Information Enhanced Graph Convolutional Networks for Skeleton-based Action Recognition

Dengdi Sun, Fanchen Zeng, Bin Luo, Jin Tang
Anhui Provincial Key Laboratory of Multimodal Cognitive Computation
School of Computer Science and Technology
Anhui University
Hefei, 230601, China
{sundengdi, zfcfuture, luobinahu}@163.com, ahhftang@gmail.com

Zhuanlian Ding*
School of Internet
Anhui University
Hefei, 230039, China
dingzhuanlian@163.com

***Abstract*—Skeleton-based action recognition has recently attracted much attention in computer vision. The latest methods are mostly based on graph convolutional networks (GCNs), which construct the human body as spatial-temporal Skeleton graphs, and has achieved excellent performance. However, previous studies only capture the local and rough information based on the physical dependencies among joints, which may miss implicit joint correlations. In this work, we propose a novel action recognition model, namely Information Enhanced Graph Convolutional Networks (IE-GCN). To improve the accuracy and robustness of recognition, this model capture higher-order dependency in the skeleton-based graph by expanding the joint neighbors, and combine second stage skeleton features (the lengths and directions of bones) to enhance the discriminative information simultaneously. In addition, an training strategy is designed to solve the framework. Extensive experiments on two large-scale public datasets, NTU-RGBD and Kinetics-Skeleton, demonstrate the superior performance of the proposed algorithms over the state-of-the-art methods.**

***Index Terms*—action recognition, graph convolutional networks, skeleton graph, high-order dependency, information enhance**

I. Introduction

Human action recognition, or HAR for short, is aimed at identifying the specific movement or action of a person based on visual sensor data. It is a fundamental and active problem in computer vision, and has a wide range of applications such as visual surveillance, video indexing, human-computer interaction (HCI), and autonomous driving [1]–[3]. Although traditional studies on action recognition mainly focus on recognizing actions in pixel matrices of video frames, articulated human pose, also referred to as skeleton can capture the dynamic structure of human actions, and actually provide more comprehensive information. These dynamic skeleton data could be represented by a time series of joint positions using depth sensors or pose estimation algorithms on videos [4], in the form of 2D or 3D spatial coordinates. Compared with RGB pixels, skeleton data are more robust to noise like background and irrelevant objects. Therefore, spatiotemporal skeleton based representations of the human body and its dynamic evolution has become an attractive option for action recognition.

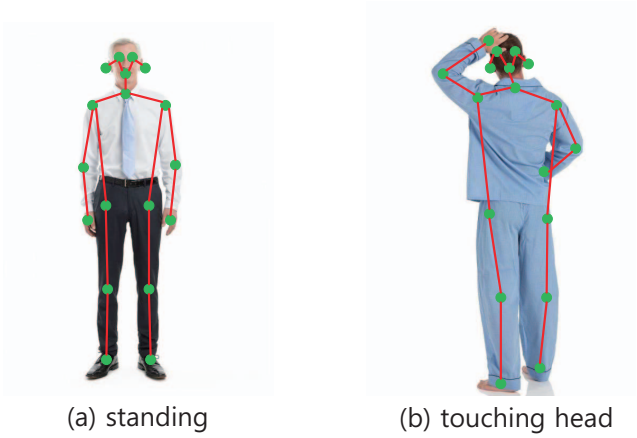

Fig. 1. For action "standing" in (a), it can be well represented by the physical structure based skeleton graph. But for action "touching head" in (b), hand and head are strong associated.

The earliest method of skeleton action recognition simply employ the positions of body joints on each frame to form feature vectors for pattern learning [5]–[8]. Obviously, these vector sequence representations seriously destroy the naturally global dependencies between human joints. In order to address this problem, a skeleton graph with vertices as joints and bones as edges is constructed in recent years (see Figure 1(a)), and introduced into graph convolutional networks (GCNs) to extract related features [9]. However, the common GCNs only can preserve the spatial structure of body joints to some extent,

D. Sun, F. Zeng, B. Luo and J. Tang are with Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, School of Computer Science and Technology, Anhui University, Hefei, China, 230601, Email: sundengdi@163.com, zfcfuture@163.com, luobinahu@163.com, ahhftang@gmai.com

Z. Ding. is with School of Internet, Anhui University, Hefei, China, 230039, Email: dingzhuanlian@163.com. (**Corresponding author is Zhuanlian Ding**)

This work was supported by the Guangdong Province Science and Technology Plan Projects (2017B010110011), the Key Natural Science Project of Anhui Provincial Education Department (KJ2018A0023), the Anhui Key Research and Development Plan (1804a09020101), the National Basic Research Program (973 Program) of China (2015CB351705) and the National Natural Science Foundation of China (61906002, 61402002, 61876002 and 61860206004).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

but the temporal dependencies of sequential skeletons are still ignored. Yan et al. [10] first propose the spatiotemporal graph convolutional networks (ST-GCN) to model skeleton data which can learn spatial and temporal features simultaneously. They construct a complete spatiotemporal graph by the natural connection between body joints on a single frame and the addition of temporal edge between the corresponding joints in consecutive frames. In addition, a distance-based sampling function is proposed for building the graph convolutional layer, which is then employed as a basic module to make the final prediction.

Unfortunately, there are still some disadvantages in ST-GCN [10]. Intuitively, the skeleton graph used in ST-GCN only depend on the physical structure of the human body, but the movements of human beings may break the restriction of natural skeletal connections. As shown in Fig. 1(b), in the gesture of "touching head", the hand (vertex) should link with the head (vertex). So it is difficult for ST-GCN to capture the dependency between the hand and head since they are located far away from each other in the physical structure based skeleton graph. Moreover, when human beings do different actions, the movement response area is usually extent for a long distance along the limbs and body. So it is significant to capture generalized higher-order adjacency information across joints. In ST-GCN, each vertex only propagates information to the neighbors, which results in a small perceived field of vertices, and is not conducive to obtaining long-distance connection information. Thus, by increasing the high order neighbors of the vertices, we can enlarge the receptive field and aggregate more discriminative information.

Inspired by Shi et al. [11], we note that the discriminative information that is important for classification exists not only in the joints, but also in the bones that connect the joints. As used in ST-GCN, the direct adjacency can be regard as the first-stage information of the skeleton data, whereas the second-stage information, which represents the feature of bones between two joints, is not exploited. For the skeleton-based human action recognitor task, the most common and useful second-stage information is the length and direction of the bones. Similar to [11], the second-stage information can be formulated as a vector pointing from the source joint to the target joint. Therefore, we can try to use the two-steam framework to fuse these two different types of discriminative data to enhance the information of our model and further improve the performance. To verify the effectiveness of the proposed model, namely, information enhanced graph convolutional network (IE-GCN), we conduct extensive experiments on two large-scale public datasets: NTU-RGBD [12] and Kinetics-Skeleton [13]. The experiments have demonstrated that IE-GCN outperforms the state-of-the-art approaches in skeleton-based action recognition.

Our main contributions in this work are summarized as follows:

- We expand the receptive field of vertices so that nodes can aggregate to a wider range of information and capture higher-order dependencies.
- The second-stage information of the skeleton data is explicitly formulated and combined with the high-order information through a two-stream framework, which bring notable improvement for the recognition performance.
- The proposed IE-GCN achieved state-of-the-art performance on two large-scale public datasets for skeleton-based action recognition.

II. Related Work

A. Skeleton-based action recognition

In the early days, a general strategy to deal with skeleton-based action recognition is to formulate the human body with designed handcrafted features [6], [8]. However, the performance of these methods based on manual features is not satisfactory since the factors considered are too segmentary. With the development of deep learning, data-driven methods such as RNNs and CNNs are increasingly becoming mainstream. Most recurrent-neural-network (RNN)-based method usually model the skeleton data as a sequence of the coordinate vectors to capture the temporal dependencies [14]–[16], such as bi-RNNs [5], deep LSTMs [17], and attention-based method [18]. On the other hand, convolutional neural networks (CNN)-based method achieve remarkable results by treat the skeleton data as a kind of pseudo-images based on the pre-designed transformation rules [19]–[21]. Moreover, compared with RNNs architecture, CNN-based methods are easier to train and have better parallelizability, so CNN-based methods tend to be more popular.

B. Graph convolutional neural networks

This work is related to the graph convolutional networks (GCNs), which is a special type of graph neural network (GNN) [22] and achieved remarkable performance on different tasks. Generally, the principle of constructing GCNs mainly follows two ways: spatial perspective and spectral perspective [9], [23]–[29]. Spatial perspective methods directly perform the convolution filters on the graph vertices and their neighbors, which are extracted and normalized based on manually designed rules [26], [27], [29], [30]. Different from the spatial perspective methods, in spectral perspective, the graph convolution is converted into the operator of spectral analysis. These methods perform the graph convolution in the frequency domain with the help of the graph Fourier transform [31], which does not need to extract locally connected regions from graphs at each convolutional step [24], [25], [28], [29]. Considering the important spatial relationships between the vertices in skeleton graph, here we follow the spirit of the spatial perspective.

III. Spatiotemporal Graph Convolutional Network

In this section, we will first introduce the spatiotemporal graph convolutional network which is closely associated with our model.

A. Graph construction

According to the spatial perspective, raw skeleton data in one frame is always represented as a sequence of vectors. Each vector represents the 2D or 3D coordinates of the corresponding human joint. Usually, a complete action consists of consecutive frames in the video, and for different samples (videos), the frame lengths are also different. Yan et al. [10] designed a complete spatiotemporal graph to model the structure information among joints in both temporal and spatial dimensions. The left sketch in Figure 2 represents a completed spatiotemporal graph based on the skeleton data, where the joints are represented as vertices and their natural connections in the human body are represented as spatial edges (the blue lines in Fig. 2, left). In addition, There are also another type of edges in the temporal dimension and consists of a connection of corresponding joints between adjacent frames, namely temporal edge (the green lines in Fig. 2, left). The feature of each joint is the coordinate vector of its corresponding vertex and the confidence.

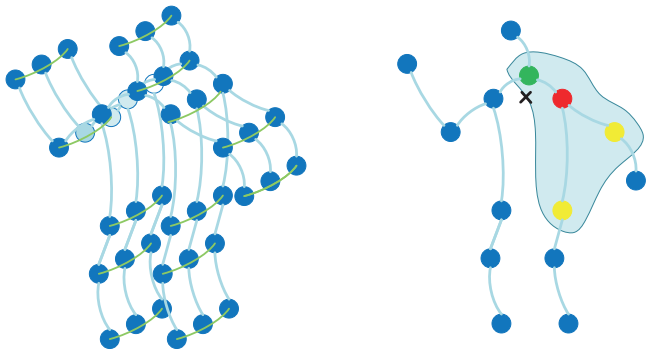

Fig. 2. (a). Illustration of the spatiotemporal graph used in ST-GCN. (b). Illustration of the mapping strategy. Different colors denote different subsets.

B. Graph convolution

According to the graph defined above, multiple layers of spatiotemporal graph convolution operations are plugged in the graph to extract the high-level features. Then, the global average pooling layer and the softmax classifier are employed to make the final action discrimination based on the extracted high-level features.

In the spatial dimension, the graph convolution operation on vertex v_i is formulated as [10]:

$$f_{out}(v_i) = \sum_{v_j \in B_i} \frac{1}{Z_{ij}} f_{in}(v_j) \cdot w(l_i(v_j)) \tag{1}$$

where f_{in} and f_{out} are the input and output feature map, respectively. B_i denotes the size of the sampling area needed to convolution for the vertex v_i, which is usually defined as 1-distance neighbor vertices (v_j) of the target vertex v_i. $l_i(\cdot)$ denotes the mapping function which maps a vertex in the neighborhood to its subset label, and will be detailed later. w are the trainable weights for each partition group to capture. Note that the number of convolution kernel (weight vectors) are fixed, yet the number of vertices in different sampling area (B_i) is unfixed.

In order to realize the operation similar to image convolution operation, each vertex is assigned a unique weight vector, the mapping function $l_i(\cdot)$ is designed specially in ST-GCN [10]. The right sketch in Figure 2 shows this special partitioning strategy, called special configuration partitioning. The $\times$ (black cross) and B_i (the blue area enclosed by curve) represent the center of gravity of the skeleton and sampling area, respectively. Specifically, the spatial strategy empirically sets the convolution kernel size as 3 and naturally divides B_i into 3 subsets: B_{i1} is the root vertex ((the red point in Fig. 2, right); B_{i2} represent the subset of centripetal vertexes, which contains the neighboring vertexes have shorter distance to the center of gravity than root vertex (the green point); B_{i3} represent the subset of centrifugal vertices, which contains the neighboring vertices have longer distance to the center of gravity than root vertex (the yellow point). Finally, Z_{ij} represents the size of the cardinality of vertexes in different subsets (B_{ik}) and it aim to balance the contribution of each subset.

C. Implementing ST-GCN

A complete spatiotemporal GCN (ST-GCN) consists of a series of the ST-GCN blocks, and each block contains a spatial graph convolution followed by a temporal convolution, which extracts spatial and temporal features simultaneously. However, as the key component in ST-GCN, spatial graph convolution operation is not straightforward to implement. In detail, let f_{in} be the input feature maps for all joints in one frame and formulated as a $C \times N \times T$ tensor, where C denotes the number of channels, N and T denotes the number of vertices and frames respectively. To implement the spatial graph convolution, Eq.1 is transformed into:

$$f_{out} = \sum_{s \in S} W_{st}^{(s)} (f_{in} A^{(s)}) \odot M_{st}^{(s)} \tag{2}$$

where S denotes the kernel size of the spatial dimension. With the partition strategy designed above, S is set to 3 corresponding to the subsets of the three types. $A^{(s)} = D^{(s)^{-\frac{1}{2}}} \Omega^{(s)} D^{(s)^{-\frac{1}{2}}}$ is the normalized adjacent matrix for each partition group, where $\Omega^{(s)}$ is a indicator matrix with same size of $N \times N$ adjacent matrix, and its element $\Omega_{ij}^{(s)}$ indicates whether the vertex v_j is in the corresponding subset B_{ik} of vertex v_i. $D^{(s)^{-\frac{1}{2}}}$ is the normalized diagonal matrix, $\odot$ denotes the Hadamard product. $M_{st}^{(s)}$ and $W_{st}^{(s)}$ are trainable weights for each partition group to capture edge weights and feature importance, respectively.

The convolution operation on the temporal dimension is much easier than spatial to implement, the number of neighbors of each vertex is fixed to 2 (temporal edge only exist in the two consecutive frames). With the feature map calculated above, the $K_t \times 1$ convolution operation can be directly applied to the temporal dimension, where K_t denotes the kernel size.

IV. Information Enhanced Graph Convolutional Network

In this section, we introduce the components of our proposed information enhanced graph convolutional network (IE-GCN) in detail.

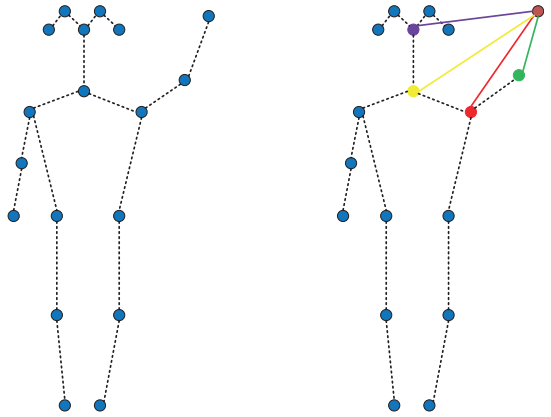

Fig. 3. The left is the original skeleton data graph defined in ST-GCN, and the right is the graph which we amplified the vertex neighbors.

A. IE graph convolution layer

As shown in Eq. 2, $f_{in}A^{(s)}$ only aggregates the 1-hop neighbor information in skeleton graph, which may not be the best choice as explained in Sec. III. In order to obtain the high-order dependencies and long-range links (see Fig. 3), we use the high-order polynomial of A, indicates augmented adjacency matrix (A denotes the raw $N \times N$ adjacent matrix, which fully describes the natural skeleton structure). Correspondingly, we set the convolution kernel of the graph to be $\hat{A}^L$, where $\hat{A}^L = D^{-1}A$ is the graph transition matrix and L is the polynomial order. With the L-order polynomial, we expand the raw skeleton graph defined in ST-GCN into a higher order structure, which can directly establish a connection to L-hop neighbors for enlarging the receptive field and aggregating more discriminative information. Based on the ideas above, we can change Eq. 2 into the following form:

$$f_{out} = \sum_{l=1}^{L} \sum_{s \in S} W_{st}^{(s,l)} (f_{in} \hat{A}^{(s,l)}) \odot M_{st}^{(s,l)} \tag{3}$$

where l denotes the polynomial order, S followed the same partitioning strategy in ST-GCN (see Eq. 2), $\hat{A}^{(s)}$ is the graph transition matrix for s-th parted graph. $M_{st}^{(s,l)}$ and $W_{st}^{(s,l)}$ are trainable weights to capture edge weights and feature importance. Apparently, more important features will be assigned larger weight. The weights are evenly distributed, which introduced for each subgraph of different partition and each polynomial order. It is worth noting that, in order to stabilize the learning of $M_{st}^{(s,l)}$, the graph transition matrix $\hat{A}^{(s)}$ and degree normalization provide a naturally good initialization for edge weights. The value of L are also flexible, when L = 1, our model perform the same operation with the original spatial graph convolution in [10]. For $L > 1$, our model acts like the Chebyshev filter and is able to approximate the convolution designed in the graph spectral domain [32].

B. IE graph convolution block

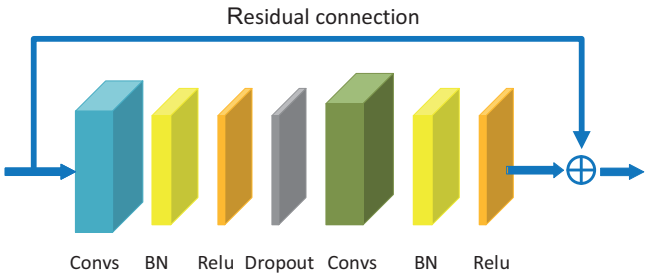

Fig. 4. Illustration of the information enhanced graph convolution block. The blue Convs represents the spatial GCN and the green Convs represents the temporal GCN, both of which are followed by a BN layer and a ReLU layer. At the same time, a residual connection is added for each block.

With the above description of spatial graph convolution, for temporal dimension, we perform the similar operation with ST-GCN. In detail, a batch normalization (BN) layer and a ReLU layer is added for both the spatial GCN and temporal GCN. As shown in Figure 4, a complete graph convolution block consists of a spatial GCN (blue Convs), a temporal GCN (green Convs) and an additional dropout layer with the drop rate set as 0.5. Similar to [10], in order to get a stable training processing, we add a residual connection for each block.

C. IE graph convolutional network

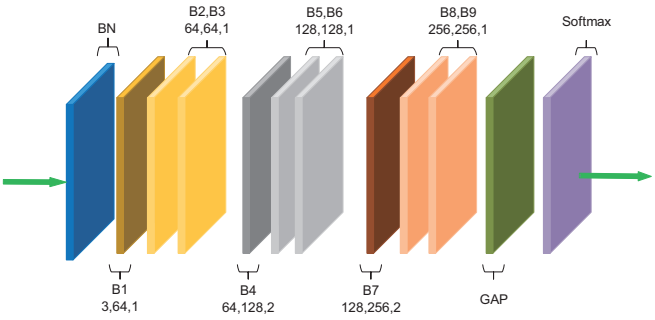

Fig. 5. Illustration of the IE-GCN, our network framework consists of 9 blocks (B1 - B9). The three numbers under each block denotes the number of input channels, the number of output channels and stride, respectively. We replaced the full connection layer with global average pooling and employ $softmax$ at the final layer to calculate the classification probability.

The information enhanced graph convolutional network (IE-GCN) is the stack of these basic blocks as shown in Figure 5, and is composed of 9 blocks of spatial temporal graph convolution operators. Although the movement stages of human beings are few, the movements in each stage are relatively

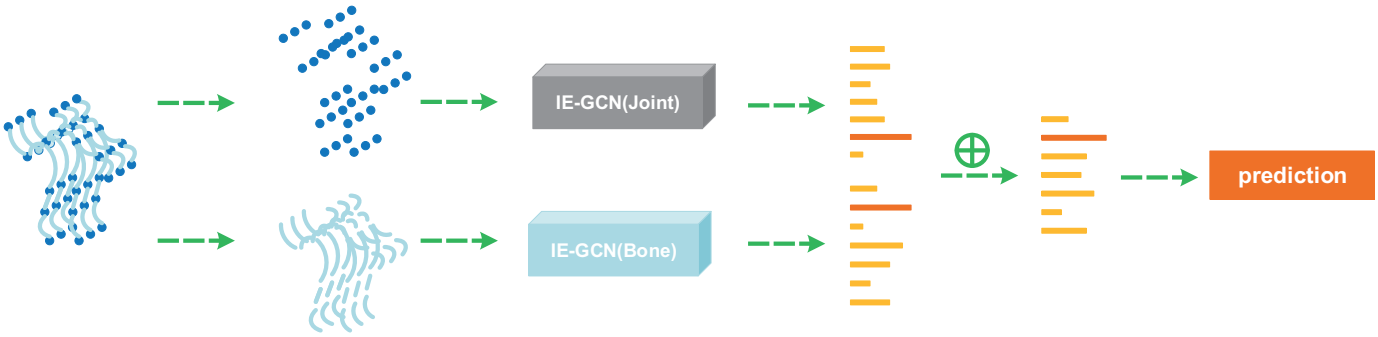

Fig. 6. Illustration of the overall pipeline of the IE-GCN. The scores of two branches are added to obtain the final prediction.

complicated. In order to obtain higher feature dimensions, we adjusted the number of channels in each block. Specifically, we set the number of output channels as 64 for the first three blocks, 128 for the middle three blocks and 256 for the last three blocks. Moreover, we also add a data BN layer at the beginning of the network to normalize the input data and help the model better convergence. Moreover, we also replaced the full connection layer with global average pooling and employ $softmax$ at the final layer to calculate the classification probability.

D. Second-stage information for IE-GCN

The purpose of this work is to extract more discriminative information to enhance the recognition capability of our model. As described above, we can capture the long-distance dependencies among joints and aggregate more information. However, the information available is still limited, so we need to further consider the information from the second stage mentioned in Section 1, namely, the bone information.

Empirically, the most convenient and useful second-stage information is the length and direction of the bones. In detail, each bone is joined by two joints, similar to [11], the joint close to the center of gravity of the skeleton is defined as source joint, otherwise, target joint. Each bone is represented as a vector pointing to its target joint from its source joint, which contains not only the length information, but also the direction information. For instance, the 3D coordinates of a given source joint $\mathbf{v_1}$ and target joint $\mathbf{v_2}$ are represented as (x_1, y_1, z_1) and (x_2, y_2, z_2), respectively. The vector of the bone is calculated as $\mathbf{e}_{(v_1,v_2)} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$.

To enable our model to process two types of data on two branches simultaneously, we conduct some preprocessing for input second stage data. Considering that there is no cycles in human skeleton data, theoretically every bone only can be assigned a unique source joint. However, the number of joints is one more than the number of bones because the central joint is not assigned to any bones. To solve this problem, we add an empty bone with its value as 0 to the central joint. By this means, we can make bone data and joint data share the ways to construct graphs and graph convolutional networks. Then, we use the networks of joints and bones to process the data of joints and bones, respectively. The overall pipeline of IE-GCN is shown in Figure 6. In short, we extract bone data and joint data for a given sample, and then the two types data are fed into the corresponding network to get the results of their respective classifications. Finally, the $softmax$ classifier is added to obtain the fused score and recognize the action label.

V. Experiments

To verify the effectiveness of proposed IE-GCN, we conduct our experiments on two large-scale human action recognition datasets: NTU-RGBD [12] and Kinetics-Skeleton [13]. Compared to state-of-the-arts, the experimental results demonstrate the powerful recognition ability of our model.

A. Dataset

Kinetics-Skeleton: Kinetics [13] is a large dataset derived from YouTube, containing about 300,000 video clips in 400 action classes. Since it only provides video in RGB mode, [10] estimate the locations of 18 joints on every frame of the clips using the publicly available OpenPose toolbox [4]. We represent each joint as 3-element feature vector: $[x, y, c]^T$, where x, y denotes the 2D pixel coordinates and c denotes the confidence score. For the multiple-person cases, we select the man with the highest average joint confidence in each video clip. Therefore, a tensor $[3, 18, T]$ can be represented a clip with T frames. And the dataset is divided into a training set (240,000 clips) and a validation set (20,000 clips). After training, we use the same evaluation method (top1 and top5) as [10] on the validation set to report the accuracy.

NTU-RGBD: NTU-RGBD [12] is currently the largest skeleton-based action dataset which completed by one or two performers indoors. This dataset contains 56, 000 action clips in 60 action classes. And each human in an action is represented by the 3D spatial coordinates of 25 joints which detected by Kinect depth sensors. To better evaluate the robustness of the model, two benchmarks are recommended: Cross-Subject and Cross-View. In the Cross-subject benchmark, 40, 320 samples performed by 20 subjects are separated into training set, and the rest belong to test set. For another, the training set contains 37,920 videos that are captured by cameras 2 and 3, and the validation set contains 18,960 videos that are captured by camera 1. We follow this convention and report the top-1 accuracy on both benchmarks.

B. Traning phase

We choose PyTorch 1.0 as our basic deep learning framework to conduct all experiments and train the model on the 4 RTX-2080Ti GPUs. We use Stochastic Gradient Descent (SGD) with Nesterov momentum (0.9) as our optimizer. The batch size is 64. Cross-entropy is selected as the loss function to back propagate gradients. The weight decay is set to 0.0001. For the Kinetics-Skeleton dataset, we set the same size of input tensor as [10], which contains 150 frames and 2 men in each frame. We also conduct the same data-augmentation strategy as in [10]. The maximum training epoch is set to 65, the initial learning rate is 0.1 and divided by 10 at the 45th and 55th epoch. For the NTU-RGBD dataset, there are at most two people in each sample of the dataset. When there is only one person in the sample, we set the second person to 0. Different from Kinetics, here the max number of frames in each sample is 300. For samples with less than 300 frames, we repeat the samples until it reaches 300 frames. The maximum training epoch is set to 50, the initial learning rate is 0.1 and divided by 10 at the 30th and 40th epoch.

C. Ablation Study

To examine the effectiveness of each individual component of IE-GCN, we conduct extensive experiments on Cross-Subject benchmark and Cross-View benchmark of the NTU-RGB+D dataset.

TABLE I
COMPARISON RESULTS OF DIFFERENT l ON THE CROSS-SUBJECT BENCHMARK OF THE NTU-RGBD DATASET

Polynomial order	Accuracy (%)
1	81.5%
2	83.8%
3	84.5%
4	86.3%

Effect of high-order. Here we focus on validating the proposed high-order dependency. In the experiment, we respectively set the polynomial order L = 1, 2, 3, 4 in the model. Note that when $L = 1$, the corresponding graph is exactly the skeleton itself (raw graph in ST-GCN). Table 1 shows the classification accuracy of action recognition. From Table 1, we can see that (1) high-order dependencies can further improve the performance; (2) when $L = 4$, we achieve the best performance; (3) These results validate the limitation of the original skeleton graph and the effectiveness of the proposed high-order dependency.

TABLE II
COMPARISON RESULTS OF DIFFERENT INPUT MODALITIES ON THE CROSS-VIEW BENCHMARK OF THE NTU-RGBD DATASET

Methods	Accuracy (%)
IE-GCN (Joint)	93.5%
IE-GCN (Bone)	93.2%
fusion	95.0%

Effect of second-stage: Another important information of skeleton data is the second-stage information, namely, the bone information. Here, we compare the performance of using each type of input data alone, shown as IE-GCN (Joint) and IE-GCN (Bone) in Table 2, and the performance when combining them shown as fusion in Table 2. Clearly, the two-branch-fusion method outperforms the one-branch-based methods.

D. Comparison with the state-of-the-art

Our proposed model achieves competitive performance with current state-of-the-art methods on both the NTU-RGBD dataset and Kinetics-Skeleton dataset. The results of these two comparisons are shown in Tabel 3 and Tabel 4, respectively. We can see that IE-GCN outperforms the other competitive methods in both top-1 and top-5 accuracies.

TABLE III
COMPARISONS OF THE VALIDATION ACCURACY WITH STATE-OF-THE-ART METHODS ON THE NTU-RGBD DATASET.

Methods	X-Sub (%)	X-View (%)
STA-LSTM [18] (AAAI 17)	73.4	81.2
Two-Stream 3DCNN [33] (CVPR 17)	66.8	72.6
Clips+CNN+MTLN [19] (CVPR 17)	79.6	84.8
VA-LSTM [34] (AAAI 18)	79.2	87.7
ST-GCN [10] (AAAI 18)	81.5	88.3
Ind-RNN [16] (CVPR 18)	81.8	88.0
DPRL+GCNN [35] (CVPR 18)	83.5	89.8
SR-TSL [36] (ECCV 18)	84.8	92.4
HCN [37] (IJICA 18)	86.5	91.1
ARRN-LSTM [15] (ICME 19)	80.7	88.8
Bayesian-LSTM [38] (ICCV 19)	81.8	89.0
AS-GCN [39] (CVPR 19)	86.8	94.2
2s-AGCN [11] (CVPR 19)	88.5	**95.1**
IE-GCN (ours)	**89.2**	95.0

TABLE IV
COMPARISONS OF THE VALIDATION ACCURACY WITH STATE-OF-THE-ART METHODS ON THE KINETICS-SKELETON DATASET.

Methods	Top-1 (%)	Top-5 (%)
Feature Enc [6]	14.9	25.8
Deep LSTM [12]	16.4	35.3
TCN [20]	20.3	40.0
ST-GCN [10]	30.7	52.8
SLnL-rFA [40]	36.6	59.1
DGNN [41]	**36.9**	**59.6**
IE-GCN (ours)	35.0	57.6

VI. CONCLUSION

In this work, we propose a novel information enhanced graph convolutional network (IE-GCN) for skeleton-based action recognition. We expand the high order neighbor of the vertex to enlarge the receptive field and aggregate more discriminative information. Furthermore, the traditional methods always ignore or underestimate the importance of second-stage information of skeleton data, i.e., the bone information. For better performance, we fuse the results of two complementary information, which further enhances the proposed model. The final prediction is evaluated on two large-scale action recognition datasets, NTU-RGBD and Kinetics, and it achieves the state-of-the-art performance on both of them.

REFERENCES

[1] Z. Duric, W. D. Gray, R. Heishman, F. Li, A. Rosenfeld, M. J. Schoelles, C. Schunn, and H. Wechsler, "Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1272–1289, 2002.

[2] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, "A string of feature graphs model for recognition of complex activities in natural videos," in *2011 International Conference on Computer Vision*, pp. 2595–2602, IEEE, 2011.

[3] M. Sudha, K. Sriraghav, S. G. Jacob, S. Manisha, *et al.*, "Approaches and applications of virtual reality and gesture recognition: A review," *International Journal of Ambient Computing and Intelligence (IJACI)*, vol. 8, no. 4, pp. 1–18, 2017.

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291–7299, 2017.

[5] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1110–1118, 2015.

[6] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5378–5387, 2015.

[7] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition*, vol. 68, pp. 346–362, 2017.

[8] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 588–595, 2014.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[10] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12026–12035, 2019.

[12] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1010–1019, 2016.

[13] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[14] C. Cao, C. Lan, Y. Zhang, W. Zeng, H. Lu, and Y. Zhang, "Skeleton-based action recognition with gated convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[15] W. Zheng, L. Li, Z. Zhang, Y. Huang, and L. Wang, "Skeleton-based relational modeling for action recognition," *arXiv preprint arXiv:1805.02556*, 2018.

[16] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (indrnn): Building a longer and deeper rnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5457–5466, 2018.

[17] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *European Conference on Computer Vision*, pp. 816–833, Springer, 2016.

[18] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[19] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3288–3297, 2017.

[20] 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)*, pp. 1623–1631, IEEE, 2017.

[21] 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)*, pp. 601–604, IEEE, 2017.

[22] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[23] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1993–2001, 2016.

[24] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[25] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, pp. 3844–3852, 2016.

[26] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, pp. 2224–2232, 2015.

[27] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

[28] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[29] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," *arXiv preprint arXiv:1802.04687*, 2018.

[30] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5115–5124, 2017.

[31] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[32] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[33] H. Liu, J. Tu, and M. Liu, "Two-stream 3d convolutional neural network for skeleton-based action recognition," *arXiv preprint arXiv:1705.08106*, 2017.

[34] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2117–2126, 2017.

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

[36] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, "Skeleton-based action recognition with spatial reasoning and temporal stack learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 103–118, 2018.

[37] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," *arXiv preprint arXiv:1804.06055*, 2018.

[38] R. Zhao, K. Wang, H. Su, and Q. Ji, "Bayesian graph convolution lstm for skeleton based action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6882–6892, 2019.

[39] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3595–3603, 2019.

[40] G. Hu, B. Cui, and S. Yu, "Skeleton-based action recognition with synchronous local and non-local spatio-temporal learning and frequency attention," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1216–1221, IEEE, 2019.

[41] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7912–7921, 2019.