redundant features, thus giving rise to better generalization and performance.

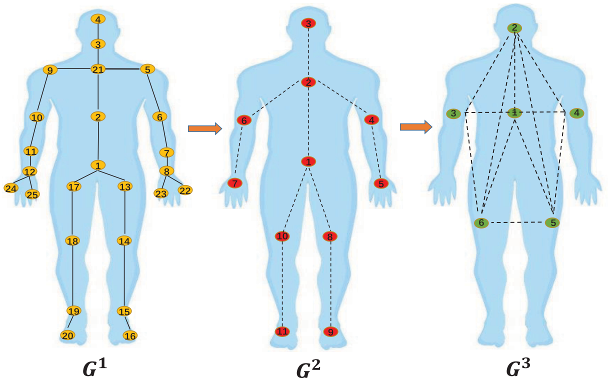

FIGURE 2. The illustration of the partition strategies for human skeleton graph of NTU-RGB+D.

Firstly, we describe how to construct the multi-scale graph G^p, where $p = 3$ is the number of graph. SGCN [35] attempts to manually design the partition strategies according to the natural human skeleton joints' semantics properties and the principle of gradual progress. As show in Figure 2 and 3, we similarly partition body into eleven parts and six parts at the two-level graph G^2 and G^3, which both partition one more sub-part than SGCN. The reason is that we argue the head and torso should be distinguished. For example, some actions such as "put on/take off a hat or glasses" are more relevant to "head", while other actions such as "hugging other person or torch chest" are more relevant to "torso". According to the partition strategies, we can get masked grouping/pooling matrix $Z^p \in \mathbb{R}^{V\times U}$ to represent how we manually group V nodes into U groups. The element in $Z^p \in \{0, 1\}$ indicates whether or not the v-th joint belongs to the next level u-th pooling group. Here, we can see that $Z^2 \in \mathbb{R}^{25\times 11}$ and $Z^3 \in \mathbb{R}^{11\times 6}$. Then, we use A^p to denote the adjacency matrix of the p-th hierarchical graph G^p. $A^1 \in \mathbb{R}^{25\times 25}$ is the adjacency matrix of predefined skeleton graph G^1 according to physical connections of human articulations. Following [36], we can get the new adjacency matrix A^p for the pooled graph G^p using A^{p-1} and Z^p in the following manner:

$$A^p = (Z^p)^T A^{p-1} Z^p \tag{3}$$

The formulation ensures that any two sub-parts i and j in G^p are connected if any of the constituent joints are neighbors in the up-level graph G^{p-1}. It is noticed that the diagonal elements of A^p are needed to set to 1 manually. We use $\widehat{Z}^p \in \mathbb{R}^{U\times V}$ to denote the normalized matrix of pooling matrix Z^p.

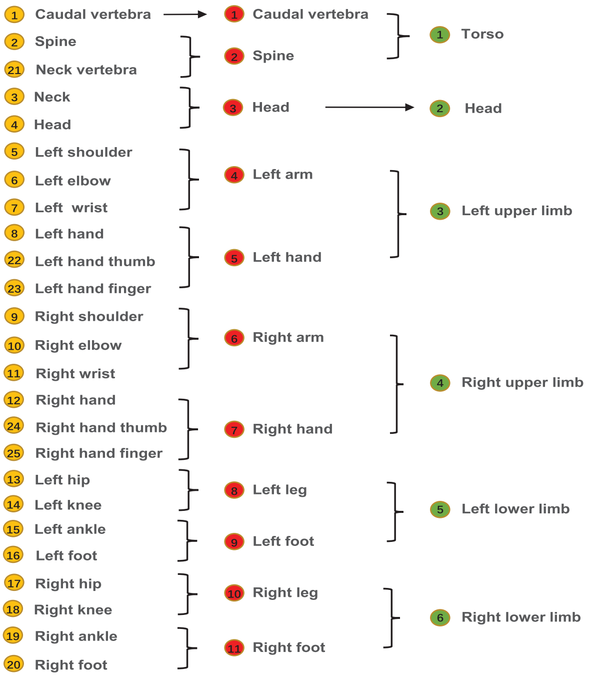

FIGURE 3. The illustration of the partition strategies for human skeleton graph of NTU-RGB+D.

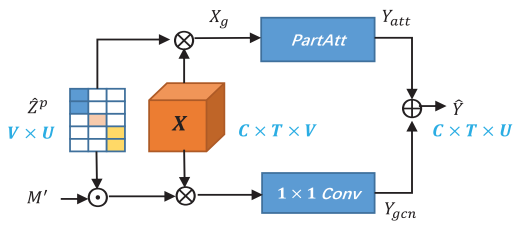

FIGURE 4. The illustration of hybrid pooling module. ⊙ denotes the element-wise multiplication, ⊗ denotes the matrix multiplication and ⊕ denotes the element-wise addition.

As show in Figure 4, we propose a hybrid pooling module to realize the efficient merging operation of the multi-scale skeleton graph. The module can be divided into two components, the one based on graph convolution is for merging body-parts features adaptively, and the other based on the attention mechanism is for features refinement. In the first component, given skeleton features $X \in \mathbb{R}^{C \times T \times V}$, we pool these V joints into U groups via graph convolution, which can be formulated as:

$$Y_{gcn} = (\hat{Z}^p \odot M')XW' \tag{4}$$

where $\odot$ is element-wise multiplication, $M' \in \mathbb{R}^{U \times V}$ is the trainable weight for contribution of joint v in group u, $W' \in \mathbb{R}^{C \times C}$ is the weight of the convolution operation. In the second component, we employ the Part-wise Attention (PartAtt) [31] to work on the average features of each group. The PartAtt based on the global contextual feature maps can discover the importance of different groups and enhance the downsample spatial-temporal features. The second component can be formulated as:

$$\begin{aligned} X_g &= X\widehat{Z}^p \\ Y_{att} &= X_g \odot \delta(\sigma(pool(X_g)W_1)W_2) \end{aligned} \tag{5}$$

where X_g is the average feature of each group, *pool* means the average pooling of all frames and joints in each channel, $W_1 \in \mathbb{R}^{C \times (C/r)}$ and $W_2 \in \mathbb{R}^{(C/r) \times (C \times U)}$ are the weights of two fully connected layers, respectively. r is reduction ratio, σ and δ represent the ReLU and Softmax activation function.

Finally, we fuse the features of two components in a sum manner. Two pooling modules are embedded between STBB in our model, which can broaden the spatial receptive fields and extract useful discriminative information.

D. SPATIAL GRAPH CONVOLUTION MODULE

According to Equation (3), we can obtain the corresponding adjacency matrix A^p of the hierachical graph G^p. Following the PA-ResGCN [31], we employ the graph convolution layer based on a distance sampling function to extract spatial features, the spatial GCN operation is formulated as:

$$f_{out} = \sum_{d=0}^{D} W_d^p f_{in}((\Lambda_d^p)^{-\frac{1}{2}} A_d^p (\Lambda_d^p)^{-\frac{1}{2}} \odot M_d^p) \tag{6}$$

where D is the predefined maximum graph distance, f_{in} and f_{out} denote the input and output feature maps, $\odot$ means element-wise multiplication, A_d^p represents the d-th order distance sampling of the adjacency matrix A^p that marks the pairs of joints with a graph distance d, and Λ_d^p is used to normalized A_d^p. W_d^p and M_d^p are both learnable parameters, the former is utilized to implement the convolution operation and the latter is utilized to tune the importance of each edge. In order to reduce the parameters and calculation, we apply the bottleneck structure, which inserts two 1×1 convolutional layers to adjust the number of feature channels before and after the graph convolution layer.

E. MULTI-SCALE TEMPORAL CONVOLUTION MODULE

In most of existing methods [23], [24], [27], [31], the temporal convolution module applies a fixed $\Gamma \times 1$ convolution filter to extract high-level temporal features, where Γ denotes the kernel size. Then the temporal convolution can be formulated as:

$$\begin{aligned} f_{temporal}(v_t) &= \sum_{v_q \in R(v_t)} f_{in}(v_q)w(v_q) \\ R(v_t) &= \left\{ v_q \middle| -\frac{\Gamma}{2} \leq q - t \leq \frac{\Gamma}{2} \right\} \end{aligned} \tag{7}$$

where R the sampling region for the temporal convolution along the temporal dimension. v_t denotes the joints in time t. w_q denotes the weight for v_q.

However, considering the characteristics of different actions, the classification depends on data at different time scales. Some actions require only a few frames of data, while the others may do not. Therefore, merely use fixed kernel size

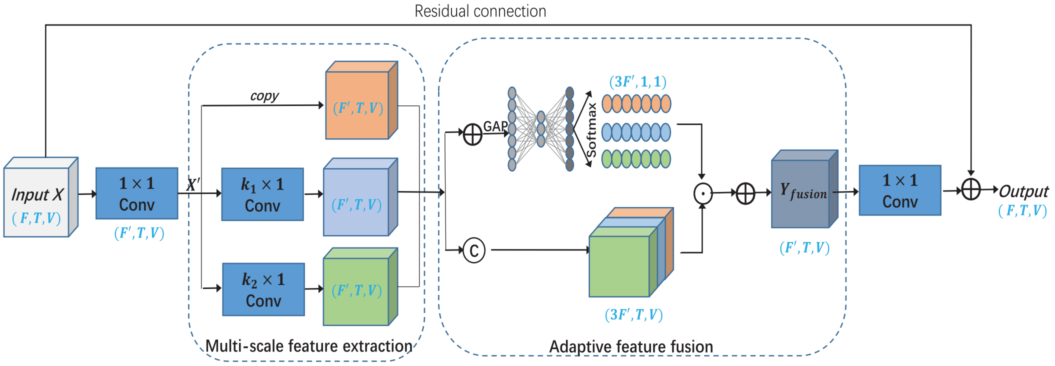

FIGURE 5. **The illustration of our proposed SGCN architecture.**

is insufficient for different action classification. The work in literature [30] employs Inception-Resnet TCN (IR-TCN) layer with three diverse temporal convolution kernel sizes to obtain more discriminative temporal features. However, IR-TCN directly concatenates all the output from different kernels for temporal dynamics, which brings a lot of redundant features that is not conducive to build an efficient model. To address the issue, we employ SKNet [33] to fuse different scales of information and enhance the expressiveness of the model. Here, we introduce a multi-scale temporal module as shown in Figure 5. In detail, there are two parts: multi-scale feature extraction and adaptive feature fusion.

Similar to the spatial graph convolution module, the temporal module also applies the bottleneck structure to reduce the number of parameters. Given the input $X \in \mathbb{R}^{F\times T\times V}$ and the down-sample output $X' \in \mathbb{R}^{F'\times T\times V}$ through 1×1 convolution of the bottleneck structure, where $F' = F/r$ is the number of reduction feature channels with a reduction rate r. In the part of multi-scale feature extraction, suppose there are m continuous temporal convolution with different kernel sizes, in which each temporal convolution takes X' as input and produces output $f_m \in \mathbb{R}^{F'\times T\times V}$ according to Equation (7). Moreover, we set $f_0 = X'$, which preserves the intrinsic feature maps from a 1×1 convolution layer. The second part is adaptive feature fusion which is similar to the PartAtt aforementioned. Specifically, we firstly fuse the results from all feature maps f_m via an element-wise summation. Then we reduce the resolutions and compress the channels through a global pooling layer and a 1×1 convolution layer, respectively. Finally, the softmax operator works channel-wise after another 1×1 convolution layer, which generates a re-weighting matrix $Q \in \mathbb{R}^{(m+1)\times F'}$. The final fusion feature Y_{fusion} is aggregated from different scale features, as follows:

$$Y_{fusion} = \sum_{i=0}^{m+1} (Q_i \odot f_i) \tag{8}$$

At the end, we restore the channels F' of Y_{fusion} to output channels F through a 1×1 convolution layer. The introduction of residual connection is to eliminate the problem of gradient disappearances or explosion, Through this kind of dynamic weight distribution, the model can automatically respond and select features from different scale inputs.

F. ONE-STREAM ARCHITECTURE

Multi-stream architecture is widely adopted to boost the classification performance in many current methods. However, the multi-stream architecture has two disadvantages. On the one hand, the multi-stream architecture shares the same network, which doubles or even quadruples the number of parameters. On the other hand, due to the network is trained independently, the training time and memory consumption is growing when increase the number of stream. To solve this issue, motivated by Multiple Input Branches (MIB) [31] architecture, we fuse two branches including position branch denoted as P_{branch} and motion branch denoted as M_{branch} at the early stage, leading to a one-stream architecture, as shown in Figure 1. In this way, the parameters are dramatically reduced, resulting in less training time and memory consumption. To be specific, position branch is $P_{branch} = \{J_{i,t}, B_{i,j,t}\}$ and motion branch is $M_{branch} = \{J\text{-}M_{i,t}, B\text{-}M_{i,t}\}$. Formally, given the source joint $J_{i,t}$ and the target joint $J_{j,t} = (x_{j,t}, y_{j,t}, z_{j,t})$, the bone data can be calculated as:

$$B_{i,j,t} = (x_{i,t} - x_{j,t}, y_{i,t} - y_{j,t}, z_{i,t} - z_{j,t}) \tag{9}$$

Moreover, joint motions which can provide kinematic cues are calculated by the joint coordinates differences between two adjacent frames:

$$J - M_{i,t} = J_{i,t+1} - J_{i,t}, \quad \forall i \in N,\ t \in T \tag{10}$$

The bone motions are also obtained in the same way:

$$B - M_{i,t} = B_{i,t+1} - B_{i,t}, \quad \forall i \in N,\ t \in T \tag{11}$$

Note that the two branches are complementary and we will show that the fusion can achieve better results in our experiments.

IV. EXPERIMENTS

To verify the efficiency of our proposed MSTGCN, we conduct our experiments on two real-world datasets: NTU-RGB+D [17] and NTU-RGB+D 120 [37]. To investigate the contributions of each important component in MSTGNN, we perform exhaustive ablation experiments on the smaller dataset NTU-RGB+D. Finally, we compare the performance of MSTGCN with other state-of-the-art approaches.

A. DATASETS

(1) **NTU-RGB+D Dataset**. NTU-RGB+D is the most widely used dataset for skeleton action recognition tasks. It consists of 56,800 action clips(samples) from 60 action classes. The samples are performed by 40 different subjects in a lab environment and captured by three cameras with different view angles. There are 25 joints with 3D coordinates in each human skeleton, and one or two subjects in each sample. For classification task, we follow the benchmark evaluations in the original work, which are **cross-subject (X-Sub)** and **cross-view (X-View)** evaluations. For X-Sub evaluation, 40 subjects are divided into training group and testing group. There are 40,320 samples performed by 20 subjects in training set and 16,560 samples performed by the rest subjects in test set. For X-view evaluation, the dataset is divided into training group and testing group according to camera views. The training and testing set have 37,920 and 18,960 samples, respectively.

(2) **NTU-RGB+D 120 Dataset**. NTU-RGB+D 120 dataset is an extended version of NTU-RGB+D. It contains 114,480 skeleton samples which are categorized into 120 action classes. These samples are performed by 106 subjects and captured from 32 different camera setups. Similar to NTU-RGB+D, the evaluation metrics for this dataset are suggested under two settings:
(1) **cross-subject (X-sub120)**, the samples performed by 53 subjects are used for training, and the rest are used for testing. The training and testing set have 63,026 and 50,922 samples, respectively.
(2) **cross-setup (X-set120)**, the samples captured from the camera setups with even IDs are used for training, and the rest are used for testing. The training and testing set have 54,471 and 59,477 samples, respectively.

B. IMPLEMENTATION DETAILS

Our experiments are carried out on the Pytorch deep learning framework with two NVIDIA GTX 2080Ti GPU. In our experiments, the maximum graph distance D is set to 2, and the temporal kernel sizes are set 5 and 9. All experiments use stochastic gradient descent (SGD) with Nesterov momentum (0.9) as the optimization strategies of our method. The batch size is 32, the weight decay is 0.0002 and the initial learning rate is 0.1. The training process include 50 epochs in total. A warmup strategy is utilized at the first 5 epochs to make the training procedure more stable. We adopt the cosine annealing scheduler to reduce the learning rate gradually. The cross-entropy is applied as the loss function.

C. STRATEGIC ANALYSIS AND ABLATION STUDY

There are three key points in our model, *i.e.*, two-branch input, multi-scale temporal convolution module and hybrid spatial pooling module. In this section, we verify the effectiveness of each component. The experiments are conducted on the NTU-RGB+D dataset under the cross-subject **(X-sub)** protocol.

TABLE 1. The comparison of accuracies(%) with different input data on NTU-RGB+D datasets.

Architecture	Input data	Params(M)	X-sub(%)
One-branch	Joint Joint & Bone	0.89 0.89	87.0 90.1
Two-branch	Position&Motion	0.94	**91.3**

TABLE 2. The comparison of accuracies(%) with multiple temporal convolution kernels with different sizes on NTU-RGB+D datasets.

Architecture	Kernel sizes	Params(M)	X-sub(%)
one-kernel	9	0.81	90.4
two kernel	3,9 5,9 7,9	0.91 0.88 0.97	90.9 **91.3** 90.6
three kernel	3,5,9 3,7,9 5,7,9	1.0 1.03 1.05	91.0 90.9 91.0

TABLE 3. The comparison of accuracies(%) with using HSPM on NTU-RGB+D datasets. The G^1, G^2, G^3 refer to the three-scale graph introduced in Figure 2.

Architecture	Graph data	Params(M)	X-sub(%)
One-graph	only G^1	0.70	90.1
Two graph	G^1,G^2 G^1,G^3	0.76 0.88	90.3 90.6
Three graph	G^1 &,G^2, G^3 w/o PartAtt G^1, G^2, G^3	0.78 0.94	91.1 **91.3**

1) THE INFLUENCE OF THE INPUT BRANCH

As shown in Figure 1, MSTGNN is designed to fuse the position branch and motion branch at the early stage of the model. The position branch containing joints and bones data considers the locations information, while the motion branch containing their second-order features for the temporal dimension is designed to model the action dynamic cues. Table 1 presents the ablation studies of MSTGNN with the different input data. Note that the one-branch architecture which only contains the position branch can be divided into two group: 1) one-branch architecture with only joins data. 2) one-branch architecture with joins and bones data. As show in the table, the one-branch architecture only with joints data obtains the worst accuracy, and the one-branch architecture with joins and bones data obtain the mediate accuracy. The two branches architecture achieve the best accuracy than the others. The results reveal that the position branch and motion branch are necessary and complementary to each other.

TABLE 4. Comparison with the GCN-based state-of-the-art methods on NTU-RGB+D and NTU-RGB+D 120 datasets in top-1 accuracy (%) and parameters number (million).

Models	Years	Params	NTU-RGB+D		NTU-RGB+D 120	
			X-sub	X-view	X-sub120	X-set120
ST-GCN [23]	2018	3.10	81.5	88.3	70.7	73.2
AS-GCN [25]	2019	6.99	86.8	94.2	77.9	78.5
2s-AGCN [24]	2019	6.94	88.5	95.1	82.5	84.2
RA-GCN [38]	2020	6.25	87.3	93.6	81.1	82.7
NAS-GCN [27]	2020	6.57	89.4	95.7	–	–
STIGCN [39]	2020	1.60	90.1	96.1	–	–
MV-IGNet [40]	2020	1.84	89.2	**96.3**	83.9	85.6
PA-ResGCN (Bottleneck) [31]	2020	1.14	90.3	95.6	86.6	87.1
PA-ResGCN (Basic) [31]	2020	3.26	90.9	96.0	87.3	**88.3**
MSTGNN (Ours)	-	0.94	**91.3**	95.9	**87.4**	87.6

2) THE INFLUENCE OF MULTI-SCALE TEMPORAL CONVOLUTION MODULE

As the typical kernel size $\Gamma = 9$ has been proven effective in many methods, we keep it in our proposed methods. At the same time, considering larger kernel sizes and the more kernels both bring more parameters and heavier calculation, we employ no larger than 9 in kernel sizes and no more than three kernels to keep the efficiency of our model. We choose two or three kernels from the sizes of 3,5,7,9 to evaluate the performance, which are shown in Table 2. We can see that using only one kernel $\Gamma = 9$ achieve the worst performance (90.4%). Increasing the number of kernels brings a certain amount of improvement, ranging from 0.2% to 0.8%. Overall, using three kernels is generally better than using two kernels. However, compare to using three kernels, the settings of 5 and 9 achieve the best performance (91.3%). We argue that using three kernels silghtly cause overfitting due to bring more parameters. Finally, considering that kernel setting of 5 and 9 achieve the best performance and reduce the number of parameters by about 0.12M, we adopt this combination.

3) THE INFLUENCE OF HYBRID SPATIAL POOLING MODULE

As introduced in Section 3.2, we apply two hybrid spatial pooling module (HSPM) after the 2nd and 4th STBB in the main stream. To evaluate the efficiency of the HSPM, we design three other version of MSTGCN which manually remove one or two pooling module. To simplify, we utilize the G^1, G^2, G^3 to denote the different version of MSTGNN. For example, the G^1, G^2 represent only using one first HSPM and removing the second HSPM, and the G^1 indicates that no any HSPM is used. The results are listed in Table 3, we can see that the version of no HSPM (the first row) achieves the worst accuracy, while using one HSPM (the second and third row) is beneficial to improve accuracy. The version using two HSPM (the fifth row) achieves the best accuracy which proves the necessity of HSPM. To further verify the efficiency of HSPM, we remove the PartAtt of HSPM in the architecture of using three graph, and its performance is also shown in Table 3 (the fourth row). Specifically, compared to the former three version, the version without the PartAtt still boosts the performance by 0.5% at least. When integrating PartAtt into the pooling module, the model obtains the best performance (91.3%), which implies the PartAtt can help to extract discriminate features. Therefore, from the above experiments, we can infer that the HSPM can capture diverse features and enhance the explainability and stability for the classification results of action recognition.

D. COMPARISONS TO OTHER STATE-OF-THE-ART METHODS

To further verify the superiority and generality of our proposed method MSTGCN, we compare with other state-of-the-art methods on both the NTU-RGB+D dataset and NTU-RGB+D 120 dataset. The methods which we selected for comparison are all recent GCN-based methods, in which STIGCN [39], MV-IGNet [40] and PA-ResGCN [31] are one-stream-architecture based methods, while other methods are multi-stream-architecture based models. Note that the MSTGCN can be viewed as an one-stream-architecture based approach. The results are reported in Table 4. We obtain three important observations. First, the multi-stream-architecture based models such as 2s-AGCN [24] and NAS-GCN [27], bring more parameters than one-stream-architecture based models. Obviously, the parameter size of MSTGCN is the smallest.

Second, in terms of X-sub and X-sub120, MSTGCN achieves the best accuracies, 91.3% and 87.4% respectively. In terms of X-view and X-set120, MSTGCN obtains competitive performance 95.9% and 87.6%, respectively. For example, alough MSTGCN is 0.4% lower than MV-IGNet [40], the parameter size of our model is almost half of MV-IGNet.

Third, MSTGCN is similar to PA-ResGCN in architecture structure. We can also observe that MSTGCN outperforms PA-ResGCN (Bottleneck) on both two datasets, and is comparable with PA-ResGCN (Basic) with less parameters. This is because, 1) We reduce the input features branch, which reduces the time and memory consumption and makes the training procedure easier to converge. 2) We propose a hybrid spatial pooling module to extract multi-scale and high-level discriminative information, while PA-ResGCN just focus on the single-scale information on the level of joints graph. 3) For the temporal aspects, we apply multiple convolution kernels instead of one kernel to extract temporal discriminative features adaptively.

V. CONCLUSION

In this paper, we propose a multi-scale spatial temporal graph neural network (MSTGNN) for the skeleton-based action recognition task. In the spatial domain, we develop a multi-scale graph structure to establish a more comprehensive body-part relationship model. A hybrid pooling module combining graph convolution and attention mechanism is then proposed to dynamically exploit the global and comprehensive information step-by-step. In the temporal domain, we design a multi-scale temporal convolution module to fuse the temporal features extracted by different scale convolution kernels adaptively. In addition, we fuse two branches including position branch and motion branch at the early stage, leading to a one-stream architecture. In this way, the parameters are dramatically reduced, resulting in less training time and memory consumption. Due to the contribution of above three key points, our MSTGNN achieve the state-of-the-art performance with less computation complexity on two large-scale action recognition datasets.

REFERENCES

[1] M. Ziaeefard and R. Bergevin, “Semantic human activity recognition: A literature review,” *Pattern Recognit.*, vol. 48, no. 8, pp. 2329–2345, Aug. 2015.

[2] Y. Kong and Y. Fu, “Human action recognition and prediction: A survey,” 2018, *arXiv:1806.11230*. [Online]. Available: http://arxiv.org/abs/1806.11230

[3] P. M. Pilarski, A. Butcher, M. Johanson, M. M. Botvinick, A. Bolt, and A. S. R. Parker, “Learned human-agent decision-making, communication and joint action in a virtual reality environment,” 2019, *arXiv:1905.02691*. [Online]. Available: http://arxiv.org/abs/1905.02691

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

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2D pose estimation using part affinity fields,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7291–7299.

[6] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, “RMPE: Regional multi-person pose estimation,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2334–2343.

[7] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, “Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations,” in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1–4.

[8] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human action recognition by representing 3D skeletons as points in a lie group,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 588–595.

[9] B. Fernando, E. Gavves, M. Jose Oramas, A. Ghodrati, and T. Tuytelaars, “Modeling video evolution for action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5378–5387.

[10] Y. Du, Y. Fu, and L. Wang, “Skeleton based action recognition with convolutional neural network,” in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 579–583.

[11] H. Liu, J. Tu, and M. Liu, “Two-stream 3D convolutional neural network for skeleton-based action recognition,” 2017, *arXiv:1705.08106*. [Online]. Available: http://arxiv.org/abs/1705.08106

[12] B. Li, Y. Dai, X. Cheng, H. Chen, Y. Lin, and M. He, “Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN,” in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2017, pp. 601–604.

[13] M. Liu, H. Liu, and C. Chen, “Enhanced skeleton visualization for view invariant human action recognition,” *Pattern Recognit.*, vol. 68, pp. 346–362, Aug. 2017.

[14] T. S. Kim and A. Reiter, “Interpretable 3D human action analysis with temporal convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1623–1631.

[15] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, “A new representation of skeleton sequences for 3D action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3288–3297.

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

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

[18] J. Liu, A. Shahroudy, D. Xu, and G. Wang, “Spatio-temporal LSTM with trust gates for 3D human action recognition,” in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 816–833.

[19] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, “An End-to-End spatio-temporal attention model for human action recognition from skeleton data,” 2016, *arXiv:1611.06067*. [Online]. Available: http://arxiv.org/abs/1611.06067

[20] I. Lee, D. Kim, S. Kang, and S. Lee, “Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1012–1020.

[21] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, “View adaptive recurrent neural networks for high performance human action recognition from skeleton data,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2117–2126.

[22] H. Wang and L. Wang, “Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 499–508.

[23] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” 2018, *arXiv:1801.07455*. [Online]. Available: http://arxiv.org/abs/1801.07455

[24] L. Shi, Y. Zhang, J. Cheng, and H. Lu, “Two-stream adaptive graph convolutional networks for skeleton-based action recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12026–12035.

[25] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, “Actional-structural graph convolutional networks for skeleton-based action recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3595–3603.

[26] Y.-H. Wen, L. Gao, H. Fu, F.-L. Zhang, and S. Xia, “Graph CNNS with motif and variable temporal block for skeleton-based action recognition,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8989–8996.

[27] W. Peng, X. Hong, H. Chen, and G. Zhao, “Learning graph convolutional network for skeleton-based human action recognition by neural searching,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 2669–2676.

[28] L. Shi, Y. Zhang, J. Cheng, and H. Lu, “Skeleton-based action recognition with multi-stream adaptive graph convolutional networks,” *IEEE Trans. Image Process.*, vol. 29, pp. 9532–9545, 2020.

[29] F. Ye, S. Pu, Q. Zhong, C. Li, D. Xie, and H. Tang, “Dynamic GCN: Context-enriched topology learning for skeleton-based action recognition,” in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 55–63.

[30] W. Li, X. Liu, Z. Liu, F. Du, and Q. Zou, “Skeleton-based action recognition using multi-scale and multi-stream improved graph convolutional network,” *IEEE Access*, vol. 8, pp. 144529–144542, 2020.

[31] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, “Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition,” in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1625–1633.

[32] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian, “Dynamic multiscale graph neural networks for 3D skeleton based human motion prediction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 214–223.

[33] X. Li, W. Wang, X. Hu, and J. Yang, “Selective kernel networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 510–519.

[34] L. Huang, Y. Huang, W. Ouyang, and L. Wang, “Part-level graph convolutional network for skeleton-based action recognition,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 11045–11052.

[35] W. Yang, J. Zhang, J. Cai, and Z. Xu, "Shallow graph convolutional network for skeleton-based action recognition," *Sensors*, vol. 21, no. 2, p. 452, Jan. 2021.

[36] E. Ranjan, S. Sanyal, and P. Talukdar, "Asap: Adaptive structure aware pooling for learning hierarchical graph representations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 5470–5477.

[37] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, Oct. 2020.

[38] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Richly activated graph convolutional network for robust skeleton-based action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Aug. 7, 2020, doi: 10.1109/TCSVT.2020.3015051.

[39] Z. Huang, X. Shen, X. Tian, H. Li, J. Huang, and X.-S. Hua, "Spatio-temporal inception graph convolutional networks for skeleton-based action recognition," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2122–2130.

[40] M. Wang, B. Ni, and X. Yang, "Learning multi-view interactional skeleton graph for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Oct. 21, 2020, doi: 10.1109/TPAMI.2020.3032738.

DONG FENG received the M.S. degree from the School of Information and Computer, Anhui Agricultural University, in 2013. He is currently pursuing the dual Ph.D. degree with the Hefei Institutes of Physical Science, Chinese Academy of Sciences, and the University of Science and Technology of China (USTC). His current research interests include data mining on the Internet of Vehicles, graph convolution networks, and deep learning.

ZHONGCHENG WU received the Ph.D. degree from Institute of Plasma Physical, Chinese Academy of Science (ASIPP), in 2001. From 2001 to 2004, he did his postdoctoral research with the University of Science and Technology of China (USTC). He was a Visiting Professor Researcher with the Computer Science Department, Hong Kong Baptist University, in 2005. Since 2008, he has been a Professor with the Hefei Institutes of Physical Science, Chinese Academy of Sciences, and a Doctoral Supervisor with the University of Science and Technology of China and the University of Chinese Academy of Sciences. He has published more than 140 papers in journals and international conference. His current research interests include standardization of the sensor interface, sensor technology, machine perception, pen computing and pen inferencem, and natural human–computer interaction.

JUN ZHANG received the Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2020. He is currently an Engineer with the Hefei Institutes of Physical Science, Chinese Academy of Sciences. His current research interests include the Internet of Things, machine learning, and pattern recognition.

TINGTING REN received the Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2017. She is currently an Engineer with the Hefei Institutes of Physical Science, Chinese Academy of Sciences. Her current research interests include pattern recognition and machine learning.