# JT-MGCN: Joint-temporal Motion Graph Convolutional Network for Skeleton-Based Action Recognition

Suekyeong Nam
Computer Sience and Engineering
Kyung hee University
Republic of Korea
Email: akasha989@khu.ac.kr

Seungkyu Lee
Computer Sience and Engineering
Kyung hee University
Republic of Korea
Email: seungkyu@khu.ac.kr

***Abstract*—Recently, action recognition methods using graph convolutional networks (GCN) have shown remarkable performance thanks to its concise but effective representation of human body motion. Prior methods construct human body motion graph building edges between neighbor or distant body joints. On the other hand, human action contains lots of temporal variations showing strong temporal correlations between joint motions. Thus the characterization of an action requires a comprehensive analysis of joint motion correlations on spatial and temporal domains. In this paper, we propose Joint-temporal Motion Graph Convolutional Networks (JT-MGCN) in which joint-temporal edges learn the correlations between different joints at different time. Experimental evaluation on large public data sets such as NTU rgb+d data set and kinetics-skeleton data set show outstanding action recognition performance.**

## I. Introduction

Human action recognition has been studied in many computer vision topics such as human machine interaction, behavior and situation understanding, medical diagnosis, and video surveillance. Traditionally, action recognition has been performed on RGB-video data extracting and tracking human body based on visual appearance clues [1]. Recently, many skeleton-based action recognition methods have been proposed using 3D sensors like Microsoft Kinect which is able to obtain human body joints and their clean movement information directly. Traditionally, action recognition on body skeleton data has been performed with handcrafted motion features. For example, lie group represents a set of skeleton changes as a single point following lie algebra for action classification. On the other hand, deep learning methods have been proposed in action recognition showing outstanding performance on skeleton data. Action recognition has been implemented based on diverse deep neural networks [2]. Algorithms are largely divided into RNN-based [3]–[16], CNN-based [14], [17]–[24] and GCN based [10], [15], [16], [25]–[29], as shown in table I. Long-Short term memory (LSTM) based deep learning networks have been successfully performed recognition tasks with sequential data such as human behavior, speech, and sentences of writings. There are many prior work using LSTM for action recognition I. Zhu et.al. [30] propose to incorporate LSTM for observing the co-occurrences of joints in human actions. Zhang et.al. [7] propose a novel view adaptation scheme to automatically regulate observation viewpoints during the occurrence of an action using LSTM architecture to address large view variation problem in human action. On the other hand, LSTM shows limited performance in extracting the correlation between two joints within a frame or separated frames. In order to address the problem, there are several approaches converting body movement information to 2D image to apply CNN based deep network. Joint Trajectory Map(JTM) [17] is proposed to express motions as images and is used as enter map for CNN to classify action. Ke

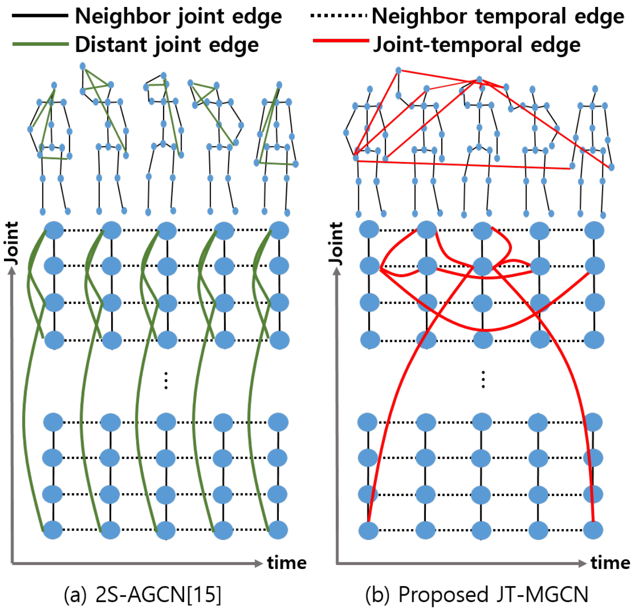


Fig. 1. Joint-temporal motion graphs with and without joint-temporal edges: Neighbor Joint Edge (NJ Edge), Neighbor Temporal Edge (NT Edge), Distant Joint Edge (DJ Edge), Joint-temporal Edge (JT Edge) (a) Joint-temporal motion graph of existing methods with NJ edges, NT edges, and DJ edges (b) Joint-temporal motion graph of our method with NJ edges, NT edges, and JT edges

| | HBRNN[3] | PA-LSTM[4] | ST-LSTM+TG[5] | GCA-LSTM[6] | VA-LSTM[7] | STA-LSTM[8] | MTLN[18] | 2S 3D CNN[19] | Res-TCN[20] | MS deep CNN[21] | MS CNN[22] | Motion+Trans[13] | ST-NBMIM[9] | DPRN+GCNN[25] | SR-TSL[10] | ARRN-LSTM[11] | IndRNN[12] | AE-LSTM[14] | MTCNN[24] | ST-GCN[26] | AS-GCN[27] | AGC-LSTM[28] | 2S-AGCN[15] | DGNN[29] | BGC-LSTM[16] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RNN | v | v | v | v | v | v | | | | | | | v | | v | v | v | v | | | | v | | | v |
| CNN | | | | | | | v | v | v | v | v | v | | | | | | v | v | | | | | | |
| GCN | | | | | | | | | | | | | | v | v | | | | | v | v | v | v | v | v |
| | | 2016 | | 2017 | | | | | | | | | 2018 | | | | | | | | 2019 | | | | |

TABLE I
3D SKELETON-BASED ACTION RECOGNITION ALGORITHMS USING EACH NETWORKS.

et.al. [18] propose to transform each skeleton sequence into 2D image clips for spatio-temporal feature learning. They apply deep convolutional neural networks to extract motion features. And Multi-Task Learning Network (MTLN) is used to jointly process all frames of extracted features in parallel to incorporate spatial structural information. Their improved method [24] learns optimal convolutional network model for training features with motion clips. However, these CNN-based approaches have limitation in extracting dependencies between body joints located away to each other.

Recently, graph convolutional networks (GCN) has been successfully adopted for action recognition in many methods. Human body skeleton is represented as a structure of graph. Body joints and parts are represented by graph nodes and edges connecting two neighbour joints, respectively. Thanks to its concise but effective representation of human body motion, action recognition methods using GCN have shown improved recognition performance. Most of the methods [15], [26]–[29] extract features using convolutional computation on the body graph encoding body motion of neighbor joints. ST-GCN [26] also extracts correlation of joint motion between two connected body joints for action recognition. Actional Structural GCN method [27] claims that in many body motions, two distant and not connected joints also frequently show strong correlation. For example, in walking motion, the correlation between right hand and left foot (and vice versa) plays an important role in classifying the motion. To include such distant joint correlation, they add Actional-links(A-links) modules calculating associations between distant joint pairs using encoder-decoder networks. Finally using the encoder part, they build A-links and add new edges to body motion graph to perform GCN operations. 2S-AGCN [15] defines two additional adjacent matrices. First one is data dependent adjacent matrix using correlation between two joints. Second one is data independent adjacent matrix allowing different topology on each layer. DGNN [29] represents the relationship between parent and child nodes by changing the basic graph structure to the directed acyclic graph (DAG). In addition, trainable adjacent matrices are added to find new edges that represent relationships between distant joints, as well as adjacent joints.

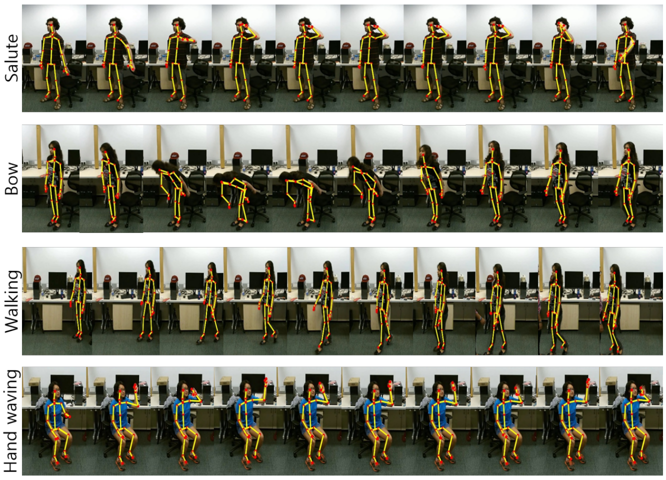


Fig. 2. Samples with some correlation between joints and time.

We focused temporal variations and correlations in human actions. First, some actions consist of unique symmetric or repeated motion patterns. For example, walking and hand waving in figure 2 contain repeated temporal motion patterns that are discriminative temporal features. 'Bow' shows bilateral symmetry motion pattern over temporal axis. Second, some actions can be characterized by only certain temporal motion segments or combination of such motion segments rather than entire temporal motions. For example, salute action can be easily recognized by few key frames as indicated in figure 2. Based on our observations, action can be described better if temporal correlations of optimal joint pairs are learned. However, state of the art methods ignore the relationship between joints from certain different time. In figure 1(a), state of the art prior method 2S-AGCN trains distant joint edges, but same edges are applied to all temporal frames. On the other hand, our proposed method makes distant joint-temporal edges as shown in the figure 1(b). For example, in 'drinking' action, right hand of first frame shows strong correlation with head of second frame.

In this paper, we introduce joint-temporal edges extracting the correlation between distant joints at distant time. We propose joint-temporal motion graph convolutional networks (JT-MGCN) for skeleton-based action recognition. When con-

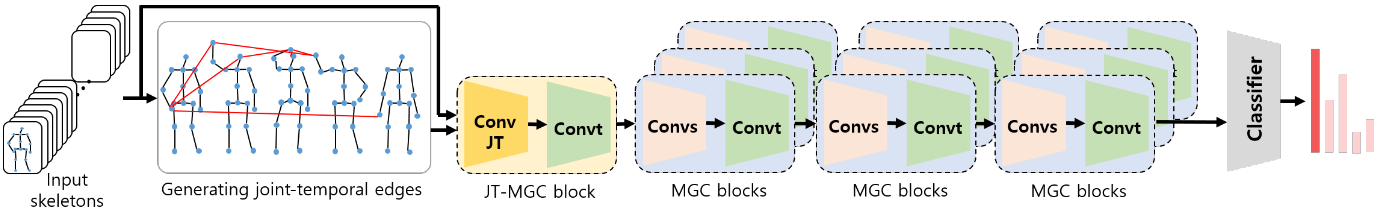


Fig. 3. The pipeline of the proposed JT-MGCN. It consist of a JT-MGC block and 9 MGC blocks. There are temporal pooling function per 3 MGC blocks.

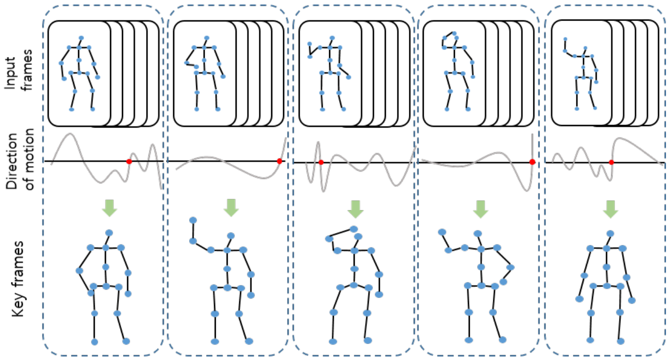


Fig. 4. Frame Grouping and Sampling

structing a graph, not just the relationship between joints on one body graph, but also the relationship between joints at distant time are considered. Fixed number of key frames of an action are extracted and joint-temporal motion graph is constructed. Five adjacent matrices define four types of edges on human body graphs. We perform experimental evaluation on NTU RGB-D dataset [4] and Kinetics dataset [31] to demonstrate the performance of proposed algorithm.

## II. Proposed Method

Figure 3 summarizes our proposed method. Proposed JT-MGCN first performs frame sampling to extract key frames and reduce the computational cost. Four types of joint-temporal edges are generated over key frames building joint-temporal motion graph on human body action. Features obtained through five adjacent matrices are added and, finally, two stream JT-MGCN is trained and action classification is performed.

### A. Frame Sampling

If motion samples consist of $F$ frame images and $N$ body joints are used for recognition, $(N \times F) \times (N \times F)$ adjacent matrix has to be constructed to manage all edges of any pair of joints from any time frame, which is $F^2$ times bigger compared to prior $N \times N$ adjacent matrix without our Joint-temporal edges. This will incur huge computation cost in training step. Because not all of image frames are equally important in describing certain human body action, we perform frame sampling to extract fixed number of key frames that play critical role in characterizing action types. We define that key frame contains a unique body pose that cannot be interpolated by other frames. We divide entire image frames into $G$ consecutive groups and detect one key frame per group (Figure 4). Three different sampling methods are presented and tested in our experimental evaluation.

**Sampling with uniform averaging**: We make a new graph by averaging the location values of each joint along the time axis within each frame group. $G$ averaged frames become representative key frame of $G$ groups. As a result, joint-temporal motion graph is built on $G$ key frames. However, frame averaging sometimes makes non-existing or strange body pose frame that is nothing to do with original action.

**Sampling with peak of joint location**: Alternative sampling method is selecting a frame with the largest position values of each joint within a group. By setting the key frame in this way, stretched poses in each body action are selected as key frames. However, such absolute locations of body joint are not always critical pose for action recognition.

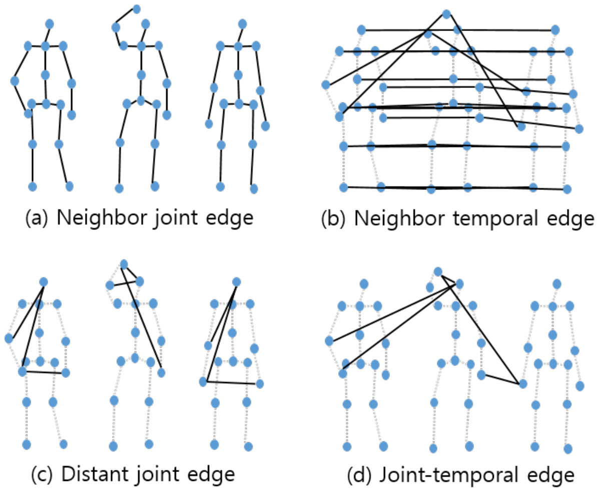
**Sampling with direction change of velocity(zc)**: In this case, we select a frame of maximum absolute slop in the velocity (first derivative of location) plot out of all zero crossing frames. The zero crossing points of the velocity plot are candidates for key frames.

The length of an action varies according to the type of action and even subject. For example, ’wearing a jacket’ action is usually longer than ’sitting down’ action and the length of ’wearing a jacket’ varies from subject to subject. In this work, temporal length variation of instances are adjusted in two ways. First, all action clips are repeated up to certain fixed number of frames, T (Repeating). In this case, discontinuous motion occurs at the boundary of two repeated actions, whereas original speed of action is preserved. Second, we resize the length of all samples as T (Resizing). In other words, we increase or decrease the length of actions to T. In this case, we have no repeated identical action in a sample without any discontinuity, however it lose its original speed of action that could be an important characterstics for some action. In our evaluation, we compare results of both cases.

### B. Joint-Temporal Edge

Proposed joint-temporal motion graph convolutional networks (JT-MGCN) employs four types of edges: Neighbor Joint Edge (NJ Edge), Neighbor Temporal Edge (NT Edge), Distant Joint Edge (DJ Edge), Joint-temporal Edge (JT Edge).

Figure 5 visualizes each edge type with corresponding edge samples. Neighbor joint edge is the conventional graph of



(a) Neighbor joint edge (b) Neighbor temporal edge

(c) Distant joint edge (d) Joint-temporal edge

Fig. 5. Four types of edges in our JT-MGCN

human body structure connecting each pair of neighbor joints extracting features of body pose at each time frame. Neighbor temporal edge detects temporal correlation of respective joint over time. Distant joint edge gives additional connections over NJ and NT edges connecting two joints located away to each other within single time frame. For this edge, the relationship between joints in all frames is represented at once. That is, the relationship between joints that is judged important in an action is connected. However, human action do not mean that certain joints have the same importance for all frames. In the case of 'taking off jacket', the relationship between the right hand and the torso is important at the beginning of the action, but at the end of the action, location of the right hand does not distinguish between the corresponding action class. In this paper, the following edge is used to solve this problem. Joint-temporal edge provides joint connection over time and space without any structural limitation. Through this edge it is possible to explicitly represent the relationship between different frames and joints.

### C. JT-MGCN

The overall structure of our JT-MGCN is shown in figure 3. JT-MGCN consists of one JT-MGC block and nine cascaded MGC blocks. JT-MGC block generates JT edges that are used in all following graphs extracting features. JT-MGC block processing occurs once in earliest block. After the first block has passed, temporal frame order will be inter-mixed in following the feature space. Therefore, we build JT edges only in the first block. After the JT-MGC block followed by three MGC blocks, a pooling along time axis occurs. The model finds the global features of action through pooling layer. Once all blocks have been passed, fully connected layers are attached for action classification.

*1) Motion Graph Convolution Block:* The motion graph convolutional block (MGCB) extracts features using the edges connected to the graph through the following figure 6. MGCB is similar to AGCB of [15] as shown in figure 6 (a). Simple Convs is a regular motion graph convolution layer for extracting features in joint domain and Convt is a regular graph convolution layer for extracting features in time domain. Between each layer, there is a batch normalization and a relu operation. The overall structure of the joint-temporal motion graph convolution block (JT-MGCB) is shown in the following figure 6 (b). It has basically the same structure as regular MGCB. Difference is ConvJT. comprehensive features in time and space is extracted through ConvJT and additional feature in time is obtained through temporal GCN, Convt. At this time, there is a batch normalization and relu after the convJT and convT, and a residual path is added to learn stably.

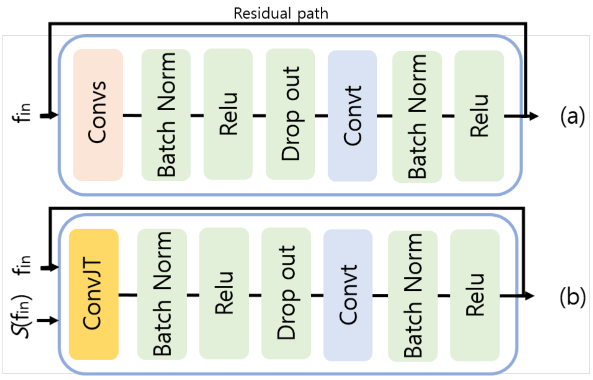


Fig. 6. Graph Convolutional Blocks (a) Regular MGC block, (b) Joint-temporal MGC block.

*2) Motion Graph Convolution Layer with Joint-temporal edges:* In our JT-MGCN, we define five adjacent matrices $(A_k, B_k, C_k, D_k, E_k)$ that encode four types of edges representing human actions. Equation (1) is the formula used in previous work 2S-AGCN [15] with $A_k, B_k, C_k$. They added DJ eedges on basic GCN formula for extracting feature between distant joints.

$$f_{out} = \sum_{k}^{K_v} W_k f_{in}(A_k + B_k + C_k) \tag{1}$$

where, $K_v$ denotes the kernel size of spatial dimension. $A_k$ is an adjacent matrix representing NJ edges, which describes human body structure. $B_k$ and $C_k$ describe DJ edges. It gives a new graph topology to the results after each convolution layer by this adjacent matrix. DJ edges described by $C_k$ represents data dependent correlation between joints. For calculating $C_k$, two joint responses are multiplied to measure the similarity in an embedding space using convolution function. Joint-temporal edges are defined by two new adjacent matrices: $D_k$ and $E_k$. $D_k$ and $E_k$ consist of $(N \times G) \times (G \times N)$, indicating which joint of frame is associated with which joint of other frame. $D_k$ describes JT edges based on data independent correlation between the joints, which is made up of weights that are learned for all samples to best classification. For example, if the relationship between the first frame and the last frame is crucial in action recognition, then corresponding JT edge will get higher weight. $E_k$ describes JT edges based

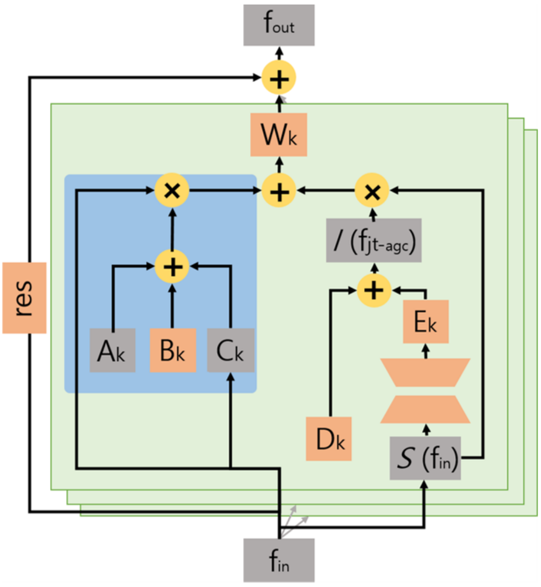


Fig. 7. Structure of convJT. Red boxes mean learnable parameters. Blue box shows ConvS. ⊕ denotes elementwise summation and ⊗ matrix multiplication.

on data dependent correlation between the joints, For example, in 'walking' action, an edge is added to the left hand of the first frame and the right foot of the last frame after training the network. This means that the correlation between the two joints or respective time is important clue in recognizing 'walking' action. On the other hand, in 'pick up' action, the head of the first frame and foot of the second frame is connected.

Now, we add new adjacent matrices $D_k$ and $E_k$ to equation (1).

$$f_{mgc} = f_{in} \cdot (A_k + B_k + C_k) \tag{2}$$

$$f_{JT-mgc} = \mathcal{I}(\mathcal{S}(f_{in}) \cdot (D_k + E_k)) \tag{3}$$

where S indicates frame sampling using three different methods mentioned in II-A. $f_{mgc}$ and $f_{JT-mgc}$ have different temporal resolution, because $f_{JT-mgc}$ are calculated after sampling. Therefore, we use simple interpolation function, I, to fill the blank frames.

$$f_{out} = \sum_{k}^{K_v} (f_{mgc} + f_{JT-mgc}) \cdot W_k \tag{4}$$

Finally, we obtain new $f_{out}$ from $f_{mgc}$ and $f_{JT-mgc}$ with convolution weight $W_k$. $B_k, D_k, E_k$ are all optimized through back propagation during training the model. Matrices $B_k, C_k, D_k, E_k$ are able to be used as attention map indicating strongly correlated joint pairs. If any value is assigned, there would be an edge connecting two joints. And higher value means higher importance of the two joints in recognizing an action.

### *D. Two-stream JT-MGCN*

In our implementation of the network, we organize the JT-MGCN as two-streams. Each stream is learned from a different order of dataset. First stream is trained using the location of the joints as follows figure 3. Second stream uses bone data connected between joints. Each joint is organized hierarchically so that when the upper joint moves, it moves along with the lower joint. For example, if the shoulder is moved, the elbow also change position. So, learning about bones helps in action recognition. In this paper, we obtain the difference between two neighbor joints and use it as a value for bone. when source joint is $v_1 = (x_1, y_1, z_1)$ and target joint is $v_2 = (x_2, y_2, z_2)$, the vector of bone $e_1$ is $(x_1 - x_2, y_1 - y_2, z_1 - z_2)$. Using these two kinds of data, we construct each stream and learn each one. Both models ensemble to make final classification result using each probability maps when learning is completed.

## III. Experimental Evaluation

We use two widely used datasets to evaluate our proposed algorithms and made comparisons.

**NTU+RGBD Dataset(NTU)** This dataset is the largest action recognition dataset obtained with kinect v2. There are about 50,000 sequences from 40 people and 60 classes. Two evaluations are performed (cross view and the cross subject) proposed by [4]. Cross view uses one of the three camera views distributed on the dataset as test, and 2 and 3 camera as training. Cross subject uses data from 20 people as training and the others as test.

**Kinectics-skeleton** Kinectics is an action dataset published by Deep Mind, consisting of about 300,000 rgb video samples, 400 classes and at least 400 samples per class. In this paper, the experiment was carried out using kinetics-skeleton dataset, which obtained the coordinates of skeleton using open source, OpenPose [32], in rgb video. The coordinates of the obtained skeletons consist of X and Y coordinates in the rgb images and the joint's confidence score in the (X,Y,C). 240,000 set clips are used for training and 20,000 samples are used for test. For frame sampling, five frames are extracted from total 300 frames. We conduct experiments using two Nvidia Titan RTX GPUs. The batch size is 80. At JT-MGCN, SGD is used as an optimization function for learning. To avoid overfitting weight decay is set to 0.001. Our model uses cross entropy as loss function.

### *A. Ablation Study*

*1) Frame Sampling Results:* As described in II-A, either repeating or resizing can be performed for data pre-processing. Frame sampling results are shown in figure 8. In the cases of 'peak' or 'zero crossing (zc)' with repeating, 'throw' motion shows multiple iterations. Especially for 'uniform' case of 'pick up' motion, the first frame is coincidentally identical, so all the same frames are selected. This would reveal almost nothing of the action. On the other hand, selected frames of resizing represent a series of movements well. The results of the frame sampling method also show different aspects. In the case of 'uniform', very similar frames are selected as key frames because five frames are picked evenly regardless of the amount of variation in action. Comparison of 'peak' with

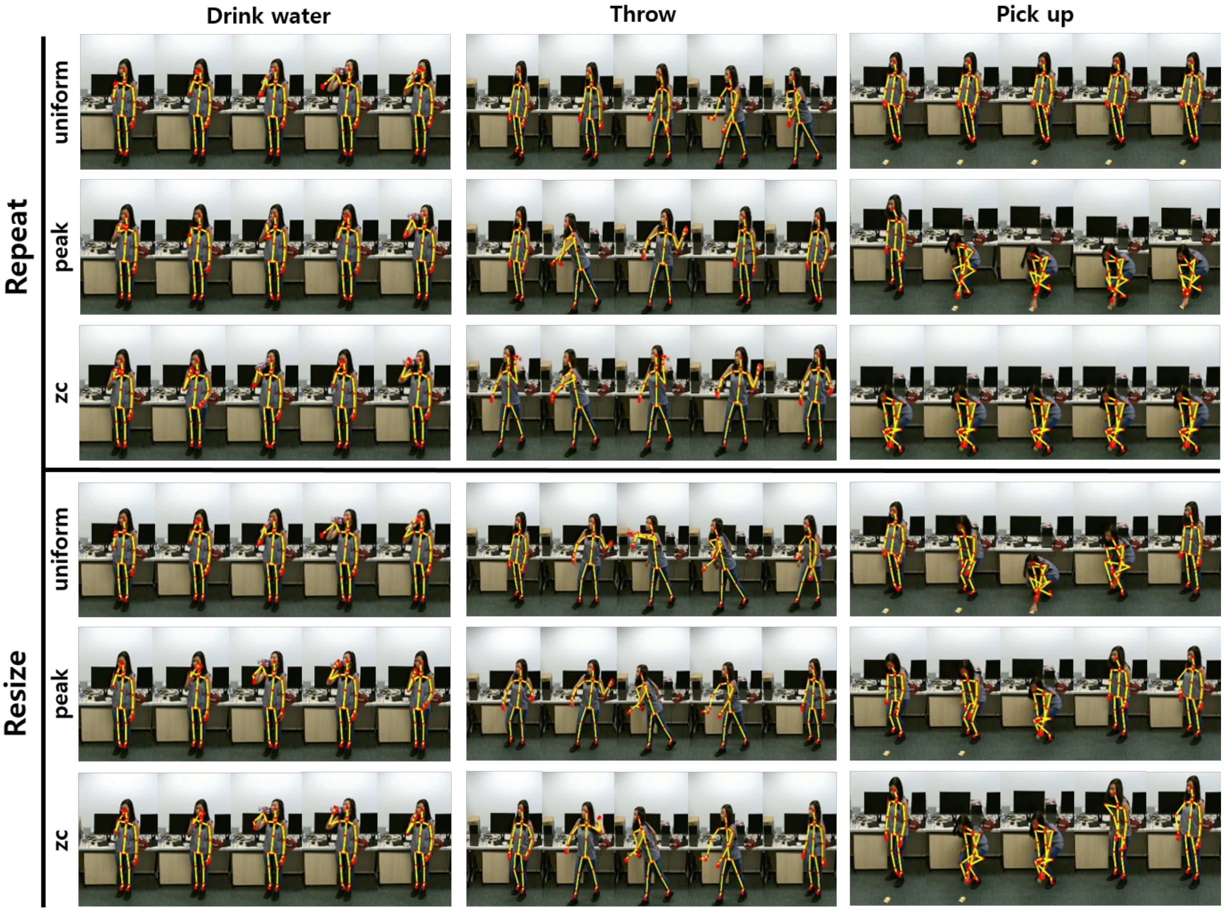


Fig. 8. Results of frame sampling. 'Uniform','Peak' and 'ZC' means 'uniformly averaging','peak of location' and 'zero crossing'. In 'Throw', repeating makes mixed order. On the other hand, for resizing, one motion continues smoothly. Looking at 'Pick up' in repeating, the same frames are selected repeatedly for 'Uniform' and 'ZC'

'zero crossing' of resizing shows that the 'peak' does not find middle part of an action because 'peak' simply finds a large position value in an action.

*2) Classification Results:* First, we compare experimental results using two pre-processing methods, repeating and resizing. Kernel size is the size of temporal median filter for body motion smoothing. Repeating with kernel size of '5' shows best performance (table II). Bigger kernel size may remove important clue of motions. The classification results using two types of JT edges, $D_k$ and $E_k$ are shown in table III. Compared to the accuracy of existing algorithms that do not include JT edges proposed in this paper, using $D_k$ increased by about 2.5% and $E_k$ by about 2%. This shows that there are specific data-dependent joint-temporal relationship, as well as common relationship to classify in all action. Both edges performed best, indicating that the two edges did not conflict with each other.

The table IV is a comparison of the three methods of frame sampling. The 'peak of location' method shows the best performance because selected frames contain more various time changes than other methods.

For demonstrating how JT-edges are effected to classify action, we shuffled each samples along the temporal axis and input to each pre-trained JT-MGCN models. As follows table V, results without shuffling show that using $D_k$ and $E_k$ improves classification performance, on the contrary results with shuffling don't. If the orders of human action are mixed, it means different action. Thus, when actions are classified,

TABLE II
RESULTS OF CLASSIFICATION ACCURACY(%) ON EACH PREPROCESSING USING CROSS-SUBJECT OF NTU RGBD DATASET.

| Method | Kernel size | |
|---|---|---|
| | **5** | 15 |
| **Repeating** | **90.40** | 90.19 |
| Resizing | 89.63 | 89.20 |

TABLE III
RESULTS ON EACH JOINT-TEMPORAL EDGES USING NTU RGBD DATASET.

| Adjacent matrices | Accuracy(%) |
|---|---|
| $(A_k + B_k + C_k)$ | 88.5 |
| $(A_k + B_k + C_k) + D_k$ | 90.28 |
| $(A_k + B_k + C_k) + E_k$ | 90.23 |
| $\boldsymbol{(A_k + B_k + C_k) + D_k + E_k}$ | **90.40** |

TABLE IV
RESULTS ON EACH FRAME SAMPLING METHODS USING NTU RGBD DATASET.

| Frame Sampling Method | Accuracy(%) |
|---|---|
| Uniform averaging | 89.48 |
| **Peak of joint location** | **90.40** |
| Zero crossing | 89.66 |

TABLE V
RESULTS WITH SHUFFLED FRAMES USING NTU RGBD DATASET.

| | Adjacent matrices | Accuracy(%) |
|---|---|---|
| without shuffling | $A_k + B_k + C_k$ | 88.5 |
| | $A_k + B_k + C_k + D_k + E_k$ | 90.4 |
| with shuffling | $A_k + B_k + C_k$ | **13.4** |
| | $A_k + B_k + C_k + D_k + E_k$ | **8.8** |

extracting temporal features is important. JT-edges make easy to extract temporal relationships than the previous one. Comparison between our proposed networks and state of the art on the NTU rgbd dataset is as follows table VI. It has improved accuracy about 2% compared to baseline, 2S-AGCN with NTU dataset. And also JT-MGCN is competitive with state of the art. The JT edge, we propose, is available in all other GCN algorithms. Thus, if the performance of the GCN algorithm is improved, it will show better accuracy.

TABLE VI
ACTION RECOGNITION RESULT ON NTU DATASET. IT SHOWS EACH ACCURACY ALONG THE MEASUREMENT, CROSS-SUBJECT AND CROSS-VIEW.

| Method | Year | CS(%) | CV(%) |
|---|---|---|---|
| PA-LSTM [4] | 2016 | 62.9 | 70.3 |
| ST-LSTM+TS [5] | 2016 | 69.2 | 77.7 |
| STA-LSTM [8] | 2017 | 73.4 | 81.2 |
| VA-LSTM [7] | 2017 | 79.4 | 87.6 |
| MTLN [18] | 2017 | 79.6 | 84.8 |
| ST-NBMIM [9] | 2018 | 80.0 | 84.2 |
| MTCNN [24] | 2018 | 81.1 | 87.4 |
| ST-GCN [26] | 2018 | 81.5 | 88.3 |
| DPRL+GCNN [25] | 2018 | 83.5 | 89.8 |
| SR-TSL [10] | 2018 | 84.8 | 92.4 |
| AS-GCN [27] | 2019 | 86.8 | 94.2 |
| 2S-AGCN [15] | 2019 | 88.5 | 95.1 |
| DGNN [29] | 2019 | 89.9 | 96.1 |
| **JT-MGCN(proposed)** | **2020** | **90.40** | **95.78** |

TABLE VII
ACTION RECOGNITION RESULT ON KINETICS-SKELETON DATASET. IT SHOWS TOP-1 AND TOP-5 ACCURACIES.

| Method | Year | Top1(%) | Top5(%) |
|---|---|---|---|
| ST-GCN [26] | 2018 | 30.7 | 52.8 |
| AS-GCN [27] | 2019 | 34.8 | 56.5 |
| 2S-AGCN [15] | 2019 | 36.1 | 58.7 |
| DGNN [29] | 2019 | 36.9 | 59.6 |
| **JT-MGCN(proposed)** | **2020** | **37.0** | **59.8** |

Also, JT-MGCN shows better accuracy than others on Kinetics-skeleton Dataset as follows table VII. Compared to baseline algorithms, our methods improves performance by 1%.

*3) Visualization of the Joint-temporal Motion Graph:* Figure 9 shows adjacent matrices trained in our JT-MGCN. Each red block represents one time frame. For example, block (2,1) represents the relationship between the joint of the second frame and the first frame. Figure 9 (a) is the adjacent matrix of NJ edges and DJ edges. Therefore, there is no edge between different frames (Outside of diagonal blocks). On the other hand, for $D_k$ and $E_k$ as proposed in this paper, there are joint relationships between frames learned as JT edges.

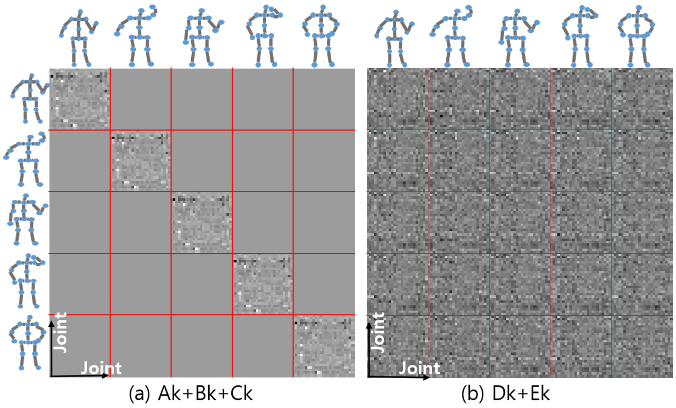


Fig. 9. (a) shows $A_k + B_k + C_k$ that represent NJ and DJ edges. (b) shows $D_k + E_k$ that represent JT edges. A block is an adjacent matrix within a frame.

Figure 10 shows JT edges of Top 10 highest weight value of $(D_k + E_k)$ on the body skeleton. Figure 10 (a) 'drinking water', the movement of the upper body is mainly used. (b) 'kicking' on the other hand is mainly used a lower body movement. Nevertheless, the edges corresponding to the two samples are the same. This shows that the overall characteristics of the samples, $D_k$, generally become more pronounced than the data dependent matrix, $E_k$. Especially when the legs move in a person's movements, the arms of the person move together. Therefore, the relationship between upper body joints is important in a person's action classification. This result is the same for the classification result above table III. And the relationship between the first frame and the other frames has more influence in classifying action, as shown in the figure 10. This indicates that a starting point is important in a person's action.

## IV. CONCLUSIONS

There exist a complex relationship between time and space in a person's action. To express this relationship, we proposed a Joint-temporal edge. Using this edge, we made to train the JT-MGCN model and presented improved results. There were certain important edges in human movement and learning them using the corresponding edges confirmed that their performance was improving.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1A2C1015146).

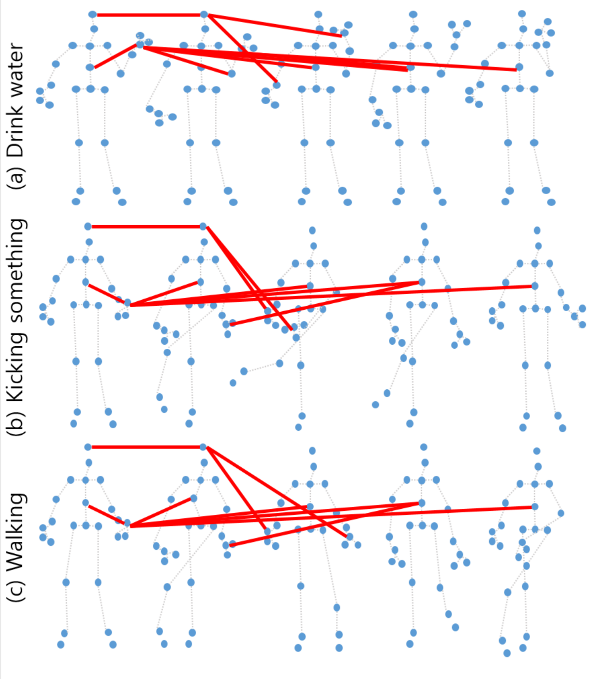


Fig. 10. Visualization JT-egdes (red lines) with the highest weight value of NTU rgbd dataset. Skeleton in each frame from left to right in order.

## References

[1] Z. Luo, B. Peng, D.-A. Huang, A. Alahi, and L. Fei-Fei, "Unsupervised learning of long-term motion dynamics for videos," *arXiv preprint arXiv:1701.01821*, vol. 2, 2017.

[2] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "Rgb-d-based human motion recognition with deep learning: A survey," *Computer Vision and Image Understanding*, vol. 171, pp. 118–139, 2018.

[3] 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*, 2015, pp. 1110–1118.

[4] 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*, 2016, pp. 1010–1019.

[5] 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*. Springer, 2016, pp. 816–833.

[6] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 7, 2017, p. 43.

[7] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng *et al.*, "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," *arXiv, no. Mar*, 2017.

[8] 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 *AAAI*, vol. 1, no. 2, 2017, pp. 4263–4270.

[9] J. Weng, C. Weng, J. Yuan, and Z. Liu, "Discriminative spatio-temporal pattern discovery for 3d action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[10] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, "Skeleton-based action recognition with spatial reasoning and temporal stack learning," in *The European Conference on Computer Vision (ECCV)*, September 2018.

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

[12] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (indrnn): Building a longer and deeper rnn," 2018.

[13] 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.

[14] S. Nam and S. Lee, "Motion regeneration using motion texture and autoencoder," in *SIGGRAPH Asia 2018 Posters*. ACM, 2018, p. 24.

[15] 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*, 2019, pp. 12 026–12 035.

[16] R. Zhao, K. Wang, H. Su, and Q. Ji, "Bayesian graph convolution lstm for skeleton based action recognition."

[17] P. Wang, Z. Li, Y. Hou, and W. Li, "Action recognition based on joint trajectory maps using convolutional neural networks," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 102–106.

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

[19] H. Liu, J. Tu, and M. Liu, "Two-stream 3d convolutional neural network for skeleton-based action recognition," *arXiv preprint arXiv:1705.08106*, 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)*. IEEE, 2017, pp. 1623–1631.

[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)*. IEEE, 2017, pp. 601–604.

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

[23] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2017, pp. 597–600.

[24] 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.

[25] 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*, 2018, pp. 5323–5332.

[26] 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.

[27] 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*, 2019, pp. 3595–3603.

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

[29] 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*, 2019, pp. 7912–7921.

[30] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks," in *AAAI*, 2016, pp. 3697–3703.

[31] 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.

[32] 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*, 2017, pp. 7291–7299.