Novel Motion Patterns Matter for Practical Skeleton-Based Action Recognition

Mengyuan Liu[1]*, Fanyang Meng[2], Chen Chen[3], Songtao Wu[4]
[1] Key Laboratory of Machine Perception, Peking University, Shenzhen Graduate School
[2] Peng Cheng Laboratory
[3] University of Central Florida
[4] Sony R&D Center China
nkliuyifang@gmail.com, mengfy@pcl.ac.cn, chen.chen@crcv.ucf.edu, Songtao.Wu@sony.com

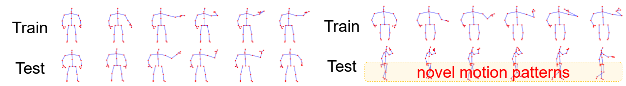
Abstract

Most skeleton-based action recognition methods assume that the same type of action samples in the training set and the test set share similar motion patterns. However, action samples in real scenarios usually contain novel motion patterns which are not involved in the training set. As it is laborious to collect sufficient training samples to enumerate various types of novel motion patterns, this paper presents a practical skeleton-based action recognition task where the training set contains common motion patterns of action samples and the test set contains action samples that suffer from novel motion patterns. For this task, we present a Mask Graph Convolutional Network (Mask-GCN) to focus on learning action-specific skeleton joints that mainly convey action information meanwhile masking action-agnostic skeleton joints that convey rare action information and suffer more from novel motion patterns. Specifically, we design a policy network to learn layer-wise body masks to construct masked adjacency matrices, which guide a GCN-based backbone to learn stable yet informative action features from dynamic graph structure. Extensive experiments on our newly collected dataset verify that Mask-GCN outperforms most GCN-based methods when testing with various novel motion patterns.

Introduction

Action recognition is of great importance to human-robot interaction, which enables a robot to understand the meaning of human movements. Existing work can be roughly categorized into RGB-based methods (Tu et al. 2019; Liu and Yuan 2018), depth-based methods (Chen et al. 2016), and skeleton-based methods (Liu, Liu, and Chen 2017; Liu, Meng, and Liang 2022). Compared with RGB data, skeleton representation of actions suffers less from clutter backgrounds and additionally encodes depth information (Zhang et al. 2022; Tu et al. 2022). Compared with depth data, skeleton data directly capture human body structure meanwhile having less redundant information. Moreover, skeleton data can be accessed in real-time with the spread of depth sensors. Therefore, skeleton-based action recognition has attracted increasing attention (Yan, Xiong, and Lin 2018; Chen et al. 2021).

*Corresponding Author.



(a) Samples used in previous methods (b) Samples used in our method

Figure 1: Comparison between samples used in previous methods and ours. We take an action “wave” as an example. (a) Previous methods assume that samples in both training and test sets share similar motion patterns. (b) In our method, samples in the test set contain novel motion patterns, e.g. leg movements (named as “walking”), which are unobserved in the training set.

Method	CS_1 (%)	CS_2 (%)	CV_1 (%)	CV_2 (%)
CTR-GCN	46.90	62.48	44.46	66.23
Ours	**51.05**	**66.90**	**57.79**	**73.00**

Table 1: Comparison of performances between state-of-the-art CTR-GCN (Chen et al. 2021) method and ours on our dataset, where test samples contain novel motion patterns. (CS: cross subject; CV: cross view)

Skeleton-based action recognition methods assume that the same type of action samples in the training set and the test set share similar motion patterns. This ideal assumption is widely adopted by existing skeleton-based action recognition datasets (Wang et al. 2014; Chen, Jafari, and Kehtarnavaz 2015; Shahroudy et al. 2016; Wang et al. 2020), including the most popular dataset, i.e., NTU RGB+D dataset. We take an action “wave” as an example. The training and test samples used in previous methods are shown in Fig. 1 (a). They contain similar motion patterns, i.e., raising one hand and moving the hand from one side to another. Meanwhile, remaining body parts, e.g., legs, keep nearly still.

Practical Problem: Compared with samples in Fig. 1 (a), action samples in real scenarios are more complex. We observe that action samples usually contain novel motion patterns which are not involved in the training set. We take an action “wave” from our newly collected dataset as an example. The training and test samples used in our method are shown in Fig. 1 (b), where the test sample contains novel motion patterns, e.g. leg movements, which are unobserved in our training set. One natural question arises: How do the novel motion patterns in the test stage affect the existing

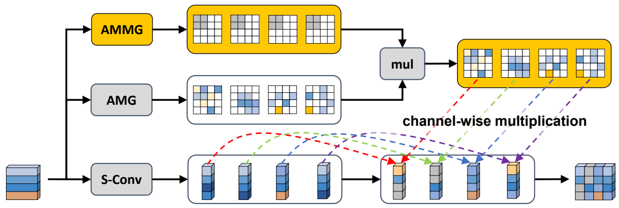


Figure 2: Pipeline of mask graph convolution, which contains three blocks, namely “AMG”, “S-Conv” and “AMMG”. Specifically, AMG is short for Adjacency Matrix Generation. S-Conv is short for Spatial Convolution. AMMG is short for Adjacency Matrix Mask Generation. We follow graph convolution to use AMG block for generating adjacency matrix and to use S-Conv block for spatial feature learning. Different from graph convolution, we present AMMG to generate an adjacency matrix mask, which guides the adjacency matrix to learn channel-wise local features.

skeleton-based action recognition methods?

We answer this question by evaluating the state-of-the-art CTR-GCN (Chen et al. 2021) method on our dataset. Despite that CTR-GCN dominates current skeleton-based action recognition methods, CTR-GCN only achieves inferior results in Table 1. To explain these results, we carefully analyze the network architecture of CTR-GCN. In general, this kind of GCN-based method typically stacks multi-layer graph convolution for feature extraction. Each graph convolution uses an Adjacency Matrix Generation (AMG) block for generating adjacency matrix and uses a Spatial Convolution (S-Conv) block for spatial feature learning and finally uses channel-wise multiplication operation to fuse adjacency matrix and spatial feature. Since the adjacency matrix is built upon all skeleton joints, it suffers from novel motion patterns in test samples. As the adjacency matrix dominates the function of graph convolution, the noisy adjacency matrix naturally induces the inferior performance of GCN.

Our Solution: To solve the limitation of GCN-based methods on this challenging problem setting, we first divide skeleton joints into two categories, namely *action-specific joints* which mainly convey action information, and *action-agnostic joints* which present rare action information. As shown in Fig. 1 (b), we observe that people mainly use action-specific skeleton joints to perform actions meanwhile use action-agnostic skeleton joints to generate novel motion patterns. Inspired by this observation, we present a mask graph convolution that focuses on learning action-specific skeleton joints meanwhile masking action-agnostic joints. In this way, we can learn motion patterns from actions and suffer less from novel motion patterns. Specifically, to solve the defect of the traditional adjacency matrix, we take advantage of the general GCN method and present a Mask GCN framework by stacking multi-layer mask graph convolution. Fig. 2 shows the pipeline of our proposed mask graph convolution. It also highlights that mask graph convolution differs from common graph convolution by involving a new Adjacency Matrix Mask Generation (AMMG) block, which generates a channel-wise mask to guide an adjacency matrix to learn stable yet informative spatial features. Our main contributions are summarized as three-fold.

- To facilitate action recognition in real applications, we present a practical skeleton-based action recognition task, which introduces novel motion patterns into test samples. To bridge the gap between training and test samples, we present a Mask-GCN framework that uses multiple mask graph convolution layers for deep feature learning from action-specific skeleton joints.
- To implement our Mask-GCN, we develop an Adjacency Matrix Mask Generation (AMMG) block for learning action-dependant adjacency matrix mask. Specifically, a Policy Network (PN) is designed to aggregate global and local features from an input feature, and Gumbel Softmax is used to generate a body mask, based on which we formulate the adjacency matrix mask.
- We collect the first large-scale dataset for evaluating different methods of handling novel motion patterns. It serves as a new benchmark to facilitate the research in this direction. Extensive experiments on our dataset verify the effectiveness of our Mask-GCN by outperforming most GCN-based action recognition methods.

Related Work

An end-to-end hierarchical RNN (Du, Wang, and Wang 2015) was proposed to model the long-term contextual information of skeleton sequences. Based on RNN, a Spatio-temporal LSTM (Liu et al. 2016) was presented to encode hidden information of skeleton over both spatial and temporal domains concurrently. On top of RNN, a spatial and temporal attention model (Song et al. 2017) was also proposed to selectively focus on discriminative joints. To enhance the sequential modeling ability of RNN, a spatial-temporal transformer network (Plizzari, Cannici, and Matteucci 2021) was used to understand intra-frame interactions between different body parts and to model inter-frame correlations. Another spatial-temporal specialized transformer (Zhang et al. 2021) was also used to model skeleton sequences in spatial and temporal dimensions respectively. Instead of using a sequential modeling network, a CNN model (Du, Fu, and Wang 2015) was used to model the hidden spatial-temporal information of skeleton sequences from an image, which is the concatenation of the joint coordinates. Moreover, multiple images (Ke et al. 2017; Liu, Liu, and Chen 2017) were also used as inputs of CNN models to extract spatial-temporal skeleton features.

Noticing that a skeleton sequence is a graph, a Spatial-Temporal Graph Convolutional Network (ST-GCN) (Yan, Xiong, and Lin 2018) was proposed to model dynamic skeletons, which moves beyond the limitations of previous methods by automatically learning spatial and temporal patterns. To improve the inference speed of ST-GCN, a simple yet effective semantics-guided neural network (SGN) (Zhang et al. 2020) was proposed to describe a long-term skeleton sequence as multiple short-term sequences. Based on SGN, an adaptive SGN (Shi et al. 2021) was developed to further reduce the computational cost of the inference process by adaptively controlling the number of skeleton joints on-the-fly. Instead of modifying inputs, the regular GCN structure was improved to formulate an efficient shift graph convolutional network (Shift-GCN) (Cheng et al. 2020b), which is composed of novel shift graph operations and lightweight point-wise convolutions. To improve the performance of ST-GCN, a directed graph neural networks (Shi et al. 2019a) was proposed to represent the skeleton data as a directed acyclic graph based on the kinematic dependency between the joints and bones in the natural human body. To capture part-level information, a part-level graph convolutional network (Huang et al. 2020a) was developed, which uses a graph pooling operation to automatically aggregate body joints into body parts. Noting that previous GCN methods separately encode spatial and temporal information, unified graph convolutions (Liu et al. 2020) were developed to extract spatial-temporal features at the same time. Besides, Bayesian inference (Zhao et al. 2019), attention mechanism (Si et al. 2019), residual connection (Song et al. 2020), and context encoding (Zhang, Xu, and Tao 2020) were jointly used with GCN to explore more discriminate features.

Recent work focuses on designing an adjacency matrix so that the graph convolution network can effectively learn topology information. Instead of using a single adjacency matrix, multiple high-order adjacency matrices (Huang et al. 2020b) can be used in an inception module. Rather than design hand-crafted adjacency matrix, actional-structural graph convolutional network (Li et al. 2019), two-stream adaptive graph convolutional network (Shi et al. 2019b) and dynamic graph convolutional network (Ye et al. 2020) used data-driven adjacency matrix that can be optimized through backward propagation. Inspired by CNN which uses an independent spatial aggregation kernel for every channel to capture different spatial information, the decoupling graph convolutional network (Cheng et al. 2020a) adopted an independent adjacency matrix for every channel to boost the graph modeling ability with no extra computation. To reduce the difficulty of modeling channel-wise topologies, channel-wise topology refinement graph convolution (CTR-GCN) (Chen et al. 2021) models channel-wise topologies by learning a shared topology as a generic prior for all channels and then refining it with channel-specific correlations for each channel. Our method differs from CTR-GCN in several aspects. First, we design a mask-guided adjacency matrix for topology learning from action-specific skeleton joints. Second, we develop a policy network using local and global modeling for implementing our adjacency matrix mask generation block. Moreover, experimental results verify that our method outperforms CTR-GCN by a large margin in handling novel motion patterns in the test stage.

Mask Graph Convolutional Network

Mask Graph Convolutional Network (Mask-GCN) consists of multiple Mask Graph Convolution (Mask-GC) layers. In the following, we first introduce the general idea of Mask-GC, then detail the Adjacency Matrix Mask Generation (AMMG) block to implement the Mask-GC, and finally formulate the Mask-GCN framework for practical skeleton-based action recognition.

Mask-GC. A human skeleton is naturally a graph, where vertices are joints and edges are bones. To describe the graph, Graph Convolutional Network (GCN) has achieved high success. GCN involves multiple graph convolutions, where each graph convolution contains two main steps, namely, spatial feature learning and global topology learning. Global topology learning is used to further enhance the extracted spatial features. Suppose a GCN contains M graph convolutions. Taking the m-th graph convolution as an example, its operation is formulated as:

$$\mathbf{X}_{m+1} = \mathcal{E}\big(\mathcal{S}(\mathbf{X}_m),\ \mathbf{A}_m\big), \tag{1}$$

where $m \in (0, ..., M-1)$, $\mathbf{X}_m$ is the input feature, and $\mathbf{X}_{m+1}$ is the output feature that is used for next graph convolution. When m equals 0, $\mathbf{X}_m$ is the input skeleton sequence. Otherwise, $\mathbf{X}_m$ is the output of the previous graph convolution. $\mathcal{S}$ is the spatial convolution operation that is used for spatial feature learning. $\mathcal{E}$ is the aggregation operation which is combined with an adjacency matrix $\mathbf{A}_m$ for global topology learning. Each element of the adjacency matrix reflects the correlation strength between pairwise joints. The original adjacency matrix is defined according to the physical connections between joints. Recent work verifies that the adjacency matrix can be directly learned from the input feature. The learned adjacency matrix can be shared by different channels of $\mathcal{S}(\mathbf{X}_m)$. Moreover, the channel-specific adjacency matrix can be directly learned from the input feature, which achieves state-of-the-art results on the skeleton-based action recognition task.

Our Mask-GC is built upon the channel-specific adjacency matrix. The data flow of the mask graph convolution is shown in Fig. 2, which contains a S-Conv (Spatial Convolution) block, an AMG (Adjacency Matrix Generation) block and an **AMMG** (Adjacency Matrix Mask Generation) block. The combination of the S-Conv block and the AMG block implements the standard graph convolution operation. Our motivation is to use the **AMMG** block to constrain the reception field of adjacency matrices extracted by the AMG block. Multiple mask graph convolutions are used to implement a Mask Graph Convolutional Network (Mask-GCN). Suppose a Mask-GCN contains M mask graph convolutions. Taking the m-th mask graph convolution as an example, its operation is defined as:

$$\mathbf{X}_{m+1} = \mathcal{E}\big(\mathcal{S}(\mathbf{X}_m),\ \mathbf{A}_m \cdot \mathcal{M}(\mathbf{X}_m)\big), \tag{2}$$

where $\mathcal{M}$ is the function of **AMMG** block, $\mathcal{M}(\mathbf{X}_m)$ is the adjacency matrix mask, and other operations and variables are defined following Eq. (1). We follow CTR-GCN (Chen et al. 2021) to implement the S-Conv block and AMG block. Specifically, S-Conv takes $\mathbf{X}_m$ as input and outputs $\mathcal{S}(\mathbf{X}_m)$. Let $\mathbf{X}_m = [J \times T_m \times C_m]$, where "[ ]" denotes a vector, J is the joint number of a skeleton, T_m is the temporal dimension and C_m is the spatial dimension. Specially, the input feature for the 0-th graph convolution is $\mathbf{X}_0 = [J \times T_0 \times C_0]$, where T_0 is the length of the skeleton sequence, C_0 equals 3, which denotes three coordinates of skeleton joints. The output feature vector $\mathcal{S}(\mathbf{X}_m)$ can be denoted as $[J \times T_m \times C_n]$. Generally, S-Conv extracts deeper spatial features for each

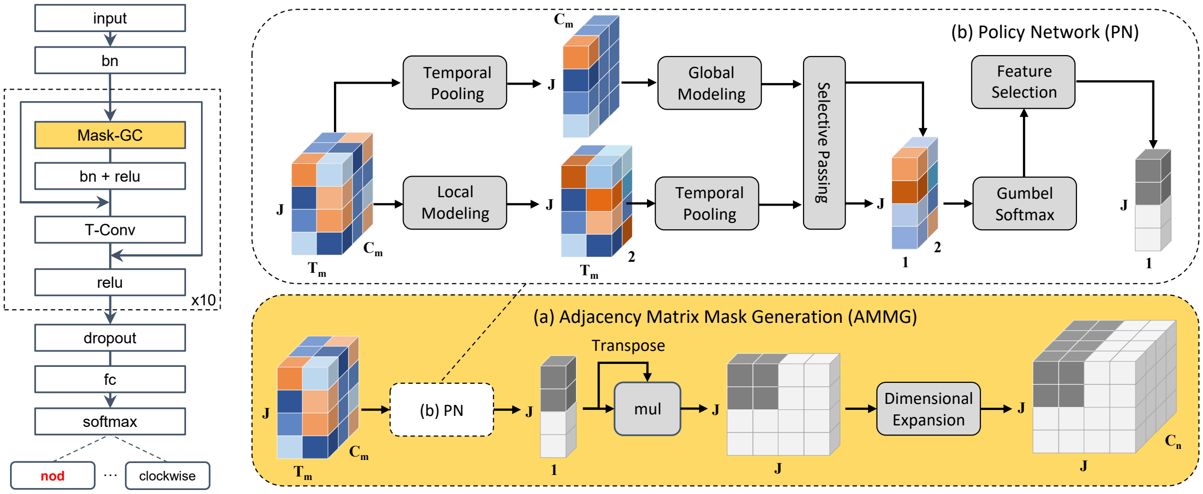


Figure 3: Illustration of our proposed Mask-GCN and our proposed Adjacency Matrix Mask Generation (AMMG) block, which consists of a Policy Network (PN) block and Dimensional Expansion (DE) operation.

spatial-temporal joint and extends the feature channel number from C_m to C_n. We can use 1×1 convolution to implement S-Conv. AMG takes $\mathbf{X}_m$ as input and outputs the learnable adjacency matrix $\mathbf{A}_m$. The key idea of constructing $\mathbf{A}_m$ is to measure the correlation of pairwise joints, e.g., the p-th joint and the q-th joint, where p and q belong to $(0, ..., J-1)$. For the p-th joint, the corresponding feature is $\mathbf{X}_{m,p} = [1 \times T_m \times C_m]$. Noting that the temporal dimension contains redundant information, we further expand the spatial dimension and then compress the temporal dimension to obtain a more representative joint feature, which is denoted as $\mathbf{X}^{'}_{m,p} = [1 \times C_h]$, where C_h is the new channel number. Similarly, the q-th joint can be denoted as $\mathbf{X}^{'}_{m,q} = [1 \times C_h]$. The correlation score is calculated as $tanh(\mathbf{X}^{'}_{m,p} - \mathbf{X}^{'}_{m,q})$, where $tanh$ activation function is used to mapping the correlation score to the scope of $[-1, 1]$. By concatenating correlation scores of all pairs, we can obtain a correlation matrix, which can be denoted as $[J \times J \times C_h]$. We further use 1×1 convolution to map the correlation matrix to the adjacency matrix $[J \times J \times C_n]$.

AMMG. As a core component of Mask-GC, our proposed **AMMG** (Adjacency Matrix Mask Generation) block mainly consists of a PN (Policy Network) block and a DE (Dimensional Expansion) operation. The general pipeline of **AMMG** is illustrated in Fig. 3, which can be formulated as:

$$\mathcal{M}(\mathbf{X}_m) = \mathcal{D}\big(\mathcal{B}(\mathbf{X}_m)\big), \tag{3}$$

where $\mathcal{B}$ denotes function of PN block and $\mathcal{D}$ denotes the function of the DE operation.

PN is a policy network that takes a deep feature $\mathbf{X}_m = [J \times T_m \times C_m]$ as input and outputs a body mask $\mathbf{B}_m = [J \times 1]$. Specifically, local modeling is firstly applied on $\mathbf{X}_m$ to aggregate information across different channels, and generate a new feature $[J \times T_m \times 2]$. Simple 1x1 convolution can be used to implement the local modeling function. Second, we use temporal pooling to aggregate information across the temporal axis, and obtain a new feature $\mathbf{P}_m = [J \times 2]$. For the j-th row, the feature $\mathbf{P}_{m,j} = [1 \times 2]$ indicates the probability of selecting each skeleton joint. We can obtain the discrete actions to select skeleton joints through argmax. Considering that argmax is not differentiable, we introduce the Gumbel Softmax trick (Jang, Gu, and Poole 2016) to solve this problem. In the forward stage, we calculate action as:

$$a_{m,j} = \arg\max_i(\mathbf{P}^i_{m,j} + G^i_{m,j}), \tag{4}$$

where $a_{m,j}$ denotes the action for the j-th joint in the m-th layer. When $a_{m,j}$ equals one, $\mathbf{B}_{m,j}$ is set to one. Otherwise, $\mathbf{B}_{m,j}$ is set to zero. $G^i_{m,j}$ is defined as:

$$G^i_{m,j} = -log(-logU^i_{m,j}), \tag{5}$$

where $U^i_{m,j}$ is sampled from a uniform i.i.d distribution – $Uniform(0, 1)$. In the back-propagation stage, we use the continuous Gumbel Softmax to relax Eq. (4) as:

$$\tilde{a}_{m,j} = \frac{exp(\mathbf{P}^i_{m,j} + G^i_{m,j})/\tau}{\sum_k exp(\mathbf{P}^k_{m,j} + G^k_{m,j})/\tau}, \tag{6}$$

where τ is the temperature parameter. When τ is set to a small value, samples from the Gumbel Softmax are close to one hot vector. Otherwise, the variance of samples' gradients from Gumbel Softmax becomes small. We select the proper value for τ by ablation studies on our dataset.

Besides using local modeling for generating features to indicate the probability of selecting each skeleton joint, we also provide an alternative way that takes advantage of global modeling. Specifically, the input feature $\mathbf{X}_m = [J \times T_m \times C_m]$ is firstly compressed by temporal pooling to generate a compact feature $[J \times C_m]$. Then, global modeling is applied to this feature to further aggregate global information across different channels and generate a new feature $\mathbf{P}_{m,j} = [1 \times 2]$, which is an alternative to the branch that

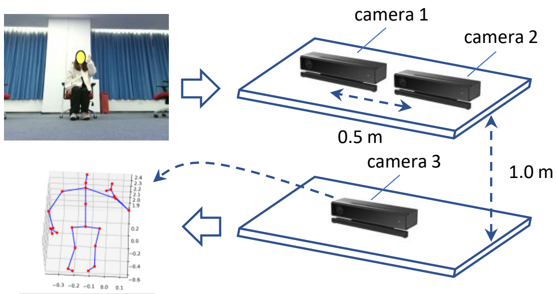


Figure 4: Pipeline of collecting our dataset with three cameras. We show one snap captured by camera 3.

uses local modeling. Noting that we use a selective passing mechanism to determine which feature is selected. The following Gumbel Softmax operation remains unchanged.

DE is a dimensional expansion operation that is used to generate adjacency matrix mask $[J \times J \times C_n]$ from $\mathbf{B}_m$, which can be formulated as:

$$\mathcal{D}(\mathbf{B}_m) = [\overbrace{\mathbf{B}_m \times \mathbf{B}_m^{-1}||...||\mathbf{B}_m \times \mathbf{B}_m^{-1}}^{C_n}], \tag{7}$$

where || is the concatenate operation.

Mask-GCN. The pipeline of Mask-GCN (Mask Graph Convolutional Network) is shown in Fig. 3. We first use the batch normalization layer to normalize the input skeleton sequence, then use multiple feature extraction blocks to extract deep features, and finally use a fully connected (FC) layer for the classification task. Each feature extraction block consists of a mask graph convolution and a T-Conv operation which denotes temporal convolution. We follow CTR-GCN (Chen et al. 2021) to implement the T-Conv, which contains four branches and each contains a 1 x 1 convolution to reduce channel dimension. The first branch uses 5 x 1 convolution with dilation equals 1. The second branch uses 5 x 1 convolution with dilation equals 2. The third branch uses 3 x 1 convolution and max pooling. The mask graph convolution extracts spatial features from normalized skeleton sequences. Then T-Conv further extracts temporal features. Batch normalization is used between the mask graph convolution and T-Conv to alleviate the overfitting problem. ReLU is used before and after T-Conv to increase the non-linear fitting capability. Residual connection is applied before and after T-Conv to avoid the degradation problem of the deep neural network. After applying multiple feature extraction blocks, the extracted deep feature is processed by an FC layer to generate a prediction value for each action type. The dropout layer is used before the FC layer to avoid overfitting.

Dataset. To evaluate our method, we use the pipeline shown in Fig. 4 to collect a new dataset. There are 21780 3D skeleton sequences in our dataset. Each action was repeatedly performed by 22 subjects 5 times and was observed by three Kinect V2 sensors from different viewpoints. These sensors are fixed on a robot platform to capture different robot views. Our dataset contains 10 types of actions, i.e., “clockwise”, “counterclockwise”, “keepClose”, “keepFar”, “left”, “right”, “nod”, “shake”, “raiseUp”, “wave”, and 5

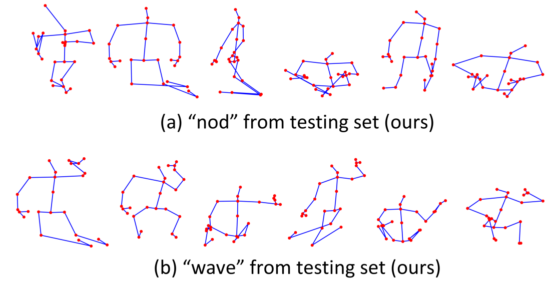


Figure 5: Randomly selected snaps from our dataset

Dataset	Sample	Noise
MSR-Action3D (Li, Zhang, and Liu 2010)	567	S,P
CAD-60 (Sung et al. 2011)	60	S,P
MSRDailyActivity3D (Wang et al. 2012)	320	S,P
UTKinect (Xia, Chen, and Aggarwal 2012)	200	S,P
Berkeley MHAD (Ofli et al. 2013)	660	S
CAD-120 (Koppula, Gupta, and Saxena 2013)	120	S,P
MSRAction-Pair (Oreifej and Liu 2013)	360	S,P
Northwestern-UCLA (Wang et al. 2014)	1475	S,P
UWA3D Multiview (Rahmani et al. 2014)	900	S,P
UWA3D Multiview II (Rahmani et al. 2016)	1075	S,P
NTU RGB,D (Shahroudy et al. 2016)	56880	S,P
PKU-MMD (Liu et al. 2017)	1076	S,P
RGB-D Varying-view (Ji et al. 2018)	25600	S,P
EV-Action (Wang et al. 2020)	7000	S
IKEA ASM (Ben-Shabat et al. 2021)	16764	S,P
Our newly collected dataset	**21780**	**S,P,NMP**

Table 2: Comparison between datasets, where “S” denotes noise from depth sensors, “P” denotes noise from pose estimation methods, and “NMP” denotes novel motion patterns.

types of novel motion patterns, i.e., “walking”, “sitDown”, “standUp”, “squat”, “squatUp”. Noting that novel motion patterns have various types, we just adopt 5 typical types as an example. Our training set contains samples with 10 types of actions. Our testing set contains samples with a mixture of actions and novel motion patterns. Taking action “nod” as an example, our training set only contains samples with action “nod”. Our test set contains samples that action “nod” and novel motion patterns concurrently happen, i.e., “nod + walking”, “nod + sitDown”, “nod + standUp”, “nod + squat”, “nod + squatUp”. Noting that novel motion patterns are invisible in the training phrase, therefore multi-label action recognition task cannot be used for this task.

Table 2 compares our dataset with the existing datasets for the skeleton-based action analysis task. As can be seen, the scale of our dataset is comparable with recent RGB-D Varying-view (Ji et al. 2018) and IKEA ASM (Ben-Shabat et al. 2021) datasets. In our dataset, each action type contains more than 1k samples, which ensures sufficient training and test samples. Moreover, our dataset contains three types of noise, i.e., “S”, “P” and “M”, where “S” stands for noise from depth sensors, “P” stands for noise from pose estimation methods, and “NMP” stands for novel motion patterns. While, most previous datasets only suffer from two types of noise, i.e., “S” and “P”. Instead of using pose estimation methods to estimate skeleton joints from depth sensors,

Protocol	CS_1 (%)		CS_2 (%)		CV_1 (%)		CV_2 (%)	
Action Type	CTR-GCN	Ours	CTR-GCN	Ours	CTR-GCN	Ours	CTR-GCN	Ours
clockwise	51.16	**66.86**	96.00	**98.00**	59.77	**83.55**	86.76	**89.89**
counterclockwise	62.21	**66.08**	80.54	**83.22**	66.34	**76.00**	79.58	**91.60**
keepClose	65.64	63.36	51.72	46.90	46.51	**63.64**	67.72	66.79
keepFar	52.95	**59.55**	71.23	**82.19**	78.50	54.55	70.46	69.36
left	64.34	54.36	88.00	85.33	55.41	**70.87**	79.39	77.67
right	36.54	**44.56**	48.32	42.95	48.04	**48.81**	70.92	**81.61**
nod	1.97	**2.56**	14.19	**14.19**	11.23	**49.42**	41.02	**42.72**
shake	32.53	30.57	6.04	**48.99**	0.09	**9.24**	0.00	**32.65**
raiseUp	48.72	**54.42**	89.58	84.03	36.02	**60.24**	77.34	**90.23**
wave	52.89	**68.17**	79.19	**83.22**	42.74	**61.55**	89.09	87.48
Average	46.90	**51.05**	62.48	**66.90**	44.46	**57.79**	66.23	**73.00**

Table 3: Comparison of per action recognition accuracy between CTR-GCN and ours

Protocol	CS_1 (%)		CS_2 (%)		CV_1 (%)		CV_2 (%)	
NMP	CTR-GCN	Ours	CTR-GCN	Ours	CTR-GCN	Ours	CTR-GCN	Ours
walking	41.58	**46.88**	60.49	**66.46**	51.85	**59.40**	57.71	**59.90**
sitDown	51.64	**51.73**	65.00	64.00	46.28	**55.78**	72.25	**76.94**
standUp	45.37	**52.48**	60.07	**65.02**	40.47	**59.76**	72.02	**75.82**
squat	54.04	**57.00**	68.57	**74.57**	44.76	**56.83**	69.65	**78.81**
squatUp	41.47	**46.73**	57.70	**64.22**	39.36	**57.40**	59.15	**73.04**

Table 4: Comparison of performances against novel motion patterns (short for NMP) between CTR-GCN and ours

Berkeley MHAD (Ofli et al. 2013) and EV-Action (Wang et al. 2020) use wearing sensors to generate skeleton joints, therefore these datasets do not contain noise from pose estimation methods. Compare with previous datasets, samples in our test set to contain more noise. As shown in Fig. 5 (b), these are randomly selected snaps from samples indicating action "wave", where the target motion patterns of "wave" suffers severe effect from novel motion patterns.

Experiments

We evaluate existing state-of-the-art skeleton-based action recognition methods and our proposed method on the newly collected dataset. Four types of evaluation protocols are performed, i.e., cross-subject recognition with low training data (CS_1), cross-subject recognition with more training data (CS_2), cross-view recognition with low training data (CV_1), cross-view recognition with more training data (CV_2). Specifically, CS_1 uses 10 subjects for training, CS_2 uses 20 subjects for training, CV_1 uses 1 view for training, and CV_2 uses 2 views for training. We report evaluation results of ST-GCN (Yan, Xiong, and Lin 2018) (AAAI 2018), CTR-GCN (Chen et al. 2021) (CVPR 2021) and info-GCN (Chi et al. 2022) (CVPR 2022), as these methods are either typical methods or currently the best methods for skeleton-based action recognition task. Our method is most comparable with CTR-GCN (Chen et al. 2021), as both methods use the same channel-wise topology refinement method for graph convolution and the same temporal convolution for temporal information aggregation. We use "Ours (Global)" to denote our proposed Mask-GCN with global modeling for implementing policy network. "Ours (Local)" denotes our proposed Mask-GCN with local modeling for implementing policy network. "Ours (Global & Local)" denotes our proposed Mask-GCN with both global and local modeling for implementing policy network. As "Ours (Local)" achieves the best performance, we use it as the final solution, which is short for "Ours".

Comparisons with State-of-the-arts

Table 3 compares our method with CTR-GCN, where We have three observations. First, despite the superior performances of CTR-GCN on skeleton-based action recognition task, its performance on our dataset is far from satisfactory. On our dataset, CTR-GCN only achieves an accuracy of 46.90% using CS_1 protocol and achieves an accuracy of 62.48% using CS_2 protocol. The main reason comes from the gap between training and test samples. The imperfect performances of CTR-GCN reflect that our dataset is more challenging than the existing datasets. Second, our method achieves obvious improvements over CTR-GCN, which verifies the effectiveness of our proposed mask graph convolution operation. Taking action "clockwise" as an example, our method achieves an accuracy of 66.86%, meanwhile, CTR-GCN only achieves an accuracy of 51.16%. Taking action "counterclockwise" as another example, our method achieves an accuracy of 66.08%, meanwhile CTR-GCN achieves an accuracy of 62.21%. Generally, our method outperforms CTR-GCN on most action types including "clockwise", "counterclockwise", "keepFar", "right", "nod", "raiseUp" and "wave". Third, our method achieves better results on CS_2 protocol than CS_1 protocol, which reflects that using more training data facilities the recognition task. Table 4 compares our method with CTR-GCN, and reports the recognition accuracy for all actions affected by novel motion patterns. Our method achieves a mean accuracy of 46.88% using CS_1 protocol,

Method	CS_1 (%)	CS_2 (%)	CV_1 (%)	CV_2 (%)
ST-GCN	49.82	61.73	53.18	68.94
CTR-GCN	46.90	62.48	44.46	66.23
info-GCN	49.80	**70.93**	46.35	65.95
Ours (Global)	**51.37**	65.79	45.53	68.81
Ours (Local)	51.05	66.90	**57.79**	**73.00**
Ours (Global, Local)	51.25	64.84	56.53	72.08

Table 5: Comparison between the state-of-the-arts and ours

which is 4.14% higher than CTR-GCN. Taking one type of novel motion patterns as an example, CTR-GCN achieves an accuracy of 41.58% on testing samples that are affected by novel motion patterns of "walking". Compared with CTR-GCN, our method achieves an accuracy of 46.88%, which outperforms CTR-GCN by 5.30%. Our method also achieves a mean accuracy of 66.85% using CS_2 protocol, which is 4.48% higher than CTR-GCN. These improvements show that our method can recognize actions that are severely affected by novel motion patterns, while previous skeleton-based action recognition method finds difficulty in directly using a well-trained model on test samples with novel motion patterns. Table 5 compares our method with ST-GCN (Yan, Xiong, and Lin 2018), CTR-GCN (Chen et al. 2021), and info-GCN (Chi et al. 2022). We find an interesting phenomenon that ST-GCN achieves comparable results with the most recent CTR-GCN and info-GCN methods. We infer that complex networks that perform well on skeleton-based action recognition task may show poor generalization probability to handle novel motion patterns. Our method with local modeling for implementing policy network is short for Ours (Local), which outperforms ST-GCN and CTR-GCN by a large margin using all kinds of protocols. Our method also outperforms info-GCN using CS_1, CV_1, and CV_2 protocols. The action "nod" and "shake" are difficult cases since their motion patterns are weak and therefore can be easily destroyed by noises.

Ablation Studies

We first evaluate the effect of our proposed **AMMG** block. Our model achieves accuracy of 51.05% using CS_1 protocol and achieves accuracy of 66.90% using CS_2 protocol. After removing **AMMG** block, the performance of our model drops by nearly 4% using either CS_1 protocol or CS_2 protocol. Second, we evaluate different designs for implementing our policy networks. As shown in Table 5, Ours (Local) outperforms Ours (Global & Local). We infer that a complex policy network may cause overfitting. Ours (Local) outperforms Ours (Global) using CS_2, CV_1, and CV_2 protocols. Therefore, we choose the policy network with local modeling as our solution. Noting that, our method using all these policy networks outperforms the performances of CTR-GCN, which verifies the effectiveness of using policy networks for generating adjacency matrix masks.

Parameter Selection & Feature Visualization

We evaluate the effect of parameter τ on the performance of Gumbel Softmax. We set τ to 0.1, 0.01 and 0.001, and find that the performance raises from 50.32% to 51.05% and then drops to 33.52%. We infer that a smaller value of τ benefits the Gumbel Softmax to generate one-hot-like vectors, therefore the performance raises when τ changes from 0.1 to 0.01. Since training with a much smaller value of τ is difficult, the performance drops when τ changes from 0.01 to 0.001. We set τ to 0.01 as the default for our policy network.

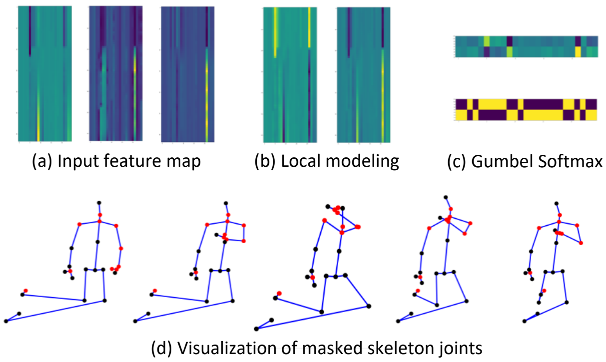


Figure 6: Visualization of feature maps and masked skeleton joints of action "keepClose" with novel motion patterns "walking". Masked skeleton joints are colored in black. Although skeleton joints are noisy (due to self-occlusions), our method suffers less by ignoring masked skeleton joints.

Fig. 6 takes an action "keepClose" which is affected by novel motion patterns "walking" as an example and shows the visualization of feature maps and masked skeleton joints. Fig. 6 (a) is the input feature map for our policy network located in the first mask convolution layer. After local modeling and Gumbel Softmax, our policy network selects skeleton joints that are colored in red (masked skeleton joints are colored in black). As can be seen, most action-specific skeleton joints are preserved.

Conclusion

This paper presents a Mask Graph Convolutional Network (Mask-GCN) framework to handle novel motion patterns in a practical skeleton-based action recognition task. Our Mask-GCN takes advantage of a new Adjacency Matrix Mask Generation (AMMG) block to learn action-dependant adjacency matrix mask, which guides graph convolution to learn spatial features from action-specific skeleton joints and to block disturbing from action-agnostic skeleton joints. We collect a challenging dataset and set up different protocols for evaluation. Extensive experiments on our dataset verify the effectiveness of our proposed Mask-GCN, which outperforms the most related CTR-GCN method by a large margin and also achieves comparable results with the state-of-the-art info-GCN method using most protocols.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62203476, 61906103, 62031013) and the Major Key Project of PCL (No. PCL2021A03-1).

References

Ben-Shabat, Y.; Yu, X.; Saleh, F.; Campbell, D.; Rodriguez-Opazo, C.; Li, H.; and Gould, S. 2021. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 847–859.

Chen, C.; Jafari, R.; and Kehtarnavaz, N. 2015. UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *IEEE International conference on image processing (ICIP)*, 168–172.

Chen, C.; Liu, M.; Zhang, B.; Han, J.; Jiang, J.; and Liu, H. 2016. 3D action recognition using multi-temporal depth motion maps and fisher vector. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 3331–3337.

Chen, Y.; Zhang, Z.; Yuan, C.; Li, B.; Deng, Y.; and Hu, W. 2021. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 13359–13368.

Cheng, K.; Zhang, Y.; Cao, C.; Shi, L.; Cheng, J.; and Lu, H. 2020a. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *European Conference on Computer Vision (ECCV)*, 536–553.

Cheng, K.; Zhang, Y.; He, X.; Chen, W.; Cheng, J.; and Lu, H. 2020b. Skeleton-based action recognition with shift graph convolutional network. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 183–192.

Chi, H.-g.; Ha, M. H.; Chi, S.; Lee, S. W.; Huang, Q.; and Ramani, K. 2022. InfoGCN: Representation Learning for Human Skeleton-Based Action Recognition. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 20186–20196.

Du, Y.; Fu, Y.; and Wang, L. 2015. Skeleton based action recognition with convolutional neural network. In *IAPR Asian Conference on Pattern Recognition (ACPR)*, 579–583.

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

Huang, L.; Huang, Y.; Ouyang, W.; and Wang, L. 2020a. Part-level graph convolutional network for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence (AAAI)*, 11045–11052.

Huang, Z.; Shen, X.; Tian, X.; Li, H.; Huang, J.; and Hua, X.-S. 2020b. Spatio-temporal inception graph convolutional networks for skeleton-based action recognition. In *ACM International Conference on Multimedia (ACM MM)*, 2122–2130.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Ji, Y.; Xu, F.; Yang, Y.; Shen, F.; Shen, H. T.; and Zheng, W.-S. 2018. A large-scale RGB-D database for arbitrary-view human action recognition. In *ACM international Conference on Multimedia (ACM MM)*, 1510–1518.

Ke, Q.; Bennamoun, M.; An, S.; Sohel, F.; and Boussaid, F. 2017. A new representation of skeleton sequences for 3d action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3288–3297.

Koppula, H. S.; Gupta, R.; and Saxena, A. 2013. Learning human activities and object affordances from rgb-d videos. *International Journal of Robotics Research*, 32(8): 951–970.

Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; and Tian, Q. 2019. Actional-structural graph convolutional networks for skeleton-based action recognition. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 3595–3603.

Li, W.; Zhang, Z.; and Liu, Z. 2010. Action recognition based on a bag of 3d points. In *IEEE/CVF International Conference on Computer Vision Workshops (CVPRW)*, 9–14.

Liu, C.; Hu, Y.; Li, Y.; Song, S.; and Liu, J. 2017. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*.

Liu, J.; Shahroudy, A.; Xu, D.; and Wang, G. 2016. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision (ECCV)*, 816–833.

Liu, M.; Liu, H.; and Chen, C. 2017. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68: 346–362.

Liu, M.; Meng, F.; and Liang, Y. 2022. Generalized Pose Decoupled Network for Unsupervised 3D Skeleton Sequence-Based Action Representation Learning. *Cyborg and Bionic Systems*, 2022: 0002.

Liu, M.; and Yuan, J. 2018. Recognizing human actions as the evolution of pose estimation maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1159–1168.

Liu, Z.; Zhang, H.; Chen, Z.; Wang, Z.; and Ouyang, W. 2020. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 143–152.

Ofli, F.; Chaudhry, R.; Kurillo, G.; Vidal, R.; and Bajcsy, R. 2013. Berkeley mhad: A comprehensive multimodal human action database. In *IEEE Workshop on Applications of Computer Vision (WACV)*, 53–60.

Oreifej, O.; and Liu, Z. 2013. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 716–723.

Plizzari, C.; Cannici, M.; and Matteucci, M. 2021. Skeleton-based action recognition via spatial and temporal transformer networks. *Computer Vision and Image Understanding*, 208: 103219.

Rahmani, H.; Mahmood, A.; Du Huynh, Q.; and Mian, A. 2014. HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition. In *European Conference on Computer Vision (ECCV)*, 742–757.

Rahmani, H.; Mahmood, A.; Huynh, D.; and Mian, A. 2016. Histogram of oriented principal components for cross-view action recognition. *IEEE Transactions on Pattern Analysis and Machine iIntelligence*, 38(12): 2430–2443.

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

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019a. Skeleton-based action recognition with directed graph neural networks. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 7912–7921.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019b. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 12026–12035.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2021. AdaSGN: Adapting Joint Number and Model Size for Efficient Skeleton-Based Action Recognition. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 13413–13422.

Si, C.; Chen, W.; Wang, W.; Wang, L.; and Tan, T. 2019. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 1227–1236.

Song, S.; Lan, C.; Xing, J.; Zeng, W.; and Liu, J. 2017. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI Conference on Artificial Intelligence (AAAI)*, 4263–4270.

Song, Y.-F.; Zhang, Z.; Shan, C.; and Wang, L. 2020. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *ACM International Conference on Multimedia (ACM MM)*, 1625–1633.

Sung, J.; Ponce, C.; Selman, B.; and Saxena, A. 2011. Human activity detection from RGBD images. In *Workshops at AAAI conference on Artificial Intelligence (AAAIW)*.

Tu, Z.; Li, H.; Zhang, D.; Dauwels, J.; Li, B.; and Yuan, J. 2019. Action-stage emphasized spatiotemporal VLAD for video action recognition. *IEEE Transactions on Image Processing*, 28(6): 2799–2812.

Tu, Z.; Zhang, J.; Li, H.; Chen, Y.; and Yuan, J. 2022. Joint-bone fusion graph convolutional network for semi-supervised skeleton action recognition. *IEEE Transactions on Multimedia*.

Wang, J.; Liu, Z.; Wu, Y.; and Yuan, J. 2012. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 1290–1297.

Wang, J.; Nie, X.; Xia, Y.; Wu, Y.; and Zhu, S.-C. 2014. Cross-view action modeling, learning and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2649–2656.

Wang, L.; Sun, B.; Robinson, J.; Jing, T.; and Fu, Y. 2020. Ev-action: Electromyography-vision multi-modal action dataset. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, 160–167.

Xia, L.; Chen, C.-C.; and Aggarwal, J. K. 2012. View invariant human action recognition using histograms of 3d joints. In *IEEE/CVF International Conference on Computer Vision Workshops (CVPRW)*, 20–27.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence (AAAI)*, 7444–7452.

Ye, F.; Pu, S.; Zhong, Q.; Li, C.; Xie, D.; and Tang, H. 2020. Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In *ACM International Conference on Multimedia (ACM MM)*, 55–63.

Zhang, J.; Jia, Y.; Xie, W.; and Tu, Z. 2022. Zoom Transformer for Skeleton-Based Group Activity Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12): 8646–8659.

Zhang, P.; Lan, C.; Zeng, W.; Xing, J.; Xue, J.; and Zheng, N. 2020. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1112–1121.

Zhang, X.; Xu, C.; and Tao, D. 2020. Context aware graph convolution for skeleton-based action recognition. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 14333–14342.

Zhang, Y.; Wu, B.; Li, W.; Duan, L.; and Gan, C. 2021. STST: Spatial-temporal specialized transformer for skeleton-based action recognition. In *ACM International Conference on Multimedia (ACM MM)*, 3229–3237.

Zhao, R.; Wang, K.; Su, H.; and Ji, Q. 2019. Bayesian graph convolution LSTM for skeleton based action recognition. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 6882–6892.