# Hierarchically Decomposed Graph Convolutional Networks for Skeleton-Based Action Recognition

Jungho Lee[1] , Minhyeok Lee[1] , Dogyoon Lee[1] , Sangyoun Lee[1,2]

[1]School of Electrical and Electronic Engineering, Yonsei University
[2]AIonFlow Research
{2015142131, hydragon516, nemotio, syleee}@yonsei.ac.kr

## Abstract

*Graph convolutional networks (GCNs) are the most commonly used methods for skeleton-based action recognition and have achieved remarkable performance. Generating adjacency matrices with semantically meaningful edges is particularly important for this task, but extracting such edges is challenging problem. To solve this, we propose a hierarchically decomposed graph convolutional network (HD-GCN) architecture with a novel hierarchically decomposed graph (HD-Graph). The proposed HD-GCN effectively decomposes every joint node into several sets to extract major structurally adjacent and distant edges, and uses them to construct an HD-Graph containing those edges in the same semantic spaces of a human skeleton. In addition, we introduce an attention-guided hierarchy aggregation (A-HA) module to highlight the dominant hierarchical edge sets of the HD-Graph. Furthermore, we apply a new six-way ensemble method, which uses only joint and bone stream without any motion stream. The proposed model is evaluated and achieves state-of-the-art performance on four large, popular datasets. Finally, we demonstrate the effectiveness of our model with various comparative experiments. Code is available at* https://github.com/Jho-Yonsei/HD-GCN.

## 1. Introduction

Human action recognition (HAR) is a task that categorizes action classes by receiving video data as input. HAR is used in many applications, such as human–computer interaction and virtual reality. Recently, several RGB-based and skeleton-based HAR methods have been proposed with the development of deep learning technology. However, RGB-based methods [31, 29] cannot robustly recognize human actions because they are strongly influenced by environmental noises such as background color, brightness of light, and clothing. Therefore, methods using skeleton modality [35, 24, 36, 26, 5, 4, 18, 2, 15] have attracted attention because they are not affected by these noises. These methods recognize action by receiving 2D or 3D coordinates of major human joints as time-series inputs.

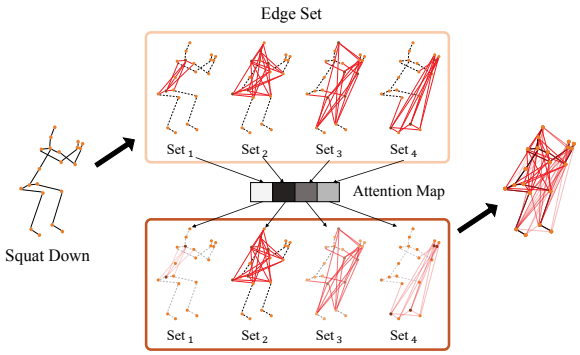


Figure 1. **The framework of HD-GCN.** The input skeleton is applied with various edge sets through a hierarchically decomposed graph (HD-Graph). The red lines are the edges included in the corresponding hierarchy edge set. The network highlights the major edge sets through the attention map. The darker the color of red line, the more highlighted the edge set, and dotted lines denote unconnected edges.

Recent approaches [24, 18, 4, 2] have adopted graph convolutional networks (GCNs) to apply human-skeleton graphs to convolutional layers. However, existing GCN-based methods [35, 24, 25, 4, 2] have the following limitations. (1) With the widely used handcrafted graph, the relationships between distant joint nodes are not identified since they use only the relationships of PC edges in the human skeleton. Although the graph with PC edges has a semantic significance, the graph with only PC edges suffers from long-range dependency problem as they are heuristically fixed. However, for humans to recognize actions, relationships between structurally distant joints as well as between adjacent joints are strongly correlated. Although several methods [24, 2] have attempted to solve such limita-

tion by training attention-guided learnable graphs, they still use [35]'s handcrafted graph with their learnable graphs. Moreover, as the element values of [35]'s graph are more dominant than those of the learnable graphs, they do not adequately highlight the relationships between distant nodes. (2) Some recent methods [35, 24, 4, 18] risk falling into suboptimality by simply aggregating the edge features and ignoring the contribution of each edge, thus incompletely recognizing which edges are significant for each skeleton sample. For example, in the case of a 'squat down' action, the interactions between the legs and arms should be highlighted.

Motivated by these limitations, we propose a hierarchically decomposed graph convolutional network (HD-GCN) with a hierarchically decomposed graph (HD-Graph) and attention-guided hierarchy aggregation (A-HA) module. In addition, we present a six-way ensemble method to effectively utilize our HD-Graph. The framework of our proposed methods is shown in Fig. 1 for 'squat down' action.

The HD-GCN incorporates GCNs with our HD-Graph, which identifies the relationships between distant joint nodes in the same semantic spaces (*e.g.* right and left hands, right and left feet). The same semantic spaces are formed by moving out step by step from the Center of Mass (CoM) node of the graph. For example, if belly is a CoM node, the first semantic space includes the belly node, the next space includes the chest and hip nodes, and the subsequent space includes the left and right shoulder and the left and right hip nodes. The nodes in the same semantic space are defined as hierarchy node set. To detect the relationships between distant joint nodes, network should have large receptive field. The proposed HD-Graph contains both meaningful adjacent and distant joint nodes by connecting all the nodes in neighboring hierarchy node sets and identifies the connectivity between those nodes for large receptive field. We adopt rooted tree-like structure to effectively represent every edges. We apply a spatial edge convolution (S-EdgeConv) layer to consider semantically close edges which cannot be captured by the HD-Graph for each sample. To create the S-EdgeConv layer, we borrow the structure of [33], which is widely used in 3D point clouds.

To consider the contribution of each edge set, the process of selecting the dominant hierarchical information should depend on the action data sample to give proper attention to the most dominant edge sets. For example, in order to recognize the "clapping" action, a hierarchy edge set that includes both hands must be emphasized. To tackle this issue, we propose an attention-guided hierarchy aggregation (A-HA) module, which consists of two submodules: representative spatial average pooling (RSAP) and hierarchical edge convolution (H-EdgeConv). A scaling bias problem occurs if we use the spatial average pooling module without any node extraction process because each node has a different number of adjacent nodes. To prevent this, we apply RSAP, which includes a representative node extraction process that triggers features after the pooling layer to represent each node. To effectively manage hierarchical features obtained by RSAP, we apply a hierarchical edge convolution (H-EdgeConv) layer. The H-EdgeConv treats each hierarchical feature as a graph node and identifies which hierarchical features should be highlighted via the Euclidean distance in feature space. With the RSAP and the H-EdgeConv, our model successfully determines which hierarchy edge sets and joints should be emphasized among the input features.

The existing ensemble method uses four-stream data composed of the joint, bone, joint motion, and bone motion streams, which are the original skeletal coordinates, spatial differential between joint coordinates, and temporal differential of joint, and temporal differential of the bone, respectively. Most existing ensemble methods [25, 2] use additional motion data, but models that solely utilize motion data exhibit relatively inferior performance. To address this problem, we present a new method, a six-way ensemble. We apply this ensemble method by setting three HD-Graphs with joint and bone stream data. Each graph has different CoM nodes to extract features of different semantic spaces (see **Appendix**).

We conduct extensive experiments on four benchmark action recognition datasets: NTU-RGB+D 60 [22], NTU-RGB+D 120 [16], Kinetics-Skeleton [11], and Northwestern-UCLA [30].

Our main contributions are summarized as follows:

- We propose a hierarchically decomposed graph (HD-Graph) to thoroughly identify the significant distant edges between the same hierarchy node sets.
- We propose an attention-guided hierarchy aggregation (A-HA) module to highlight the key edge sets with representative spatial average pooling (RSAP) and hierarchical edge convolution (H-EdgeConv).
- We use a new six-way ensemble method for skeleton-based action recognition with HD-Graphs that have different center of mass (CoM), which outperforms regular ensemble without any motion data.
- Our HD-GCN outperforms the state-of-the-arts on four benchmarks for skeleton-based action recognition.

## 2. Related Work

### 2.1. Action Recognition with GCNs

In skeleton-based action recognition, human skeletal data are represented by a graph with joint nodes. Most recent approaches use GCN-based methods [35, 24, 5, 18, 2] with [35]'s graph structure, which identifies physical connections in the human skeleton. Those GCN-based meth-

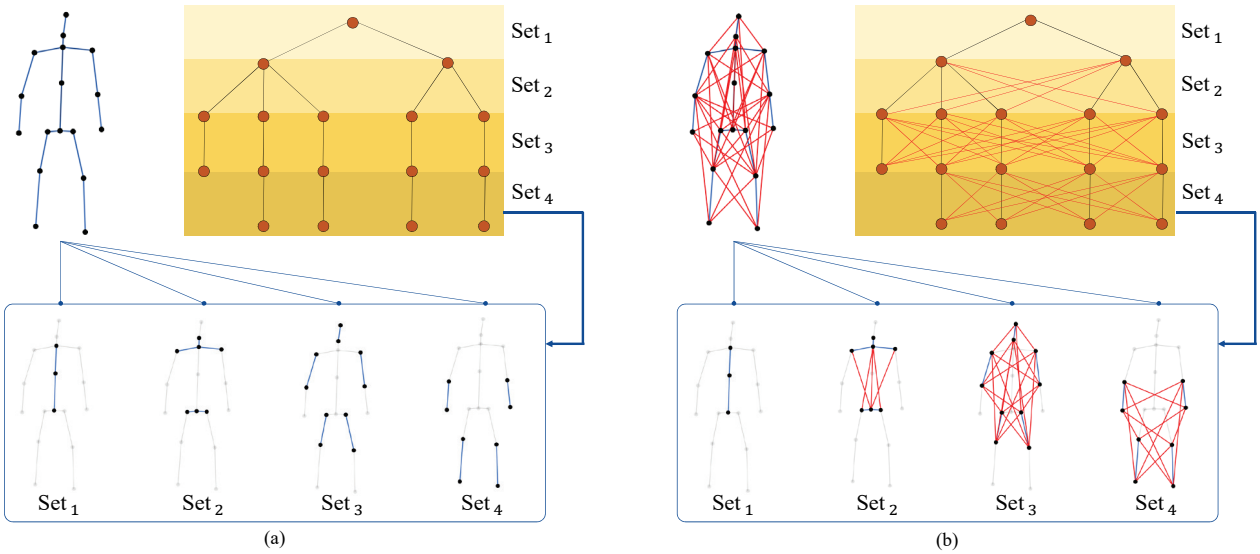


Figure 2. **(a) Structure of HD-Graph with physically connected (PC) edges.** The human skeleton graph is decomposed into a rooted tree, where PC edges are included in hierarchy sets. **(b) Structure of HD-Graph with fully connected (FC) edges.** Edges between all nodes in the same semantic space are obtained by connecting all the nodes in adjacent hierarchy edge sets. Blue and red lines stand for PC and FC edges, respectively.

ods perform remarkably better than methods using handcrafted features [7, 8, 13, 20, 21]. They extract the spatial features representing the relationships between physically connected edges among human skeleton, and they outperform other methods by using them to construct the major relationships between joint nodes in the human skeleton. In particular, [25] and [2] propose adaptive attention-based graph structures to learn the sample-wise topological features. However, they might fall into suboptimality because they do not consider the physical prior of the human skeletal structure and allow too much flexibility in network training. To address this issue, we introduce a novel HD-Graph, referencing the known tendencies of human perception

### 2.2. Attention Modules for Action Recognition

The attention mechanism is an essential element for constructing a deep neural network. Using recent attention modules [10, 34], networks emphasize important information along a specific dimension. For example, Hu *et al*. [10] applies channel-wise attention, and Woo *et al*. [34] applies both channel-wise and spatial-wise attentions. These techniques are divided into two categories for GCNs: (1) attention-based graph construction [24, 2] which is a method of forming topologies using a non-local block [32] or customized correlation matrices, and (2) spatial-wise, temporal-wise, channel-wise attention, which are commonly used attentions in [25, 27], and several other networks.

## 3. Methodology

In Sec. 3.2, we detail the HD-Graph convolution to solve the problems of the conventional human-skeleton graph [35], which includes only PC edges. We also explain the A-HA module in Sec. 3.3 to highlight dominant hierarchical features. In Sec. 3.4, we replace the widely used four-stream ensemble method [25, 2, 4] with a six-way ensemble without motion data streams. Finally, we introduce the HD-GCN, which uses these proposed methods.

### 3.1. Preliminaries

**Notations.** The spatio-temporal graph for human skeleton is represented by $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the joint and edge groups, respectively. Physically connected edges and fully-connected edges used in Sec. 3.2 are denoted as PC-edges and FC-edges, respectively.

**Graph Convolutional Networks.** 3D time-series skeletal data are represented by $\mathbf{X} \in \mathbb{R}^{3\times T\times V}$, where $V$ and $T$ are the number of joint nodes and the temporal window size, respectively. GCN's operation with input feature map $\mathbf{F}_{in} \in \mathbb{R}^{C\times T\times V}$ is as follows:

$$\mathbf{F}_{out} = \sum_{s\in S} \widehat{\mathbf{A}}_s \mathbf{F}_{in} \Theta_s, \tag{1}$$

where $S = \{s_{id}, s_{cf}, s_{cp}\}$ denotes graph subsets, and $s_{id}$, $s_{cf}$, and $s_{cp}$ indicate identity, centrifugal, and centripetal joint subsets, respectively. $\Theta_s$ denotes the pointwise convolution operation. The normalized adjacency matrix $\widehat{\mathbf{A}}$ is

initialized as $\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{A}\mathbf{\Lambda}^{-\frac{1}{2}} \in \mathbb{R}^{N_S \times V \times V}$, where $\mathbf{\Lambda}$ is a diagonal matrix for normalization and $N_S = 3$.

### 3.2. Hierarchically Decomposed Graph

Most recent methods have adopted the handcrafted graph proposed by Yan *et al*. [35], but the HD-Graph is derived through a newly presented method. Fig. 2 shows the framework of the HD-Graph.

**Decomposition into a Rooted Tree.** The first step is to decompose the graph with PC edges and construct a rooted tree. To decompose a given skeleton into the tree, we need to determine a CoM node, which allows nodes in the same hierarchy edge set to exist in the same semantic space. For example, nodes in the same semantic space, such as elbow and knee joints, or hands and feet, must exist in a hierarchy node set. After choosing the CoM node, the graph is converted into a rooted tree, which includes the hierarchical information of the graph, and defines the directed adjacency matrix $\overrightarrow{\mathbf{A}}_{\text{HD}} \in \mathbb{R}^{N_L \times V \times V}$ with $N_L$ hierarchy layers for $N_H$ hierarchy edge sets:

$$\overrightarrow{\mathbf{A}}_{\text{HD}} = \left[\mathcal{E}(H_1 \rightarrow H_2),\ \cdots,\ \mathcal{E}(H_{N_H-1} \rightarrow H_{N_H})\right], \quad (2)$$

where $H_k$ denotes the $k$-th hierarchy node set and $\mathcal{E}(H_k \rightarrow H_{k+1})$ denotes a set of edges from $H_k$ to $H_{k+1}$. $N_L$ and $N_H$ are the number of hierarchy layers and hierarchy edge sets, respectively, and $N_L = N_H - 1$. However, $\overrightarrow{\mathbf{A}}_{\text{HD}}$ includes only the directed centrifugal edges. For consistency with existing methods, all the reverse-directed edges from the leaf nodes of the rooted tree in Fig. 2 to the CoM node must be reflected in the adjacency matrices to cover the centripetal edges. In addition, to get the features of the nodes themselves, the identity edges for each hierarchy node set must be considered. Thus, the adjacency matrices $\overleftrightarrow{\mathbf{A}}_{\text{HD}} \in \mathbb{R}^{N_L \times N_S \times V \times V}$ are defined as follows:

$$\overleftrightarrow{\mathbf{A}}_{\text{HD}} = \left[\ \mathcal{E}_1,\ \mathcal{E}_2,\ \cdots,\ \mathcal{E}_{N_L}\ \right], \quad (3)$$

$$\mathcal{E}_k = \mathcal{E}(\underbrace{H_k \cup H_{k+1}}_{s_{id}},\ \underbrace{H_k \rightarrow H_{k+1}}_{s_{cp}},\ \underbrace{H_{k+1} \rightarrow H_k}_{s_{cf}}), \quad (4)$$

where $\mathcal{E}_k$ denotes the concatenation of the three edge subsets of $S = \{s_{id}, s_{cp}, s_{cf}\}$ and $s_{id}, s_{cp}, s_{cf}$ indicate the identity, centripetal, and centrifugal edge subsets, respectively. Through this construction policy, we create a skeletal graph with bidirectional and identity edges.

**Fully Connected Inter-Hierarchy Edges.** Decomposed graph $\overleftrightarrow{\mathbf{A}}_{\text{HD}}$ has a different number of edge sets from the conventional graph, but the edges are all the same. To identify the relationships between major distant joint nodes, especially those in the same semantic space, we connect all nodes between neighboring hierarchy node sets. In addition, since [35]'s graph contains the connectivity of only PC edges, not distant relationships, the receptive field is very small with this sparse graph. Applying our fully connected (FC) edges to the rooted tree, the graph becomes denser and makes the receptive field larger than before with more meaningful distant connectivity as shown in Fig. 2 (b). Then, the adjacency matrices are normalized with degree matrices for training stability and we leave all elements of the matrices as learnable parameters for training adaptability.

**HD-Graph Convolution.** Our HD-Graph convolution includes four parallel branch operations: three graph convolution through HD-Graph and an additional EdgeConv [33] operation. To reduce the computational complexity, a linear transformation is applied to all four operations. For three of these operations, our method performs a subset-wise GCN operation in the same way as [35, 24] for each hierarchy edge set with three edge subsets. However, rather than summing the output values for each subset as in Eq. (1), we concatenate these output values to the channel dimension:

$$\mathbf{F}_{\text{HD}}^{(k)} = \Big\|_{s \in S} \left\{ \overleftrightarrow{\mathbf{A}}_{\text{HD};s}^{(k)} \Phi(\mathbf{F}_{in}) \Theta_s^{(k)} \right\}, \quad (5)$$

where $\mathbf{F}_{\text{HD}}^{(k)}$ denotes the output feature map of the HD-Graph convolution and function $\Phi$ denotes a linear transformation with parameter $\mathbf{W} \in \mathbb{R}^{C' \times C}$. Note that $\|$ is a concatenation operation.

Although our HD-Graph defines more meaningful node relationships than conventional graph, it may still not be able to extract sample-wise key relationships that reflect the similarities between all nodes in the feature space. To improve this limitation, we adopt EdgeConv [33] as the remaining operation, which is used for extracting graphical features through local neighborhood graphs in the feature space. With spatial EdgeConv (S-EdgeConv), our network extracts sample-wise node connectivity, which the HD-Graph does not capture. For our method, S-EdgeConv initially takes the average pooling as the temporal dimension for computational efficiency. Local graphs with local edges are then formed via k-nearest neighbor (k-NN) based on the Euclidean distance, and the local edges as well as identity edges based on the graphs are aggregated via trainable parameters $\mathbf{W}_{edge} \in \mathbb{R}^{C' \times 2C'}$. For the deep neural network, physically close edges are reflected to the initial shallow layers, but as they become deeper, the relationship between semantically similar edges in the feature space are identified and learned. Our whole GCN process is shown in

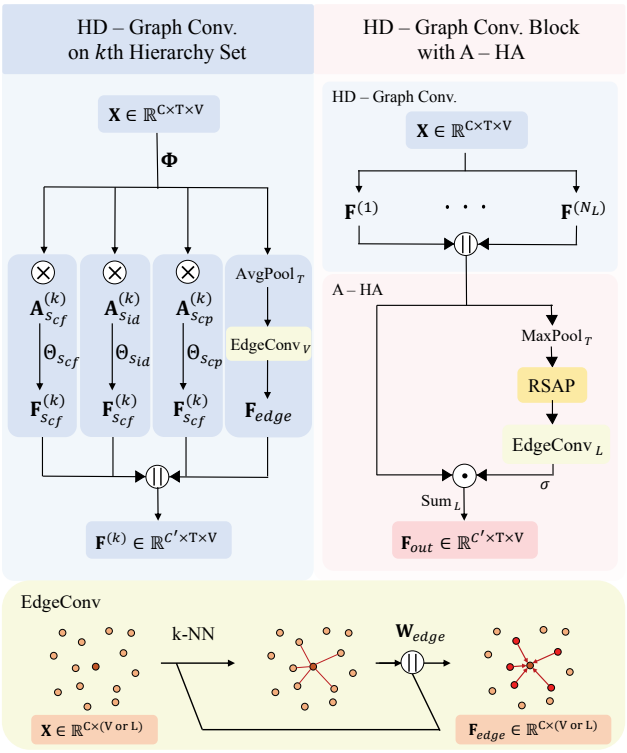


Figure 3. **HD-Graph convolution operation block with an A-HA module.** The left side shows an operation for one hierarchy edge set, and the right side shows an operation block that concatenates the results for $N_L$ edge sets and applies A-HA. The lower part of the figure is EdgeConv, where the EdgeConv subscript indicates the feature space to extract graphical features. The $\times$ and $\cdot$ operations denote matrix and element-wise multiplication.

Fig. 3 and computed as follows:

$$\mathbf{F}_{\text{HD}} \leftarrow \sum_{k=1}^{N_L} \left[ \mathbf{F}_{\text{HD}}^{(k)} \,\|\, z_{\mathcal{V}}^{(k)} \left( \frac{1}{T} \sum_{t=1}^{T} \Phi(\mathbf{F}_{in}) \right) \right], \tag{6}$$

where $z_{\mathcal{V}}$ denotes the S-EdgeConv operation.

All four branch outputs are concatenated to the channel dimension, with all four computed in the same way for $N_L$ hierarchy edge sets. Due to the inherent characteristics of skeletal data, the number of joint nodes included in each dataset is different, and, consequently, the number of hierarchy sets is different. Therefore, we adopt an addition policy for $N_L$ hierarchy-wise outputs and a concatenation policy for $N_S$ subset-wise outputs. In this way, the dimensionality is maintained, and the common hierarchy-wise aggregation policy is followed for every skeletal dataset by adding all the outputs for different numbers of hierarchical sets.

### 3.3. Attention-Guided Hierarchy Aggregation

The HD-Graph convolution uses an aggregation policy of adding all the hierarchy-wise outputs. However, because each data sample has relationship between specific major edges, we propose an attention-guided hierarchy aggregation (A-HA) module, which applies a weighted-sum policy to the hierarchy dimension with proper attention to the hierarchy-wise outputs. The framework of the HD-Graph convolution with the A-HA module is shown in Fig. 3.

**Representative Spatial Average Pooling.** Our A-HA module is applied to feature map $\mathbf{F}_{\text{HD}} \in \mathbb{R}^{C \times N_L \times T \times V}$ after the HD-Graph convolution. The first step is to extract the temporal frame with the highest score on $\mathbf{F}_{\text{HD}}$. RSAP, $\Psi$, is then applied, which is preceded by the extraction of representative nodes in each hierarchy layer. If spatial average pooling is applied without this extraction process, scaling bias occurs because the number of edges connected to each node is different. Therefore, representative node extraction is essential to obtain an appropriate score for attention without any bias. After the extraction, spatial average pooling is applied to hierarchy-wise outputs. Our pooling function $\Psi$ is as follows:

$$\Psi(\mathbf{F}_{\text{HD}}^{(k)}) = \frac{1}{N_k + N_{k+1}} \sum_{v \in H_k \cup H_{k+1}} \max_t \left( \mathbf{F}_{\text{HD}}^{(k)}(v) \right), \tag{7}$$

where $N_k$ denotes the number of vertices in the $H_k$ set.

**Hierarchical Edge Convolution.** After the RSAP layer, $N_L$ hierarchy-wise features in attention feature map $\mathbf{M}$ have not yet shared their information with each other. We treat all $N_L$ features as nodes on a graph to learn and reflect similarities in the hierarchical feature space. To apply this process, representative features of these nodes are fed into EdgeConv [33], and the similarities of those nodes are learned based on the Euclidean distance. We also include the self-loop shown in the bottom section of Fig. 3 so that the node's own features can be reflected. Our attention map $\mathbf{M}$ operates as follows:

$$\mathbf{M} = \sigma \left( z_L \left( \Big\| _{k \in L} \left\{ \Psi \left( \mathbf{F}_{\text{HD}}^{(k)} \right) \right\} \right) \right), \tag{8}$$

where $z_L$ and $\sigma$ denote H-EdgeConv and the sigmoid function, respectively.

The attention map $\mathbf{M}$ obtained is multiplied by the HD-Graph convolution output feature map $\mathbf{F}_{\text{HD}}$, and the output feature map $\mathbf{F}_{out}$ is obtained through a weighted sum to the hierarchy axis as shown in Fig. 3. Similar to S-EdgeConv in Sec. 3.2, The H-EdgeConv method incorporates the concept of hierarchical edge sets in a physically proximal manner in earlier layers, while in the deeper layers, it emphasizes the presence of semantically similar edge sets. This approach highlights different hierarchical edge sets for each sample and enables the model to learn meaningful representations that capture both physical and semantic properties of the input data.

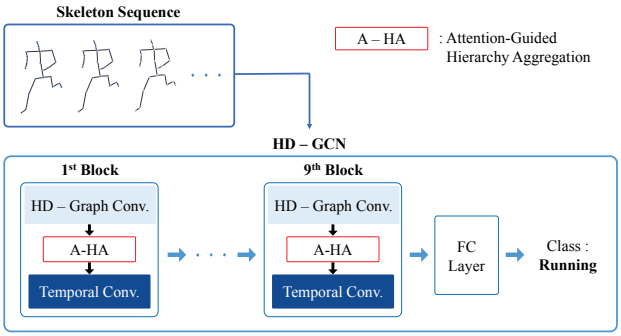


Figure 4. **The architecture of HD-GCN.** HD-GCN receives the skeleton sequence as input and obtains the class label through nine GCN blocks, and an FC layer, and the softmax function.

### 3.4. Six-Way Ensemble

Shi *et al*. [24, 25] have applied a four-stream ensemble method using streams for joints, bones, joint motion, and bone motion. However, as the performances of motion streams are relatively poorer than the performances of joint and bone streams, we adopt an ensemble method with the joint and bone streams without any motion streams. We use three different HD-Graphs, and each graph is used for training with joint and bone streams. The three HD-Graphs have different CoM nodes, which are chest, belly, hip nodes, respectively. In other words, we train joint and bone streams with HD-Graph with the CoM node of chest, and we train the same when the CoM node is belly or hip node. As models with the three different graphs should be trained in different aspects, each of the graphs is composed of different edge sets. For example, if the CoM node is belly, both thigh edges and both upper arm edges are included in the same edge set, whereas when the CoM node is chest, both thigh edges and both forearm edges are included in the same edge set. The details of our six-way ensemble are specified on our **Appendix**.

### 3.5. Network Architecture

As shown in Fig. 4, we adopt [24] as our baseline network architecture with a total of nine stacked GCN blocks. The numbers of output channels for the blocks are 64, 64, 64, 128, 128, 128, 256, 256, and 256. Each block contains a residual connection [9] and is divided into a spatial module, in which the GCN operation proceeds, and a temporal module, which includes the temporal convolutions. Our method use the temporal module of [2], whose baseline module is [18, 28]. This module consists of four branch operations. Two are dilated temporal convolutions with kernel size five and dilation one and two, respectively. The remaining branch operations are pointwise convolution and max pooling with kernel size three. Our spatial module consists of an HD-Graph convolution operation and an A-HA module, as introduced in Sec. 3.2 and Sec. 3.3. After passing through all GCN layers with attention to the hierarchy-wise features, the network compresses the feature map through the global average pooling layer and classifies the action sample through the softmax function.

## 4. Experiments

### 4.1. Datasets and Experimental Settings

**NTU-RGB+D 60.** NTU-RGB+D 60 [22] is a large dataset used in skeletal action recognition. It contains 56,880 skeleton action samples, performed by 40 different participants and classified into 60 classes. The authors of this dataset recommend two benchmarks. (1) Cross-Subject (X-Sub): 20 of the 40 subjects' actions are used for training, and the remaining 20 are for validation. (2) Cross-View (X-View): Two of the three camera-views are used for training, and the other one is used for validation.

**NTU-RGB+D 120.** NTU-RGB+D 120 [16] is a dataset in which 57,367 new action samples are added to the NTU-RGB+D 60 dataset. It contains a total of 114,480 skeleton action samples over 120 classes, performed by 106 different subjects. The authors of this dataset recommend two benchmarks: (1) Cross-Subject (X-Sub): 53 of the 106 subjects' actions are used for training, and the remaining 53 are used for validation. (2) Cross-Setup (X-Set): Of the 32 setups, data with even setup IDs are used for training, and the remaining data with odd IDs are used for validation.

**Kinetics-Skeleton.** The Kinetics-Skeleton dataset is derived from the Kinetics 400 video dataset [11], utilizing the OpenPose pose estimation [1] to extract 240,436 training and 19,796 testing skeleton sequences across 400 classes. The dataset restricts the number of skeletons per time step to two and eliminates skeletons with lower confidence scores, ensuring high-quality sequences for human action recognition and pose estimation research.

**Northwestern-UCLA.** The Northwestern-UCLA skeleton dataset [30] contains 1494 video clips over 10 classes. Each action is captured through three Kinect cameras with different camera views and is performed by 10 subjects. We adopt the same protocol as NW-UCLA: Two of the three camera-views are used for training, and the other one is used for validation.

**Experimental Settings.** In our experiments, we adopt [24] as the backbone. The SGD optimizer is employed with a Nesterov momentum of 0.9 and a weight decay of 0.0004. The number of learning epochs is set to 90, with a warm-up strategy [9] applied to the first five epochs for more stable learning. We set the learning rate to decay with cosine annealing [19], with a maximum

| Methods | Publication | Motion Stream | NTU-RGB+D 60 | | NTU-RGB+D 120 | | Kinetics-Skeleton | | Northwestern |
|---|---|---|---|---|---|---|---|---|---|
| | | | X-Sub (%) | X-View (%) | X-Sub (%) | X-Set (%) | Top-1 (%) | Top-5 (%) | UCLA (%) |
| ST-GCN [35] | AAAI 2018 | ✗ | 81.5 | 88.3 | 70.7 | 73.2 | 30.7 | 52.8 | - |
| 2s-AGCN [24] | CVPR 2019 | ✗ | 88.5 | 95.1 | 82.5 | 84.2 | 36.1 | 58.7 | - |
| SGN [36] | CVPR 2020 | ✓ | 89.0 | 94.5 | 79.2 | 81.5 | - | - | 92.5 |
| AGC-LSTM [26] | CVPR 2019 | ✗ | 89.2 | 95.0 | - | - | - | - | 93.3 |
| DGNN [23] | CVPR 2019 | ✓ | 89.9 | 96.1 | - | - | 36.9 | 59.6 | - |
| Shift-GCN [5] | CVPR 2020 | ✓ | 90.7 | 96.5 | 85.9 | 87.6 | - | - | 94.6 |
| DC-GCN+ADG [4] | ECCV 2020 | ✓ | 90.8 | 96.6 | 86.5 | 88.1 | - | - | 95.3 |
| DDGCN [14] | ECCV 2020 | ✓ | 91.1 | 97.1 | - | - | 38.1 | 60.8 | - |
| MS-G3D [18] | CVPR 2020 | ✗ | 91.5 | 96.2 | 86.9 | 88.4 | 38.0 | 60.9 | - |
| MST-GCN [3] | AAAI 2021 | ✓ | 91.5 | 96.6 | 87.5 | 88.8 | 38.1 | 60.8 | - |
| CTR-GCN [2] | ICCV 2021 | ✓ | 92.4 | 96.8 | 88.9 | 90.6 | - | - | 96.5 |
| EfficientGCN-B4 [27] | TPAMI 2022 | ✓ | 91.7 | 95.7 | 88.3 | 89.1 | - | - | - |
| STF [12] | AAAI 2022 | ✗ | 92.5 | 96.9 | 88.9 | 89.9 | 39.9 | - | - |
| InfoGCN (4-ensemble) [6] | CVPR 2022 | ✓ | 92.7 | 96.9 | 89.4 | 90.7 | - | - | 96.6 |
| InfoGCN (6-ensemble) [6] | CVPR 2022 | ✓ | 93.0 | 97.1 | 89.8 | 91.2 | - | - | 97.0 |
| HD-GCN (2-ensemble) | | ✗ | 92.4 | 96.6 | 89.1 | 90.6 | 38.9 | 61.7 | 96.6 |
| HD-GCN (4-ensemble) | | ✗ | 93.0 | 97.0 | 89.8 | 91.2 | 40.3 | 63.0 | 96.9 |
| **HD-GCN (6-ensemble)** | | ✗ | 93.4 | 97.2 | 90.1 | 91.6 | 40.9 | 63.5 | 97.2 |

Table 1. **Comparisons of the top-1 accuracy (%) against state-of-the-art methods on the NTU-RGB+D 60, 120, Northwestern-UCLA, and Kinetics-Skeleton datasets.** The orange and yellow cells respectively indicate the highest and second-highest value.

learning rate of 0.1 and a minimum learning rate of 0.0001. For the NTU-RGB+D datasets, we set the batch size to 64 and use the data preprocessing method from [36]. For Kinetics-Skeleton, the batch size is set to 128. In addition, to overcome the absence of belly and hip nodes in the Kinetics-Skeleton, we define the center of both hip joints as CoM hip node, and the center of chest and the hip node as CoM belly node, resulting in a total of 20 nodes. For the Northwestern-UCLA dataset, we set the batch size to 16 and use the data preprocessing method from [5]. All our experiments are conducted on a single RTX 3090 GPU.

### 4.2. Comparison with State-of-the-Arts Methods

Most recent state-of-the-art networks [25, 5, 4, 2] adopt a four-way ensemble method, but we adopt the six-way ensemble method described in Sec. 3.4.

We compare ours with state-of-the-art networks on three datasets: NTU-RGB+D 60 [22], NTU-RGB+D 120 [16], Northwestern-UCLA [30], and Kinetics-Skeleton [11]. Comparisons for each dataset are shown in Tab. 1. The recognition performance of our HD-GCN has exceeded the state-of-the-arts on every dataset without any motion streams, as shown in Tab. 1. With our proposed ensemble method, HD-GCN outperforms the state-of-the-art and shows comparable performance to the 6-way ensemble state-of-the-art using only 4-way ensemble method.

### 4.3. Ablation Study

In this section, we demonstrate the effectiveness of the proposed HD-GCN. Performance is specified as the cross-subject and cross-setup classification accuracy on the NTU-RGB+D 120 [16] joint stream data.

| Graph type | Edges | S-EdgeConv | X-Sub (%) | X-Set (%) |
|---|---|---|---|---|
| Conventional | PC | ✗ | 83.5 | 85.4 |
| HD-Graph | | | | |
| A | PC | ✗ | 84.3 (↑ 0.8) | 86.1 (↑ 0.7) |
| B | PC | ✓ | 84.6 (↑ 1.1) | 86.3 (↑ 0.9) |
| C | FC | ✗ | 84.9 (↑ 1.4) | 86.5 (↑ 1.1) |
| D | FC | ✓ | **85.1 (↑ 1.6)** | **86.7 (↑ 1.3)** |

Table 2. **Comparison of the conventional graph and four types of HD-Graph.**

| Method | H-EdgeConv | X-Sub (%) | X-Set (%) |
|---|---|---|---|
| Baseline | ✗ | 83.5 | 85.4 |
| HD-GCN | | | |
| w/o A-HA | ✗ | 85.1 (↑ 1.6) | 86.7 (↑ 1.3) |
| w/ SAP | ✗ | 85.2 (↑ 1.7) | 86.7 (↑ 1.3) |
| w/ SAP | ✓ | 85.4 (↑ 1.9) | 87.0 (↑ 1.6) |
| w/ RSAP | ✗ | 85.5 (↑ 2.0) | 87.0 (↑ 1.6) |
| w/ RSAP | ✓ | **85.7 (↑ 2.2)** | **87.3 (↑ 1.9)** |

Table 3. **Comparison of various types of attention modules.** SAP denotes the spatial average pooling.

**Hierarchically Decomposed Graph.** To proceed with the ablation study for HD-Graph, we set Yan *et al.* [35]'s graph as the conventional graph. Here, we use the temporal convolution module of [18], as mentioned in Sec. 3.5, to compare the performance of networks fairly with various graphs. The experimental results are shown in 2.

We set the edges of the HD-Graph in different ways to show a gradual performance increase according to the type of graph. There are four main versions of HD-Graph, the first of which is graph A containing only the PC edges. Unlike the conventional graph with one edge set including three fixed subsets, HD-Graph has a flexible number of edge sets, each divided by hierarchy layers with three subsets. Graph B is an extension of A, with the additional operation S-EdgeConv. Graph C contains FC edges for $N_H$ hierarchy node sets, and graph D is similar to C but includes

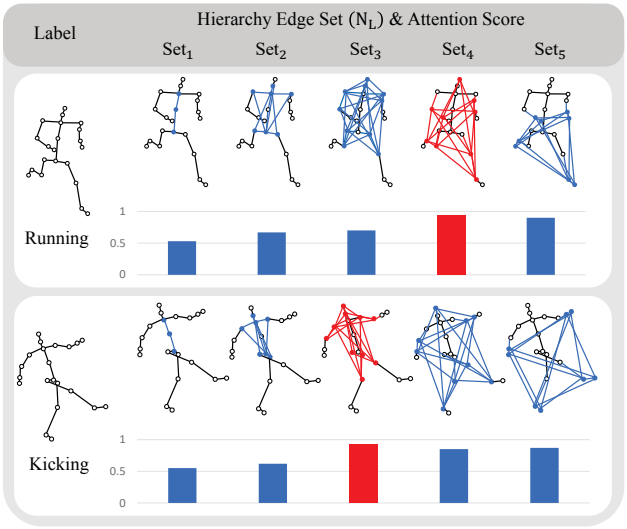


Figure 5. **Hierarchy-wise attention scores by A-HA for "Running" and "Kicking" class.** Each score indicates the value of the attention map $\mathbf{M}$ of the A-HA module.

S-EdgeConv. The HD-Graph with only PC edges performs better than the conventional graph by a large margin, even though they share the same edges. This proves that it is meaningful to divide the joint nodes by hierarchy edge sets. In addition, the HD-Graph with FC edges and S-EdgeConv performs better for every datasets.

**Attention-Guided Hierarchy Aggregation.** To prove the effectiveness of the A-HA module, we use a method to change or remove specific parts of our attention module, with the results shown in Tab. 3. Spatial average pooling (SAP) simply averages along the spatial axis without the representative node extraction process, which performs worse than our RSAP. The poorer performance is due to two factors: (1) scaling bias occurs because the number of nodes in each hierarchy node set is different, and (2) attention through SAP does not represent the corresponding hierarchy node set because it brings the average of the feature vectors of all nodes, not a specific node set. Furthermore, it performs better with H-EdgeConv, which recognizes each hierarchy edge set as a graph node. This proves that because the major edge sets are different for each data sample, it is important to find and highlight edge sets with high similarity based on the Euclidean distance through H-EdgeConv.

The results of the attention score $\mathbf{M}$ of our A-HA module are shown in Fig. 5. These results show that our module scores edge sets 4 and 5 higher for the "running" class, which includes knees and feet, elbows and hands. For the "Kicking" class, A-HA gives the highest score to edge set 3, which includes shoulders and hips, followed by edge set 4 and 5. It is reasonable for human visual recognition that the dynamically moving edge set 4, 5 are more important than the stationary and barely moving edge set 3 when running rather than when kicking something.

| | X-Sub (%) | X-Set (%) | GFLOPs | # Param. (M) |
|---|---|---|---|---|
| DC-GCN [4] | 84.0⋆ | 86.1⋆ | 2.74 | 3.45 |
| MS-G3D [18] | 84.9⋆ | 86.8⋆ | 5.22 | 3.22 |
| CTR-GCN [2] | 84.9 | 86.5⋆ | 1.97 | **1.46** |
| InfoGCN [6] | 85.1 | 86.3 | 1.68 | 1.57 |
| HD-GCN | **85.7** | **87.3** | **1.60** | 1.68 |

Table 4. **Comparison of complexity of the single-stream state-of-the-arts.** ⋆: results obtained by the released codes.

| | X-Sub (%) | X-Set (%) | GFLOPs | # Param. (M) |
|---|---|---|---|---|
| CTR-GCN † [2] | 88.9 | 90.6 | 7.88 | **5.84** |
| InfoGCN † [6] | 89.4 | 90.7 | 6.72 | 6.28 |
| InfoGCN ‡ [6] | 89.8 | 91.2 | 10.08 | 9.42 |
| HD-GCN † | 89.8 | 91.2 | **6.40** | 6.72 |
| HD-GCN ‡ | **90.1** | **91.6** | 9.60 | 10.08 |

Table 5. **Comparison of complexity of the multi-stream state-of-the-arts.** †: 4-ensemble, ‡: 6-ensemble

**Six-Way Ensemble.** We use the ensemble method to which three graphs with different CoM nodes are applied, excluding motion streams. Tab. 1 shows that the HD-GCN with 4-way ensemble outperforms the state-of-the-art [6] 4-way methods with motion data and shows comparable performance with the state-of-the-art 6-way method. In addition, when the 6-way ensemble with three different graphs is applied to HD-GCN, it outperforms the state-of-the-art methods. This proves that the features extracted with different CoM nodes are learned in different learning aspects.

### 4.4. Comparison of Complexity with Other Models

Although our model has multiple branch layers for multiple edge sets, it does not cause high complexity because it precedes channel reduction layers. Comparisons of computational complexity with other models are shown in Tab. 4, where the window size is fixed to 64. Our model shows the best performance on NTU-RGB+D 120 joint stream by a large margin even though the computational complexity of our model is the lowest. For multi-stream ensemble, our 4-stream HD-GCN shows almost similar performance to 6-stream InfoGCN [6] while having 3.68G fewer FLOPs and 2.70M fewer parameters as shown in Tab. 5.

## 5. Conclusions

In this work, we propose a novel hierarchically decomposed graph convolutional network (HD-GCN) for skeleton-based action recognition. We also propose a new framework (HD-Graph) that replaces the existing framework, decomposes all the joint nodes by hierarchy edge sets and considers the connectivity between major distant nodes, which is difficult to identify naturally. We also present an effective attention module (A-HA) composed of representative spatial average pooling (RSAP) layer and hierarchical

edge convolution (H-EdgeConv), which applies hierarchy-wise attention for the HD-Graph. In addition, our HD-GCN learns graph-wise features with different patterns through a six-way ensemble method. We derive an effective feature extractor by combining these three methods and empirically verify its effectiveness. Our approach outperforms current state-of-the-art methods on four benchmark datasets.

**Acknowledgements.** This work was supported by AIon-Flow Research and the KIST Institutional Program (Project No.2E32283-23-064).

## References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 6

[2] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368, 2021. 1, 2, 3, 6, 7, 8

[3] Zhan Chen, Sicheng Li, Bing Yang, Qinghan Li, and Hong Liu. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1113–1122, 2021. 7

[4] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3, 7, 8

[5] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192, 2020. 1, 2, 7

[6] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20186–20196, 2022. 7, 8

[7] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015. 3

[8] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[10] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[11] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 6, 7

[12] Lipeng Ke, Kuan-Chuan Peng, and Siwei Lyu. Towards to-at spatio-temporal focus for skeleton-based action recognition. *arXiv preprint arXiv:2202.02314*, 2022. 7

[13] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697. PMLR, 2018. 3

[14] Matthew Korban and Xin Li. Ddgcn: A dynamic directed graph convolutional network for action recognition. In *European Conference on Computer Vision*, pages 761–776. Springer, 2020. 7

[15] Jungho Lee, Minhyeok Lee, Suhwan Cho, Sungmin Woo, and Sangyoun Lee. Leveraging spatio-temporal dependency for skeleton-based action recognition. *arXiv preprint arXiv:2212.04761*, 2022. 1

[16] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2684–2701, 2019. 2, 6, 7

[17] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European conference on computer vision*, pages 816–833. Springer, 2016.

[18] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020. 1, 2, 6, 7, 8

[19] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[20] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael 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*, pages 5115–5124, 2017. 3

[21] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016. 3

[22] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang 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*, pages 1010–1019, 2016. 2, 6, 7

[23] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference*

*on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019. 7

[24] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019. 1, 2, 3, 4, 6, 7

[25] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29:9532–9545, 2020. 1, 2, 3, 6, 7

[26] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019. 1, 7

[27] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Constructing stronger and faster baselines for skeleton-based action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3, 7

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 6

[29] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 4041–4049, 2015. 1

[30] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014. 2, 6, 7

[31] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 1

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3

[33] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2, 4, 5

[34] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3

[35] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI Conference on Artificial Intelligence*, 2018. 1, 2, 3, 4, 7

[36] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1112–1121, 2020. 1, 7