Multi-Stream and Enhanced Spatial-Temporal Graph Convolution Network for Skeleton-Based Action Recognition

FANJIA LI\textsuperscript{1,2}, AICHUN ZHU\textsuperscript{3}, YONGGANG XU\textsuperscript{1}, RAN CUI\textsuperscript{1}, AND GANG HUA\textsuperscript{1}

\textsuperscript{1}School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221008, China
\textsuperscript{2}Jiangsu Province Xuzhou Technician Institute, Xuzhou 221151, China
\textsuperscript{3}School of Computer Science and Technology, Nanjing Tech University, Nanjing 211800, China

Corresponding author: Gang Hua (ghua@cumt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51574232 and Grant 61901475, and in part by the State Key Research Development Program of China under Grant 2017YFC0804401.

\section*{ABSTRACT}
In skeleton-based human action recognition, spatial-temporal graph convolution networks (ST-GCNs) have achieved remarkable performances recently. However, how to explore more discriminative spatial and temporal features is still an open problem. The temporal graph convolution of the traditional ST-GCNs utilizes only one fixed kernel which cannot completely cover all the important stages of each action execution. Besides, the spatial and temporal graph convolution layers (GCLs) are serial connected, which mixes information of different domains and limits the feature extraction capability. In addition, the input features like joints, bones, and their motions are modeled in existing methods, but more input features are needed for better performance. To this end, this article proposes a novel multi-stream and enhanced spatial-temporal graph convolution network (MS-ESTGCN). For each basic block of MS-ESTGCN, densely connected multiple temporal GCLs with different kernel sizes are employed to aggregate more temporal features. To eliminate the adverse impact of information mixing, an additional spatial GCL branch is added to the block and the spatial features can be enhanced. Furthermore, we extend the input features by employing relative positions of joints and bones. Consequently, there are totally six data modalities (joints, bones, their motions and relative positions) that can be fed into the network independently with a six-stream paradigm. The proposed method is evaluated on two large scale datasets: NTU-RGB+D and Kinetics-Skeleton. The experimental results show that our method using only two data modalities delivers state-of-the-art performance, and our methods using four and six data modalities further exceed other methods with a significant margin.

\section*{INDEX TERMS}
Skeleton based action recognition, graph convolution network, multi-stream, enhanced spatial-temporal.

\section*{I. INTRODUCTION}
Human action recognition which aims to accurately classify human actions \cite{1}, plays an essential role in video surveillance, pedestrian tracking, health care systems, virtual reality, and human-computer interaction \cite{2}--\cite{13}. The data of actions are presented in the form of RGB videos or skeleton data. The skeleton data are demonstrated to be comprehensive and informative for representing human actions by biological studies \cite{14}. They are represented by 2D or 3D coordinates of joint locations, which means they are more succinct than RGB data. Moreover, they are more robust against variations in viewpoints, body scales, motion speeds, illumination, clothing textures, and background clutter \cite{15}--\cite{17}. Thanks to these superiorities, action recognition based on skeleton becomes an attractive and popular research domain \cite{18}--\cite{23}.

Conventional methods in this domain \cite{24}--\cite{26} usually use handcraft features such as joint angles, distances, and kinematics for human body modeling. But these features have certain limitations in action representing and suffer from lack of universality. In addition, these methods use shallow architectures which constrain learning capabilities and cannot comprehensively capture spatial-temporal features. With the development of deep learning, Convolutional...
Neural Network (CNN) and Recurrent Neural Network (RNN) have been implemented to the task. Although these two networks have achieved substantial improvements, they both neglect the internal dependencies between correlated joints, because the skeleton is naturally non-Euclidean spatial graph-structured with joints as vertexes and bones as edges. To this end, Graph Convolutional Networks (GCNs), which generalize classic convolution from image to arbitrary graph, are implemented to the task. The spatial-temporal GCN (ST-GCN) by Yan et al. [27] is the groundbreaking work among them. ST-GCN achieves encouraging results and many subsequent ST-GCNs are constructed based on it [28]–[31]. However, how to further exploit discriminative spatial and temporal features and retrieve better semantic information are still challenging problems. Specifically, three issues in recent ST-GCNs need to be addressed.

First of all, the movements of joints along the temporal dimension contain crucial cues for representing the underlying action. Due to the different duration and number of discriminative stages of each action execution, the fixed kernel size of traditional temporal convolution [27]–[30] cannot fully cover all cases. Secondly, in each basic block of ST-GCNs, the spatial graph convolution layer (GCL) and the temporal GCL are connected with a tandem structure. Although the tandem structure is concise and costless in computation, it mixes information of different domains and limits the feature extraction capability. Thirdly, the public datasets used in skeleton-based tasks only provide 3D or 2D coordinates of the joint locations [21], [32]. The coordinates are regarded as the first-order information, and the bone information calculated as the difference between source joint and target joint coordinates [28] is regarded as the second-order information. Moreover, the motion information is extracted by the temporal displacements of the joints and bones in two consecutive frames [33]. All of the above features are discriminative and can be used as input of ST-GCNs. However, relative positions of the joints and bones which are also informative are ignored.

To address the aforementioned issues, we propose a novel model named as multi-stream and enhanced spatial-temporal graph convolution network (MS-ESTGCN). For the temporal convolution of each basic block, we employ multiple kernels with different sizes instead of a fixed kernel to aggregate more temporal features. The corresponding temporal GCLs are densely connected for parameter-efficient and dense knowledge propagation [34].

In order to enhance the spatial features, we change the tandem structure of the original block to a parallel structure with two branches. One branch is the serial connected spatial GCL and temporal GCLs. The other is an additional spatial GCL. The outputs of the two branches are added and then transmitted to the next block. In this way, spatial features can be enhanced.

Besides the joint and bone information together with their motion information, we further extend the input features by employing relative positions of the joints and bones. Specifically, we calculate the difference between two coordinates of adjacent joints along the traversal sequence defined by the datasets, and the same is true of the bones. In that case, the input features consist of six modalities in total: joints, bones, their motions and relative positions. This makes our model a six-stream network, and the final results can be obtained by adding the softmax scores of each stream.

To verify the superiority of MS-ESTGCN, elaborate experiments are conducted on two large-scale datasets: NTU-RGB+D [21] and Kinetics-Skeleton [32]. The experimental results prove that the novel construction of spatial-temporal blocks and the multiple streams bring notable performance gains.

Overall, the main contributions of our work are summarized as follows:

- We propose MS-ESTGCN which consists of multiple novel constructed blocks with both spatial and temporal features enhanced. In each block, densely connected temporal GCLs with different kernel sizes are employed to extract more precise and informative temporal features.
- The block is transformed from a tandem structure to a parallel structure by adding an additional spatial GCL branch. Spatial features are enhanced as a result.
- To the best of our knowledge, we are the first to propose a six-stream network using six modalities (joints and bones, their motions and relative positions) to take full advantage of the low-level features.
- Compared to other models, the proposed model using two modalities delivers state-of-the-art results on the two datasets. Our models using four and six modalities further exceed other models with a significant margin.

II. RELATED WORK

In this section, we review the work related to this article, such as action recognition and pose estimation. There are substantial methods that focus on skeleton-based action recognition based on deep learning. We divide them into three categories: RNN-based, CNN-based, and GCN-based methods.

A. RNN-BASED ACTION RECOGNITION

The RNN, which has an advantage in learning temporal dependencies, structures skeleton data as sequences of joint-coordinate vectors [35]–[37]. An end-to-end hierarchical RNN is proposed in [20] to model the temporal dependencies between positional configurations of joints. According to the human physical structure, the skeleton is divided into five parts and fed to five subnets separately. But RNNs are difficult to train because of gradient vanishing and exploding problems. A new type of RNN, named IndRNN [37] is proposed to address the problems, in which neurons are connected across layers and are independent of each other in the same layer. Since Long Short-term Memory (LSTM) network can model long term contextual information and capture the co-occurrences of human joints, it is more popular.
for the task. Zhang et al. [38] design a view adaptive model based on LSTM which can decrease the influence of the angle variations by automatic learning and choosing the most optimal angle. Liu et al. [17] extend the usage of LSTM to both spatial and temporal domains. The proposed ST-LSTM network can better analyze the dynamics and dependency relations within the skeleton sequences in both of the two domains. Similarly, an attention model based on LSTM [16] is built to extract spatial-temporal features by adaptively choosing discriminative joints of one frame and assigning different importance to each frame. To combine cross-domain attention, a joint training strategy is employed with a regularized cross-entropy loss. Si et al. [39] further extend the LSTM with an attention enhanced graph convolution. The LSTM is used to capture features in temporal dynamics while the graph convolution is employed to capture features in spatial configuration. For each layer, the information of key joints is enhanced by the attention network.

### B. CNN-BASED ACTION RECOGNITION

The CNN, widely used in image processing, structures skeleton data as pseudo-images [40]–[42]. Compared to RNN-based methods, CNN-based methods are easier to train and parallelize. Ke et al. [23] propose a method to transform the skeleton sequences into three clips which are generated from the three channels of joint coordinates. The frames of the clips represent temporal information of the sequences and spatial information of the joints. To effectively extract spatial-temporal features and eliminate the adverse impacts of noisy data and angle variations, [42] proposes an enhanced skeleton visualization method. The method describes the skeleton sequences as a series of color images to encode temporal and spatial cues. [40] is the first work to apply 3D CNN to the task. The two-stream network maps skeleton joints to 3D coordinate space and extracts deep features by a multi-temporal structure. However, the aforementioned methods only aggregate local co-occurrence features, thus have limited ability to capture actions involving long-range joint interactions. To solve this problem, Li et al. [43] propose an end-to-end two-stream framework to learn both local and global co-occurrence features. The skeleton sequences are presented as three-dimensional tensors [frames, joints, 3]. They first extract point-level features by treating the last dimension as the channel of the convolution, and then extract global-level features by making the joints dimension as the channel. To further explore low-level features, Wang et al. [44] propose a four-stream network with a temporal convolutional sub-network and a co-occurrence convolutional sub-network.

### C. GCN-BASED ACTION RECOGNITION

Due to the superiority in modeling the graph-structured human skeleton data, GCNs have been successfully applied in action recognition. [27] is the first work of applying GCN to the task. It constructs a spatial graph based on the physical connections of joints and adds edges between the same joints across adjacent frames in the temporal dimension. Hence it has the ability to simultaneously extract features embedded in spatial configuration and temporal dynamics. But it is based on a pre-defined graph with fixed topology constraint, which ignores implicit correlations of joints that are not physically connected. To transform the topology of the skeleton graphs from fixed to variable, [29] proposes an A-link inference module to capture action-specific dependencies and a pose prediction sub-network to further improve the accuracy by self-supervision. [28] parameterizes the graph and makes it adaptive to each layer and each action. [31] proposes motif-based GCN with a weight-sharing strategy for the joints of the same semantics. Variable temporal dense block and attention mechanism are employed to enhance the local and global temporal dependencies. [45] formulates a general GCN which divides the graph to four subgraphs instead of treating the whole skeleton as a single graph. Each subgraph corresponds to a part of the body. The model captures the high-level properties of subgraphs and learns the relations between them. To comprehensively model coherent skeleton information, Wu et al. [30] propose SDGCN which includes two novel components. One is a cross-domain spatial residual layer which can enhance the spatial-temporal information. The other one is a dense connection block to learn global information effectively. [46] is the first work to infer the relations of joints in the task. Specifically, a joints relation inference network is employed to globally aggregate the optimal relation between two arbitrary joints and infer the optimal adjacency matrices. A skeleton graph convolutional network is then used to make the final classification.

The aforementioned GCNs are all formulated in the spatial-temporal domain. [47] extends the task to the spectral domain. A residual frequency attention block is proposed to better classify the actions with characteristic frequency patterns which are restricted in the spatial-temporal domain.

### D. POSE ESTIMATION

Pose estimation plays an important role in computer vision and can be used for higher-level tasks (e.g., action recognition). The skeleton data can be acquired using pose estimation algorithms [48]. [49] introduces an end-to-end architecture which incorporates convolutional networks into pose machine framework. The architecture can learn both spatial models and image features without graphical model-style inference. [50] proposes a network which is composed of multiple stacked hourglass modules and allows repeated bottom-up and top-down reasoning by pooling and upsampling. Although the network is very powerful, it is confused by the occlusion or overlap of human bodies. [51] addressed the problem by hard joints mining technique. The proposed GAN-based network consists of two stacked hourglasses, one is the generator and the other is the discriminator. The difficult joint points in the generator are given more attention during the training period benefiting from the hard joints mining technique. Moccia et al. [52] are the first to investigate infants’ pose estimation by using depth videos acquired in the actual clinical practice. They propose an innovative network...
FIGURE 1. The architecture of the multi-stream enhanced spatial-temporal graph convolution network. There are a total of 10 blocks which are marked as B1 to B10. Take B5 as an example to introduce the meaning of the three numbers. There are 64 input channels, 128 output channels, and the stride is 2. There are six modalities fed into the network separately: joint, bone, joint motion(j-m), bone motion(b-m), joint relative position (j-rp), bone relative position(b-rp). Each modality corresponds to a stream. The final prediction is obtained using the weighted sum of the softmax scores. The details of the block are shown in Fig.2.

FIGURE 2. The architecture of the proposed enhanced spatial-temporal graph convolution block. The block consists of spatial GCLs and densely connected multiple temporal GCLs. ⊕ denotes the elementwise addition operation and ⊗ denotes the matrix multiplication operation. BN denotes batch normalization. Convt denotes the temporal graph convolution. The details of the adjacency matrix are shown in Fig.5.

which exploits spatio-temporal features with a detection and a regression CNN. They also release a new dataset named babyPose which is the first in the field.

III. METHODS
In this section, we introduce the structure and components of our proposed MS-ESTGCN in detail.

A. NETWORK ARCHITECTURE
Fig.1 illustrates the pipeline of MS-ESTGCN. The network is composed of 10 enhanced spatial-temporal graph convolution blocks which are connected in series. Blocks B1 to B4 have 64 output channels. Blocks B5 to B7 have 128 output channels and blocks B8 to B10 have 256 output channels. The detailed architecture of the blocks is shown in Fig. 2. Specially, we follow the setting of [27] which removes the shortcut of the first block. The input data is normalized by a BN layer at the beginning of the network. After block B10, the feature maps of samples are pooled to the same size by a global pooling average (GPA) layer. Finally, the result is obtained by a softmax layer.

To improve network performance, we make an intensive study in input features. The skeleton sequences are formulated as a four-dimensional tensor \[N \times M \times C \times T \times V\]. \(N\) denotes the batch size, \(M\) denotes the number of persons, \(C\) denotes the number of channels, \(T\) denotes the number of frames in each sample, and \(V\) denotes the number of joints for a person. For the \(C\) dimension, it provides the original joint coordinates which can be represented as:

\[
J_{i,t} = (x_{i,t}, y_{i,t}, z_{i,t}), \quad \forall i \in V, \ t \in T
\]  

(1)

The most extensive methods for skeleton-based action recognition are conducted based on joint coordinates which are shown in Fig.3(a). In addition, the joint displacements across time can provide kinematic cues for the task. Along the \(T\) dimension, we can extract joint motion information which
is shown in Fig.3(b) by coordinate differences between the same joints in two adjacent frames:

$$J - M_{i,t} = J_{i,t+1} - J_{i,t}, \quad \forall i \in V, \ t \in T$$  \hspace{1cm} (2)$$

Furthermore, in order to extract discriminative geometric features, [45] calculates the joint relative positions with respect to shoulders (left and right) and hips (left and right) by difference operation. But in this way, each joint should do the difference four times and will get four sets of coordinates which leads to twelve channels. This will obviously increase network complexity. So we adopt an easier and immediate method without changing the channels. Along the V dimension, we calculate the coordinate differences between two adjacent joints in each frame following the traversal sequences defined by datasets. The joint relative positions which are shown in Fig.3(c) can be formulated as:

$$J - RS_{i,t} = J_{i,t+1} - J_{i,t}, \quad \forall i \in V, \ t \in T$$  \hspace{1cm} (3)$$

The effectiveness of this modality will be demonstrated in Section IV-C. Besides the joints based modalities, bones based modalities are also discriminative. We follow [28] and calculate the vectors of bones which are shown in Fig.3(d).

As shown in Fig.1, the six modalities: joints, bones, their motions (J-M, B-M), and their relative positions (J-RP, B-RP) are fed into the network separately and constitute a six-stream architecture. The softmax scores of the six streams are fused by weighted sum and the final prediction can be made.

**B. SPATIAL GRAPH CONVOLUTION LAYER**

ST-GCN [27] formulates the spatial graph convolution as:

$$f_{\text{Convs}} = \sum_{k}^{K_c} W_k (f_{\text{in}} A_k) \odot M_k$$  \hspace{1cm} (4)$$

where \(f_{\text{in}}\) denotes the input feature map, \(W_k\) and \(K_c\) denote the weight vector and kernel size of the convolution operation respectively. As the nodes are divided into 3 subsets (root, centripetal, and centrifugal nodes) following the spatial configuration partitioning strategy, \(K_c\) is set to 3 accordingly. \(M_k\) is a trainable weight to capture edge weights. \(A_k\) is a fixed normalized adjacency matrix which represents the physical structure of the human skeleton. The human skeleton graph and the corresponding adjacency matrix are shown in Fig.4.

As introduced in Section II, many methods are devoted to change the skeleton graph from predefined structure to variable structure. Here we formulate the spatial graph convolution based on the adaptive graph [28]. Specifically, the graph convolution operation (illustrated in Fig.2) is formulated as:

$$f_{\text{Convs}} = \sum_{k}^{K_c} W_k f_{\text{in}} (A_k + B_k + C_k)$$  \hspace{1cm} (5)$$

Compared with (4), the main difference lies in the adjacency matrix of the graph which turns to the sum of \(A_k\), \(B_k\), and \(C_k\). See Fig.5 for illustration. Their meanings are as follows [28].

\(A_k\) is the same as (4). \(B_k\) is a layer adaptive adjacency matrix with all the values parameterized and initialized to 0. \(C_k\) is a sample adaptive adjacency matrix which measures the similarity of two nodes by normalized embedded Gaussian function. Concretely, the input feature map \(f_{\text{in}}\) is firstly embedded by two embedding functions, \(\theta\) and \(\varphi\). Each function is a \(1 \times 1\) convolutional layer. Then the outputs of \(\theta\) and \(\varphi\) are multiplied and the result is the similarity matrix.
Finally, $C_k$ is obtained after a softmax normalization. In summary, the use of the three types of graphs enables the network to adaptively learn the topology and achieve better performance in the task.

In addition, the spatial GCL includes a residual connection which can guarantee the model stability. Hence the output of spatial GCL can be formulated as:

$$f_{\text{spatial}} = B(f_{\text{Conv}}) + I(f_{\text{in}})$$

where $B$ denotes batch normalization (BN), $I$ denotes the identity mapping.

**C. TEMPORAL GRAPH CONVOLUTION LAYERS**

In order to effectively extract the spatial-temporal context, the spatial graph convolution is followed by the temporal graph convolution as shown in Fig.2. [27] formulates temporal graph convolution by densely concatenating the consecutive temporal GCLs as shown in Fig.2. The output of the $l$-th layer is denoted as:

$$x_l = H_l ([x_0, x_1, \ldots, x_{l-1}])$$

where $[x_0, x_1, \ldots, x_{l-1}]$ is the concatenation of feature outputs from preceding layers. $H$ is a composite function consisting of a batch normalization (BN), a rectified linear unit (ReLU) activation function, and a temporal graph convolution. Let the feature map extracted by each layer have the same scale with $k$ channels (i.e., growth rate of DenseNet), the input channels of the temporal graph convolution be set to $k_0$. Then the number of channels for the $l$-th layer can be calculated as:

$$k_l = k_0 + k \times (l - 1)$$

where the values of $k_0$ and $k$ are associated with the number of output channels for each block which will be discussed in Section IV-C. A transition layer following the dense connection is utilized to match the output channel setting of each block. It is made up of a BN, a RELU, and a $1 \times 1$ convolution layer.

In brief, by introducing dense connection to the multiple temporal GCLs which use different kernel sizes, the model can make full use of temporal information from shorter to longer terms. As to how many temporal GCLs should be employed and how to choose the kernel sizes are also discussed in Section IV-C.
D. ENHANCED SPATIAL-TEMPORAL GRAPH CONVOLUTION BLOCK

In the RGB-based action recognition domain, Tran et al. [53] demonstrate that 3D CNNs can achieve better performance than 2D CNNs. Furthermore, they show the superiority of decomposing 3D CNNs to separately 2D spatial convolution followed by 1D temporal convolution. The ST-GCNs [27]–[29] adopt this idea, accordingly construct the spatial-temporal graph convolution block. Our MS-ESTGCN is also composed of multiple basic blocks. To better contrast the differences between blocks of other ST-GCNs and ours, we show their architectures in Fig.7 and Fig.2 respectively. In Fig.7, the traditional block is structured as a residual network with a shortcut. The residual mapping part is used to extract spatial-temporal features with a tandem structure including a spatial GCL and a temporal GCL.

Comparing Fig.2 with Fig.7, we can intuitively observe that our block replaces the identity mapping part with a spatial GCL branch. In this way, the residual mapping part is transformed into a parallel structure. The motivation for this design is that our temporal features are greatly enhanced as introduced in III-C. This makes the spatial features relatively weak and restricts the performance of the network. To this end, we add the additional spatial GCL branch to enhance the spatial features. Meanwhile, as introduced in Section III-B, identity mapping is an inherent part of the spatial GCL. Hence the shortcut of the block is actually preserved and our novel block maintains the residual network structure. The output of our block can be formulated as:

\[
\begin{align*}
    f_{\text{block}} &= T \left[ \sigma \left( f_{\text{spatial}} \right) \right] + \sigma \left( f_{\text{spatial}} \right) \\
    &= T \left[ \sigma \left( f_{\text{spatial}} \right) \right] + \sigma \left[ B \left( f_{\text{Conv}} \right) \right] + \sigma \left[ I \left( f_{in} \right) \right]
\end{align*}
\]

where \( \sigma \) denotes the ReLU activation function, \( T \) denotes temporal graph convolution. The sum of the first two additions is the residual mapping part and represents the enhanced spatial-temporal features. The last addition can be seen as the identity mapping part.

IV. EXPERIMENTS

To evaluate the performance of our model and make a head-to-head comparison with other ST-GCN based methods [27]–[29], extensive experiments are conducted on two large-scale datasets: NTU-RGB+D [21] and Kinetics-Skeleton [32]. As the NTU-RGB+D dataset is the smaller one, we perform ablation studies on it to empirically evaluate the effectiveness of our model components.

A. DATASETS

1) NTU-RGB+D

NTU-RGB+D [21] is the most widely used and currently the largest in-door-captured dataset for skeleton-based human action recognition. It contains 56800 skeleton samples in 60 action classes which are divided into three categories: daily actions, mutual actions, and medical conditions. The samples are performed by 40 volunteers with different ages from 10 to 30 in a lab environment. Three cameras at the same height are used to capture the samples from different perspectives: −45°, 0°, and 45°. The duration of each sequence is no more than 10 seconds with a 30 fps frame rate. There are no more than two subjects in each sample and 25 joints in each subject. The dataset provides 3D joint coordinates which are collected by the Microsoft KinectV2 depth sensor. Two benchmarks are recommended by the authors of the dataset: cross-subject (X-Sub) and cross-view (X-View). For the X-Sub benchmark, 37920 samples captured by cameras 2 and 3 are classified to training set while 18960 samples captured by camera 1 are classified to the validation set. For the X-View benchmark, the training set includes 40320 samples which are performed by 20 subjects. 16560 samples performed by the other 20 subjects form the validation set. We conduct experiments following conventional settings.

2) KINETICS-SKELETON

Kinetics [32] is a much larger human action recognition dataset than NTU-RGB+D. The samples of this dataset are collected from YouTube with 300000 video samples in 400 human action classes. Each class corresponds to at least 400 samples. As the original dataset only provides raw video samples without skeleton data, [27] extracts skeleton data using the OpenPose [48] toolbox. The samples are firstly extracted at a 30 fps frame rate and then resized to a resolution of \(340 \times 256\). No more than two persons are selected in each sample, and a total of 18 joints are estimated for every person. The joint is represented by a tuple of \((X, Y, C)\). \((X, Y)\) denote the 2D coordinates of the joint and \(C\) denotes the confidence score. The training set of the dataset includes 240000 samples and the validation set includes 20000 samples.

B. TRAINING DETAILS

All of the experiments are conducted on PyTorch 1.0 with 4 GeForce RTX1080Ti GPUs. The optimization strategy of our model is stochastic gradient descent (SGD) with Nesterov momentum which is set to 0.9. The loss function is cross-entropy and the weight decay is set to 0.0001. The batch size is set according to the size of GPU memory. For the NTU-RGB-D dataset, the batch size is set to 32 and the initial learning rate is set to 0.1. When training to the 30th epoch and 45th epoch, the learning rate will be divided by 10. There are 65 epochs in each training process. For Kinetics–Skeleton dataset, the batch size is set to 84 and the initial learning rate is also set to 0.1. When training to the 45th epoch and...
60th epoch, the learning rate will be divided by 10. There are 75 epochs in each training process.

C. ABLATION STUDY

We implement the following methods using all the six modalities on both X-View and X-Sub benchmarks for better comparison:

AGCN [28]. We use this baseline because our model follows the graph construction of it. The original paper only reports results on joint and bone modalities. For a fair comparison, we strictly re-implement the model on our experimental platform.

ETGCN (Enhanced Temporal GCN). This is our proposed network which includes dense connected temporal GCLs with different kernel sizes. The additional spatial GCL branch which is introduced in Section III-D is not added to the block. That is to say, the spatial features are not enhanced.

ESGCN (Enhanced Spatial GCN). This is our proposed network which adds an additional spatial GCL branch to the block. The temporal graph convolution follows [28] which means the temporal features are not enhanced.

ESTGCN. This is our proposed network with both temporal and spatial features enhanced.

1) MULTIPLE TEMPORAL CONVOLUTION KERNEL SIZES

In this section, we evaluate the necessity of multiple temporal convolution kernels with different sizes described in Section III-C using the joint modality for the X-Sub benchmark. For the densely connected temporal GCLs, we set the number of input channels \( k_0 \) in (10) equals to the number of output channels of each block, growth rate \( k \) in (10) equals half of that. The fixed kernel size used in [27] is 9 which is widely demonstrated to be effective [28]–[30]. Thus we set the temporal convolution kernel sizes of our model around 9.

Table 1. Action recognition accuracies using multiple temporal convolution kernels with different sizes by ETGCN on the X-Sub benchmark.

| kernel sizes | Top-1 (%) | Top-5 (%) |
|--------------|-----------|-----------|
| one kernel   |           |           |
| 9            | 86.27     | 97.09     |
| 1,9          | 87.58     | 97.80     |
| 5,9          | 87.49     | 97.54     |
| 9,13         | 87.76     | 97.86     |
| 9,17         | 87.06     | 97.55     |
| 1,5,9        | 87.86     | 97.93     |
| three kernels|           |           |
| 5,9,13       | 87.47     | 97.79     |
| 9,13,17,21   | 86.57     | 97.65     |
| 1,5,9,13,17  | 87.04     | 97.83     |
| four kernels |           |           |
| 5,9,13,17    | 86.60     | 97.60     |

Table 2. The performance of ETGCN with different combinations of input channels and growth rate on the X-Sub benchmark.

| k_0/NOC | k/k_0 | Top-1 (%) | Top-5 (%) | Training time/epoch |
|---------|-------|-----------|-----------|--------------------|
| 1       | 1/2   | 87.86     | 97.93     | 16'33"             |
| 1       | 1/4   | 87.66     | 97.85     | 13'44"             |
| 1       | 1/8   | 87.53     | 97.76     | 12'1"              |
| 1/2     | 1/2   | 87.74     | 97.59     | 12'2"              |
| 1/2     | 1/4   | 87.59     | 97.69     | 10'21"             |
| 1/2     | 1/8   | 86.87     | 97.63     | 8'56"              |
| 1/4     | 1/2   | 87.04     | 97.79     | 10'51"             |
| 1/4     | 1/4   | 86.86     | 97.70     | 8'9"               |
| 1/4     | 1/8   | 86.84     | 97.63     | 7’7”               |

We can see that using only one fixed kernel achieves 86.27% accuracy which is the worst result in Table 1. The effectiveness of using multiple kernels can be verified. When using two kernels, the combination of 9 and 13 achieves 87.76% accuracy which is better than others. When three kernels are used, the combination of 1, 5, and 9 shows the best performance throughout Table 1 with 87.86% accuracy. When four kernels are used, the performance degrades significantly. In general, using two and three kernels all achieve better performance than using one and four kernels. Moreover, we can observe that combinations including large kernel sizes (i.e. 17 and 21) lead to weak performance. The Top-5 accuracies also validate the tendency. Our final model adopts the combination of 1, 5, and 9.

2) INPUT CHANNELS AND GROWTH RATE OF THE DENSE CONNECTION

In this section, we fix temporal kernel sizes to the combination of 1, 5, and 9, then change \( k_0 \) and \( k \) in (10) to evaluate the performance of the dense connection. The results are shown in Table 2. The parameters \( k_0 \) and \( k \) are associated with the number of output channels (NOC) of each block. Note that as shown in Fig.2, the output of spatial GCL is the input of temporal graph convolution. As \( k_0 \) denotes the number of input channels of dense connection, it can also denote the number of output channels of spatial GCL. We set \( k_0 \) equal, 1/2, 1/4 of NOC separately. The equal setting obviously gets the best performance which can make the most of spatial features. \( k \) is set 1/2, 1/4, 1/8 of NOC. We can observe that as \( k \) increases, so does network performance. Specifically, when \( k_0 = \text{NOC} \), the best result of 87.86% is achieved when \( k = 1/2 \). Nevertheless, there’s no clear performance gap whatever \( k \) equals. This benefits from the advantage of dense connection which can have very narrow layers [34]. When \( k_0 \) is set half of NOC, approximate performances are achieved when \( k \) is set half or a quarter of NOC.

97764
TABLE 3. The accuracy comparisons between AGCN [28], ESGCN, ETGCN, ESTGCN with six modalities on X-Sub and X-View benchmarks.

| methods               | accuracy          | X-View |           |           |           | X-Sub |           |           |           |
|-----------------------|-------------------|--------|-----------|-----------|-----------|-------|-----------|-----------|-----------|
|                       |                   | joint  | bone      | j-m       | b-m       | j-rp  | b-rp      | joint     | bone      | j-m       | b-m       | j-rp  | b-rp      |
| AGCN [28]             | Top-1(%)          | 93.94  | 93.88     | 92.45     | 92.41     | 93.43 | 92.88     | 86.27     | 86.65     | 84.24     | 84.71     | 85.70 | 85.22     |
|                       | Top-5(%)          | 99.14  | 99.07     | 98.92     | 98.98     | 99.08 | 99.04     | 97.09     | 97.26     | 96.77     | 96.99     | 97.04 | 97.15     |
| Ours (ESGCN)          | Top-1(%)          | 94.11  | 94.07     | 93.22     | 92.95     | 94.01 | 93.18     | 87.52     | 87.76     | 84.95     | 85.92     | 87.24 | 86.13     |
|                       | Top-5(%)          | 99.08  | 99.18     | 99.03     | 99.02     | 99.12 | 99.18     | 97.51     | 97.65     | 96.93     | 97.43     | 97.56 | 97.31     |
| Ours (ETGCN)          | Top-1(%)          | 94.44  | 94.13     | 93.32     | 93.24     | 94.05 | 93.36     | 87.86     | 88.23     | 86.19     | 86.23     | 88.05 | 86.98     |
|                       | Top-5(%)          | 99.18  | 99.28     | 99.05     | 99.13     | 99.33 | 99.17     | 97.93     | 98.11     | 97.48     | 97.62     | 98.16 | 97.78     |
| Ours (ESTGCN)         | Top-1(%)          | 95.18  | 94.73     | 93.90     | 93.36     | 94.91 | 94.06     | 88.51     | 88.95     | 86.83     | 87.09     | 88.15 | 87.51     |
|                       | Top-5(%)          | 99.31  | 99.29     | 99.20     | 99.10     | 99.24 | 99.20     | 97.74     | 97.85     | 97.59     | 97.80     | 97.81 | 97.67     |

are suppressed severely. Table 2 also reports a comparison of training time for one epoch. We can see that larger values of k0 and k lead to longer training time but better performance. The combination for best performance (k0 = NOC, k = 1/2 NOC) needs 16’ 33’’ per epoch while the combination of k0 = 1/2 NOC, k = 1/4 NOC only needs 10’ 21’’. So the training time reduces by 37.5% and the accuracy drops only 0.27%. In brief, our model adopts the former combination for the best performance. But when the model is oriented towards efficiency, the latter combination is optimal.

3) ENHANCED TEMPORAL GRAPH CONVOLUTION

To further evaluate the effectiveness of enhanced temporal graph convolution, we conduct ETGCN with all the six modalities on both X-View and X-Sub benchmarks. The results are reported in Table 3 and Fig.8. Comparing with AGCN [28], ETGCN brings significant improvements with ranges from 1.52% (b-m) to 2.35% (j-rp) on X-Sub, 0.25% (bone) to 0.87% (j-m) on X-View for all of the six modalities. The accuracies on X-View are already very high, so improvements on X-Sub are relatively more potential.

As discussed in Section III-C, our enhanced temporal GCLs can aggregate more temporal discriminative cues and make a more accurate classification. Table 4 shows the confusion matrix for the action “clapping” on X-Sub. AGCN [28] only achieves 67% accuracies and our ETGCN achieves 81% accuracies with 14% improvement. In AGCN [28], the “clapping” samples are mainly misclassified as other six classes of actions: “reading”, “writing”, “playing with phone/tablet”, “check time(watch)”, “rub hands together”, and “put the palms together”. All of these actions are performed mainly by both hands, and it is very difficult to accurately distinguish them with one glimpse. Nevertheless, the six classes of actions show distinct differences with “clapping” in the temporal dimension. The first three actions do not include significant relative movements of hands. The action of “check time (watch)” keeps hands apart throughout execution whereas
TABLE 4. Confusion matrix for the action of “clapping”. 10 denotes “clapping”, 11 denotes “reading”, 12 denotes “writing”, 29 denotes “playing with phone/tablet”, 33 denotes “check time (watch)”, 34 denotes “rub hands together”, and 39 denotes “put the palms together”. The predicted label percentages greater than or equal to 1% are reported.

| methods       | predicted label percentages |
|---------------|----------------------------|
|               | 10 | 11 | 12 | 29 | 33 | 34 | 39 |
| AGCN[28]      | 67%| 1% | 3% | 3% | 1% | 16%| 2% |
| ETGCN         | 81%| 0% | 1% | 0% | 0% | 11%| 0% |

the action “rub hands together” is the opposite. The last action has been analyzed in Section III-C. Benefits from the enhanced temporal graph convolution, our ETGCN has the ability to classify them more accurately and Table 4 illustrates the effectiveness.

4) ENHANCED SPATIAL-TEMPORAL GRAPH CONVOLUTION
From Table 3 and Fig.8, we can observe that our ESGCN which enhances spatial features, brings notable improvements to all of the six modalities. Comparing with AGCN [28], the accuracies improves 0.71% (j-m) to 1.54% (j-rp) on X-sub, 0.17% (joint) to 0.77% (j-m) on X-View. This demonstrates our ESGCN is very powerful, although the improvements are slightly lower than ETGCN. Our ESTGCN combines the superiority of these two networks, with both temporal and spatial features enhanced, achieves the best performances. Comparing with AGCN [28], ESTGCN gets 2.24% (joint) to 2.59% (j-m) improvements on X-Sub, 0.85% (bone) to 1.48% (j-rp) improvements on X-View.

5) MULTI-STREAM FRAMEWORK
In this section, we evaluate the necessity of using six modalities and report the results in Table 5. Clearly, for one-stream methods using joint-related modalities, the joint modality achieves the best performance. Joint position modality achieves better performance than joint motion modality. The same is true of corresponding bone-related modalities. In this paper, we denote our one-stream method using joint modality as 1S-ESTGCN.

Two-stream methods bring encouraging improvements. Comparing with 1S-ESTGCN, the combination of joint and bone modalities achieves improvements of 1.84% on X-Sub and 0.84% on X-View, which is the best. Similar to one-stream methods, relative position modalities achieve
better performance than motion modalities. We denote our method using joint and bone modalities as 2S-ESTGCN.

When four modalities are fused, further improvements are achieved. Comparing with 1S-ESTGCN, our method using joint, bone, and their motion modalities, which is denoted as 4S-ESTGCN, are the best with 2.7% improvement on X-Sub and 1.32% improvement on X-View. The joint, bone, and their relative position modalities all provide geometric cues whereas the motion modalities provide kinematic cues. So joint, bone, and their motion modalities are more complementary than other combinations.

The six-stream method, which is our final model, denoted as MS-ESTGCN, achieves the best performance of all the methods. Comparing with 1S-ESTGCN, it achieves improvements of 2.85% on X-Sub and 1.58% on X-View.

Furthermore, we make an in-depth analysis of our multi-stream models based on each action. Fig.9 presents the improvements of 2S, 4S, and MS-ESTGCN over 1S-ESTGCN on the X-View benchmark. We can observe that most of the 60 actions gain improvements in the three models. For 2S-ESTGCN, there are 38 actions that achieve improvements by 4% to 1%, 18 actions have no changes. The remaining 4 actions (i.e. “eat meal/snack”, “brushing hair”, “falling”, and “nausea or vomiting”) get lower accuracies than 1S-ESTGCN. We take the former two actions as examples to explain the performance decline. In our model using
bone modality, 2% samples of “eat meal/snack” are misclassified as “drink water”, 2% samples of “brushing hair” are misclassified as “wipe face”. The misclassifications do not appear in 1S-ESTGCN. These two sets of actions are presented in Fig.10, and the two actions in each set look very similar. We argue that they are not discriminative by using bone modality because of being performed all by one arm. The accuracies of these actions in 2S-ESTGCN are driven down as a result.

For 4S-ESTGCN, there are 47 actions that achieve improvements by 8% to 1%, 13 actions have no changes. The best performance comes from MS-ESTGCN. There are 50 actions that achieve improvements by 11% to 1%, only 10 actions have no changes. Note that our 1S-ESTGCN is already powerful. Finally, we visualize the confusion matrix of MS-ESTGCN in Fig.11. There are 54 actions with accuracies of 95% or more, only 3 actions with accuracies lower than 90%; “typing on keyboard” (88%), “reading” (82%), “writing” (77%). As shown in Fig.12, the skeletons of these 3 actions are very similar. The main differences are the fingers, but there are only two joints (“tip of the hand” and “thumb”) are marked. Therefore, it is very challenging or even impossible to precisely tell apart these actions only depends on skeleton data. If the network incorporates the appearance information, these three actions can be distinguished according to whether there is a keyboard, pen or notebook in hands.

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

To demonstrate the superiority and robustness of our model, we compare it with other state-of-the-art models on both NTU-RGB+D dataset and Kinetics-Skeleton dataset. These models are divided into four categories: handcraft-feature-based models [19], [25], RNN-based models [16], [17], [20], [21], [22], [36], [37], [38], CNN-based models [23], [41], [42], [43], [44], GCN-based models [27], [28], [29], [33].

| Methods               | X-Sub (%) | X-View (%) |
|-----------------------|-----------|------------|
| Lie Group [19]        | 50.1      | 82.8       |
| HBRNN [20]            | 59.1      | 64.0       |
| Deep LSTM [21]        | 60.7      | 67.3       |
| ST-LSTM [17]          | 69.2      | 77.7       |
| STA-LSTM [16]         | 73.4      | 81.2       |
| VA-LSTM [38]          | 79.2      | 87.7       |
| ARRN-LSTM [36]        | 80.7      | 80.7       |
| Ind-RNN [37]          | 81.8      | 88.0       |
| SR-TSL[22]            | 84.8      | 92.4       |
| Clips+CNN+MTLN [23]   | 79.6      | 84.8       |
| Synthesized CNN [42]  | 80.0      | 87.2       |
| CNN+Motion+Trans [41] | 83.2      | 89.3       |
| HCN[43]               | 86.5      | 91.1       |
| MF-Net[44]            | 90.0      | 95.4       |
| ST-GCN [27]           | 81.5      | 88.3       |
| AS-GCN[29]            | 86.8      | 94.2       |
| 2s-AGCN[28]           | 88.5      | 95.1       |
| DGNN[33]              | 89.9      | 96.1       |
| **1S-ESTGCN (ours)**  | **88.5**  | **95.2**   |
| **2S-ESTGCN (ours)**  | **90.4**  | **96.1**   |
| **4S-ESTGCN (ours)**  | **91.2**  | **96.5**   |
| **MS-ESTGCN (ours)**  | **91.4**  | **96.8**   |

For the NTU-RGB+D dataset, we report the results of the models on both X-Sub and X-View benchmarks as shown in Table 6. We can observe that deep learning methods perform better than handcraft-feature-based methods. In the deep learning domain, GCN-based methods are more powerful than CNN-based methods. RNN-based methods are relatively weak. Our 1S-ESTGCN performs the best among the methods using only joint modality [22], [27], [29]. Our 2S-ESTGCN achieves state-of-the-art performance
two attributes lead to a decline in recognition performance. Sensitive to the finer movements of the human body. These skeleton data lack the appearance information and are not employed to determine which order is optimal. Thirdly, the network and Neural Architecture Search (NAS) can be thus it is worth introducing higher-order connections into problems to be solved. Firstly, the multi-stream construction increases the parameter count of the network and reduces network efficiency. Therefore, we recommend more exploration on how to fuse multiple modalities into just one stream. Secondly, our skeleton graph only aggregates one-order hop neighbors’ information and neglects higher-order connections that may provide additional performance improvements. Thus it is worth introducing higher-order connections into the network and Neural Architecture Search (NAS) can be employed to determine which order is optimal. Thirdly, the skeleton data lack the appearance information and are not sensitive to the finer movements of the human body. These two attributes lead to a decline in recognition performance. To this end, giving full play to the complementarity of the skeleton data and the RGB data may be a promising method.

V. CONCLUSION
In this paper, we propose a multi-stream and enhanced spatial-temporal graph convolution network with both spatial and temporal features enhanced. In each basic block of our model, densely connected temporal graph convolution layers with different kernel sizes are employed to extract more precise and informative temporal features. An additional spatial GCL branch is added to the block to enhance spatial features. Besides, we use up to six modalities (joints and bones, their motions and relative positions) as input features of our model and the performance achieves state-of-the-art.

Despite the superiority of our model, there are still several problems to be solved. Firstly, the multi-stream construction increases the parameter count of the network and reduces network efficiency. Therefore, we recommend more exploration on how to fuse multiple modalities into just one stream. Secondly, our skeleton graph only aggregates one-order hop neighbors’ information and neglects higher-order connections that may provide additional performance improvements. Thus it is worth introducing higher-order connections into the network and Neural Architecture Search (NAS) can be employed to determine which order is optimal. Thirdly, the skeleton data lack the appearance information and are not sensitive to the finer movements of the human body. These two attributes lead to a decline in recognition performance. To this end, giving full play to the complementarity of the skeleton data and the RGB data may be a promising method.

REFERENCES
[1] S. Mathe and C. Sminchisescu, “Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 7, pp. 1408–1424, Jul. 2015.
[2] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, “A ‘string of feature graphs’ model for recognition of complex activities in natural videos,” in Proc. ICCV. Nov. 2011, pp. 2595–2602.
[3] D. Huang and K. Kitani, “Action-reaction: Forecasting the dynamics of human interaction,” in Proc. ECCV. Jul. 2014, pp. 489–504.
[4] L.-Y. Gui, K. Zhang, Y.-X. Wang, X. Liang, J. M. F. Moura, and M. Veloso, “Teaching robots to predict human motion,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 562–567.
[5] Z. Duric, W. D. Gray, R. Heishman, F. Li, A. Rosenfeld, M. J. Schoelles, C. Schunn, and H. Wechsler, “Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction,” Proc. IEEE, vol. 90, no. 7, pp. 1272–1289, Jul. 2002.
[6] M. R. Sudha, K. Sriraghav, S. S. Abisheck, S. G. Jacob, and S. Manisha, “Approaches and applications of virtual reality and gesture recognition: A review,” Int. J. Ambient Comput. Intell., vol. 8, no. 4, pp. 1–18, Oct. 2017.
