Received: 25 April 2023 | Revised: 20 June 2023 | Accepted: 30 June 2023

IET Computer Vision

DOI: 10.1049/cvi2.12223

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Position-aware spatio-temporal graph convolutional networks for skeleton-based action recognition

Ping Yang | **Qin Wang** | **Hao Chen** | **Zizhao Wu**

Department of Digital Media Technology, Hangzhou Dianzi University, Hangzhou, China

Correspondence

Zizhao Wu, Hangzhou Dianzi University, Department of Digital Media Technology, Hangzhou, China.
Email: wuzizhao@hdu.edu.cn

Funding information

Zhejiang Provincial Natural Science Foundation of Chinaunder, Grant/Award Number: LGF21F20012

Abstract

Graph Convolutional Networks (GCNs) have been widely used in skeleton-based action recognition. Though significant performance has been achieved, it is still challenging to effectively model the complex dynamics of skeleton sequences. A novel position-aware spatio-temporal GCN for skeleton-based action recognition is proposed, where the positional encoding is investigated to enhance the capacity of typical baselines for comprehending the dynamic characteristics of action sequence. Specifically, the authors' method systematically investigates the temporal position encoding and spatial position embedding, in favour of explicitly capturing the sequence ordering information and the identity information of nodes that are used in graphs. Additionally, to alleviate the redundancy and over-smoothing problems of typical GCNs, the authors' method further investigates a subgraph mask, which gears to mine the prominent subgraph patterns over the underlying graph, letting the model be robust against the impaction of some irrelevant joints. Extensive experiments on three large-scale datasets demonstrate that our model can achieve competitive results comparing to the previous state-of-art methods.

KEYWORDS

computer vision, convolutional neural nets, graph theory

1 | INTRODUCTION

In recent years, skeleton-based action recognition has been extensively investigated and has attracted considerable attention due to its robustness against dynamic circumstance and complex backgrounds [1–3]. Some earlier attempts focus mainly on designing hand-crafted features to represent the skeleton, whereas these approaches have been proven to be subject to limitations in robustness, interoperability and scalability [4, 5]. With the development of deep neural networks in computer vision field, some researchers have investigated many classical neural network architectures to gain the prediction [6]. Typical architectures include CNNs and RNNs [7, 8], where the skeleton data is usually represented as a sequence of joint-coordinate vectors or as a pseudo-image. However, these methods overlook the inherent correlations between joints. Instead, the skeleton can be naturally structured as a graph with inherently the joints as nodes and their correlations as edges, benefiting the high-level analysis of action patterns [9]. Inspired from this, ref. [10] pioneered the Spatio-Temporal Graph Convolutional Networks (ST-GCN) to model the skeleton data with GCNs [11], and achieved significant performance. Since then, in order to boost the performance of GCNs, plenty of works have been explored. At present, with the surge of attention mechanism [12], numerous researchers have explored to combine both the advantages of GCNs and attention modules, resulting in notable improvements in performance. Due to the strong modelling capability, the combination of GCNs and attention mechanism has thereafter become the de facto model in the field.

While showing encouraging results, existing attention and GCN models suffer from the following problems: (1) They are invariant to sequence ordering information. In order to capture the temporal correlations, typical GCN-based approaches have adopted the Temporal Convolutional Networks (TCN) to model the information. However, existing TCN only facilitates

to model the short-range sequence ordering information implicitly with the help of convolutional kernels, whereas we believe that the fusion of long-range and short-range ordering information is one of key elements for comprehending actions. Take the classes of stand up/sit down as an example, we can see from Figure 1a,b that both action clips contain similar content but differ in the sequence order. For these action sequences, their ordering information is a strong prior in determining which action class they belong to. Similar actions are widely available in the *NTU RGB + D* benchmark dataset, for example, put on jacket/take off jacket, put on glasses/take off glasses, etc. (2) Action recognition is dominated by the movement of some key joints, how to locate these key joints and model their interaction is one big issue to address this problem. However, equipped with GCN and attention, existing methods can teach their network 'where (joints) and when (frame) to look' for the skeleton sequence efficiently, but struggle to recognise the identity information of the nodes, that is, which node is critical and with whom it has a strong interaction? Taking the action class of clapping as an example, we illustrate the left hand joint at the centre of the graphs in Figure 1c,d. From (c), the network is aware that some nodes are important, and some of them have strong interactions. By comparison, from (d), the network knows precisely the red node (left hand joint) and the blue node (right hand joint) are important, and they have strong correlation. When proceeding with large amounts of graphs of different classes, such side information will definitely help our model to learn more discriminative features. (3) There is still some room to improve the redundancy and over-smoothing problems in GCNs. We note that typical approaches have explored the graph-level and node-level representations, however, on one hand, graph-level representation provides an overarching view of the entire graph, but at the loss of some striking local patterns. On the other hand, node-level representation encodes local structures but holds a deficiency in capturing correlations among joints. We also note that graphs encompass various structural properties, including nodes, edges, motifs and subgraphs. Among these structures, subgraphs usually contain vital characteristics and striking patterns. Despite the popularity of subgraph researches in machine learning community, little research has been conducted for action recognition.

In this paper, we propose Position-Aware Graph Convolutional Networks (PA-GCN) to address the above issues. Specifically, our method first investigates the temporal positional encoding, which is defined using the sine and cosine function frequencies. In this way, each frame in the skeleton sequence will be provided with a unique identity, thus facilitating the model to capture the sequence ordering information. Secondly, our method explores the spatial positional embedding by adopting a 1D learnable parameter to empower the model of GCNs with the explicit identity information for each joint. Equipped with this side information, our method enables the network to learn more discriminative features. And finally, our method introduces a novel subgraph mask to learn the prominent subgraph patterns and decrease the impact of other irrelevant joints. The mask is purely data-driven and initialised using the attention weights provided by the attention module. By utilising the mask, only striking subgraph patterns will be extracted, which is geared to address the redundancy and over-smoothing problems in GCNs.

We highlight the contributions of our PA-GCN as follows:

1. Our method presents a novel positional aware GCN model which facilitates to model the complex dynamics of actions.
2. Our method investigates temporal position encoding and SP encoding strategies for the task.
3. Our method introduces a subgraph mask that talents to mine prominent patterns.

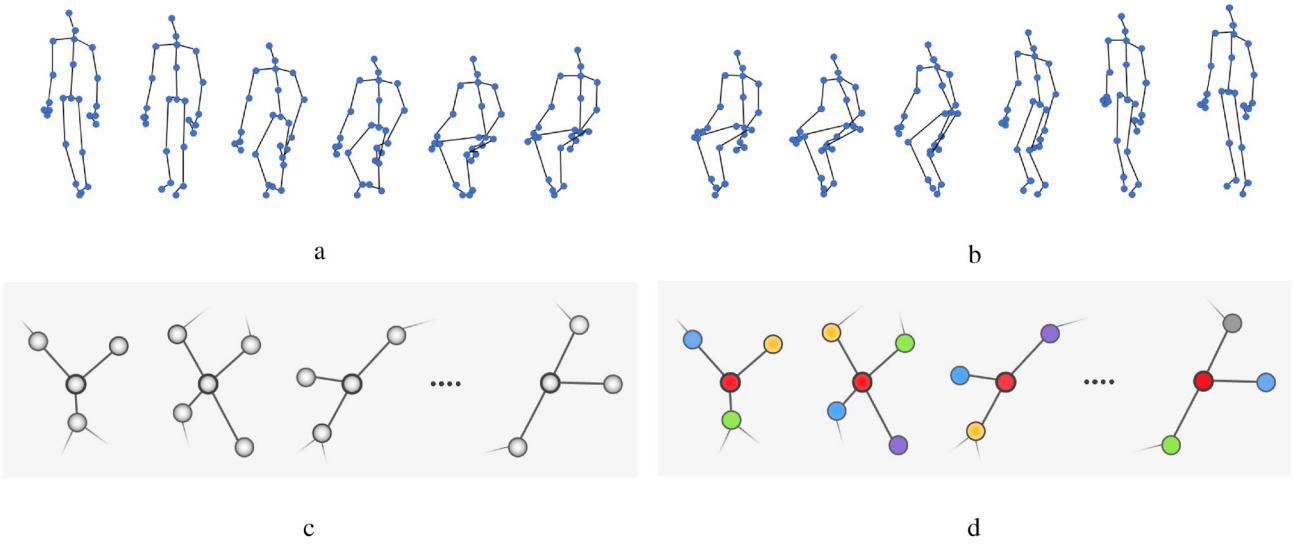

FIGURE 1 In the upper row, we show the actions with similar content, but differ in ordering. In the bottom row, we illustrate the graph embedding With (W/) and Without (W/O) Spatial Position (SP) embedding. (a). Sit Down (b). Stand Up (c). W/O SP Embedding (d). W/SP Embedding.

4. Extensive experiments on two benchmark datasets have demonstrated the significant performance of our method against some state-of-the-art algorithms.

2 | RELATED WORKS

In this section, we present a brief overview of skeleton-based action recognition, subgraph neural network, as well as the position encoding.

2.1 | Skeleton-based action recognition

There exists abundant literature on predicting human actions from skeleton data [13]. Recently, it was witnessed that a novel architecture called Graph Neural Networks (GCN) [11] generalises CNN to graphs of arbitrary structures and has beaten many established baselines in a number of applications. Ref. [10] presented ST-GCN by modelling the skeleton data as a graph structure under the spatio-temporal GCN architecture. Spatio-Temporal Graph Convolutional Networks marries the spatial graph convolutions with the temporal convolutions for modelling the spatio-temporal sequence data, becoming the de facto model in the paradigm. Upon the baseline, many variants of ST-GCN have been proposed to enhance the modelling capacity. For example, ref. [14] investigated Shift-GCN that employs shift graph operations and lightweight point-wise convolutions to alleviate the computational complexity burden. Ref. [15] proposed a channel-wise topology refinement graph convolution to dynamically model different topologies, via aggregating joint features in different channels of skeletons. Ref. [16] introduced InfoGCN, which is combined with an Information-Bottleneck learning objective and a self-attention based graph convolution module, learns the compressed latent representation of actions and infers context-dependent intrinsic topology in spatial modelling of skeletons. They also suggested a hierarchically decomposed GCN [17] that extracts major structural edges and uses them to construct a hierarchically decomposed graph. Other follow-up works include multi-scale modelling [18, 19], graph routing [20] and so on [21, 22].

Attention mechanism has been successfully applied to various tasks due to the effectiveness of modelling long-range dependencies. For the task of skeleton-based action recognition, ref. [23] introduced a spatial-temporal attention LSTM model to specify various attention weights to the skeleton joints in the temporal domain. Ref. [24] combined the LSTM module with attention mechanism. However, processing the objects in a recurrent manner will inevitably neglect the dependencies between nonadjacent frames. Some researchers have explored the combination of attention modules and GCNs, ref. [25] proposed spatial-temporal joint attention which aims to detect the prominent joints over the skeleton sequence to avoid the over-smoothing problem occurred in the vanilla GCN model. Ref. [16] investigated an architecture that utilises a self-attention mechanism to capture the intrinsic topology. Ref. [26] also constructed a Transformer-based spatial-temporal network that captures both local and global attention of human motion skeleton sequences.

In this paper, we introduce a novel architecture termed PA-GCN for skeleton-based action recognition, which enhances the performance of the typical attention and GCNs by exploring some side and structural information.

2.2 | SubGraph neural networks

Generally, graph contains many structural properties, ranging from nodes, edges, motifs to subgraphs, which have been extensively examined in the existing architectures of GCN and GNN. In one sense, motifs and subgraphs share something in common as both of them capture local substructures. However, motif-based approaches are normally concerned with small structures holding prominent features. Instead, subgraphs are designed to exploit high-order structural information. Aiming to meet this goal, some attempts exploit subgraphs by motif combination [27], subgraph isomorphism counting [28], rule-based extraction [29], etc. Besides these works, ref. [30] introduced Ego-CNN, which uses ego graphs to precisely detect prominent structures. Ref. [31] investigated a SEAL framework that learns a mapping function from subgraph patterns to link existence. Ref. [32] proposed a subgraph neural network that investigates routing mechanism to propagate messages between different subgraphs. Ref. [33] presented SUGAR, a novel hierarchical sub-graph level selection, which extracts prominent subgraphs to reconstruct a sketched graph, based on which, a representation that can better reveal the subgraph patterns can be obtained.

Drawing inspiration from these works, we introduce the idea of sub-graph to our skeleton-based action recognition paradigm by suggesting a sub-graph mask to the baselines.

2.3 | Position encoding

Position encoding is firstly introduced by the vanilla Transformer to overcome the inherent deficiency of self-attention, which cannot capture the sequence ordering information. To date, there are mainly two classes of position encoding strategies: absolute position encoding [12] that focuses on capturing absolute position information, and relative position encoding [34, 35] that emphases on relative position between the input elements.

In recent years, several researchers have also explored the combination of position encoding and GCN [36–38]. For example, ref. [36] investigated the position-aware graph neural network which enables to compute position information of one node with respect to other nodes in the graph by sampling the anchor nodes. Ref. [38] proposed a new method that enables GNNs to learn both structural and positional representations with the help of learnable positional encoding.

In our work, we mainly adopt the absolute position encoding for our task to capture both the temporal position information and SP information.

3 | METHOD

In this section, we first introduce the preliminaries including data notations and some basic building blocks, then we elaborate the modules of position encoding and subgraph masking operator and finally, the overall architecture of our model is presented.

3.1 | Preliminaries

Notation. A 3D skeleton-based action sequence can be represented as $X = x_n^t \in R^{T\times N\times C}$ with T frames, N joints and the C coordinates of each joint. In this work, following the previous work [25] that has shown promising results based on multi-stream input, we also take the position features, velocity features and bone features of skeleton data as the input to our method. Specifically, we use X_P to denote the position features, which is defined as the xyz-coordinates of absolute feature vector of each joint, that is, $C = 3$. The velocity and bone features can analogously be defined as X_V and X_B based on the coordinate difference in the temporal and spatial dimensions. Since our method tackles multiple stream data without any difference, without losing the generality, in the following we only discuss the case $X = X_P$. For convenience, we denote single frame skeleton as $X^t \in R^{C\times N}$ for tth frame and node trajectory as $X_n \in R^{C\times T}$ for nth joint.

We model the human skeleton as a graph $G = (V, E, X)$, where $V = \{v_1, v_2, \ldots, v_N\}$ is the graph node set representing the skeleton joints, and E is the graph edge set that reflects the correlations between the skeleton joints, and the correlation is formulated as an adjacency matrix $A \in R^{N\times N}$. $X \in R^{N\times C}$ is the feature set where each element $x_i \in R^C$ denotes the feature of v_i.

Self-Attention. Self-attention is a particular implementation of attention mechanism in which the queries, keys and values are the same sequence of symbol representations. Formally, self-attention can be described as a mapping function that projects a query Q, key K and value V to an output attention representation in the following form:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{1}$$

where d is their dimension, the $1/\sqrt{d}$ is used for numerical stability for attentions, preventing extremely small gradients caused by the large values of d. In our framework, we employ the self-attention to compute the initial correlation matrix among skeleton joints both in temporal and spatial dimensions.

Graph Convolutional Network. Graph convolutional network is motivated via a first-order approximation of spectral graph convolution which aggregates node representations through their neighbours. Specifically, given a graph with adjacency matrix A, and data matrix X, the basic formulation of GCN is to produce an output x' via the layer-wise propagation rule:

$$x' = Wx\left(\Lambda^{-\frac{1}{2}}A\Lambda^{-\frac{1}{2}} \odot M\right), \tag{2}$$

where W are the trainable parameters of the layer, $\odot$ refers to element-wise product, and Λ is adopted to normalised A. Each element of Λ can be defined as $\Lambda_{ii} = \sum_j (A_{ij}) + \lambda$, λ is usually set to 0.001 to avoid empty row. M is an attention map that indicates the importance of each vertex, here we defined as the subgraph masking operator. In our framework, we utilise the above GCN block to learn the optimised graph representation at each frame.

3.2 | Position encoding

In this subsection, we first introduce our temporal position encoding strategy and then the SP embedding.

Temporal position encoding. As mentioned earlier, on one hand self-attention has an inherent deficiency — it cannot capture the ordering of input sequence. On the other hand, the TCN can only model short-range correlations implicitly. Therefore, incorporating explicitly representations of position information is especially important for the task. Currently, there are two popular positional encoding strategies: absolute positional encoding and relative positional encoding, depending on the tasks they face. In our case, we mainly focus on the absolute position encoding, as such information is critical for us to capture the sequence ordering information.

Specifically, given a skeleton X^t of tth frame, our method computes temporal position encoding with the dimension size dt using sine and cosine functions of different frequencies [12], it is formulated as follows:

$$\begin{aligned} E_{pt}(t, 2i) &= \sin\left(t/10000^{2i/dt}\right), \\ E_{pt}(t, 2i+1) &= \cos\left(t/10000^{2i/dt}\right), \end{aligned} \tag{3}$$

where i is the dimension index of the output tensor. The temporal position encoding can then be injected into the original feature representations at each frame, and fed into the TCN block, to serve for the fusion of implicit and explicit sequence ordering information, that is, $\tilde{X}^t = X^t + E_{pt}$.

In a multi-layer network with varying sequence lengths, adding temporal encoding to the global network header has limited noting capabilities. Therefore, in order to enhance the noting ability of the embedding, we propose multi-layer and multi-scale temporal encoding and add the temporal encoding in each layer. In the multi-layer network, the multi-layer temporal encoding of different scales can still identify time information when the length of time series changes, so that the temporal information of different lengths can be better captured.

Spatial Position embedding The SP embedding is devised to let the GCNs be conscious of which node is prominent and with whom it interactive more closely, as the identities of joints are all lost when the joint features are fed into the graph. To address this problem, instead of using fixed

temporal absolute position encoding, we explore an alternative approach: a learnable position embedding. This is based on the understanding that joints do not possess absolute ordering information, and their correlations vary dynamically over time.

Specifically, given a node trajectory X_n for nth joint, our method suggests a learnable 1D embedding as the SP embedding defined as $E_{ps}(n) \in R^{N\times ds}$, where ds is the dimension size of the embedding, N is the number of joints and n is the spatial index of each joint in the sequence. We randomly initialise $P(n)$, this value will lately be optimised based on the back-propagation of the networks. In the subsequent processing, we inject the position embedding of each joint to the features, serving as a unique identitity for joints among different frames, that is, $X_n = X_n + E_{ps}$.

3.3 | Subgraph masking operator

In practice, the difference between different categories of action sequence often lies in small distinct joints, ideally, we should pay much attention to these prominent joints and their correlations while overlooking the irrelevant joints. In a typical pipeline, this process is frequently modelled using the attention mechanism to assign attention weights to all joints, which will inevitably incur some noise contributed by the irrelevant joints. To resolve this issue and improve concentration in attention, we investigate a subgraph masking operation upon the self-attention weight matrix.

Specifically, as illustrated in Figure 2, given the input skeleton features X, our subgraph masking operator is achieved through three steps: (1) extract the initial attention weight matrix and updates the initial features X; (2) perform temporal-wise topology modelling; and (3) conduct threshold judgement. In the following, we list the details:

Self-attention layer. We employ the self-attention mechanism to compute the initial attention weight matrix based on Equation (1). Following the definition, our method employs the Q, K, V to denote the query, key and values, where Q and K are computed from the input feature X by linear transformations W_θ and W_ϕ, and $V = X$, that is, $(Q, K, V) = (W_\theta X, W_\phi X, X)$. The output of the self-attention layer is achieved through the following formulation:

$$\tilde{X} = \alpha X, \alpha = Softmax\left(\frac{QK^T}{\sqrt{d}}\right), \tag{4}$$

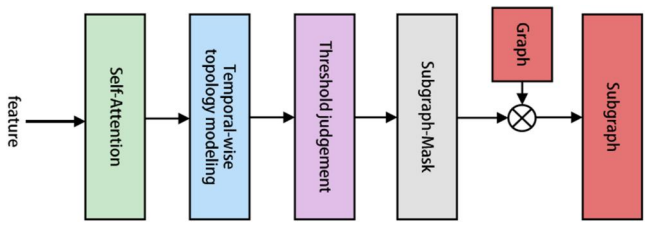

FIGURE 2 The pipeline of Subgraph masking operator. Subgraph masking operator takes action sequence as input and outputs a subgraph.

where α is utilised to denote the attention weight matrix.

Temporal-wise topology modelling. Though we have reached an initial attention weight matrix α, yet this matrix is shared across all temporal dimensions. To achieve a subgraph mask at each frame, we proceed to model the specific relationships between vertices in each frame in the following formulation:

$$W_t = \sigma\left(\delta\left(\tilde{X}\right) + \alpha\right), \tag{5}$$

where the temporal-wise topology matrix $W_t \in R^{N\times N\times T}$ is obtained by first employing a linear function $\delta(\cdot)$ to adjust the dimension of $\tilde{X}$ to be consistent with α, and then performing the addition operation with the shared weight matrix α in a broadcast way, that is, α is added to each temporal of $\tilde{X}$, akin to channel-wise topology matrix definition in ref. [39], finally sigmoid activation function σ is conducted on the value.

Threshold judgement. In order to obtain a sparse version of the attention matrix, we employ a threshold judgement to mask the temporal-wise attention matrix W_t. Specifically, given a threshold value v and for each element $W_t^{ij} \in W_t$ $(0 \le i, j \le N)$, if W_t^{ij} is larger than v, it will be reserved, and otherwise it will be reset to 0. The output mask matrix M can be formulated as:

$$M = M(W_t) = \begin{cases} 1 & W_t^{ij} \ge v \\ 0 & W_t^{ij} < v, \end{cases} \tag{6}$$

where we also use $M(\cdot)$ to denote the masking operator function. However, we note that the exact value of v is irrelevant to the final results, since the attention weights are learnable. If we specify the threshold with a larger value, then the network will learn the attention matrix with larger elements in respond, and vice versa.

Finally, this masking operator M can be replaced with the M in the Equation (2), enabling the masking operation on the attention weight matrix in the GCN block.

3.4 | Model architecture

The overall architecture of our method is illustrated in Figure 3. It is composed of a stack of $N = 10$ identical PA-GCN layers, followed by a global average pooling and a softmax classifier to make the prediction. Each PA-GCN layer has two blocks: the first is a GCN block, and the second is a TCN block. We employ a skip connection of residual structure in each of the two blocks, followed by layer normalisation.

Inside the GCN block, our method injects the temporal position encoding module, SP embedding module and subgraph masking operator into the GCN block. Specifically, given an input feature F_{in}, which is initialised by X, the GCN block first lets it pass through a subgraph masking operator layer (Section 3.3) to obtain a subgraph mask M and the attention matrix that is also defined as the correlation matrix A of GCNs. The SP embedding E_{ps} (Section 3.3) is also achieved and

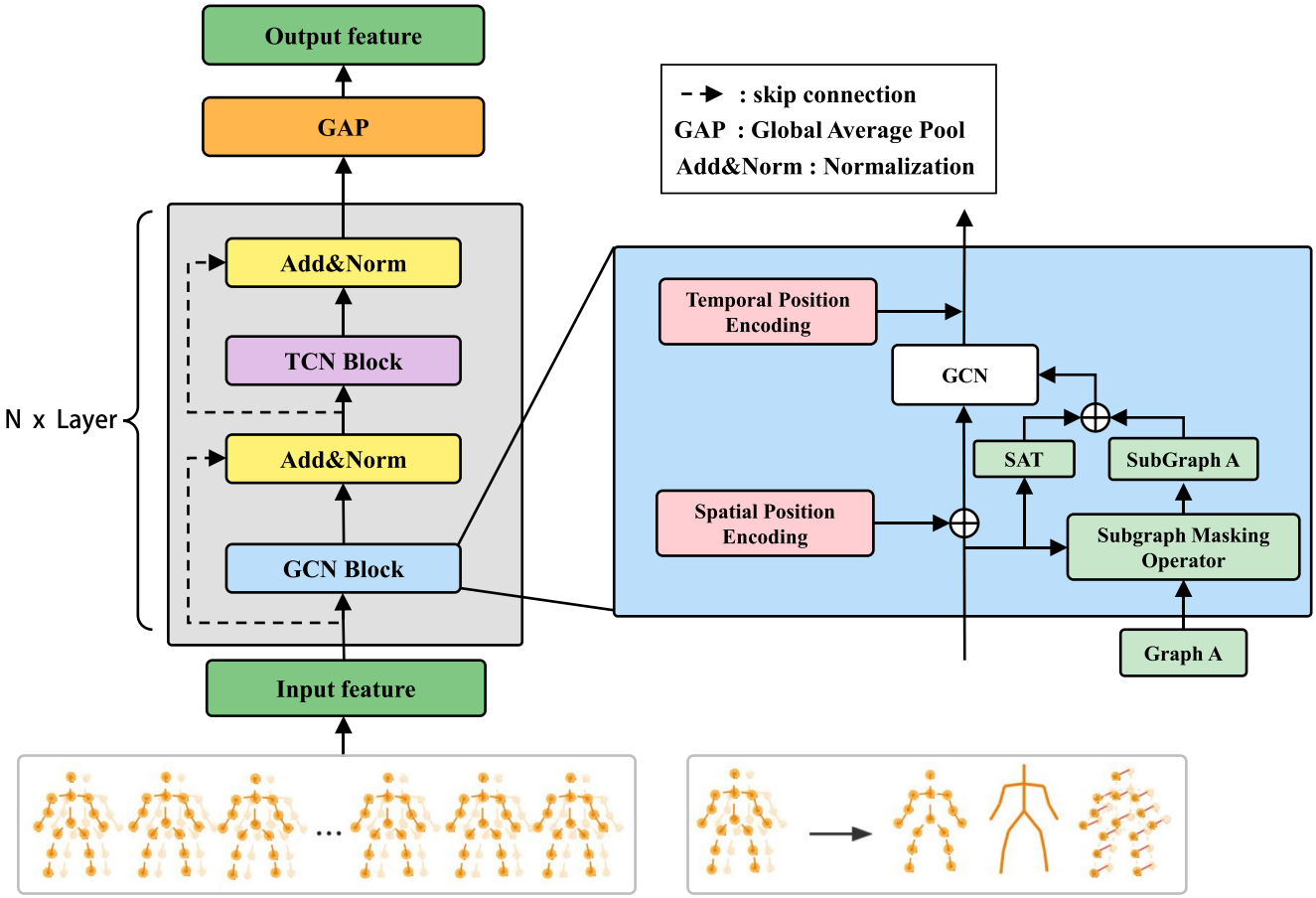

FIGURE 3 The pipeline of our architecture. Our network takes multi-stream data, that is, Joint, Bone and Velocity as input, and then feeds them into the PA-GANs layers one by one to generate discriminative features, the recognition results are finally obtained based on these features by passing them through a global average pooling and a softmax classify layer. The core of our model lies in the graph convolutional network block, where temporal position encoding, spatial position embedding and subgraph masking operator are investigated to mine the side and structure information.

combined with the initial feature as the final input feature matrix $F_{in} + E_{ps}$ to the GCNs. After that, the GCNs module (Equation 2) is conducted to optimise the graph representations F'_{in} at each frame, which is then fed into the temporal position encoding layer to compute the temporal position encoding E_{pt} (Section 3.2). Eventually, the GCN block combines temporal position encoding with the output feature of GCN to form a new feature $F'_{in} + E_{pt}$ as input to the TCN block.

Inside the TCN block, we use the regular 1D convolution on the temporal dimension as the TCN for modelling the relationship between adjacent frames. In our case, we set the kernel size of the 1D convolution to 9, and the stride size is set to 2 when performing downsampling, and 1 otherwise.

4 | Expriments

To evaluate our method, we conduct extensive experiments on three skeleton-based action recognition datasets: *NTU RGB + D* 60 [40], *NTU RGB + D* 120 [41] and *Northwestern-UCLA* [42], which have been widely utilised in previous works. In the following, we first describe the datasets we used and the implementation details, followed with the ablation study to verify the effectiveness of our model. Finally, we compare our model with some state-of-the-art approaches.

4.1 | Datasets

NTU RGB + D 60. NTU RGB + D 60 is a widely used large scale skeleton-based action recognition dataset, containing 56,880 action clips with 60 classes. These classes are roughly divided into health-related actions, daily actions, as well as mutual actions. The action clips are performed by 40 users with different age groups spanning from 10 to 35. Each action is captured by three Microsoft Kinect v2 depth sensors from 3 different horizontal views but at the same height concurrently, and is represented as 3D coordinates of 25 joints. Mutual action classes contain two subjects while the others contain only one subject. Two evaluation benchmarks are provided by the dataset: (1). Cross-subject (X-sub): the dataset in this benchmark contains the training set with 40,320 clips from 20 subjects; and the validation set with 16,560 clips from the other 20 subjects. (2). Cross-view (X-view): the dataset in this benchmark contains the training set with 37,920 clips from two views; and the validation set with 18,960 clips from another

view. In this study, akin to many other baselines, we report the top-1 accuracy on both benchmarks.

NTU RGB + D 120. NTU RGB + D 120 extends the NTU RGB + D 60 by providing 114,480 video clips performed by 106 subjects from 155 viewpoints. This dataset covers 120 classes, where 60 classes are inherited from the previous NTU RGB + D dataset. The dataset contains 32 setups, each of them denotes a specific location and background. Akin to NTU RGB + D, two benchmarks are suggested by the dataset: (1). Cross-subject (X-sub): the dataset in this benchmark covers the training set with 63,026 clips and a validation set with 50,922 clips respectively. (2). Cross-setup (X-view): the dataset in this benchmark contains the training set with 54,471 clips and a validation set with 59,477 clips respectively.

Northwestern-UCLA. Northwestern-UCLA dataset covers video clips captured by 3 different Kinect sensors from different viewpoints. A total of 1494 video clips belonging to 10 action categories were collected, where each action is captured by 10 different subjects. Following the typical evaluation methods, we divide the dataset into the training set from the first two cameras and the test set from the other camera.

TABLE 1 Ablation study on the NTU-RGB 60 dataset.

Model	Acc
Backbone	88.5
Backbone + TPE (single)	90.6
Backbone + TPE (multi)	91.2
Backbone + SPE	90.4
Backbone + subgraph (without threshold judgement)	90.8
Backbone + subgraph	91.1
PA-GCN(full)	92.1

Note: Top-1 accuracy is reported. TPE, SPE and subgraph denote the temporal position encoding, spatial position embedding and subgraph masking operator blocks respectively.

4.2 | Implementation

We implement our framework in PyTorch and the code will be released later. For the hyper-parameters, we use stochastic gradient descent with a Nesterov momentum of 0.9, weight decay 0.0004 to train all experiments. Specifically, we set the training epoch to 65 and define a warm-up strategy in the first 5 epochs to make the training procedure more stable. We resample each sample to 64 frames, and set the batch size to 64. For NTU RGB + D, the learning rate is defined to 0.1 and let decays to be a factor 0.1 at epoch 35 and 45. For NTU RGB + D 120, we set the learning rate to 0.1 and decay it by a factor of 0.1 at epoch 35 and 55. The numbers of channels for each PA-GCN layer are 64,64,64,64,128,128,128,256,256 and 256. The dimension sizes of dt and ds are the same as the numbers of channels.

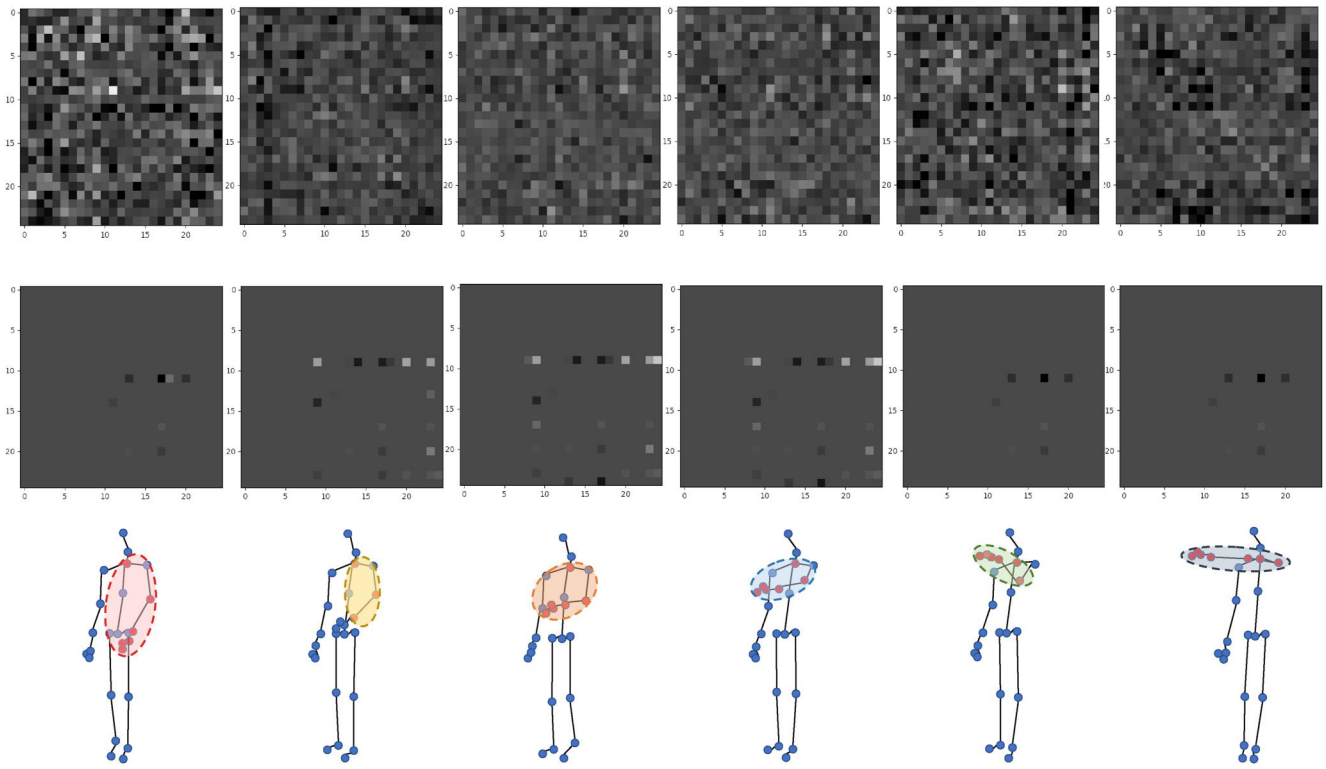

FIGURE 4 We take the *point to something* action as an example. In the upper row, we show the original attention adjacency matrices. In the middle row, we show the learnt subgraph matrices at different frames. In the bottom row, we illustrate the activated joints at different frames, where we colour the activated joints with red dots, and the non-activated joints with blue dots. We also use circles of different colours to represent the different topological structures of the subgraphs.

4.3 | Ablation study

In this subsection, we examine how different components of the model affects the final performance. The experiments are conducted on X-sub benchmark on the NTU RGB + D 60 dataset.

Position encoding. We first evaluate how the position encoding affects the final performance. Specifically, our method adopts the backbone (AGCN) in our architecture as the baseline, and then trying to incorporate the different modules into the baseline to evaluate the results. Table 1 presents the results. From the table, we can see that by injecting the temporal position encoding into the backbone, our method obtains a higher accuracy record with 2.7% improvement over the backbone. Similarly, when testing the SP embedding block, we achieve results with 1.9% improvement. This results indicate the importance of the temporal position encoding and SP embedding in mining the side information to enhance the performance of skeleton-based action recognition.

Subgraph-mask. Evaluation has been conducted on how subgraph mask and threshold judgement contribute to the final performance. Table 1 shows the results, we first test threshold judgement operator and get an improvement of 2.3% points. Similarly, subgraph mask contributed 2.6% improvement on the backbone. The results validate the effectiveness of the subgraph masking operator block.

We also visualise the attention weights of one action sequence in different ways in Figure 4, where in upper row, the original attention weights is illustrated, we can see that such weight matrix is dense, suggesting that each query has context correlations to all the other values. In the middle row, the subgraph-based attention weights are depicted. As can be seen from these figures, our subgraphs are sparse, and change dynamically in temporal dimension. In the bottom row, we illustrate the activated joints in several contextual frames of different actions, where we can see that only prominent patterns have been extracted, letting our model be more robust against the redundancy problem commonly encountered in typical GCNs. These figures validate the effectiveness of the subgraph block.

In addition to the above display, we visualise the attention matrix before and after threshold judgement in Figure 5. The left attention matrix is the attention weights of backbone, we can see that when compared to the attention matrix with subgraph operations, the original attention matrix has no clearly prominent joints. Comparing the middle image and the right image, it can be seen that the attention matrix with threshold determination highlights more important nodes.

4.4 | Comparison with state-of-the-art

We compare our final model with some state-of-the-art skeleton-based action recognition methods in both *NTU RGB + D* 60 and *NTU RGB + D* 120 datasets under the given evaluation metrics. In Table 2,' J′ refers to Joint features, 'B' refers to Bone features and 'V' refers to Velocity features.

On *NTU RGB + D* 60 dataset, we compared with 23 State-Of-The-Art (SOTA) approaches including RNN-based method [43], LSTM-based methods [23, 44], Transformer-

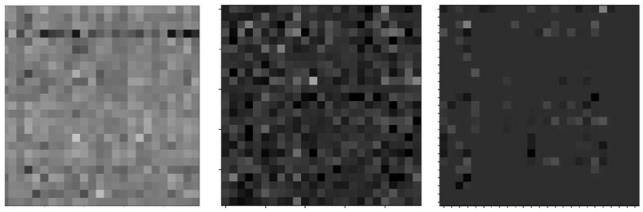

FIGURE 5 Left figure we show is the original attention adjacency matrices without subgraph masking operator. The middle picture is the attention adjacency matrices without threshold judgement. On the right, We show the subgraph attention matrix resulting from full1 subgraph masking operations.

TABLE 2 Evaluation on NTU-RGB + D 60/120 dataset.

	NTU-RGB + D 60[a]		NTU-RGB + D 120[b]	
Models	***X*sub**	***X*view**	***X*sub**	***X*set**
HBRNN [43]	59.1	64.0	-	-
ST-LSTM [44]	69.2	77.7	55.7	57.9
STA-LSTM [23]	73.4	81.2	-	-
PA-LSTM [40]	-	-	25.5	26.3
GL-Transformer [26]	76.3	83.8	66.0	68.7
ST-GCN [10]	59.1	64.0	70.7	73.2
AS-GCN [19]	86.8	94.2	77.9	78.5
2s-AGCN [45]	88.5	95.1	82.5	84.2
DGNN [46]	89.9	96.1	-	-
SGN [47]	89.0	94.5	79.2	81.5
4s-shift-GCN [14]	90.7	96.5	85.9	87.6
MS-G3D [18]	91.5	96.2	86.9	88.4
DC-GCN+ADG [48]	90.8	96.6	86.5	88.1
PA-ResGCN-B19 [25]	90.9	96.0	87.3	88.3
Dynamic-GCN [49]	91.5	96.0	87.3	88.6
RA-GCNv2 [50]	87.3	93.6	81.1	82.7
Colourisation model [51]	88.0	94.9	-	-
MCC+2s-AGCN [52]	89.7	96.3	81.3	83.3
MST-GCN [53]	91.5	96.6	-	-
Skeletal GNN [54]	91.6	96.7	87.5	89.2
CTR-GCN [15]	92.4	96.8	88.9	90.6
EfficientGCN-B4 [55]	92.1	96.1	88.7	89.1
Info-GCN [16]	93.0	97.1	89.8	91.2
PA-GCN(J+B))	91.4	96.1	86.9	88.9
PA-GCN(J+B+V))	92.1	96.7	87.4	89.8

[a]Results on NTU-RGB + D 60 dataset.

[b]Results on NTU-RGB + D 120 dataset.

TABLE 3 Comparison with SOTA methods on the northwestern-UCLA dataset.

Model	NW-UCLA
DRNN [56]	74.2
TS-LSTM [57]	89.2
GL-Transformer [26]	90.4
SGN [58]	92.5
AGC-LSTM [24]	93.3
Shift-GCN [14]	94.6
DC-GCN+ADG [48]	95.3
Colourisation model [51]	94.6
CTR-GCN [15]	96.5
Info-GCN [16]	97.0
PA-GCN(J+B+V)	96.1

Note: Top-1 accuracy is reported.

based method [26] and many GCN-based methods [10, 53, 55]. Table 1 shows that the GCN-based methods have superior performance than the RNN-, LSTM- and Transformer-based methods, and the ST-GCN model have becoming the de facto baseline in the field. And among the SOTA approaches, our model achieves remarkable performance with the accuracy of 92.1% on the X-sub and 96.7% on the X-view evaluations of the dataset respectively, the results verify the effectiveness of our method for the task.

On *NTU RGB + D* 120 dataset, we compared with our final model with 18 SOTA approaches including LSTM-based methods [40, 44], Transformer-based method [26] and GCN-based methods [10, 54, 55] as well. The evaluation results are reported in Table 1. As can be seen from the table, which are identical to the experiments on NTU-RGB + D 60, our model shows markable performance. The results confirm the generality capability of our model for large-scale datasets.

On *Northwestern-UCLA* dataset, we also compare our model with 10 SOTA approaches including RNN-based methods DRNN [56], LSTM-based methods TS-LSTM [57], AGC-LSTM [24], Transformer-based method GL-Transformer [26] and many GCNS-based methods Shift-GCN [51]. In Table 3, evaluation results have shown that our model outperforms many other SOTA models. The results show that on small-scale datasets, our model performs well even without data augmentation methods.

5 | CONCLUSION

In this work, we propose a novel PA-GCN for skeleton-based action recognition. Our model investigates the position encoding modules and the subgraph masking operator to address some defects holding in existing models, including long-term sequence ordering, GCN position detecting, prominent pattern mining, etc. Experimental results on three publicly available datasets demonstrate the significant performance of the proposed approach. For future work, we plan to further explore the potential of the self-attention and subgraph methods for skeleton-based action recognition.

AUTHOR CONTRIBUTIONS

Ping Yang: Conceptualization; resources; supervision; validation; writing – review & editing. **Qin Wang**: Conceptualization; data curation; formal analysis; investigation; methodology; project administration; visualization; writing – original draft; writing – review & editing. **Hao Chen**: Conceptualization; data curation; formal analysis; investigation; methodology; validation; writing – original draft. **Zizhao Wu**: Conceptualization; formal analysis; project administration; resources; writing – review & editing.

ACKNOWLEDGEMENTS

This work was supported by the Zhejiang Provincial Natural Science Foundation of Chinaunder Grant No. LGF21F20012.

CONFLICT OF INTEREST STATEMENT

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled "PA-GCN: Position-Aware Spatio-Temporal Graph Convolutional Networks for Skeleton-based Action Recognition."

DATA AVAILABILITY STATEMENT

The NTU-RGB + D 120 dataset and NW-UCLA dataset that support the findings of this study are available from ROSE Lab. Restrictions apply to the availability of these data, which were used under license for this study. Data are available at https://rose1.ntu.edu.sg/dataset/actionRecognition/ with the permission of ROSE Lab. The NW-UCLA dataset that support the findings of this study are openly available at http://wangjiangb.github.io/my_data.html.

ORCID

Qin Wang https://orcid.org/0009-0002-4740-3686

REFERENCES

1. Poppe, R.: A survey on vision-based human action recognition. Image Vis Comput. 28(6), 976–990 (2010). https://doi.org/10.1016/j.imavis.2009.11.014
2. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: a review. ACM Comput. Surv. 43(3), 16:1–16:43 (2011). https://doi.org/10.1145/1922649.1922653
3. Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action representation, segmentation and recognition. Comput. Vis. Image Underst. 115(2), 224–241 (2011). https://doi.org/10.1016/j.cviu.2010.10.002
4. Hussein, M.E., et al.: Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations. In: Rossi, F. (ed.) International Joint Conference on Artificial Intelligence IJCAI, pp. 2466–2472 (2013)
5. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3D skeletons as points in a lie group. In: IEEE Conference

on Computer Vision and Pattern Recognition, CVPR, pp. 588–595. IEEE Computer Society (2014)
6. Rahmani, H., Bennamoun, M., Ke, Q.: Human Action Recognition from Various Data Modalities: A Review (2021)
7. Chéron, G., et al.: Pose-based CNN features for action recognition. In: IEEE International Conference on Computer Vision, ICCV, pp. 3218–3226. IEEE Computer Society (2015)
8. Lev, G., et al.: RNN Fisher vectors for action recognition and image annotation. In: Leibe, B., et al. (eds.) ECCV. vol. 9910 of Lecture Notes in Computer Science, pp. 833–850. Springer (2016)
9. Johansson, G.: Visual perception of biological motion and a model for its analysis. Percept. Psychophys. 14(2), 201–211 (1973). https://doi.org/10.3758/bf03212378
10. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Association for the Advancement of Artificial Intelligence, pp. 7444–7452 (2018)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR (2017)
12. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
13. Presti, L.L., Cascia, M.L.: 3D skeleton-based human action classification: a survey. Pattern Recogn. 53, 130–147 (2016). https://doi.org/10.1016/j.patcog.2015.11.019
14. Cheng, K., et al.: Skeleton-based action recognition with shift graph convolutional network. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR. p. 180–189 (2020)
15. Chen, Y., et al.: Channel-Wise topology refinement graph convolution for skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13359–13368 (2021)
16. Chi, Hg, et al.: Infogcn: representation learning for human skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20186–20196 (2022)
17. Lee, J., et al.: Hierarchically Decomposed Graph Convolutional Networks for Skeleton-Based Action Recognition. (2022). arXiv preprint arXiv: 220810741
18. Liu, Z., et al.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, p. 140–149 (2020)
19. Li, M., et al.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 3595–3603 (2019)
20. Li, B., et al.: Spatio-temporal graph routing for skeleton-based action recognition. In: Association for the Advancement of Artificial Intelligence, pp. 8561–8568 (2019)
21. Cao, Y., et al.: Skeleton-based action recognition with temporal action graph and temporal adaptive graph convolution structure. Multimed. Tool. Appl. 80(19), 29139–29162 (2021). https://doi.org/10.1007/s11042-021-11136-z
22. Li, X., et al.: Two-stream adaptive-attentional subgraph convolution networks for skeleton-based action recognition. Multimed. Tool. Appl. 81(4), 4821–4838 (2022). https://doi.org/10.1007/s11042-021-11026-4
23. Song, S., et al.: An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In: Association for the Advancement of Artificial Intelligence, pp. 4263–4270 (2017)
24. Si, C., et al.: An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1227–1236 (2019)
25. Song, Y., et al.: Stronger, faster and more explainable: a graph convolutional baseline for skeleton-based action recognition. In: ACM International Conference on Multimedia. ACM, pp. 1625–1633 (2020)
26. Kim, B., et al.: Global-local motion transformer for unsupervised skeleton-based action learning. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV, pp. 209–225. Springer (2022)
27. Yang, C., et al.: Node, motif and subgraph: leveraging network functional blocks through structural convolution. In: IEEE/ACM 2018 International Conference on Advances in Social Networks Analysis and Mining, ASONAM, pp. 47–52 (2018)
28. Bouritsas, G., et al.: Improving Graph Neural Network Expressivity via Subgraph Isomorphism Counting. CoRR (2020). abs/2006.09252
29. Xuan, Q., et al.: Subgraph networks with application to structural feature space expansion. IEEE Trans. Knowl. Data Eng. 33(6), 2776–2789 (2021). https://doi.org/10.1109/tkde.2019.2957755
30. Tzeng, R., Wu, S.: Distributed, egocentric representations of graphs for detecting critical structures. In: Proceedings of the 36th International Conference on Machine Learning, ICML, vol. 97, pp. 6354–6362 (2019)
31. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. In: Neural Information Processing Systems, pp. 5171–5181 (2018)
32. Alsentzer, E., et al.: Subgraph neural networks. In: Neural Information Processing Systems (2020)
33. Sun, Q., et al.: SUGAR: subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In: International World Wide Web Conference, pp. 2081–2091 (2021)
34. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: Walker, M.A., Ji, H., Stent, A. (eds.) ACL. Association for Computational Linguistics, pp. 464–468 (2018)
35. Wu, K., et al.: Rethinking and improving relative position encoding for vision transformer. In: IEEE/CVF International Conference on Computer Vision, ICCV, pp. 10013–10021. IEEE (2021)
36. You, J., Ying, R., Leskovec, J.: Position-aware graph neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, ICML. vol. 97 of Proceedings of Machine Learning Research, pp. 7134–7143. PMLR (2019)
37. Yan, R., et al.: Position-aware participation-contributed temporal dynamic model for group activity recognition. IEEE Transact. Neural Networks Learn. Syst. 33(12), 7574–7588 (2021). https://doi.org/10.1109/tnnls.2021.3085567
38. Dwivedi, V.P., et al.: Graph Neural Networks with Learnable Structural and Positional Representations. ICLR (2022)
39. Chen, Y., et al.: Channel-Wise topology refinement graph convolution for skeleton-based action recognition. In: IEEE International Conference on Computer Vision, ICCV (2021)
40. Shahroudy, A., et al.: NTU RGB+D: a large scale dataset for 3D human activity analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1010–1019 (2016)
41. Liu, J., et al.: NTU RGB+D 120: a large-scale benchmark for 3D human activity understanding. IEEE Trans. Pattern Anal. Mach. Intell. 42(10), 2684–2701 (2020). https://doi.org/10.1109/tpami.2019.2916873
42. Wang, J., et al.: Cross-view action modeling, learning, and recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2649–2656 (2014)
43. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1110–1118 (2015)
44. Liu, J., et al.: Spatio-temporal LSTM with trust gates for 3D human action recognition. In: The European Conference on Computer Vision ECCV, pp. 816–833 (2016)
45. Shi, L., et al.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 12026–12035 (2019)
46. Shi, L., et al.: Skeleton-based action recognition with directed graph neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 7912–7921 (2019)
47. Zhang, P., et al.: Semantics-guided neural networks for efficient skeleton-based human action recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1109–1118. CVPR (2020)
48. Cheng, K., et al.: Decoupling GCN with DropGraph module for skeleton-based action recognition. In: ECCV, pp. 536–553 (2020)
49. Ye, F., et al.: Dynamic GCN: context-enriched topology learning for skeleton-based action recognition. In: ACM MM, pp. 55–63 (2020)
50. Song, Y., et al.: Richly activated graph convolutional network for robust skeleton-based action recognition. IEEE Trans. Circ. Syst. Video Technol. 31(5), 1915–1925 (2021). https://doi.org/10.1109/tcsvt.2020.3015051

51. Yang, S., et al.: Skeleton Cloud Colorization for Unsupervised 3D Action Representation Learning (2021)
52. Su, Y., Lin, G., Wu, Q.: Self-supervised 3D skeleton action representation learning with motion consistency and continuity. In: International Conference on Computer Vision (2021)
53. Chen, Z., et al.: Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI, pp. 1113–1122 (2021)
54. Zeng, A., et al.: Learning skeletal graph neural networks for hard 3D pose estimation. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 11416–11425 (2021)
55. Song, Y.F., et al.: Constructing stronger and faster baselines for skeleton-based action recognition. IEEE Trans. Pattern Anal. Mach. Intell. 45(2), 1474–1488 (2022). Available from. https://doi.org/10.1109/TPAMI.2022.3157033
56. Veeriah, V., Zhuang, N., Qi, G.: Differential recurrent neural networks for action recognition. In: 2015 IEEE International Conference on Computer Vision, ICCV, pp. 4041–4049 (2015)
57. Lee, I., et al.: Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks. In: IEEE International Conference on Computer Vision, ICCV, pp. 1012–1020 (2017)
58. Zhang, P., et al.: Semantics-guided neural networks for efficient skeleton-based human action recognition. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020). Available from. https://doi.org/10.1109/cvpr42600.2020.00119

How to cite this article: Yang, P., et al.: Position-aware spatio-temporal graph convolutional networks for skeleton-based action recognition. IET Comput. Vis. 1–11 (2023). https://doi.org/10.1049/cvi2.12223