Motion-Driven Spatial and Temporal Adaptive High-Resolution Graph Convolutional Networks for Skeleton-Based Action Recognition

Zengxi Huang, Yusong Qin, Xiaobing Lin, Tianlin Liu, Zhenhua Feng, Yiguang Liu

***Abstract*—Graph convolutional networks (GCN) have attracted increasing interest in action recognition in recent years. GCN models human skeleton sequences as spatio-temporal graphs. Also, attention mechanisms are often jointly used with GCNs to highlight important frames or body joints in a sequence. However, attention modules learn parameters offline and are fixed, so may not adapt well to various action samples. In this paper, we propose a simple but effective motion-driven spatial and temporal adaptation strategy to dynamically strengthen the features of important frames and joints for skeleton-based action recognition. The rationale is that the joints and frames with dramatic motions are generally more informative and discriminative. We decouple and combine the spatial and temporal refinements by using a two-branch structure, in which the joint and frame-wise feature refinements perform in parallel. Such a structure can also lead to learn more complementary feature representations. Moreover, we propose to use the fully connected graph convolution to learn the long-range spatial dependencies. Besides, we investigate two high-resolution skeleton graphs by creating virtual joints, aiming to improve the representation of skeleton features. By combining the above proposals, we develop a novel motion-driven spatial and temporal adaptive high-resolution GCN. Experimental results demonstrate that the proposed model achieves state-of-the-art (SOTA) results on the challenging large-scale Kinetics-Skeleton and UAV-Human datasets, and it is on par with the SOTA methods on the two NTU-RGB+D 60&120 datasets. Additionally, our motion-driven adaptation method shows encouraging performance when compared with the attention mechanisms.**

***Index Terms*—Graph Convolutional Networks, Skeleton-based Action Recognition, Spatial and Temporal Adaptation, Skeleton Motion, Fully Connected Graph Convolution, High-resolution Graph.**

I. Introduction

HUMAN action recognition (HAR) aims to understand human behaviors and can be used in a wide range of applications [1], [2], such as video surveillance [3], [4], retrieval, entertainment, human-computer interaction [5], smart home/healthcare, etc. In recent years, we have witnessed a shift from using RGB and gray-level videos with hand-crafted features [6]–[10] to the use of deep learning with various modalities, such as skeleton, depth, infrared sequence, point cloud, event stream, audio, radar, and WiFi for HAR [1]. Among them, skeleton data represents the human action trajectories with a set of predefined coordinates of body key joints. Skeleton-based HAR methods are generally more efficient and robust to the variations of a video sequence in illumination, viewpoints, and background.

Early skeleton-based methods focus on extracting hand-crafted features [11]–[13], or encode the skeleton data into sequential vectors or pseudo images, and then model and classify them with recurrent neural networks (RNN) [14], [15], long short-term memory (LSTM) networks [16]–[18], or convolutional neural networks (CNN) [19]–[24]. However, these methods either break the natural spatial graph structure of skeleton data or make it difficult to extract temporal features, and thus cannot fully model the complex spatio-temporal configurations and correlations of the body joints for HAR [1]. On the other hand, graph convolutional networks (GCN) can preserve the spatio-temporal skeleton graph structure by representing body joints as vertices and using their connections of intra/inter-frames as edges. Efficient spatial and temporal graph convolutional operators are then applied to exploit spatio-temporal features in skeleton sequences [25]. Due to the strong spatio-temporal feature learning capability, GCNs have been widely used for skeleton-based HAR and has become the mainstream method in the community [26]–[34].

Generally, an action sequence consists of multiple stages of different importance. How to make GCN networks focus on the important stages and joints is expected to boost action recognition performance. For example, the action 'sit down' can be roughly divided into multiple stages, *e.g.*, stand, sitting down, and sit, as shown in Fig. 1. The frames at the 'sitting down' stage should be much more informative and discriminative than the remainders for identifying the 'sit down' action. Tang *et al*. [35] selected the key frames for recognition with a deep progressive reinforcement learning method, while the remaining frames are directly discarded. This strategy may not only lose some certain discriminative information, but also undermine the integrity of temporal information. Shi *et al*. [27] utilized spatio-temporal attention mechanisms to help the model pay more attention to important joints and frames. Kong *et al*. [31] proposed a multi-perspective attention fusion module to combine the spatial and temporal feature streams. Plizzari *et al*. [36] proposed to use transformer self-attention mechanisms on both spatial and temporal dimensions. However, note that deep learning attention mechanisms generally

Z. Huang, Y. Qin, and T. Liu are with the School of Computer Science and Software Engineering, and the Sichuan Xihua Jiaotong Forensics Center, Xihua University, Chengdu, China (e-mail: huangzx001@mail.xhu.edu.cn).

X. Lin is with School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China (e-mail:lxb155@my.swjtu.edu.cn).

Z. Feng is with the School of Computer Science and Electronic Engineering, University of Surrey, Guildford, UK. (e-mail: z.feng@surrey.ac.uk).

Y. Liu is with the College of Computer Science, Sichuan University, Chengdu, China (e-mail: liuyg@scu.edu.cn).

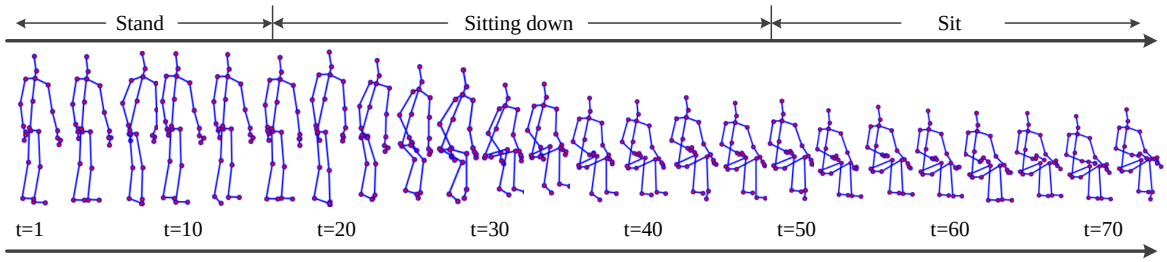


Fig. 1. Illustration of the procedure for the action 'sit down', in which the 'sitting down' stage should be more informative and discriminative than the others.

involve convolution layers, fully connected layers, parameterized activation functions, etc [37], [38]. Their parameters are learned offline and fixed. The learning of attention module parameters is also affected by the amount and diversity of the training data, as well as the network architecture and training tricks [38]. Therefore, the use of attention mechanisms in HAR may not adapt well to unseen samples.

Moreover, many GCN models only consider the first-order information of skeleton data and use a small kernel size of graph convolution [25]–[27], [39], such as physically adjacent joints. They focus on the local features but ignore, to a large extent, the long-range joint correlation information. For example, it is important to capture the correlation between hands and feet for recognizing some specific classes, such as 'put on a shoe' and 'take off a shoe'. Furthermore, only the human body and facial key points are usually annotated to form a skeleton, hence the skeleton has a very small number of key points and the distribution is uneven. For example, the Kinetics-skeleton dataset [40] offers only 18 body joints and such a definition of skeletons may not represent human actions accurately. Consequently, the GCNs created by the original skeleton data have very low spatial resolution and may not be able to effectively extract discriminative features for HAR.

In this work, we address the above limitations from three aspects. First, we propose a skeleton motion-driven spatial and temporal adaptation strategy to guide the GCN networks to dynamically strengthen the features of important frames and joints for action recognition. The rationale is that the joints and the frames with dramatic motion are generally more informative and discriminative. The skeleton motion can be defined by the coordinate differences of the save joints between consecutive frames. The joint-wise and frame-wise weights can be then calculated based on skeleton motion by using a weighting function like $sigmoid$. In each graph convolution block, skeleton features can be refined in the same way as attention mechanisms by using these weights. Second, we propose to use the fully connected graph convolution (FCGC) to exploit long-range joint correlation information. In this context, the receptive field of graph convolution is extended from the physically adjacent joints to all the body joints. Third, we propose a high-resolution skeleton graph (HRG) by creating virtual joints between each pair of physically connected joints.

In the end, we propose to use a two-branch model to separately utilize the spatial and temporal motion-driven adaptation methods, and to achieve better feature learning diversity. Both branches have the same network architecture and adopt the proposed FCGC and HRG. We finally derive a novel motion-driven spatial and temporal adaptive high-resolution graph convolutional network, namely MS&TA-HGCN-FC. In the experiments on four large-scale skeleton datasets, the proposed MS&TA-HGCN-FC achieves the state-of-the-art (SOTA) performance on Kinetics-Skeleton [40] and UAV-Human [41], and it is no par with the SOTA methods on NTU-RGB+D 60 [42] and NTU-RGB+D 120 [43].

The main contributions of the proposed method include:

(1) We propose a motion-driven spatial and temporal attention strategy to adaptively strengthen the features of important joints and frames for HAR. Compared with the attention mechanisms whose parameters are learned offline and fixed, our motion-driven adaptation method is simple yet effective and highly flexible.

(2) We propose to use a high-resolution skeleton graph and fully connected graph convolution. FCGC can capture long-range spatial dependencies, while HRG improves the representation of skeleton features, facilitating the extraction of discriminative human action features.

(3) Extensive experiments are carried out on four large-scale skeleton datasets to validate the effectiveness of the proposed model and its innovative components. Overall, our model achieves SOTA performance on all these benchmarks. Also, the proposed motion-driven adaptation shows better performance than the attention mechanisms used in [27].

The rest of this paper is organized as follows. Section II briefly reviews the existing skeleton-based deep action recognition methods. Section III introduces the basic graph convolutional network and the proposed MS&TA-HGCN-FC model. Section IV reports the experimental results. Last, the conclusion is drawn in Section V.

II. Related Work

A. RNN-Based Action Recognition

RNN and its gated variants, such as LSTM, aim to learn the dynamic dependencies of sequential data. Du *et al*. [14] divided the human skeleton into five parts and separately fed them into five bidirectional RNNs to extract spatio-temporal features. Si *et al*. [15] proposed an RNN model that contains a hierarchical spatial reasoning network (HSRN) and a temporal stack learning network (TSLN). Zhang *et al*. [16] proposed an adaptive RNN using two LSTM sub-networks, which can

dynamically transform the skeleton coordinates to a more suitable observation view. Liu *et al*. [17] proposed an attention-based LSTM network called GCA-LSTM, including a global context memory cell and two LSTM layers. Fan *et al*. [18] transformed the input skeleton into several possible view observations by using attention LSTM networks.

B. CNN-Based Action Recognition

CNN-based methods generally encode skeleton sequences into pseudo images (RGB or gray images) and then classify them via standard CNN networks. Hou *et al*. [20] encoded the spatio-temporal information of a skeleton sequence into color texture images, referred to as skeleton optical spectra. Wang *et al*. [21] encoded the joint trajectories and their dynamics in 3D skeleton sequences into color distribution in 2D images. Li *et al*. [44] focused on data augmentation with translation-scale invariant image mapping and used multi-scale CNN for action recognition. Ke *et al*. [19] transformed each skeleton sequence into three clips, corresponding to the three channels of joint coordinates, each clip consisting of several gray images. Kim and Reiter [45] utilized temporal CNN (TCN) to explicitly learn interpretable spatio-temporal representations for 3D action recognition. Banerjee *et al*. [22] extracted geometrical features from skeleton sequence, including distance, angle, and their temporal differences, and then separately encoded them into 4 gray images. A multi-stream CNN network is used to combine them for classification. Dhiman *et al*. [23] projected a skeleton sequence onto an image and encrypted the human poses using different colors.The pseudo image is then classified using a part-wise spatio-temporal attention-driven CNN network. Xu *et al*. [24] proposed a pure CNN architecture named topology-aware CNN (Ta-CNN) with a cross-channel feature augmentation module. Although a lot of efforts have been made in CNN-based methods, it is still a challenge to model spatio-temporal information for action recognition.

C. GCN-Based Action Recognition

Yan *et al*. [25] proposed the first GCN network called ST-GCN, which models skeleton sequence with its natural spatio-temporal graph structure and uses spatial and temporal graph convolutional operators to extract action features. Inspired by ST-GCN, a great amount of GCN methods have been proposed in recent years. Many studies focus on improving the spatio-temporal topology of GCN networks, aiming to capture the long-range spatial dependencies and the temporal correlations between different joints. Li *et al*. [46] used a spatio-temporal graph routing scheme to produce new connections among joints. Shi *et al*. [26] proposed an adaptive graph convolutional network (AGCN) to learn optimal spatial topology. Obinata *et al*. [47] extended the temporal topology by adding connections to multiple neighboring vertices of inter-frames. Liu *et al*. [28] proposed a unified spatio-temporal graph convolutional operator named G3D, coupled with a disentangled multi-scale aggregation scheme.Ye *et al*. [30] proposed a dynamic GCN (Dynamic GCN) by introducing a convolutional neural network named context-encoding network to learn skeleton topology automatically. Peng *et al*. [48] used a neural architecture search (NAS) scheme to build a graph convolutional network for skeleton-based action recognition. Cheng *et al*. [29] proposed a shift graph convolutional network (Shift-GCN) by using shift graph operations and lightweight point-wise convolutions, so as to make the receptive fields on both spatial and temporal dimensions are flexible. Wu *et al*. [32] used a local-global GCN and a region-aware GCN to model the spatial and temporal information separately.

Attention mechanisms have also attracted much attention in GCN-based methods. Shi *et al*. [27] proposed attention-enhanced AGCN (AAGCN) by embedding spatial-temporal-channel attention modules into AGCN model [26]. Kong *et al*. [31] proposed a multi-perspective attention fusion module to combine the spatial and temporal feature streams. Plizzari *et al*. [36] used transformer self-attention mechanisms on both spatial and temporal dimensions and combined them in a two-stream network. Song *et al*. [49] proposed a sequential multi-stream model called RA-GCN to reduce the impact of noisy or incomplete skeletons. RA-GCN uses class activation maps (CAM) to learn the activation degrees of joints, and only passes the inactivated joints to the next GCN stream.

Multi-stream strategy is commonly used to exploit complementary skeleton information to boost action recognition performance. Shi *et al*. [26] combined an additional AGCN network using the second-order skeleton information (the lengths and directions of human bones) as input to become a two-stream network named 2s-AGCN. Later, they [27] further proposed a multi-stream network (MS-AAGCN) by incorporating two more streams respectively with the joint motion and bone motion. Most of the recent SOTA GCN-based methods follow the multi-stream network architecture [28], [29], [31], [32], [48]–[52].

III. Motion-driven Adaptive GCN

In this section, we first introduce GCN preliminaries and then present our motion-driven spatial and temporal adaptation modules, fully connected graph convolution, high-resolution skeleton graph, and the overall network architecture of the proposed model.

A. Preliminaries

A human skeleton sequence can be modeled as a spatio-temporal graph $G(V, E)$. The vertices V denote human body joints represented by their 2D or 3D joint coordinates. E denotes both spatial and temporal edges. The spatial edges are the natural connections of the human body joints within a frame. The corresponding joints in two consecutive frames are connected as temporal edges. Typically, a GCN network includes several spatio-temperal graph convolution (ST-GC) blocks, each ST-GC block contains a layer that operates spatial graph convolution (SGC) and temporal graph convolution (TGC) [25], [26].

Spatial Graph Convolution: The spatial graph convolution operation on vertex v_i involves the vertices in the same frame, according to [25], [26] , it can be formulated as:

$$f_{out}(v_i) = \sum_{v_j \in B(v_i)} \frac{1}{Z_i(v_j)} f_{in}(v_j) w(l_i(v_j)), \tag{1}$$

where v_i denotes the i-th joint, f_{in} and f_{out} denotes the corresponding input and output features. B_i denotes the sampling area of the convolution for v_i. w is a weight function that allocates weights to the neighboring joints of v_i involved in convolution. Z_i is set to balance the contribution of different neighbors.

The spatial graph convolution is similar to the classical 2D image convolution operation in CNN. B_i can be seen as the receptive field, and w can be seen as a collection of weights of the convolution kernel. However, unlike the 2D image convolution where a rigid grid naturally exists around the center location, it is trickier to define the weight function w in spatial graph convolution [25]. Once the sampling distance is fixed, the number of weights for spatial convolution is fixed, but the number of vertexes in B_i is varied. So, a label function l_i is required to map all neighboring vertexes into a fixed number of subsets each of which is associated with a unique weight vector [27]. According to [25], there are three mapping strategies, we follow the strategy used in [26], [27]. We refer readers to [25]–[27] for more details of mapping strategies and label functions.

For implementation, we denote the feature map of GCN as a tensor $\mathbf{f} \in \mathbb{R}^{N \times T \times C}$, where N is the number of vertexes, T is the number of frames in a sequence, C is the number of channels. Eq. (1) can be transformed as:

$$\mathbf{f}_{out} = \sum_{k}^{K_t} (\mathbf{A}_k \mathbf{f}_{in}) \mathbf{W}_k, \tag{2}$$

where $\mathbf{f}_{in} \in \mathbb{R}^{N \times T \times C_{in}}$ and $\mathbf{f}_{out} \in \mathbb{R}^{N \times T \times C_{out}}$ denote the input and output feature maps, respectively. C_{in} and C_{out} are their channels. K_t denotes the kernel size of the spatial convolution operation and is set to 3 by default. $\mathbf{A}_k \in \mathbb{R}^{N \times N}$ is the normalized adjacency matrix. $\mathbf{A}_k = \mathbf{\Lambda}_k^{-\frac{1}{2}} \overline{\mathbf{A}}_k \mathbf{\Lambda}_k^{-\frac{1}{2}}$, where $\overline{\mathbf{A}}_k$ is similar to the $N \times N$ adjacency matrix, and its element indicates whether one vertex is in the subset of another vertex. $\mathbf{\Lambda}_k^{i,j} = \sum_k (\overline{\mathbf{A}}_k^{i,j}) + \varepsilon$ is the normalized diagonal matrix, where ε is set as 0.001 to avoid empty rows. $\mathbf{W}_k \in \mathbb{R}^{C_{in} \times C_{out} \times 1 \times 1}$ is the weight vector of the 1×1 convolution operation, which represents the weighting function w in Eq.(1).

Temporal Graph Convolution: On the temporal dimension of GCN networks, the same joints in the consecutive frames of a skeleton sequence are connected. Hence, the basic TGC can be performed similarly to the classical convolution operation [25]. Concretely, TGC performs a $K_t \times 1$ convolution on the output feature map $\mathbf{f}_{out}$ in (2), where K_t denotes the kernel size of the temporal dimension. The K_t is set as 9 in our experiments. The above basic TGC only involves the same joint on the inter-frame. It can also be extended to include multiple neighboring vertices on the inter-frame to capture cross-spacetime correlations [28], [47].

B. Motion-driven Adaptation

It can be observed in NTU-RGB+D and Kinetics-Skeleton datasets that, generally, an action sequence can be partitioned into multiple stages. It is unlikely that all the stages play an equal role in action recognition. Intuitively, an action stage that with more dramatic body movement and a more complex skeleton trajectory should be more informative and discriminative. Likewise, the joints that conduct stronger movement should be more representative for a class of actions. Therefore, it is desirable to have a proper strategy to focus on the important stages and joints while suppressing those with noisy and redundant information. Nevertheless, it is difficult to define and slice an action sequence into a certain number of segments in temporal space. In this paper, we focus on frame-wise and joint-wise adaptation strategies and measure the importance of frames and joints or their features by using skeleton motion.

Motion information: Let $\mathbf{P}_t$ be the motion information between two consecutive frames at time t. $\mathbf{P}_t$ can be defined by the difference between the same joints as follows:

$$\mathbf{P}_t = \mathbf{f}_{in}(t) - \mathbf{f}_{in}(t-1), \tag{3}$$

where $\mathbf{f}_{in} \in \mathbb{R}^{N \times T \times C}$ is the input feature map. To align with $\mathbf{f}_{in}$, we set $\mathbf{P}_0 = \mathbf{P}_1$, thus, $\mathbf{P} \in \mathbb{R}^{N \times T \times C}$.

Motion-driven Temporal Adaptation (MTA): In the MTA module, we estimate the skeleton motion associated with frame t with the average motion of all joints within a frame, considering all feature channels. Then, a temporal adaptive weight $\alpha_{tp}(t)$ for frame t can be calculated by using a $Sigmoid$ function with the skeleton motion as input. $\alpha_{tp}(t)$ can be formulated as:

$$\alpha_{tp}(t) = \sigma(\frac{1}{N \times C} \sum_{n}^{N} \sum_{c}^{C} \|\mathbf{P}_{n,t,c}\|_1), \tag{4}$$

where σ denotes the $Sigmoid$ function. The frame-wise teporal adaptive weights for all the frames in a skeleton sequence can be represented by $\alpha_{\mathbf{tp}} \in \mathbb{R}^{1 \times T \times 1}$.

The temporal adaptive weight $\alpha_{tp}(t)$ is then multiplied to the input feature map for adaptive feature refinement. Inspired by the residual attention networks [27], [37], MTA combines the adjusted feature with the input feature using a residual connection, as shown in Fig.2. The output temporal adaptive feature for frame t can be formulated as follows:

$$\mathbf{f}_{out}(t) = \alpha_{tp}(t) \mathbf{f}_{in}(t) + \mathbf{f}_{in}(t) \tag{5}$$

To better illustrate the behaviors of MTA, we visualize the temporal adaptive weights α_{tp} in different layers (1, 4, 7, 10) for two action classes ('sit down' and 'jump up') of NTU-RGB+D dataset. According to Fig. 4, the MTA module tends to assign higher weights to the frames that are in more important action stages. For example, in the action sequence of $sitdown$, the movement among the 20-th to 50-th frames is more significant than the remaining ones. Correspondingly, these frames obtain higher adaptive weights in our MTA module as expected. We can also see that compared with the early convolution layers, the adaptive weights for different

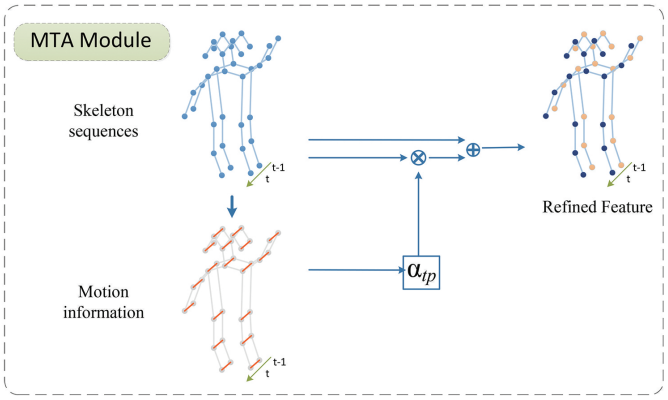


Fig. 2. Illustration of the MTA module. $\otimes$ denotes the element-wise multiplication. $\oplus$ denotes the element-wise addition. If the temporal adaptive weight α_{tp} is not zero, all the feature elements are refined accordingly.

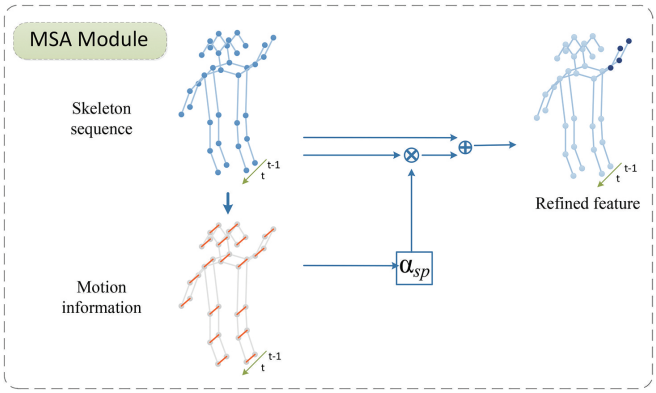


Fig. 3. Illustration of the MSA module. $\otimes$ denotes the element-wise multiplication. $\oplus$ denotes the element-wise addition. Only the features associated with the joints that get non-zero spatial adaptive weights α_{sp} are refined.

frames in the late layers become relatively even, which on the other hand reflects that the skeleton features become stable.

Motion-driven Spatial Adaptation (MSA): The MSA module focuses on the human joints, *i.e.*, the vertices of GCN on the spatial dimension. One may follow the same idea in MTA module and use the average motion of a specific joint in the entire skeleton sequence to generate a unique weight for joint-wise feature refinement. However, such a spatial adaptation strategy has low flexibility and may incorrectly strengthen the joint features in unimportant frames. Therefore, we propose to consider the movement of a joint in a certain time window that does not cover the entire sequence, and accordingly calculate a joint-wise spatial adaptive weight for feature refinement. In this work, the spatial adaptive weight $\alpha_{sp}(n,t)$ for the vertex n at frame t is computed as:

$$\alpha_{sp}(n,t) = \sigma\left(\frac{1}{2\tau \times C}\sum_{t-\tau}^{t+\tau}\sum_{c}^{C}\|\mathbf{P}_{n,t,c}\|_1\right), \tag{6}$$

where τ specifies the number of frames considered for motion calculation, and it is set as 1 by default. The joint-wise adaptive weights for all the frames in a skeleton sequence can be represented by $\alpha_{\mathbf{sp}} \in \mathbb{R}^{N\times T\times 1}$. In MSA module, the vertices corresponding to the joints with vigorous motion will be assigned with higher weights, and thus be enhanced for action recognition. As shown in Fig.3, the features of joint n on each frame are multiplied with $\alpha_{sp}(n)$ in a residual manner for spatial adaptive feature refinement. The output spatial adaptive feature of joint n on frame t can be formulated as follows:

$$\mathbf{f}_{out}(n,t) = \alpha_{sp}(n,t)\mathbf{f}_{in}(n,t) + \mathbf{f}_{in}(n,t) \tag{7}$$

C. Fully Connected Graph Convolution

As introduced in Sec. I and shown in Fig. 5(a), the graph convolution based on the adjacent joints that are physically connected mainly focuses on the local features but ignore, to a large extent, the long-range joint correlation information. Therefore, we propose to use fully connected graph convolution (FCGC) to exploit the long-range joint correlation information, seeking for the global movement pattern of the whole body in an action.

Note that, for a long time, the fully connected layer has been the standard structure of classical feedforward neural networks. Krizhevsky and Hinton *et al*. stated that, compared with the popular CNN networks with a similar scale, the theoretically-best performance of classical feedforward neural networks is likely to be better. However, because of the fully connected structure, these classical feedforward neural networks have excessive parameters that are hard to train, and they are more prone to over-fitting. Consequently, as we all know now, the popular CNN networks with very few or only a single fully connected layer dominate most image/video-based computer vision tasks. However, unlike the image-based CNN models that have dense inputs, there are much fewer vertices in the GCN for skeleton-based action recognition. The numbers of joints in a human skeleton provided by Kinetics-Skeleton [40] and NTU-RGB+D [42] datasets are only 18 and 25, respectively. Therefore, the use of fully connected layers in GCN would not be acute like in CNN models.

As shown in Fig. 5(b), in the spatial FCGC layer, the receptive field is extended to cover all the joints in a spatial skeleton graph, that is, each joint in the next layer is connected to all the joints on the same frame of the former layer. To this aim, we need to revise the adjacency matrix $\mathbf{A}$ in spatial graph convolution to a full-1 matrix. Note that, the extension of graph convolution kernel size will inevitably increase the computational complexity and the use of FCGC may also result in over-fitting or difficulty in training. To avoid these problems, we conduct FCGC only on spatial dimension and use a single layer before the final global average pooling (GAP) in GCN networks, as shown in the proposed network structure illustrated in Subsection III-E.

D. High-Resolution Skeleton Graph

Notice that current GCN models construct graph networks based on the original skeleton data available in skeleton datasets. For example, Kinetics-skeleton [40] offers only 18 body joint coordinates in each frame. The distribution of joints on the body is too sparse and is also very uneven. The action may not be represented accurately enough by the original

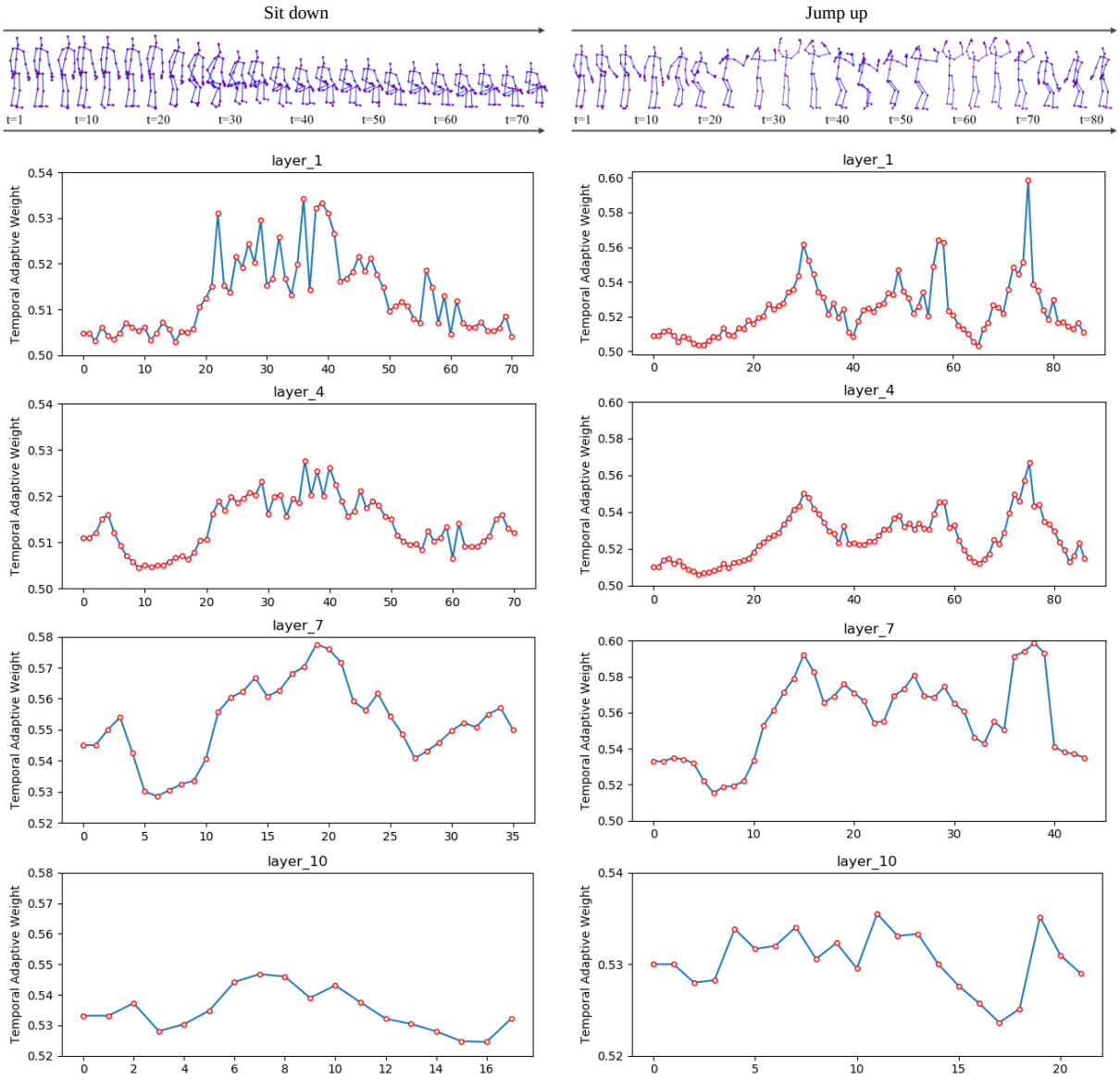


Fig. 4. Visualization of temporal adaptive weights obtained in MTA module for two action sequences ('sit down' and 'jump up') in NTU-RGB+D 60 dataset. The plots in top row are the raw skeleton sequences and the other plots illustrate the temporal adaptive weights obtained in layers 1, 4, 7, and 10.

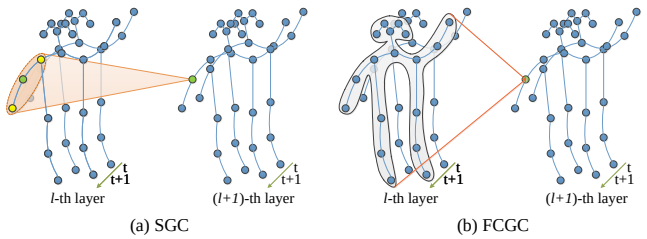
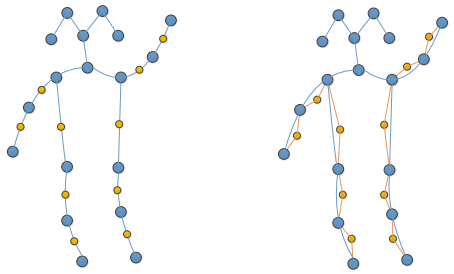


Fig. 5. (a) Spatial graph convolution on a certain neighborhood captures local information. (b)Fully connected spatial graph convolution learns global information from all joints.

skeleton data. The constructed GCN based on it thus has a low resolution and may not be able to effectively extract the discriminative action features.

Recent studies demonstrate that the second-order information (*i.e.*, the direction and length of the bones) is also helpful for improving the action recognition performance [26]–[29]. They created a skeleton bone branch to form a multi-stream GCN model for skeleton-based action recognition. However, the location information of bones has not been considered, which should also contain discriminative information for action recognition.

In this paper, we propose to use a high-resolution skeleton graph (HRG) and exploit the location information of bones. New virtual joints are created in the middle of each pair of physically connected joints. They are inserted into the original spatial-temporal skeleton graph to generate a high-resolution skeleton graph. For example, given a bone with its adjacent joints $v_1 = (x_1, y_1, z_1)$ and $v_2 = (x_2, y_2, z_2)$, the new virtual joint in the middle of the bone is calculated as:



(a)HRG1 (b)HRG2

Fig. 6. Illustration of the high-resolution skeleton graphs. They introduce a virtual node and two edges between each two physical adjacent joints. HRG1 removes the original edges while HRG2 retains them.

$$v_{1,2} = (\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}, \frac{z_1 + z_2}{2}), \tag{8}$$

We explore two different high-resolution skeleton graphs (HRG), namely HRG1 and HRG2. As shown in Fig. 6, HRG1 introduces a virtual node and two edges between two physically adjacent joints and removes their original edges, while HRG2 retains the original edges. The input data $\mathbf{f}_{in} \in \mathbb{R}^{N \times T \times C}$ is then reshaped as a tensor with the shape of $(2N-1) \times T \times C$.

E. Network Architecture

The proposed MSA and MTA modules provide joint-wise and frame-wise weights for spatial and temporal adaptive feature refinements, respectively. As described in III-B, their feature refinement operations are similar to the deep learning-based spatial and temporal attention mechanisms used in [27]. However, it may not be a good choice to use MSA and MTA to refine the features in a serial way as in [27]. In [27], the spatial attention module and temporal attention module are placed in a sequential manner. As a result, the features corresponding to each joint are adjusted twice and some of them could possibly be overly strengthened or weakened. Nevertheless, note that the parameters in the convolution and fully connected layers in both spatial and temporal attention modules are simultaneously learned in an end-to-end manner. Their mutual contradiction in feature refinement could possibly be reconciled by training. The authors in [27] reported better performance of a serial arrangement of spatial and temporal attention modules, compared with the parallel arrangement.

On the other hand, our MSA and MTA select joint-wise and frame-wise weights separately based on the skeleton motion associated with a specific joint or frame. They enjoy a high degree of flexibility but do not have a mechanism to settle the possible overly-adjustment problem. Therefore, we propose to arrange MTA and MSA in a parallel manner by using a two-branch GCN network structure, as shown in Fig. 7. The action prediction scores of two sub-networks are combined by using weighted summation for final classification. This parallel architecture can not only avoid the contradiction between the feature refinements by MTA and MSA, but can also lead to learn more complementary feature representations. The final model with a high-resolution spatial graph and fully connected graph convolution is called MS&TA-HGCN-FC for simplicity.

As shown in Fig. 7, the two spatial and temporal adaptive feature refinement branches of MS&TA-HGCN-FC use the same network constitution. They follow the well-known networks ST-GCN [25] and 2s-AGCN [26] along the temporal dimension except for an additional block of FCGC layer in the end. Both use 10 motion-driven spatial or temporal adaptive graph convolution blocks, namely MTA-GC and MSA-GC. The 10th block using a fully connected graph convolution layer is represented by MTA-FCGC or MSA-FCGC. There is a global average pooling (GAP) layer after the final graph convolution block. Before the 10th graph convolution blocks, a BN layer is used to normalize the input skeleton data. The outputs of GAP are fed into a $Softmax$ classifier to generate prediction scores. Finally, the $Softmax$ scores of two branches are fused using weighted summation to obtain the action scores for classification.

Fig. 8 illustrates a motion-driven adaptive graph convolution block. It is composed of seven layers, including a spatial graph convolution (SGConv) layer or a spatial fully connected graph convolution layer (SFCGC), two batch normalization (BN) layers, two ReLU layers, a motion-driven adaptation module (MTA or MSA), and a temporal graph convolution (TGConv) layer. Meanwhile, a residual connection is added for each block. Note that, the first block does not contain a residual connection. The numbers of output channels are respectively 64, 64, 64, 64, 128, 128, 128, 256, 256, and 256.

F. Multi-stream extensions

The second-order information of skeleton data like the direction and length of the bones, and the motion information of the joints and bones have been demonstrated complementary to the original joint skeleton data in action recognition by many up-to-date methods, such as 2s-AGCN [26], AS-GCN [50], GCN-NAS [48], MS-AAGCN [27], Shift-GCN [29], SEFN [31]. Typically, the joint and bone skeleton sequences and their motion sequences are separately fed into four GCN pathways to form a multi-stream GCN network. The $softmax$ scores of all streams are fused with Sum rule to generate the final prediction.

Inspired by these works, we also extend our MS&TA-HGCN-FC model by using various types of skeleton information and fusing their prediction scores with the weighted Sum rule. We refer a reader to [27] for the detailed definitions of the bone skeleton, joint motion and bone motion. For clarity and simplicity, we use the prefixes Js, Bs, 2s, and 4s to denote the models that use a single joint or bone stream, both the two spatial streams, and all the 4 spatial and motion streams, respectively. For example, the 4s-MS&TA-HGCN-FC model consists of joint, bone, and their corresponding motion streams. Note that, in the motion stream networks, we still use Eq. 4 and Eq. 6 to calculate the frame-wise and joint-wise adaptive weights for motion feature refinement. In this case, the $\mathbf{P}_{n,t,c}$ in these two equations is the motion change between two consecutive frames. Thus, the motion streams actually adopt an adaptive strategy guided by motion change.

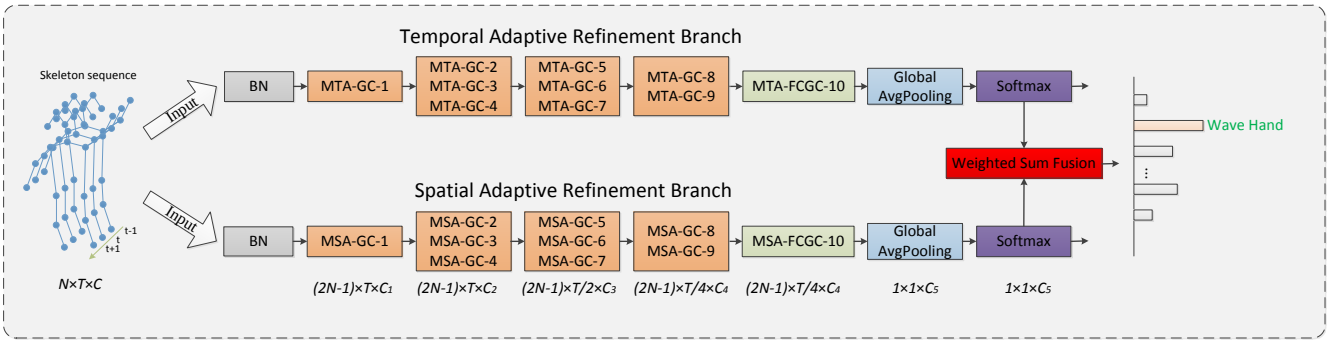


Fig. 7. The two-branch network structure of the proposed MS&TA-HGCN-FC model. $N \times T \times C$ refers to the numbers of joints, frames, and channels. Each branch consists of 10 spatio-temporal graph convolution blocks where the skeleton features are refined by MTA or MSA. The constitution of a graph convolution block is illustrated in Fig. 8. The final prediction is obtained based on the weighted summation of the $Softmax$ scores of two branches.

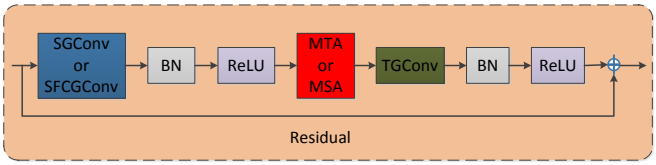


Fig. 8. The structures of spatio-temperal graph convolution blocks. SFCG-Conv refers to the spatial fully connected graph convolution layer. TGConv denotes the temporal graph convolution layer. BN refers to batch normalization. ReLU is the activation function.

IV. Experiments

The experiments are conducted on 4 popular large-scale skeleton datasets, *i.e.*, Kinetics-skeleton [40], NTU-RGB+D 60 [42], NTU-RGB+D 120 [43], and UAV-Human [41], and a small-scale MSR Action 3D dataset [53]. We follow the evaluation convention and report top-1 and top-5 accuracy on Kinetics-Skeleton and report the top-1 accuracy on the others. We will first conduct an ablation study to verify the effectiveness of the proposed MTA, MSA, FCGC, and HRG. They are embedded into the two classical baseline GCN networks, *i.e.*, ST-GCN [25] and AGCN [26], for experimental comparisons. Then, we compare the proposed MS&TA-HGCN-FC method with the SOTA methods presented in the recent 5 years.

A. Datasets

Kinetics-Skeleton: The large-scale Kinetics-Skeleton dataset was built based on the Kinetics human action dataset [40] that contains 300,000 video clips collected from YouTube. Yan. *et al*. [25] extracted the 2D locations of 18 joints each person frame by frame for each clip using the publicly available OpenPose toolbox [54]. The released dataset has 400 classes of action sequences and is divided into a training set with 240,000 skeleton sequences and an evaluation set with 20,000 skeleton sequences.

NTU-RGB+D 60: NTU-RGB+D 60 dataset consists of 56,880 video clips of 60 action classes captured from 40 subjects. It provides 3D locations of 25 human body joints each frame detected by three Kinect depth sensors. Two benchmarks are defined as Cross-Subject (CS, or X-sub) and Cross-View (CV, or X-view). In CS, the 40,320 samples from 20 subjects are used for training and the 16,560 samples from the remaining 20 subjects are used for testing. In CV, the training set has 37,920 samples captured by sensors 2 & 3, and the evaluation set has 18,960 samples captured by sensor 1.

NTU-RGB+D 120: This is an extended version of NTU-RGB+D 60 dataset. It contains totally 114,480 samples over 120 classes captured from 106 distinct subjects and 155 viewpoints. The CS benchmark is extended with 63,026 and 50,922 samples foo training and testing, respectively. The CV benchmark has 54,471 samples for training and 59,477 samples for evaluation.

UAV-Human: UAV-Human is a large-scale dataset for human behavior understanding with unmanned aerial vehicles (UAVs). The dataset was collected by a flying UAV in multiple urban and rural districts in both daytime and nighttime, hence covering extensive diversities w.r.t subjects, backgrounds, illuminations, weathers, occlusions, camera motions, and UAV flying attitudes. It contains 67,428 video sequences of 155 action classes captured from 119 subjects. 17 body joints are annotated for each skeleton. Two CS evaluation benchmarks are defined, namely CSv1 and CSv2. For each benchmark, 89 subjects are used for training and 30 subjects are used for testing.

MSR Action 3D: The MSR Action 3D dataset [53] consists of 557 valid depth sequences of 20 action classes captured from 10 subjects. Each action was performed three times by every subject. In the experiment, we follow the recent literature [23], [55], [56] and use the cross-subject evaluation protocol where half of the subjects are used for training and the rest are used for testing. We use the pretrained GCN models on NTU-RGB+D datasets to funtune for evaluation because of the insufficient training samples of this dataset.

B. Training details

All experiments are conducted on the PyTorch [57] deep learning platform by using two NVIDIA RTX 2080Ti GPUs. Stochastic gradient descent (SGD) with Nesterov momentum is applied for optimization [58], [59]. Cross-entropy is used as the loss function. The Nesterov momentum and weight decay

are set as 0.9 and 0.0001, respectively. We follow the same data preprocessing and hyperparameters as in ST-GCN [25] and 2s-AGCN [26] except the batch size. The batch size is set as 32 except that it is set as 64 on Kinetics-Skeleton. For Kinetics-Skeleton, the learning rate is set as 0.1 and is divided by 10 at the 45-th and 55-th epochs. The training process is completed at the 65-th epoch. For the other datasets, the learning rate is set as 0.1 and is divided by 10 at the 30-th and 40-th epochs. The training process is completed at the 50-th epoch.

C. Ablation Study

In ablation study, two classical models, *i.e.*, ST-GCN [25] and AGCN [26], and their variants are used as the baseline method. We embed our proposals, *i.e.*, MTA, MSA, FCGC, and HRG, into these baseline networks in the same way as used in our proposed model. The derived new models are denoted by the combination of the names of the original model and our proposals. For example, MTA-ST-GCN denotes the ST-GCN model that uses MTA.

1) Motion-driven Adaptation: In this subsection, we compare the ST-GCN and AGCN models with their extensions using the proposed MTA and MSA modules. The experimental results on NTU-RGB+D 60 dataset are summarized in Table I. It is obvious that all the methods with MTA and MSA modules are superior to their original methods. The MTA-ST-GCN outperforms ST-GCN by 3.46% and 2.12% in terms of the top-1 accuracy on CS and CV benchmarks, respectively. Compared with MTA, MSA brings about less significant improvements for ST-GCN model. However, for the AGCN and its variants, MSA can lead to better performance than MTA on CS benchmark, and they are comparable on CV benchmark.

TABLE I
COMPARISONS OF THE MODELS WITH OR WITHOUT MTA AND MSA ON THE NTU-RGB+D 60 DATASET.

Method	CS(%)	CV(%)
ST-GCN [25]	81.5	88.3
Js-AGCN [26]	86.1	93.7
Bs-AGCN [26]	86.9	93.2
2s-AGCN [26]	88.5	95.1
MTA-ST-GCN	84.96(↑3.46)	90.42(↑2.12)
MSA-ST-GCN	83.28(↑1.78)	90.26(↑1.96)
Js-MTA-AGCN	86.7(↑0.6)	94.02(↑0.32)
Bs-MTA-AGCN	87.26(↑0.36)	93.93(↑0.73)
2s-MTA-AGCN	88.85(↑0.35)	95.40(↑0.3)
Js-MSA-AGCN	86.82(↑0.72)	94.01(↑0.31)
Bs-MSA-AGCN	87.4(↑0.5)	93.84(↑0.64)
2s-MSA-AGCN	89.06(↑0.56)	95.34(↑0.24)

Note: The value in parentheses is the accuracy change brought about by the MTA or MSA module. '↑' means the value is an increase.

TABLE II
COMPARISONS OF THE MODELS WITH OR WITHOUT MTA AND MSA ON THE KINETICS-SKELETON DATASET.

Method	Top-1(%)	Top-5(%)
ST-GCN [25]	30.7	52.8
Js-AGCN [26]	35.1	57.1
Bs-AGCN [26]	33.3	55.7
2s-AGCN [26]	36.1	58.7
MTA-ST-GCN	33.11(↑2.41)	55.61(↑2.81)
MSA-ST-GCN	32.56(↑1.86)	55.01(↑2.21)
Js-MTA-AGCN	35.36(↑0.26)	57.78(↑0.68)
Bs-MTA-AGCN	34.43(↑1.13)	58.52(↑2.82)
2s-MTA-AGCN	36.53(↑0.43)	59.7(↑1.0)
Js-MSA-AGCN	35.54(↑0.44)	57.46(↑0.36)
Bs-MSA-AGCN	34.28(↑0.98)	57.03(↑1.33)
2s-MSA-AGCN	36.46(↑0.36)	59.81(↑1.11)

On the Kinetics-Skeleton dataset, as shown in Table II, the MTA-ST-GCN and MSA-ST-GCN are also superior to ST-GCN by a large margin. Both modules can also bring about significant improvements for AGCN and its variants. Although when AGCN uses only joint or bone stream, MSA shows inferior progress than MTA, they are comparable when combining both joint and bone streams. Overall, both MTA and MSA modules can significantly boost the performance of ST-GCN and AGCN.

We also compare our motion-driven adaptation with deep learning attention mechanisms by using Js-AGCN as baseline on NTU-RGB+D 60 dataset. The AAGCN model is an extension of AGCN by adding a spatial-temporal-channel (STC) attention module in each graph convolution block [27]. Note that, in [27], the authors reported higher accuracy of Js-AGCN by using some data preprocessing strategies and a learning rate scheduler. Hence, we directly use their ablation study results of STC attention module for our comparison. As shown in Table III, compared with Js-AGCN, the Js-AAGCN obtains improvements of 0.6% and 0.7% on CS and CV benchmarks, respectively. On the other hand, compared with the 86.1% obtained by Js-AGCN in our experiments on CS benchmark, our Js-MTA-AGCN and Js-MSA-AGCN improve the performance by 0.6% and 0.72%. The combination of MTA and MSA in two-branch architecture can lead to an increase of 2.38%, and the derived Js-MS&TA-AGCN outperforms Js-AAGCN by 0.48%. On the CV benchmark, the improvements obtained separately by MTA and MSA are not evident, however, Js-MS&TA-AGCN improves the performance to 95.26%, which is higher than that obtained by Js-AAGCN. Therefore, our spatial and temporal motion-driven adaptation methods and their parallel utilization show better performance than the spatial-temporal-channel attention mechanisms used in AAGCN.

What is more, since our adaption modules are driven online by the skeleton motion estimated from the input action sequences, one may be concerned about their reliability against the noise in skeleton data. Note that, in Eq. 4 and Eq. 6, the frame-wise and joint-wise adaptive weights are calculated based on the average motion of multiple joints. Thus, theoretically, the noise caused by one or a small subset of joints will be suppressed to a certain extent in frame-wise weight calculation. Likewise, the noise in some frames would not affect our joint-wise weight calculation much. To study the reliability of motion-driven adaptation methods, *i.e.*, MTA and MSA, we add 5%, 10%, 15%, or 20% random disturbance on the average motion used for the weights calculation in Eq. 4 and Eq. 6. Then, we evaluate the performance of the Js-MTA-AGCN and the Js-MSA-AGCN using these degenerated adaptation modules on CS benchmark of NTU-RGB+D 60 dataset. The results shown in Table IV demonstrate that the 5% random disturbance has little effect on the recognition

TABLE III
COMPARISONS OF THE AGCN MODELS WITH ATTENTION MECHANISMS OR OUR MOTION-DRIVEN ADAPTATION MODULES ON THE NTU-RGB+D 60 DATASET.

Method	CS(%)	CV(%)
Js-AGCN [27]	87.4	94.4
Js-AAGCN [27]	88(↑0.6)	95.1(↑0.7)
Js-AGCN [26]	86.1	93.7
Js-MTA-AGCN	86.7(↑0.6)	94.02(↑0.32)
Js-MSA-AGCN	86.82(↑0.72)	94.01(↑0.31)
Js-MS&TA-AGCN	88.48(↑2.38)	95.26(↑1.56)

TABLE IV
RANDOM DISTURBANCE EXPERIMENTS OF OUR MOTION-DRIVEN ADAPTATION MODULES ON THE CS BENCHMARK OF NTU-RGB+D 60 DATASET.

Method	0%	5%	10%	15%	20%
Js-MTA-AGCN	86.7	86.62	86.55	86.38	86.17
Js-MSA-AGCN	86.82	86.88	86.6	86.47	86.28

based on motion-driven adaptation. As the random disturbance intensifies, the effect gradually appears, but Js-MTA-AGCN and Js-MSA-AGCN can always maintain better performance than the Js-AGCN without motion-driven adaptation.

2) Fully Connected Graph Convolution: We replace the spatial graph convolution layer of the last graph convolution block of ST-GCN and AGCN networks with a spatial FCGC layer. We use FC as suffix to denote the new models. As shown in Table V, compared with ST-GCN, the ST-GCN-FC achieves significant improvements of 2.71% and 1.77% on CS and CV benchmarks. As for the AGCN networks, the use of FCGC layer can improve the accuracy by at least 0.4% on CS benchmark. On CV benchmark, FCGC leads to the best improvement of 0.78% when using bone stream.

Moreover, to examine the validity of using MTA and FCGC together, we put MTA-ST-GCN-FC and MTA-AGCN-FC in comparison and list their results in Table V. It can be seen that on CS benchmark, MTA-ST-GCN-FC achieves 85.67%, which is clearly superior to the 84.21% of ST-GCN-FC, and the 84.96% of MTA-ST-GCN in Table I. The superiority of MTA-ST-GCN-FC is also very evident on CV benchmark. As for the AGCN models, MTA-AGCN-FC consistently outperforms its counterparts that use only MTA or FCGC. These results validate that the simultaneous usage of MTA and FCGC can further improve the action recognition performance.

3) High-Resolution Skeleton Graph: We improve the spatial resolution of the baseline methods according to the constructions of HRG1 and HRG2, respectively. The new models are denoted with HRG1 or HRG2 like ST-GCN(HRG1). Note that, the joint and bone skeleton input data must be augmented by creating virtual joints according to Eq. 8 before being fed into the new networks. Our experimental results are shown in Table VI. Apparently, the models with HRG can bring significant improvement. Compared with the standard ST-GCN, ST-GCN(HRG1) gains the improvements of 2.81% and 1.75% on CS and CV benchmarks, while ST-GCN(HRG2) brings better improvements of 3.25% and 2.06%. All the AGCN variants get clear accuracy increases when applying HRG1 or HRG2. We can also see that the models with HRG2 are slightly better than the models with HRG1 in all the cases.

TABLE V
COMPARISONS OF THE MODELS WITH OR WITHOUT FCGC ON THE NTU-RGB+D 60 DATASET.

Method	CS(%)	CV(%)
ST-GCN [25]	81.5	88.3
Js-AGCN [26]	86.1	93.7
Bs-AGCN [26]	86.9	93.2
2s-AGCN [26]	88.5	95.1
ST-GCN-FC	84.21(↑2.71)	90.07(↑1.77)
Js-AGCN-FC	86.54(↑0.44)	94.13(↑0.43)
Bs-AGCN-FC	87.31(↑0.41)	93.98(↑0.78)
2s-AGCN-FC	88.92(↑0.42)	95.32(↑0.22)
MTA-ST-GCN-FC	85.67(↑4.17)	91.17(↑2.87)
Js-MTA-AGCN-FC	87.30(↑1.2)	94.61(↑0.91)
Bs-MTA-AGCN-FC	87.61(↑0.71)	94.33(↑1.13)
2s-MTA-AGCN-FC	89.51(↑1.01)	95.67(↑0.57)

TABLE VI
COMPARISONS OF THE MODELS WITH OR WITHOUT HRG ON THE NTU-RGB+D 60 DATASET.

Method	CS(%)	CV(%)
ST-GCN [25]	81.5	88.3
Js-AGCN [26]	86.1	93.7
Bs-AGCN [26]	86.9	93.2
2s-AGCN [26]	88.5	95.1
ST-GCN(HRG1)	84.31(↑2.81)	90.05(↑1.75)
Js-AGCN(HRG1)	86.73(↑0.63)	94.27(↑0.57)
Bs-AGCN(HRG1)	87.49(↑0.59)	94.03(↑0.83)
2s-AGCN(HRG1)	89.06(↑0.56)	95.38(↑0.28)
ST-GCN(HRG2)	84.75 (↑3.25)	90.36(↑2.06)
Js-AGCN(HRG2)	86.78(↑0.68)	94.12(↑0.42)
Bs-AGCN(HRG2)	87.54(↑0.64)	94.13(↑0.93)
2s-AGCN(HRG2)	89.11(↑0.61)	95.41(↑0.31)

D. Comparisons with the state-of-the-art methods

Given the superiority of HRG2 over HRG1, we select HRG2 to construct our final model. We compare the proposed 2s-MS&TA-HGCN-FC and 4s-MS&TA-HGCN-FC with more than 40 SOTA methods on five skeleton-based action recognition datasets. We would like to note that from the perspectives of fairness and generalization, except for a few classical models, most of the competitors are those who have ever been evaluated on the Kinetics-skeleton that has the largest number of action classes and at least one NTU-RGB+D dataset in the existing literature. Also, it should be noted that UAV-Human is a newly released dataset, we use all the 10 methods that have been evaluated on this dataset as competitors.

On the Kinetics-Skeleton dataset, the action recognition results in terms of top-1 accuracy and top-5 accuracy are summarized in Table VII. It is obvious that our 4s-MS&TA-HGCN-FC evidently outperforms all the competitors on both evaluation measures. The methods with higher top-1 accuracy can be divided into two groups. The best group achieves a top-1 accuracy between 38.5% and 38.7%. Our 4s-MS&TA-HGCN-FC achieves the best of 38.7%, followed by 2s-AAGCN+TEM [47] and 2s-MS&TA-HGCN-FC. The second group includes the well-known MS-AAGCN [27], MS-G3D [28], Dynamic GCN [30], etc. They achieve the top-1 accuracy between 37.7% and 38%. By comparing the two-stream and four-stream methods based on the same model, it

TABLE VII
COMPARISONS OF THE ACTION RECOGNITION ACCURACY WITH STATE-OF-THE-ART METHODS ON THE KINETICS-SKELETON DATASET.

Method	Year	Top-1(%)	Top-5(%)
ST-GCN [25]	2018	30.7	52.8
STGR-GCN [60]	2019	33.6	56.1
AS-GCN [50]	2019	34.8	56.5
2s-AGCN [26]	2019	36.1	58.7
DGNN [39]	2019	36.9	59.6
BAGCN [61]	2019	37.3	60.2
L-CAGCN [62]	2020	33.3	55.4
A-CAGCN [62]	2020	34.1	56.6
GCN-NAS [48]	2020	37.1	60.1
2s-AAGCN [27]	2020	37.4	60.4
CGCN [63]	2020	37.5	60.4
MS-AAGCN [27]	2020	37.8	61.0
Dynamic GCN [30]	2020	37.9	61.3
MS-G3D [28]	2020	38	60.9
MS-AAGCN+TEM [47]	2020	38	61.4
2s-AAGCN+TEM [47]	2020	38.6	61.6
PR-GCN [64]	2021	33.7	55.8
ST-TR [36]	2021	37	59.7
SEFN(Att) [31]	2021	37.7	N/A
SEFN(Base) [31]	2021	37.8	N/A
ST-TR-agcn [36]	2021	38	60.5
PB-GCN [65]	2022	30.9	52.8
PeGCN [66]	2022	34.8	57.2
Graph2Net [32]	2022	36.8	N/A
EGCN [67]	2022	37.1	59.7
Sybio-GNN [33]	2022	37.2	58.1
TE-GCN [68]	2022	37.5	59.7
2s-MS&TA-HGCN-FC(ours)		38.5	61.7
4s-MS&TA-HGCN-FC(ours)		**38.7**	**62.3**

TABLE VIII
COMPARISONS OF THE ACTION RECOGNITION ACCURACY WITH STATE-OF-THE-ART METHODS ON THE NTU-RGB+D 60 DATASET.

Method	Year	CS(%)	CV(%)
GCA-LSTM [17]	2018	74.3	82.8
ST-GCN [25]	2018	81.5	88.3
SR-TSL [15]	2018	84.8	92.4
HCN [69]	2018	86.5	91.1
DPRL+GCNN [35]	2018	83.5	89.8
AS-GCN [50]	2019	86.8	94.2
AGC-LSTM [70]	2019	89.2	95.0
2s-AGCN [26]	2019	88.5	95.1
DGNN [39]	2019	89.9	96.1
BAGCN [61]	2019	90.3	96.3
STGR-GCN [60]	2019	86.9	92.3
TS-SAN [71]	2020	87.2	92.7
2s-AAGCN+TEM [47]	2020	88.7	95.8
SGN [72]	2020	89.0	94.5
2s-Shift-GCN [29]	2020	89.7	96
GCN-NAS [48]	2020	89.4	95.7
2s-AAGCN [27]	2020	89.4	96.0
MS-AAGCN [27]	2020	90.0	96.2
CGCN [63]	2020	90.3	96.4
4s-Shift-GCN [29]	2020	90.7	**96.5**
MS-AAGCN+TEM [47]	2020	91	**96.5**
Dynamic GCN [30]	2020	**91.5**	96
MS-G3D [28]	2020	**91.5**	96.2
PR-GCN [64]	2021	85.2	91.7
RA-GCN [49]	2021	87.3	93.6
SEFN(Base) [31]	2021	89.2	95.8
ST-TR [36]	2021	89.9	96.1
SEFN(Att) [31]	2021	90.2	96.1
ST-TR-agcn [36]	2021	90.3	96.3
MSTGNN [73]	2021	91.3	95.5
PB-GCN [65]	2022	83.8	91.3
PeGCN [66]	2022	85.6	93.4
LAGA [51]	2022	87.1	93.2
TE-GCN [68]	2022	88.7	95.4
EGCN [67]	2022	89.1	95.5
Graph2Net [32]	2022	90.1	96
Sybio-GNN [33]	2022	90.1	95.4
CD-GCN [34]	2022	90.1	**96.5**
Ta-CNN [24]	2022	90.7	95.1
FR-AGCN [52]	2022	90.5	95.8
2s-MS&TA-HGCN-FC(ours)		90.2	96.1
4s-MS&TA-HGCN-FC(ours)		90.8	96.4

can be seen that the additional motion streams cannot bring evident improvement to the Kinetics-Skeleton dataset. Even, the MS-AAGCN+TEM is inferior to 2s-AAGCN+TEM by 0.6% top-1 accuracy.

On the NTU-RGB+D 60 dataset, our proposed models are compared with more than 40 competitors, their top-1 accuracy results on the two benchmarks are collected in Table VIII. Our 4s-MS&TA-HGCN-FC achieves 90.8% and 96.4% on CS and CV benchmarks, respectively. Although they are not the best on any benchmark, no competitor can surpass our 4s-MS&TA-HGCN-FC on both benchmarks except the MS-AAGCN+TEM [47]. MS-AAGCN+TEM is slightly better than 4s-MS&TA-HGCN-FC by 0.2% and 0.1% on CS and CV, respectively, but it is evidently inferior to 4s-MS&TA-HGCN-FC. CD-GCN [34] and CGCN [63] are slightly superior or comparable to 4s-MS&TA-HGCN-FC on CV benchmark, but they are evidently inferior to our model on CS benchmark. Although MS-G3D [28] and Dynamic GCN [30] achieve the best result of 91.5% on CS benchmark, they are not only worse than 4s-MS&TA-HGCN-FC on CV benchmark but also inferior to our model by around 0.6% on Kinetics-Skeleton.

On the NTU-RGB+D 120 dataset, as shown in Table IX, MSTGNN [73] and Dynamic GCN [30] obtain the best top-1 accuracy on one of the CS and CV benchmarks, which are 87.4% and 88.6%, respectively. Our 4s-MS&TA-HGCN-FC achieves 87% and 88.4% on CS and CV benchmarks, respectively. 4s-MS&TA-HGCN-FC outperforms MSTGNN on CV benchmark. Dynamic GCN is the only one better than 4s-MS&TA-HGCN-FC on both benchmarks, but it is much worse than ours on Kinetics-Skeleton. 4s-MS&TA-HGCN-FC is comparable to MS-G3D [28] according to their results on both benchmarks.

The UAV-Human is a new large-scale dataset that is much more challenging than the ground camera-based datasets like NTU-RGB+D 60&120 because of the unique viewpoints, movements, motion blurs, and resolution changes caused by the flying UAV. As shown in Table X, the proposed 4s-MS&TA-HGCN-FC achieves the best results of 45.72% and 71.84% on both CSv1 and CSv2 benchmarks. The well-known MS-G3D obtains comparable results with FR-AGCN on CSv1, which is evidently inferior to ours. On CSv2 benchmark, our model outperforms the second-best method, FR-AGCN, by a large margin of 2.34%.

The MSR Action 3D dataset is a very small dataset. To the best of our knowledge, so far, no GCN-based method has been evaluated on it. We compare our methods with two publicly available classical GCN-based methods (*i.e.*, ST-GCN [25] and 2s-AGCN [26]), and the SOTA CNN-based methods (*e.g.*, RIAC-LSTM [23], SPMFs [78]), point cloud-based methods (*e.g.*, SequentialPointNet [56], P4Transformer [55]), and

TABLE IX
COMPARISONS OF THE ACTION RECOGNITION ACCURACY WITH STATE-OF-THE-ART METHODS ON THE NTU-RGB+D 120 DATASET.

Method	Year	CS(%)	CV(%)
GCA-LSTM [17]	2018	61.2	63.3
RotClips+MTCNN [74]	2018	62.2	61.8
BPEM [75]	2018	64.6	66.9
ST-GCN [25]	2018	70.7	73.2
SR-TSL [15]	2018	74.1	79.9
TSRJI [76]	2019	67.9	62.8
2s-AGCN [26]	2019	82.9	84.9
SGN [72]	2020	79.2	81.5
2s-Shift-GCN [29]	2020	85.3	86.6
4s-Shift-GCN [29]	2020	85.9	87.6
MS-G3D [28]	2020	86.9	88.4
Dynamic GCN [30]	2020	87.3	**88.6**
RA-GCN [49]	2021	81.1	82.7
ST-TR [36]	2021	84.3	86.7
ST-TR-agcn [36]	2021	85.1	87.1
SEFN [31]	2021	86.2	87.8
MSTGNN [73]	2021	**87.4**	87.6
LAGA [51]	2022	81	82.2
Ta-CNN [24]	2022	85.7	87.3
Graph2Net [32]	2022	86	87.6
CD-GCN [34]	2022	86.3	87.8
FR-AGCN [52]	2022	86.6	87
2s-MS&TA-HGCN-FC(ours)		86.3	87.7
4s-MS&TA-HGCN-FC(ours)		87	88.4

TABLE X
COMPARISONS OF THE ACTION RECOGNITION ACCURACY WITH STATE-OF-THE-ART METHODS ON THE UAV-HUMAN DATASET.

Method	Year	CSv1(%)	CSv2(%)
ST-GCN [25]	2018	30.25	56.14
DGNN [39]	2019	29.9	N/A
2s-AGCN [26]	2019	34.84	66.68
HARD-Net [77]	2020	36.97	N/A
4s-Shift-GCN [29]	2020	37.98	67.04
AAGCN [27]	2020	41.43	N/A
MS-G3D [28]	2020	43.94	N/A
PB-GCN [65]	2022	37.48	N/A
TE-GCN [68]	2022	42.5	68.2
FR-AGCN [52]	2022	43.98	69.5
2s-MS&TA-HGCN-FC(ours)		44.33	70.69
4s-MS&TA-HGCN-FC(ours)		**45.72**	**71.84**

TABLE XI
COMPARISONS OF THE ACTION RECOGNITION ACCURACY WITH STATE-OF-THE-ART METHODS ON THE MSR ACTION 3D DATASET.

Method	Year	Accuracy(%)
ST-GCN [25]	2018	83.27
SPMFs [78]	2018	98.05
2s-AGCN [26]	2019	88.36
MeteorNet [79]	2019	88.5
UnifiedDeep [80]	2019	97.98
Movement polygon [13]	2020	94.13
P4Transformer [55]	2021	90.94
PSTNet [81]	2021	91.2
MMDNN [82]	2021	91.94
RIAC-LSTM [23]	2021	**98.06**
Complex Network+LSTM [83]	2022	90.7
SequentialPointNet [56]	2022	91.94
2s-MS&TA-HGCN-FC(ours)		90.54
4s-MS&TA-HGCN-FC(ours)		92.73

depth-based methods (*e.g.*, MMDNN [82]). The experimental results summarized in Table XI demonstrate that the proposed 4s-MS&TA-HGCN-FC is superior to the SOTA point cloud-based models and depth-based models, as well as ST-GCN and 2s-AGCN. As also can be seen, the GCN methods including 4s-MS&TA-HGCN-FC are much worse than the SOTA CNN-based models using pseudo images. This can be attributed to the insufficient training samples on the small-scale dataset. Note that the other deep learning methods can take advantage of data augmentation skills like random rotation and jittering operation to generate more samples for training. However, GCN methods generally conduct data preprocessing like 3D rotation to make the view angles as similar as possible. And, the jittering operation on some body joints of the 18 or 25 joints on a skeleton will also break the inherent spatio-temporal correlations of a skeleton sequence. As far as we know, there is no GCN method using data augmentation in the existing literature.

According to the above observations and analyses on the five skeleton datasets, we summarize the experimental results as follows. On the Kinetics-Skeleton dataset with the largest scale, our proposed 4s-MS&TA-HGCN-FC outperforms all the state-of-the-art competitors. On the NTU-RGB+D 60 dataset, although MS-AAGCN+TEM [47] is slightly better than 4s-MS&TA-HGCN-FC, its performance on Kinetics-Skeleton is worse than 4s-MS&TA-HGCN-FC by a large margin. The Dynamic GCN [30] achieves better results on NTU-RGB+D 120 dataset, but it is evidently inferior to 4s-MS&TA-HGCN-FC on the CV benchmark of NTU-RGB+D 60 and the Kinetics-Skeleton. On the two NTU-RGB+D datasets, there are also a few competitors that outperform 4s-MS&TA-HGCN-FC on one or two of the four benchmarks. However, it is not enough to show that they are better than 4s-MS&TA-HGCN-FC. On the more challenging UAV-Human dataset, the proposed 4s-MS&TA-HGCN-FC outperforms all SOTA methods by a large margin on both CSv1 and CSv2 benchmarks. On the small-scale MSR Action 3D dataset, the proposed 4s-MS&TA-HGCN-FC is superior to the SOTA point cloud-based models and depth-based models, as well as the classical ST-GCN and 2s-AGCN.

In addition, we visualize a part of each confusion matrix obtained by our proposed 4s-MS&TA-HGCN-FC and the classical 2s-AGCN on NTU-RGB+D 60 and Kinetics-Skeleton datasets in Fig. 9. The labels on Y-axis are true labels and the labels on X-axis are predicted labels. Note that, the 2s-AGCN model is publicly available and the results can be reproducible. We follow the literature [31] and select the same action classes for confusion matrix illustration. We can see that our 4s-MS&TA-HGCN-FC significantly outperforms 2s-AGCN on all the action classes without exception. On the CS benchmark, 4s-MS&TA-HGCN-FC exceeds 2s-AGCN between 5% and 8% on all the classes. On the CV benchmark, the improvements achieved by 4s-MS&TA-HGCN-FC become less evident on most classes, but it is up to 10.7% on class 'play with phone'. On the more challenging Kinetics-Skeleton dataset, the superiority of 4s-MS&TA-HGCN-FC is more significant. It exceeds 2s-AGCN 20.5% on class 'getting a tattoo', and at least 13% on all the other classes. Compared

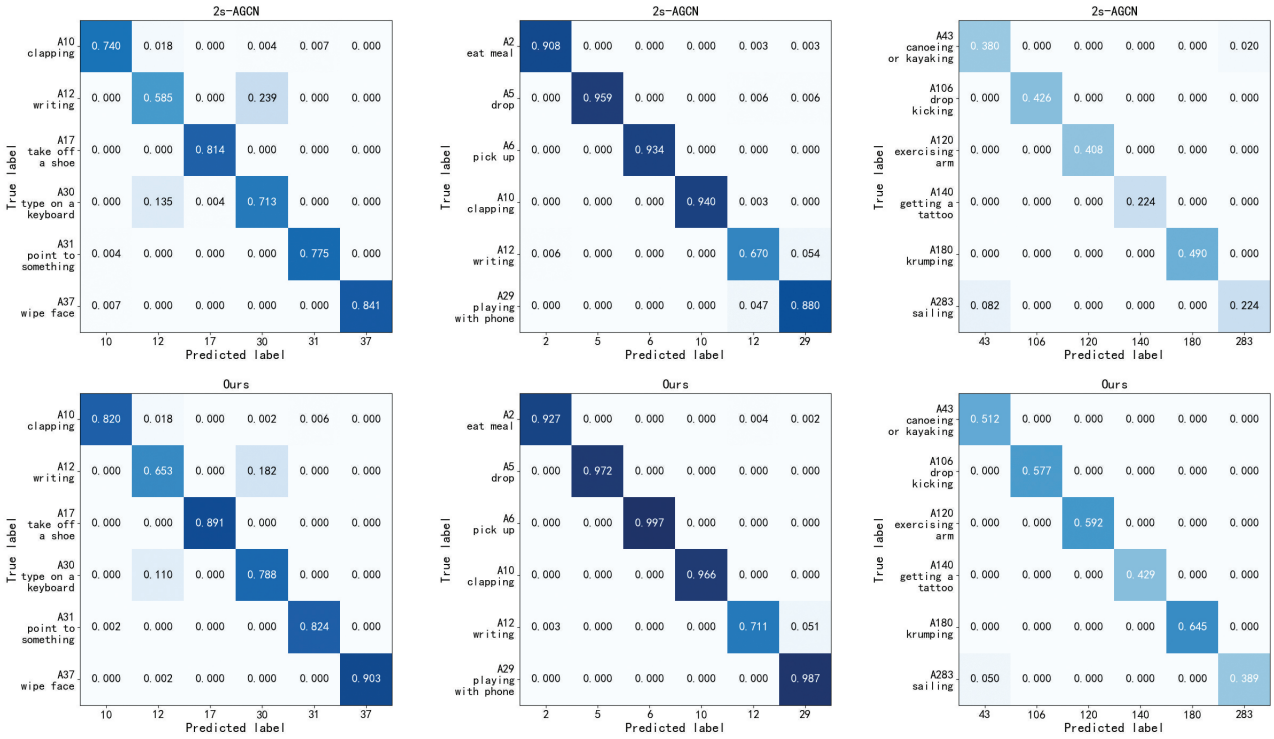


Fig. 9. The confusion matrices obtained by 2s-AGCN and 4s-MS&TA-GCN-FC on the CS (left column) and CV (middle column) benchmark of NTU-RGB+D 60 dataset and Kinetics-Skeleton dataset (right column).

the confusion matrices shown in [31], it can also be observed that our 4s-MS&TA-HGCN-FC is superior to the SEFN model on all the action classes illustrated in Fig. 9.

V. Conclusion

In this work, we propose a novel motion-driven spatial and temporal adaptive high-resolution graph convolutional network for skeleton-based action recognition, namely MS&TA-HGCN-FC. In each graph convolution block, we dynamically calculate frame-wise and joint-wise adaptive weights based on skeleton motion in order to strengthen the features of important frames and joints. Unlike the deep learning-based attention mechanisms whose parameters are learned offline and fixed in operation, our motion-driven adaptation is simpler and highly flexible. We decouple and combine the spatial and temporal refinements by using a two-branch GCN network structure. Such a parallel structure can also lead to learn more complementary feature representations. Moreover, we propose to use the fully connected graph convolution to learn the long-range joint dependency information and use a high-resolution graph with virtual joints to improve the representation of skeleton features. Our three proposals can be readily embedded into most off-the-shelf GCN networks. The ablation study based on two classical baseline models validate their effectiveness. Finally, we extend our model to be a multi-stream network that includes joints, bones, and their motion information. Extensive experimental results demonstrate that the proposed model outperforms the state-of-the-art methods on the challenging large-scale Kinetics-Skeleton and UAV-Human datasets, and it is on par with them on the two NTU-RGB+D 60&120 datasets. Given the encouraging performance, simplicity and flexibility, the proposed motion-driven adaptation can be expected to complement the deep learning-based attention mechanisms for better action feature learning and classification.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Grants 61602390, 61902153, 61860206007, U19A2071), the Xihua University Funds for Young Scholars, and the funding from Sichuan University under Grant 2020SCUNG205.

References

[1] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human action recognition from various data modalities: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2022.

[2] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *International Journal of Computer Vision*, vol. 130, no. 5, pp. 1366–1401, 2022.

[3] C. Dhiman and D. K. Vishwakarma, "A review of state-of-the-art techniques for abnormal human activity recognition," *Engineering Applications of Artificial Intelligence*, vol. 77, pp. 21–45, 2019.

[4] C. Dhiman and D. Vishwakarma, "A robust framework for abnormal human action recognition using r-transform and zernike moments in depth videos," *IEEE Sensors Journal*, vol. 19, no. 13, pp. 5195–5203, 2019.

[5] D. K. Vishwakarma and R. Kapoor, "An efficient interpretation of hand gestures to control smart interactive television," *International Journal of Computational Vision and Robotics*, vol. 7, no. 4, p. 454, 2017.

[6] D. K. Vishwakarma, R. Kapoor, and A. Dhiman, "A proposed unified framework for the recognition of human activity by exploiting the characteristics of action dynamics," *Robotics and Autonomous Systems*, vol. 77, pp. 25–38, 2016.

[7] D. Vishwakarma, R. Kapoor, and A. Dhiman, "Unified framework for human activity recognition: An approach using spatial edge distribution and r-transform," *AEU - International Journal of Electronics and Communications*, vol. 70, no. 3, pp. 341–353, 2016.

[8] D. Vishwakarma and D. Chhavi, "A unified model for human activity recognition using spatial distribution of gradients and difference of gaussian kernel," *Visual Computer*, pp. 1–19, 2018.

[9] D. K. Vishwakarma and T. Singh, "A visual cognizance based multi-resolution descriptor for human action recognition using key pose," *AEU - International Journal of Electronics and Communications*, vol. 107, pp. 157–169, 2019.

[10] D. K. Vishwakarma, "A two-fold transformation model for human action recognition using decisive pose," *Cognitive Systems Research*, vol. 61, pp. 1–13, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389041719305224

[11] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 588–595.

[12] Z. Shao and Y. Li, "Integral invariants for space motion trajectory matching and recognition," *Pattern Recognition*, vol. 48, no. 8, pp. 2418–2432, 2015.

[13] K. Vishwakarma, Dineshand Jain, "Three-dimensional human activity recognition by forming a movement polygon using posture skeletal data from depth sensor," *ETRI Journal*, vol. 44, no. 2, pp. 286–299, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.2020-0101

[14] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1110–1118.

[15] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, "Skeleton-based action recognition with spatial reasoning and temporal stack learning," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 106–121.

[16] 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 *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2136–2145.

[17] J. Liu, G. Wang, L. Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention lstm networks," *IEEE Transactions on Image Processing*, vol. 27, no. 99, pp. 1586–1599, 2018.

[18] Z. Fan, X. Zhao, T. Lin, and H. Su, "Attention-based multiview re-observation fusion network for skeletal action recognition," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 363–374, 2019.

[19] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4570–4579.

[20] Y. Hou, Z. Li, P. Wang, and W. Li, "Skeleton optical spectra-based action recognition using convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 3, pp. 807–811, 2018.

[21] P. Wang, W. Li, C. Li, and Y. Hou, "Action recognition based on joint trajectory maps with convolutional neural networks," *Knowledge-Based Systems*, vol. 158, pp. 43–53, 2018.

[22] A. Banerjee, P. K. Singh, and R. Sarkar, "Fuzzy integral-based cnn classifier fusion for 3d skeleton action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 6, pp. 2206–2216, 2021.

[23] C. Dhiman, D. K. Vishwakarma, and P. Agarwal, "Part-wise spatio-temporal attention driven cnn-based 3d human action recognition," *ACM Transactions on Multimidia Computing Communications and Applications*, 2021.

[24] K. Xu, F. Ye, Q. Zhong, and D. Xie, "Topology-aware convolutional neural network for efficient skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 2866–2874.

[25] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, p. 7444–7452, 01 2018.

[26] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 018–12 027.

[27] L. Shi, Y. Zhang, and J. Cheng, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 9532–9545, 2020.

[28] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 140–149.

[29] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 180–189.

[30] 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 *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[31] J. Kong, H. Deng, and M. Jiang, "Symmetrical enhanced fusion network for skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4394–4408, 2021.

[32] C. Wu, X.-J. Wu, and J. Kittler, "Graph2net: Perceptually-enriched graph learning for skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2120–2132, 2022.

[33] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3316–3333, 2022.

[34] S. Miao, Y. Hou, Z. Gao, M. Xu, and W. Li, "A central difference graph convolutional operator for skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4893–4899, 2022.

[35] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, "Deep progressive reinforcement learning for skeleton-based action recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5323–5332.

[36] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks," *Computer Vision and Image Understanding*, vol. 208-209, pp. 103 219–, 2021.

[37] M. H. Guo, T. X. Xu, J. J. Liu, Z. N. Liu, P. T. Jiang, T. J. Mu, S. H. Zhang, R. R. Martin, M. M. Cheng, and S. M. Hu, "Attention mechanisms in computer vision: A survey," *Computational Visual Media*, vol. 8, no. 3, pp. 331–368, 2022.

[38] M. Hassanin, S. Anwar, I. Radwan, F. S. Khan, and A. Mian, "Visual attention methods in deep learning: An in-depth survey," *arXiv e-prints*, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2204.07756

[39] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7904–7913.

[40] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *arXiv eprints*, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1705.06950

[41] T. Li, J. Liu, W. Zhang, Y. Ni, W. Wang, and Z. Li, "Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 261–16 270.

[42] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1010–1019.

[43] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2684–2701, 2020.

[44] 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 *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2017, pp. 601–604.

[45] T. S. Kim and A. Reiter, "Interpretable 3d human action analysis with temporal convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1623–1631.

[46] B. Li, X. Li, Z. Zhang, and F. Wu, "Spatio-temporal graph routing for skeleton-based action recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8561–8568, 2019.

[47] Y. Obinata and T. Yamamoto, "Temporal extension module for skeleton-based action recognition," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 534–540.

[48] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020, pp. 2669–2676.

[49] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Richly activated graph convolutional network for robust skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1915–1925, 2021.

[50] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3590–3598.

[51] R. Xia, Y. Li, and W. Luo, "Laga-net: Local-and-global attention network for skeleton based action recognition," *IEEE Transactions on Multimedia*, vol. 24, pp. 2648–2661, 2022.

[52] Z. Hu, Z. Pan, Q. Wang, L. Yu, and S. Fei, "Forward-reverse adaptive graph convolutional networks for skeleton-based action recognition," *Neurocomputing*, vol. 492, pp. 624–636, 2022.

[53] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010, pp. 9–14.

[54] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1302–1310.

[55] H. Fan, Y. Yang, and M. Kankanhalli, "Point 4d transformer networks for spatio-temporal modeling in point cloud videos," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 199–14 208.

[56] X. Li, Q. Huang, Z. Wang, Z. Hou, and T. Yang, "Sequentialpointnet: A strong parallelized point cloud sequence network for 3d action recognition," *arXiv e-prints*, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2111.08492

[57] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 32 (NIPS 2019)*, 2019.

[58] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[59] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1139–1147.

[60] B. Li, X. Li, Z. Zhang, and F. Wu, "Spatio-temporal graph routing for skeleton-based action recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8561–8568, 2019.

[61] J. Gao, T. He, X. Zhou, and S. Ge, "Focusing and diffusion: Bidirectional attentive graph convolutional networks for skeleton-based action recognition," *arXiv e-prints*, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1912.11521

[62] X. Zhang, C. Xu, and D. Tao, "Context aware graph convolution for skeleton-based action recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14 321–14 330.

[63] D. Yang, M. M. Li, H. Fu, J. Fan, and H. Leung, "Centrality graph convolutional networks for skeleton-based action recognition," *Sensors*, 2020.

[64] S. Li, J. Yi, Y. A. Farha, and J. Gall, "Pose refinement graph convolutional network for skeleton-based action recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1028–1035, 2021.

[65] M. Zhao, S. Dai, Y. Zhu, H. Tang, P. Xie, Y. Li, C. Liu, and B. Zhang, "Pb-gcn: Progressive binary graph convolutional networks for skeleton-based action recognition," *Neurocomputing*, vol. 501, pp. 640–649, 2022.

[66] Y. Yoon, J. Yu, and M. Jeon, "Predictively encoded graph convolutional network for noise-robust skeleton-based action recognition," *Applied Intelligence*, vol. 52, pp. 2317–2331, 2022.

[67] Q. Wang, K. Zhang, and M. A. Asghar, "Skeleton-based st-gcn for human action recognition with extended skeleton graph and partitioning strategy," *IEEE Access*, vol. 10, pp. 41 403–41 410, 2022.

[68] Y. Xie, Y. Zhang, and F. Ren, "Temporal-enhanced graph convolution network for skeleton-based action recognition," *IET Computer Vision*, vol. 16, no. 3, pp. 266–279, 2022. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cvi2.12086

[69] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 786–792. [Online]. Available: https://doi.org/10.24963/ijcai.2018/109

[70] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1227–1236.

[71] S. Cho, M. H. Maqbool, F. Liu, and H. Foroosh, "Self-attention network for skeleton-based human action recognition," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 624–633.

[72] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1109–1118.

[73] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian, "Multiscale spatio-temporal graph neural networks for 3d skeleton-based motion prediction," *IEEE Transactions on Image Processing*, vol. 30, pp. 7760–7775, 2021.

[74] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3d action recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2842–2855, 2018.

[75] M. Liu and J. Yuan, "Recognizing human actions as the evolution of pose estimation maps," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1159–1168.

[76] C. Caetano, F. Brémond, and W. R. Schwartz, "Skeleton image representation for 3d action recognition based on tree structure and reference joints," in *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2019, pp. 16–23.

[77] T. Li, J. Liu, W. Zhang, and L. Y. Duan, "Hard-net: Hardness-aware discrimination network for 3d early activity prediction," in *2020 European Conference on Computer Vision (ECCV)*, 2020, pp. 420–436.

[78] H.-H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin, "Skeletal movement to color map: A novel representation for 3d action recognition with inception residual networks," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 3483–3487.

[79] X. Liu, M. Yan, and J. Bohg, "Meteornet: Deep learning on dynamic 3d point cloud sequences," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9245–9254.

[80] H. H. Pham, H. Salmane, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin, "A unified deep framework for joint 3d pose estimation and action recognition from a single rgb camera," *Sensors*, vol. 20, no. 7, p. 1825, 2020.

[81] H. Fan, X. Yu, Y. Ding, Y. Yang, and M. Kankanhalli, "Pstnet: Point spatio-temporal convolution on point cloud sequences," in *International Conference on Learning Representations*, 2021.

[82] Z. Bi and W. Huang, "Human action identification by a quality-guided fusion of multi-model feature," *Future Generation Computer Systems*, vol. 116, pp. 13–21, 2021.

[83] X. Shen and Y. Ding, "Human skeleton representation for 3d action recognition based on complex network coding and lstm," *Journal of Visual Communication and Image Representation*, vol. 82, p. 103386, 2022.