electronics

MDPI

Article

Multi-Stage Attention-Enhanced Sparse Graph Convolutional Network for Skeleton-Based Action Recognition

Chaoyue Li [1], Lian Zou [1,*], Cien Fan [1], Hao Jiang [1] and Yifeng Liu [2]

[1] School of Electronic Information, Wuhan University, Wuhan 430072, China; 2015301220007@whu.edu.cn (C.L.); fce@whu.edu.cn (C.F.); jh@whu.edu.cn (H.J.)
[2] National Engineering Laboratory for Risk Perception and Prevention (NEL-RPP), Beijing 100041, China; liuyifeng3@cetc.com.cn
* Correspondence: zoulian@whu.edu.cn; Tel.: +86-1397-157-9950

Abstract: Graph convolutional networks (GCNs), which model human actions as a series of spatial-temporal graphs, have recently achieved superior performance in skeleton-based action recognition. However, the existing methods mostly use the physical connections of joints to construct a spatial graph, resulting in limited topological information of the human skeleton. In addition, the action features in the time domain have not been fully explored. To better extract spatial-temporal features, we propose a multi-stage attention-enhanced sparse graph convolutional network (MS-ASGCN) for skeleton-based action recognition. To capture more abundant joint dependencies, we propose a new strategy for constructing skeleton graphs. This simulates bidirectional information flows between neighboring joints and pays greater attention to the information transmission between sparse joints. In addition, a part attention mechanism is proposed to learn the weight of each part and enhance the part-level feature learning. We introduce multiple streams of different stages and merge them in specific layers of the network to further improve the performance of the model. Our model is finally verified on two large-scale datasets, namely NTU-RGB+D and Skeleton-Kinetics. Experiments demonstrate that the proposed MS-ASGCN outperformed the previous state-of-the-art methods on both datasets.

Keywords: graph convolutional networks; skeleton-based action recognition; spatial-temporal graphs; multi-stage streams

Citation: Li, C.; Zou, L.; Fan, C.; Jiang, C.; Liu, Y. Multi-Stage Attention-Enhanced Sparse Graph Convolutional Network for Skeleton-Based Action Recognition. *Electronics* **2021**, *10*, 2198. https://doi.org/10.3390/electronics10182198

Academic Editor: Tomasz Trzcinski

Received: 8 August 2021
Accepted: 5 September 2021
Published: 8 September 2021

1. Introduction

Human action recognition has recently attracted considerable attention in computer vision due to its wide applications, such as autonomous driving [1,2], human-robot interaction [3,4], video retrieval [5,6], and video surveillance [7,8]. Conventional action recognition methods mainly contain two steps: representation and classification. The former transforms the action video into a series of vectors, and the latter obtains a label from the vectors. Recently, deep-learning-based methods have been used to construct an end-to-end trainable framework to unify the above two steps. These methods rely on large amounts of data with multiple modalities like video, RGB image sequence, depth image sequence, and skeleton data, etc. Compared with other data modalities, skeleton data is robust to the dynamic circumstance and filters out excessive invalid information. Thus, the skeleton-based method gradually shows its superiority in action recognition.

The dynamic skeleton data is provided by depth sensors or pose estimation algorithms. Earlier deep-learning-based methods typically represent the skeleton data as a sequence of coordinate vectors [9–13] providing human joint positions or convert it into a pseudo-image [14–16] and then perform feature extraction and classification. Although these methods have some effect, they ignore the structural information of the human skeleton. To capture the dependency between the joints in the human skeleton, some methods model the skeleton data as a series of spatial-temporal graphs.

In these graphs, joints are vertices and bones are edges. After that, the stacked graph convolutional networks (GCN) are applied to extract action features, and the classifier finally outputs the action category. Yan et al. [17] first used GCNs to learn spatial and temporal features simultaneously. Although ST-GCN [17] starts to learn structural features of the human body, it pays less attention to the connections between the joints that are far apart, thus, ignoring the long-distance features. In this way, some actions coordinated by joints that are far apart, such as wearing shoes, clapping, and wearing a hat, cannot be accurately identified. In addition, it only learns the dependencies between joints, without considering the relationship of each body part. Moreover, the feature extraction method in the time domain is relatively simple, resulting in insufficient motion information captured.

To solve the above issues, we propose a multi-stage attention-enhanced sparse graph convolutional network (MS-ASGCN). In our paper, a new partition strategy for neighboring joints is proposed. We construct the spatial graph with four subgraphs. The first subgraph represents the self-loops of joints, the second and third subgraphs describe the physical edges in opposite directions, and the fourth subgraph explains the dependencies between some sparse joints. These four sub-graphs learn the movement mode adaptively during the training process. This new neighborhood partition strategy can learn more topological information of human motion. In addition, we also introduce a part attention module to enhance the feature learning of each human body part. Finally, we explore a new network structure in which streams of different stages are fused in some specific layers of the network. This supplements the information in the time dimension.

To examine the effectiveness of MS-ASGCN, we conduct extensive experiments on the NTU-RGB+D [10] and Skeleton-kinetics [18] datasets. The proposed model performed well on both datasets. We mainly make the following contributions:

1. We propose a new neighborhood partition strategy by constructing four subgraphs with different connections of joints.
2. We introduce a part attention module to learn the activation parameters of each part and perform weighted feature fusion.
3. A new network structure is proposed, which integrates streams of different stages in specific layers of the network.
4. Our final model achieved state-of-the-art performance on two large-scale skeleton-based action recognition datasets.

2. Related Work

2.1. Skeleton-Based Action Recognition

With the rapid development of depth sensors and pose estimation algorithms, skeleton data is more accessible and more accurate. Subsequently, action recognition based on skeleton data has become an emerging research field in recent years. We list the main skeleton-based action recognition methods and their pros and cons in Table 1. Traditional skeleton-based methods [19–21] usually focus on manual features, such as using relative three-dimensional rotation and translation between body joints or parts. However, the portability of these methods is poor; thus, they may perform well on one dataset and terrible on other datasets. Deep learning methods effectively compensate for this defect and replace manual feature extraction in a data-driven form.

The methods based on deep learning are normally modeled into three types: the RNNs, the CNNs, and the GCNs. The RNN-based methods [9–13], which take a sequence of vectors as input, recognize actions by extracting features mainly in the time domain. They make full use of the commonality between the skeleton data and the RNNs. In addition, to learn more temporal context information of action sequences, some variants based on RNN, such as LSTM and GRU, are also applied to action recognition to improve performance. Unlike RNN, the CNN-based methods [14–16,22] can naturally learn high-level semantic features efficiently. However, with image input as the mainstay, these methods may not be suitable for action recognition tasks based on the time-dependent skeleton data.

Therefore, it is still challenging to extract the spatiotemporal features for action recognition. To solve the problem, the GCN-based method first proposed by [17] et al. models the skeleton data as a topological graph whose edges are bones and vertices are joints. This method utilizes the dependencies between human joints, then extracts features sequentially in space and time, and finally stacks them for action recognition.

2.2. Graph Convolutional Neural Network

Unlike CNN, GCN is specially used to process graph data with a non-Euclidean structure and has a wide range of applications. Generally, GCN mainly contains two kinds of methods to learn the topological map, one extracts features on the spatial domain and the other on the spectral domain. The method based on the spatial domain [23–27] focuses on constructing a spatiotemporal graph and performing graph convolution operations to extract features. The method based on spectral-domain [28–32] utilizes the properties of the Laplacian matrix of the graph, such as eigenvalues and eigenvectors, and performs a Fourier transform on the graph data. Afterward, graph convolution networks are stacked to extract features.

Table 1. Pros and cons of skeleton-based action recognition methods.

	Methods	Pros	Cons
Manual	Actionlet Ensemble [20]	Use depth data.	Sensitive to noise.
	HOJ3D [21]	Use histograms of 3D joints.	Portability is poor.
	Lie Group [19]	Model body parts finer.	Limited to small datasets.
RNNs	HBRNN [9]	Model temporal evolution.	Easy to overfit.
	ST-LSTM [11]	Make spatiotemporal analysis.	Low recognition accuracy.
	ARRN-LSTM [13]	Achieve higher performance.	Complex network structure.
CNNs	TCN [14]	Re-designs the TCN.	Small temporal information.
	Synthesized CNN [15]	Enhance visualization.	Not flexible enough.
	3scale ResNet152 [16]	Can use pre-trained CNNs.	A large amount of calculation.
GCNs	ST-GCN [17]	Model actions as graphs.	Ignore long-range links
	AS-GCN [33]	Explore actional-structural links.	Complex network.
	2s-AGCN [34]	Increase model's flexibility.	Simple temporal domain modeling.

3. Background

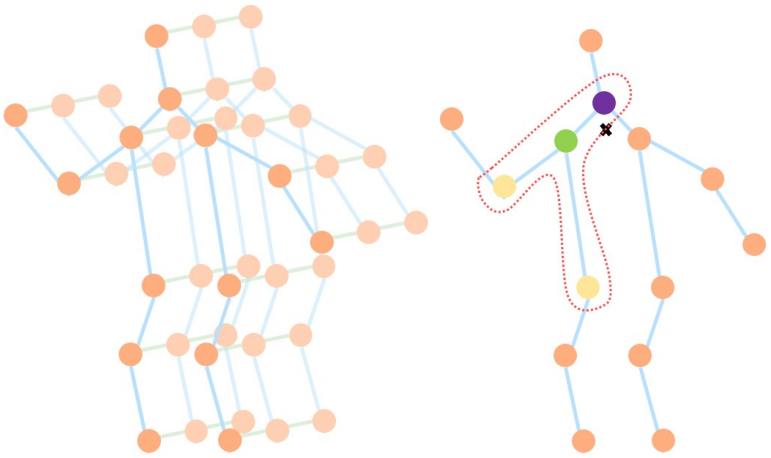
We will provide some necessary background materials in this section for a better presentation of our work.

3.1. Skeleton Graph Construction

The raw skeleton data is usually represented as a vector, providing the coordinates of each human joint in graph convolutional networks. An action sequence can generally be described as a 3D vector $\mathbf{X} \in \mathbb{R}^{C \times T \times V}$, where C is the number of channels, referring to the dimension of the joint coordinates; T denotes the length of the action sequence; and V is the number of joints of a single person in the constructed human skeleton graph. To better describe the spatial features of human actions, we use $\mathbf{X}_t$ to denote a set of joint coordinates in the t-th frame, and $\mathbf{X}_{t_i}$ to denote joint coordinates of the i-th joint in the t-th frame.

Then, we construct the raw skeleton data $\mathbf{X}$ into T spatial graphs. We denote $\mathbf{X}_t$ as a skeleton graph $\mathcal{G} = (V, E)$, where V is a collection of n vertices (joints), and E denotes a collection of m edges (bones). Afterward, an adjacency matrix $\overline{\mathbf{A}}$ is introduced to describe the connection between the joints in the skeleton graph. If $\overline{\mathbf{A}}_{ij} = 1$, the i-th joint and the j-th joint are connected, otherwise, the i-th joint and the j-th joint point are disconnected. In this way, we find the skeleton graph shown in Figure 1a. The orange circles denote the human body joints, the blue lines denote the physical connections between human body joints in the same frame, and the green lines represent the connection of the same joint in adjacent frames. To capture more refined spatial features, we often divide the neighbors of the target joint into several subsets. Some partition strategies are proposed.

Taking the strategy adopted in the ST-GCN method as an example (see Figure 1b), the root node and its neighbors are divided into three subsets: (1) the root node itself, (2) the centrifugal group, and (3) the centripetal group. The centrifugal group denotes the nodes that are farther to the gravity center of the skeleton than the root node (as shown by the yellow circles in Figure 1b). The centripetal group is the opposite (as shown by the purple circle in Figure 1b). Correspondingly, the adjacency matrix is also expanded into three adjacency matrices. The graph construction strategy of our work is introduced in Section 4.1.



(**a**) Spatiotemporal graph (**b**) Partition strategy

Figure 1. (**a**). The spatiotemporal graph modeled by input skeleton sequence used in ST-GCN. (**b**). The partition strategy of the neighboring joints in ST-GCN.

3.2. Graph-Based Convolution

Similar to the CNNs, the GCNs extract the features by stacking multiple graph convolutional layers. The high-level features are then fed into the global average pooling layer, the fully connected layer, and the Softmax classifier to predict the action category. The graph convolution layer mainly consists of spatial graph convolution operation and temporal graph convolution operation, in which the former is essential to action recognition. The spatial graph convolution operation learns the features of the neighboring joints and performs a weighted average on them to obtain the features of the target joint. For the i-th joint in the t-th frame v_{ti}, the spatial graph convolution operation is formulated as [17]:

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(v_{tj}) \cdot w(l_{ti}(v_{tj})). \tag{1}$$

where f_{in} is the input feature map and B denotes a set of neighbors of the target joint v_{ti}. w is a weight function providing a weight vector for each joint. Z is the cardinality of the subset in to normalize the data. l_{ti} is the label of the neighbors of v_{ti}, which depends on the subset partition strategy. In this way, for the entire action sequence, the spatial graph convolution is performed as Equation (2):

$$\mathbf{Y} = \sum_{k=1}^{K} \mathbf{\Lambda}_k^{-\frac{1}{2}} \overline{\mathbf{A}}_k \mathbf{\Lambda}_k^{-\frac{1}{2}} \mathbf{X} \mathbf{W}. \tag{2}$$

where $\mathbf{X}$ is the input feature map and $\mathbf{Y}$ is the output feature map. $\mathbf{W}$ is a learnable weight function. K denotes the number of spatial subsets for each joint according to ST-GCN [17]. $\overline{\mathbf{A}}$ is the adjacency matrix. $\mathbf{\Lambda}_{ii} = \sum_{j=1}^{V} \overline{\mathbf{A}}_{ij} + \alpha$ is the diagonal degree matrix for normalization, where $\overline{\mathbf{A}}$ represents the element in the i-th row and j-th column of $\overline{\mathbf{A}}$. To avoid the all-zero problem, an extra parameter α is added to the formula.

3.3. Attention-Enhanced Adaptive GCNs

MS-AAGCN [35] is an improvement of STGCN [17]. The model of AAGCN is introduced below. First, AAGCN preprocesses the original skeleton data. Second, AAGCN proposes an adaptive graph convolutional layer, which adds a subgraph for feature learning for each sample and learns the features in an adaptive way. Finally, AAGCN introduces an attention mechanism called STC-Attention, which better extracts the features by connecting a spatial attention mechanism, temporal attention mechanism, and channel attention mechanism in series. The work of this paper is carried out on this basis.

4. Method

In this section, we introduce the proposed framework in detail. We first adopt a new method to construct a spatial graph. Then, we describe the proposed part attention module. Finally, we build a multi-stage convergence network framework.

4.1. Construction of Spatial Graph

The previous methods are focused on the physical connection of joints to construct a spatial graph. Although these spatial graphs show the dependencies between joints to a certain extent, the topological information captured from the input action sequence is limited. In contrast, considering the human body structure, we propose a new partition strategy for neighbors of the target joint to learn more motion patterns. It is worth noting that the finer partition strategy produces a more expressive graph topology.

The partition strategy of neighboring joints is shown in Figure 2. We propose four modes of neighboring joints to learn the spatial features. The first subgraph shows the self-loops of each joint. The corresponding adjacency matrix is shown in Equation (3):

$$\overline{\mathbf{A}}_{self(ij)} = \begin{cases} 1, & i = j \\ 0, & otherwise \end{cases} \tag{3}$$

Then, we construct the second subgraph with the edge direction inward. Correspondingly, the edge direction of the third subgraph is outward. The adjacency matrices corresponding to these two subgraphs are represented as Equations (4) and (5):

$$\overline{\mathbf{A}}_{in(ij)} = \begin{cases} 1, & when\ joint\ i\ is\ connected\ to\ joint\ j\ and\ edge_{ij}\ is\ inward \\ 0, & otherwise \end{cases} \tag{4}$$

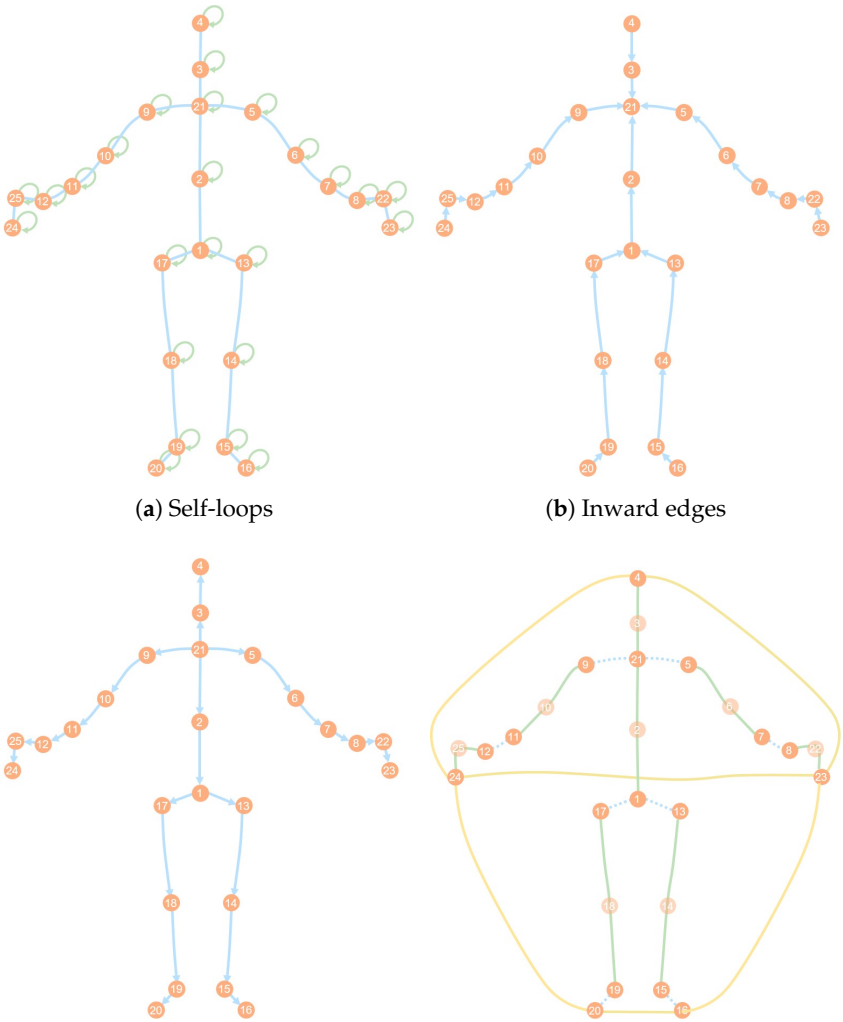
$$\overline{\mathbf{A}}_{out(ij)} = \begin{cases} 1, & when\ joint\ i\ is\ connected\ to\ joint\ j\ and\ edge_{ij}\ is\ outward \\ 0, & otherwise \end{cases} \tag{5}$$

In this way, the information transmission paths between joints are bidirectional. The two subgraphs with opposite edge directions are more conducive to the passage of joint-level information between the central joint and the far joint.

The fourth subgraph adds new dependencies between joints. Considering that 1-hop neighbors ignore remote information, we introduce a connection between sparse joints. We first add some 2-hop neighbors based on the structure of the human body, as shown by the green line in Figure 2d. In this process, we selectively ignore some joints (the light-colored joints in Figure 2d) to capture more sparse joint information. We speculate that this method can speed up the spread of information between joints and suppress the weakening in the information propagation process.

In addition, we divide the human body into five parts (right arm, left arm, torso, right leg, and left leg), as shown in Figure 3. In each part, the most marginal point is selected and connected with the marginal points of other parts (shown by the yellow line in Figure 2d). The corresponding adjacency matrix is denoted as Equation (6):

$$\overline{\mathbf{A}}_{sparse(ij)} = \begin{cases} 1, & when\ sparse\ joint\ i\ and\ joint\ j\ are\ connected\ in\ the\ defined\ subgraph \\ 0, & otherwise \end{cases} \tag{6}$$



(**a**) Self-loops (**b**) Inward edges

(**c**) Outward edges (**d**) 2-hop edges and distant edges

Figure 2. (**a**) The self-loops of each joint. (**b**) Edges whose direction is inward from the center joint. The direction of edges in (**c**) is opposite to (**b**). (**d**) 2-hop edges and the connection of several marginal joints.

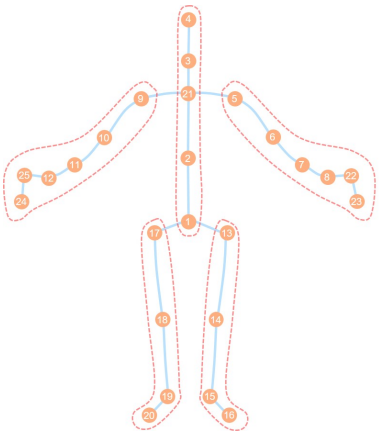


Figure 3. Human body part division strategy.

In this way, the information transmission path of the edge points is more diversified, and the connection between body parts is enhanced. According to the construction of the

above four subgraphs, we can find four normalized adjacency matrices $\mathbf{A}_{self}$, $\mathbf{A}_{in}$, $\mathbf{A}_{out}$, and $\mathbf{A}_{sparse}$, which are represented by Equations (7)–(10):

$$\mathbf{A}_{self} = \mathbf{\Lambda}_{self}^{-\frac{1}{2}} \overline{\mathbf{A}}_{self} \mathbf{\Lambda}_{self}^{-\frac{1}{2}}. \tag{7}$$

$$\mathbf{A}_{in} = \mathbf{\Lambda}_{in}^{-\frac{1}{2}} \overline{\mathbf{A}}_{in} \mathbf{\Lambda}_{in}^{-\frac{1}{2}}. \tag{8}$$

$$\mathbf{A}_{out} = \mathbf{\Lambda}_{out}^{-\frac{1}{2}} \overline{\mathbf{A}}_{out} \mathbf{\Lambda}_{out}^{-\frac{1}{2}}. \tag{9}$$

$$\mathbf{A}_{sparse} = \mathbf{\Lambda}_{sparse}^{-\frac{1}{2}} \overline{\mathbf{A}}_{sparse} \mathbf{\Lambda}_{sparse}^{-\frac{1}{2}}. \tag{10}$$

These adjacency matrices will be used in the spatial graph convolution operation. See Section 4.3 for details.

4.2. Part Attention Module

The attention mechanism has played an essential role in neural networks [36–39]. To improve the recognition accuracy of our model, the STC attention module proposed in MS-AAGCN [35] is added after the spatial graph convolution operation. However, the module only aggregates and activates features in the three dimensions of time, space, and channel, without considering the structure of the human body. Therefore, we propose a part attention mechanism.

First, as shown in Figure 3, the human body is divided into five parts. Then, the spatial-temporal feature maps extracted from each part are input into the part attention module to aggregate the part-wise features. In this way, we find the weight vector, and then we connect the weighted features of each part to obtain the final output result. The specific operation of the m-th part is shown in Figure 4. The feature map of each part is generated by Equations (11) and (12):

$$\mathbf{f}_{mid}(p_m) = \sigma(\delta(GAP(\mathbf{f}_{in}(p_m))\mathbf{W}_1)\mathbf{W}_2). \tag{11}$$

$$\mathbf{f}_{out}(p_m) = \mathbf{f}_{in}(p_m) + \mathbf{f}_{in}(p_m) \otimes \mathbf{f}_{mid}(p_m). \tag{12}$$

where f_{in} is the input feature map and f_{out} is the output feature map. $\otimes$ represents element-level multiplication. GAP denotes global average pooling, including pooling in the time dimension and space dimension. σ is the $Sigmoid$ function and δ denotes the $ReLu$ activation function. $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable weights of the two fully connected layers. $\mathbf{W}_1$ is shared for all parts, while $\mathbf{W}_2$ is specific to each part.

Different from the STC attention module, we only insert the part attention module into some specific layers in the network. In this way, the part features are activated, and the amount of calculation is relatively small with little redundant information. In these layers, the part attention module follows the spatial-temporal GCNs, which can better activate features at the part level.

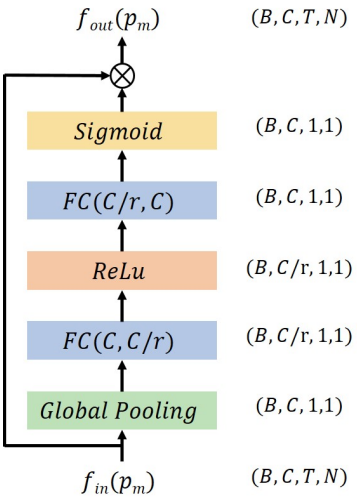


Figure 4. Architecture of the proposed part attention module.

4.3. Graph Convolutional Block

As shown in Figure 5, a basic block consists of a spatial GCN, an STC attention module, a temporal GCN, and a part attention module. Each block uses the residual connection method. In addition, a BN layer and a ReLu layer will follow immediately after both the spatial GCN and the temporal GCN. The temporal GCN is the same as that in STGCN, performing 1D convolution on the input feature map in the temporal dimension. Here, we mainly introduce the spatial GCN.

The spatial GCN extracts the spatial features of the action sequence based on the construction of the spatial graph. As shown in Figure 6, we respectively perform graph convolution operations on the four subgraphs constructed in Section 4.1 and add the outputs of the four branches to obtain the final feature map. In addition, we add a weighted self-attention adjacency matrix B to each branch. The adjacency matrix B is obtained by Equation (13):

$$\mathbf{B} = SoftMax(\mathbf{f}_{in}^{T}\mathbf{W}_{f1}^{T}\mathbf{W}_{f2}\mathbf{f}_{in}). \tag{13}$$

where $\mathbf{f}_{in} \in \mathbb{R}^{C_{in} \times T \times V}$ is the input feature map. f_1 and f_2 are convolution operations with the kernel size of 1×1, and $\mathbf{W}_{f1} \in \mathbb{R}^{C_{in} \times C_{mid} \times 1 \times 1}$ and $\mathbf{W}_{f2} \in \mathbb{R}^{C_{in} \times C_{mid} \times 1 \times 1}$ are the weights for them, respectively. $SoftMax$ denotes the $SoftMax$ function. G is a learnable weight for matrix B. Then, the spatial GCN in Equation (2) is converted to that in Equation (14):

$$\mathbf{Y} = (\mathbf{A}_{self} + \mathbf{B})\mathbf{X}\mathbf{W}_{self} + (\mathbf{A}_{in} + \mathbf{B})\mathbf{X}\mathbf{W}_{in} + (\mathbf{A}_{out} + \mathbf{B})\mathbf{X}\mathbf{W}_{out} + (\mathbf{A}_{sparse} + \mathbf{B})\mathbf{X}\mathbf{W}_{sparse}. \tag{14}$$

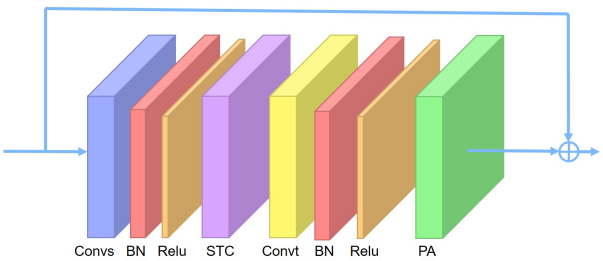


Figure 5. A basic block consists of Convs, STC, Convt, PA, and other operations: batch normalization (BN), ReLU, and the residual block. Convs represents the spatial GCN, and Convt represents the temporal GCN, both of which are followed by a BN layer and a ReLU layer. STC denotes the STC-attention module proposed in MS-AAGCN. PA is the part attention module.

In the training process, we adopt an adaptive learning strategy. In the early stage of training, the adjacency matrices corresponding to the four subgraphs are fixed. When the

training state is stable, the four adjacency matrices are learnable weights. In this way, we can learn the dependencies between more joints, not just limited to the predefined graph.

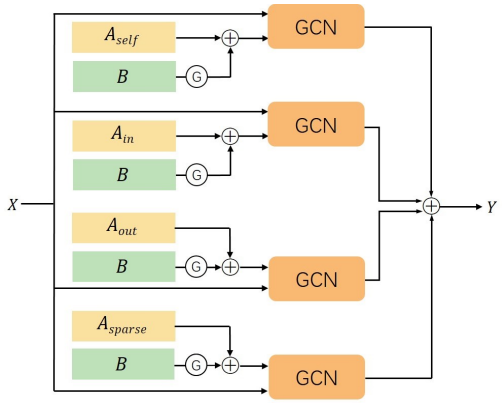


Figure 6. The spatial GCN in a basic block.

4.4. Network Architecture

The multi-stage attention-enhanced sparse graph convolutional network is shown in Figure 7. First, we divide the input skeleton sequence into T temporal stages with equal time interval. In each stage, there are 10 basic blocks (B1–B10), whose output channels are 64, 64, 64, 64, 128, 128, 128, 256, 256, and 256. A BN layer is added to the front to normalize the input data. A GAP layer and an FC layer are added to the end to adjust feature maps of different samples to the same size. The final output is sent to the Softmax classifier to obtain the action category.

It is worth noting that we send the features of the previous stage to the next stage and carry out the fusion of the features. Specifically, the primary features of the first stage are sent to the second stage, and the intermediate features of the second stage are sent to the third stage to achieve the initial integration of features. In addition, the output of the previous stage is sent to the next stage and is connected with the output of the eighth basic block to realize early action prediction. In the end, we find T outputs and perform weighted fusion on them to obtain the final action category. Each branch corresponds to a loss function, and the final loss function is the weighted result of the loss function of each branch.

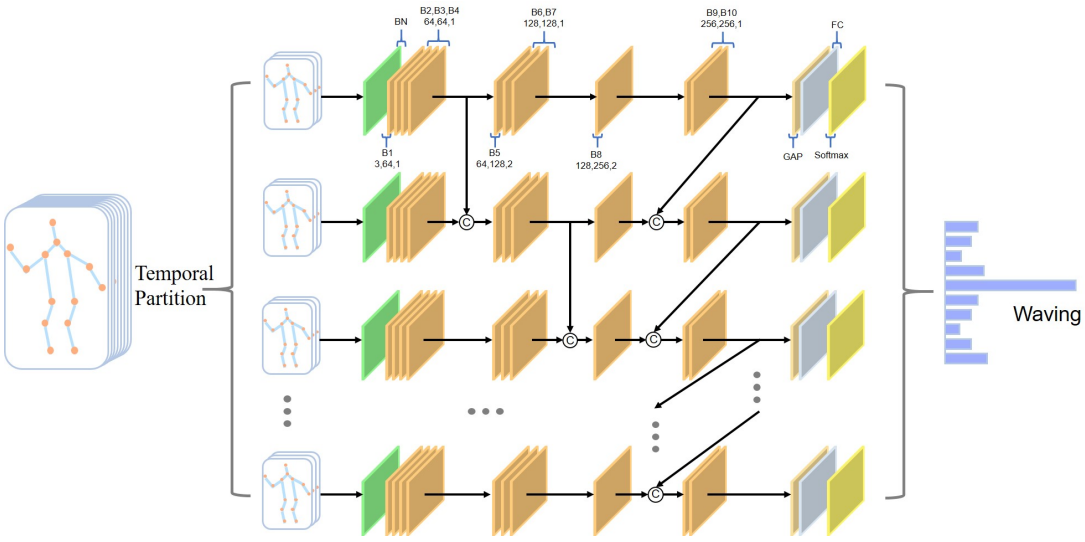


Figure 7. Network architecture of MS-ASGCN.

In addition, we add bone data, as in [35], whose network structure is the same as the joint data for feature extraction and action classification. Each bone is composed of two adjacent joints. The one near the center joint is the source joint, and the one far away from the center is the target joint. The direction of the bone is from the original joint to the target joint. Assuming that the coordinate of the original joint is (x_1, y_1, z_1) and the coordinate of the target joint is (x_2, y_2, z_2), the bone coordinate can be expressed as $(x_2 - x_1, y_2 - y_1, z_2 - z_1)$. In this way, every joint except the central joint is a target joint. To make the number of bones equal to the joints, a central bone with the value of 0 is defined.

5. Experiments

To test the accuracy of the proposed MS-ASGCN, we conducted experiments on two datasets, namely NTU-RGB+D [10] and Skeleton-Kinetics [18]. Extensive ablation studies were also performed on the NTU-RGB+D dataset to show the impact on action recognition of different components in our model. Finally, we compared the experimental results of our model with the state-of-the-art methods on the NTU-RGB+D and Skeleton-Kinetics datasets.

5.1. Datasets

NTU-RGB+D: NTU-RGB+D [10] is one of the largest datasets used in skeleton-based action recognition tasks. It contains 56,880 skeletal action sequences of 60 action classes, played by 40 volunteers aged from 10 to 35 years old. Each clip, performed by one to two performers, is captured by three cameras from different views in a laboratory environment. The position of each subject is fixed by the 3D space coordinates of 25 human joints.

To evaluate the model, we observed two standard evaluation protocols: cross-subject (X-Sub) and cross-view (X-View). In the cross-subject (X-Sub) setting, we divided 40,320 clips from 20 subjects into the training set and the rest into the testing set. In the cross-view (X-View) setting, we put 37,920 clips captured by camera 2 and camera 3 into the training set, and 18,960 clips captured by camera 1 were for testing. The top-1 accuracy is reported in the two settings on the NTU-RGB+D dataset.

Skeleton-Kinetics: Skeleton-Kinetics [18] is a larger dataset for skeleton-based human action recognition, containing 300,000 video clips in 400 categories. These clips are from YouTube videos, and each video clip lasts 10 s. To obtain the skeleton data from the original videos, we used the publicly available OpenPose toolbox [35] to estimate the position of 18 joints of the human body on each frame. Each joint is composed of a 2D coordinate (x, y) and a confidence parameter z, which is finally expressed as a 3D vector (x, y, z). In each frame, two subjects with the highest average joint confidence were selected. The dataset can be divided into a training set containing 240,000 clips and a validation set with 20,000 clips. Our model was trained on the training set, and the top-1 and top-5 accuracies are reported on the validation set.

5.2. Training Details

Here, we elaborate on the training details of the model. We performed MS-ASGCN on the PyTorch deep learning framework [40]. We adopted the same data processing strategy in 2s-AGCN [34]. The batch size was 16. The optimization strategy is Stochastic Gradient Descent (SGD) with Nesterov momentum (0.9). The loss function of the back-propagation gradient was the cross-entropy function. The weight decay was set to 0.0001, and the initial learning rate was 0.1. On the NTU-RGB+D dataset, the size of the input skeleton sequence was adjusted to a fixed length of 300. The training process contained a total of 60 epochs, where the learning rate was divided by 10 at the 30th and 45th epoch. On the Skeleton-Kinetics dataset, each input sample was set to 150 frames. The training process contained a total of 65 epochs, where the learning rate was divided by 10 at the 45th and 55th epoch.

5.3. Ablation Studies

We verified the effectiveness of different components in multi-stage attention-enhanced sparse graph convolutional network (MS-ASGCN) on the NTU-RGB+D dataset in this section. We first tested the effectiveness of the neighborhood partition strategy on the baseline. Then, the performance of the part attention module was tested on the baseline. After that, the best result of the combination of the two was used as a new baseline to test the performance of the multi-stage streams network. Finally, the bone information was added to obtain the final model.

5.3.1. Partition Strategy

Yan et al. [17] divided the spatial neighbors into three subsets and learned the features of each subset individually. We proposed a new partition strategy and achieved better performance. As described in Section 4.1, we proposed a subset of sparsely connected joints. In this new subset, two kinds of joint dependencies are introduced: the connection of 2-hop neighbors and the connection of edge joints. Several related experiments were conducted to explore the two kinds of joint dependencies. As shown in Table 2, these two joint dependencies both improved the accuracy of action recognition, and the system performed better with a subset with both dependencies.

Table 2. Performance comparison on the NTU-RGB+D dataset of the partition strategy with connections of 2-hop neighbors, edge joints and the fusion of two modalities. T: 2-hop neighbors, and E: Edge joints.

Methods	X-Sub (%)	X-View (%)
baseline (AAGCN) [35]	88.0	95.1
baseline (we train) [35]	87.75	94.83
SGCN-T	88.55	95.11
SGCN-E	88.47	95.26
SGCN-T+E	88.68	95.39

5.3.2. Part Attention Module

As described in Section 4.2, we proposed a part attention module to activate the features of each body part. Unlike the activation method of joint-level features, the part attention module is added to some specific layers in the network. Considering that different layers of the network extract different levels of features, we attempted the following configuration schemes. As shown in Table 3, the position configuration hardly affected the performance, and when it was applied to the third and eighth layers, the accuracy of action recognition was the highest.

Table 3. Comparison of action recognition accuracy on the NTU-RGB+D dataset when the part attention module is added to different layers of the network.

Method	X-Sub (%)	X-View (%)
baseline (AAGCN) [35]	88.0	95.1
baseline (we train) [35]	87.75	94.83
AGCN (2&5&8-th)	88.25	95.13
AGCN (3&6&9-th)	88.31	95.17
AGCN (4&7&10-th)	88.19	95.10
AGCN (3&8-th)	88.44	95.31
AGCN (4&9-th)	88.29	95.25
AGCN (5&10-th)	88.26	95.12

5.3.3. Multi-Stage Streams Network

In this subsection, we confirm the effectiveness of the multi-stage network structure. The original data flow of the network is the joint coordinate vector of the action sequence. As described in Section 4.2, we divide the input skeleton sequence into T temporal stages with equal time intervals and then perform feature fusion for each stage. Afterward, the output of each stage is sent to a softmax classifier respectively. Finally, a weighted fusion is performed to obtain the action category.

As a result, multi-stage feature extraction and fusion can improve the recognition accuracy. In addition, the number of stages and the parameters of information fusion also have an impact on the final result, as shown in Tables 4 and 5. Considering the two aspects of performance and computational cost, the number of time stages T was set to 5 in our model. When fusion weights satisfy $w_1 = 0.1, w_2 = 0.1, w_3 = 0.1, w_4 = 0.2$, and $w_5 = 0.5$, the performance is the best.

Table 4. Action recognition accuracy on the NTU-RGB+D dataset of MS-ASGCN with various numbers of stages.

Method	X-Sub (%)	X-View (%)
new baseline (ASGCN)	88.93	95.50
MS-ASGCN (T = 3)	89.47	95.83
MS-ASGCN (T = 4)	89.69	95.87
MS-ASGCN (T = 5)	89.80	95.94
MS-ASGCN (T = 6)	89.87	95.97
MS-ASGCN (T = 7)	89.58	95.82

Table 5. Action recognition accuracy on the NTU-RGB+D dataset of MS-ASGCN with distinct fusion weights.

w_1	w_2	w_3	w_4	w_5	X-Sub (%)	X-View (%)
0	0	0	0	1.0	89.29	95.65
0.1	0.1	0.1	0.1	0.6	89.73	95.79
0.1	0.1	0.1	0.2	0.5	89.80	95.94
0.1	0.1	0.15	0.2	0.45	89.68	95.87
0.1	0.15	0.2	0.25	0.3	89.51	95.80
0.2	0.2	0.2	0.2	0.2	89.33	95.72

5.3.4. Bone Information

We introduce the dual-stream network of joint flow and bone flow in Section 4.4 to improve the recognition accuracy of our method. We used different types of data as input to test the performance, as shown in MS-ASGCN (Js) and MS-ASGCN (Bs) in Table 6. Then, we combine the output of these two types of data to test the performance as shown in MS-ASGCN in Table 6. Clearly, the dual-stream method performed better than the single-stream method.

Table 6. Experimental results on the NTU-RGB+D dataset with different input modalities. Js: Joint stream and Bs: Bone stream.

Methods	X-Sub (%)	X-View (%)
MS-ASGCN (Js)	89.80	95.94
MS-ASGCN (Bs)	89.92	95.61
MS-ASGCN	90.87	96.53

5.4. Comparison with the State-of-the-Arts

To test the accuracy of MS-ASGCN, the final model is compared with the state-of-the-art methods in the performance of skeleton-based action recognition on both the NTU-RGB+D dataset and Kinetics-Skeleton dataset. These methods are mainly divided into four categories, including methods based on handcraft features, methods based on RNNs, methods based on CNNs, and methods based on GCNs. The action recognition accuracies are reported in Tables 7 and 8. Our method performed better on the NTU-RGB+D dataset, which is captured in a constrained environment. We trained our model on two recommended benchmarks: X-Sub and X-View. The top-1 classification accuracy is reported in the test phase. As shown in Table 7, our model performed well, i.e., 90.9% and 96.5% in the X-Sub and X-View settings of NTU-RGB+D, respectively.

Table 7. Comparison of action recognition accuracy with state-of-the-art methods on the NTU-RGB+D dataset. Top-1 accuracy is reported in both settings.

Methods	X-Sub (%)	X-View (%)
Lie Group [19]	50.1	82.8
HBRNN [9]	59.1	64.0
ST-LSTM [11]	69.2	77.7
VA-LSTM [12]	79.2	87.7
ARRN-LSTM [13]	80.7	88.8
TCN [14]	74.3	83.1
Synthesized CNN [15]	80.0	87.2
3scale ResNet152 [16]	85.0	92.3
ST-GCN [17]	81.5	88.3
DPRL [41]	83.5	89.8
HCN [42]	86.5	91.1
STGR-GCN [43]	86.9	92.3
AS-GCN [33]	86.8	94.2
2s-AGCN [34]	88.5	95.1
MS-AAGCN [35]	90.0	96.2
Shift-GCN [44]	90.7	96.5
MS-ASGCN (Ours)	90.9	96.5

Table 8. Comparison of action recognition accuracy with state-of-the-art methods on the Kinetics-Skeleton dataset. Both top-1 and top-5 accuracies are reported.

Methods	Top-1 Accuracy (%)	Top-5 Accuracy (%)
Feature Enc. [45]	14.9	25.8
Deep LSTM [10]	16.4	35.3
TCN [14]	20.3	40.0
ST-GCN [17]	30.7	52.8
STGR-GCN [43]	33.6	56.1
AS-GCN [33]	34.8	56.5
2s-AGCN [34]	36.1	58.7
MS-AAGCN [35]	37.8	61.0
MS-ASGCN (Ours)	39.0	61.8

The Kinetics-Skeleton dataset was larger than the NTU-RGB+D dataset, and the video clips from YouTube were captured in an unconstrained environment. Thus, the recognition accuracy of these methods was generally lower than that on the NTU-RGB+D dataset. As shown in Table 8, our model achieved 39.0% top-1 accuracy and 61.8% top-5 accuracy, thus, achieving the best performance. In general, the performance of methods based on deep learning was generally better than that of methods based on handcraft features. Methods

based on GCNs generally perform better than methods based on RNNs and methods based on CNNs. On both datasets, our method outperformed these methods, showing the superiority and generality of MS-ASGCN.

6. Conclusions

In this paper, we proposed a multi-stage attention-enhanced sparse graph convolutional network (MS-ASGCN) to recognize human actions. We constructed a new skeleton graph to explore the dependencies of sparse joints. In addition, we introduced a part attention module to reinforce the feature learning of each body part. Furthermore, we explored the motion information of the input action sequence and merged input data streams of different stages in the middle layer of the network. The performance of MS-ASGCN was verified on two datasets: NTU-RGB+D and Kinetics-Skeleton. Future works can focus on how to integrate multi-stage streams more efficiently and even explore other modal data fusion methods.

Author Contributions: Funding acquisition, L.Z. and H.J.; investigation, C.L.; methodology, C.L.; project administration, L.Z. and C.F.; validation, C.L.; writing—original draft, C.L.; writing—review and editing, L.Z., C.F. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China Enterprise Innovation and Development Joint Fund under Project No. U19B2004.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rasouli, A.; Yau, T.; Lakner, P.; Malekmohammadi, S.; Rohani, M.; Luo, J. PePScenes: A Novel Dataset and Baseline for Pedestrian Action Prediction in 3D. *arXiv* **2020**, arXiv:2012.07773.
2. Kong, Y.; Fu, Y. Max-Margin Action Prediction Machine. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1844–1858. [CrossRef] [PubMed]
3. Song, Z.; Yin, Z.; Yuan, Z.; Zhang, C.; Zhang, S. Attention-Oriented Action Recognition for Real-Time Human-Robot Interaction. *arXiv* **2020**, arXiv:2007.01065.
4. Koppula, H.S.; Saxena, A. Anticipating Human Activities Using Object Affordances for Reactive Robotic Response. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 14–29. [CrossRef] [PubMed]
5. Zhao, Z.; Chen, G.; Chen, C.; Li, X.; Su, F. Instance-Based Video Search via Multi-Task Retrieval and Re-Ranking. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshop, Seoul, Korea, 27 October–2 November 2019.
6. Ciptadi, A.; Goodwin, M.S.; Rehg, J.M. Movement Pattern Histogram for Action Recognition and Retrieval. In *The European Conference on Computer Vision (ECCV)*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 695–710.
7. Singh, S.; Velastin, S.A.; Ragheb, H. MuHAVi: A Multicamera Human Action Video Dataset for the Evaluation of Action Recognition Methods. In Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Boston, MA, USA, 29 August–1 September 2010; pp. 48–55.
8. Gul, M.A.; Yousaf, M.H.; Nawaz, S.; Ur Rehman, Z.; Kim, H. Patient Monitoring by Abnormal Human Activity Recognition Based on CNN Architecture. *Electronics* **2020**, *9*, 1993. [CrossRef]
9. Du, Y.; Wang, W.; Wang, L. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
10. Shahroudy, A.; Liu, J.; Ng, T.T.; Wang, G. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
11. Liu, J.; Shahroudy, A.; Xu, D.; Wang, G. Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition. In *The European Conference on Computer Vision (ECCV)*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 816–833.
12. Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J.; Zheng, N. View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition From Skeleton Data. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
13. Zheng, W.; Li, L.; Zhang, Z.; Huang, Y.; Wang, L. Relational Network for Skeleton-Based Action Recognition. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 826–831.
14. Kim, T.S.; Reiter, A. Interpretable 3D Human Action Analysis with Temporal Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, HI, USA, 21–26 July 2017.

15. Liu, M.; Liu, H.; Chen, C. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognit.* **2017**, *68*, 346–362. [CrossRef]
16. Li, B.; Dai, Y.; Cheng, X.; Chen, H.; Lin, Y.; He, M. Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME) Workshops, Hong Kong, China, 10–14 July 2017; pp. 601–604.
17. Yan, S.; Xiong, Y.; Lin, D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *arXiv* **2018**, arXiv:1801.07455.
18. Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. The kinetics human action video dataset. *arXiv* **2017**, arXiv:1705.06950.
19. Vemulapalli, R.; Arrate, F.; Chellappa, R. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 588–595.
20. Wang, J.; Liu, Z.; Wu, Y.; Yuan, J. Mining actionlet ensemble for action recognition with depth cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 1290–1297.
21. Xia, L.; Chen, C.C.; Aggarwal, J.K. View invariant human action recognition using histograms of 3D joints. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Providence, RI, USA, 16–21 June 2012; pp. 20–27.
22. Anvarov, F.; Kim, D.H.; Song, B.C. Action Recognition Using Deep 3D CNNs with Sequential Feature Aggregation and Attention. *Electronics* **2020**, *9*, 147. [CrossRef]
23. Duvenaud, D.K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R.P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in Neural Information Processing Systems 28*; Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2015; pp. 2224–2232.
24. Niepert, M.; Ahmed, M.; Kutzkov, K. Learning Convolutional Neural Networks for Graphs. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016.
25. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. *arXiv* **2017**, arXiv:1706.02216.
26. Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; Bronstein, M.M. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
27. Kipf, T.; Fetaya, E.; Wang, K.C.; Welling, M.; Zemel, R. Neural Relational Inference for Interacting Systems. *arXiv* **2018**, arXiv:1802.04687.
28. Shuman, D.I.; Narang, S.K.; Frossard, P.; Ortega, A.; Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.* **2013**, *30*, 83–98. [CrossRef]
29. Bruna, J.; Zaremba, W.; Szlam, A.; Lecun, Y. Spectral Networks and Locally Connected Networks on Graphs. *arXiv* **2013**, arXiv:1312.6203.
30. Henaff, M.; Bruna, J.; LeCun, Y. Deep Convolutional Networks on Graph-Structured Data. *arXiv* **2015**, arXiv:1506.05163.
31. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
32. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems 29*; Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2016; pp. 3844–3852.
33. Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; Tian, Q. Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
34. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
35. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks. *arXiv* **2019**, arXiv:1912.06971.
36. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; Bach, F., Blei, D., Eds.; PMLR: Lille, France, 2015; pp. 2048–2057.
37. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual Attention Network for Image Classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
39. Jie, H.; Li, S.; Albanie, S.; Gang, S.; Vedaldi, A. Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks. *arXiv* **2018**, arXiv:1810.12348

40. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; Devito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in PyTorch. In Proceedings of the Conference and Workshop on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 3–9 December 2017.
41. Tang, Y.; Tian, Y.; Lu, J.; Li, P.; Zhou, J. Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
42. Li, C.; Zhong, Q.; Xie, D.; Pu, S. Co-Occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18, Stockholm, Sweden, 13–19 July 2018.
43. Li, B.; Li, X.; Zhang, Z.; Wu, F. Spatio-Temporal Graph Routing for Skeleton-Based Action Recognition. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8561–8568. [CrossRef]
44. Cheng, K.; Zhang, Y.; He, X.; Chen, W.; Cheng, J.; Lu, H. Skeleton-Based Action Recognition with Shift Graph Convolutional Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
45. Fernando, B.; Gavves, E.; Oramas, J.M.; Ghodrati, A.; Tuytelaars, T. Modeling Video Evolution for Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.