symmetry

MDPI

Article

Action Recognition Based on GCN with Adjacency Matrix Generation Module and Time Domain Attention Mechanism

Rong Yang, Junyu Niu, Ying Xu *, Yun Wang * and Li Qiu

College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518000, China; ryang@szu.edu.cn (R.Y.); 1910294012@email.szu.edu.cn (J.N.); qiuli@szu.edu.cn (L.Q.)
* Correspondence: yxu@szu.edu.cn (Y.X.); wangyun@szu.edu.cn (Y.W.)

Abstract: Different from other computer vision tasks, action recognition needs to process larger-scale video data. How to extract and analyze the effective parts from a huge amount of video information is the main difficulty of action recognition technology. In recent years, due to the outstanding performance of Graph Convolutional Networks (GCN) in many fields, a new solution to the action recognition algorithm has emerged. However, in current GCN models, the constant physical adjacency matrix makes it difficult to mine synergistic relationships between key points that are not directly connected in physical space. Additionally, a simple time connection of skeleton data from different frames makes each frame in the video contribute equally to the recognition results, which increases the difficulty of distinguishing action stages. In this paper, the information extraction ability of the model has been optimized in the space domain and time domain, respectively. In the space domain, an Adjacency Matrix Generation (AMG) module, which can pre-analyze node sets and generate an adaptive adjacency matrix, has been proposed. The adaptive adjacency matrix can help the graph convolution model to extract the synergistic information between the key points that are crucial for recognition. In the time domain, the Time Domain Attention (TDA) mechanism has been designed to calculate the time-domain weight vector through double pooling channels and complete the weights of key point sequences. Furthermore, performance of the improved TDA-AMG-GCN modules has been verified on the NTU-RGB+D dataset. Its detection accuracy at the CS and CV divisions reached 84.5% and 89.8%, respectively, with an average level higher than other commonly used detection methods at present.

Keywords: action recognition; pose estimation; graph convolutional network; attention mechanism

Citation: Yang, R.; Niu, J.; Xu, Y.; Wang, Y.; Qiu, L. Action Recognition Based on GCN with Adjacency Matrix Generation Module and Time Domain Attention Mechanism. *Symmetry* **2023**, *15*, 1954. https://doi.org/10.3390/sym15101954

Academic Editors: João Ruivo Paulo, Cristina P. Santos and Gabriel Pires

Received: 15 September 2023
Revised: 13 October 2023
Accepted: 16 October 2023
Published: 23 October 2023

1. Introduction

Human action recognition has always been one of the most valuable topics in computer vision. There are many applications for human action recognition, such as security monitoring, video retrieval and somatosensory games. Currently, there are two main solutions for action recognition tasks: image-based methods and skeleton-based methods.

Unlike image data, skeleton data has the advantage of high information density. In recent years, with the continuous improvement of pose estimation algorithms, many skeleton-based action recognition algorithms have emerged. The most revolutionary achievement was the introduction of Graph Convolutional Networks (GCN) into skeleton-based action recognition algorithms [1].

Before the graph convolution method was proposed, most skeleton-based action recognition algorithms used Long-Short Memory Networks (LSTM) [2]. However, these methods have some obvious drawbacks. When converting joint features into vector sequences to fit the LSTM networks, there is inevitable information loss. Essentially, such methods adapt to the model by modifying the form of input data, but in the GCN model, skeleton data can be directly read without any conversion. This feature gives the GCN model an edge on action recognition tasks.

However, there are still many possibilities for improvement in the action recognition algorithms based on GCN. Firstly, in conventional GCN models, the adjacency matrices representing the connection status of key points is generated based on the human body structure in physical space. The adjacency matrix is input as a constant at the beginning of the task and does not vary during the whole task. A constant physical adjacency matrix is not conducive to GCN mining synergistic relationships between key points that are not directly connected in the physical space. However, understanding these synergistic relationships is of great significance for some recognition tasks. Secondly, in conventional GCN models, skeleton data from different frames are directly connected through time-domain edges. This simple time connection makes each frame in the video contribute equally to the recognition results, which increases the difficulty of distinguishing action stages.

To address these issues, an Adjacency Matrix Generation (AMG) module and Time Domain Attention (TDA) mechanism are proposed in this paper. The AMG module pre-analyzes the key points to generate an adaptive adjacency matrix that is more conducive to mining synergistic information, which helps improve the model's performance through spatial domain analysis. On the other hand, the TDA mechanism weighs time frame data through dual channel computation and dimension expansion. This structure enables the model to filter out high-weight frames and invest more computing resources in them. Therefore, the TDA mechanism optimizes the discrimination ability through the temporal domain. In the experiments, our proposed method was validated on the large-scale skeleton action benchmark NTU RGB+D [3]. The main contribution of our work is summarized as follows:

1. An AMG mechanism has been designed to generate an adaptive adjacency matrix for the current recognition task, helping graph convolution models to selectively mine synergistic information between key points that are not adjacent in physical space.
2. A TDA mechanism has been proposed to enhance temporal processing capabilities by adding the ability to distinguish frames that are crucial to the final result in the model.
3. The above-mentioned components are combined into a new model called AMG-TDA-GCN. In the experiment, the performance of this new model was verified to have considerable competitiveness on the state-of-the-art behavior recognition methods on the NTU-RGB+D dataset.

The remaining parts of the paper are organized as follows. We present related works of action recognition based on deep learning in Section 2. In Section 3, the AMG module and TDA mechanism are, respectively, introduced in detail. The structure and operating mechanism of the AMG-TDA-GCN model is also presented in this section. Experimental results, comparison and analysis are given in Section 4. Finally, Section 5 concludes this paper.

2. Related Works

Before deep learning was widely applied in computer vision, most solutions to action recognition used template-based methods and manually designed features, such as PCA-HOG [4] and N-gram HOG [5]. The drawback of these methods is the need to manually design behavior models for each action.

After the introduction of deep learning methods, there are two main research directions regarding human action recognition: image-based methods and skeleton-based methods. Among many image-based methods, the most representative ones are 3D graph convolution [6] and two-stream convolution networks [7]. By extending 2D convolution kernels to 3D, 3D graph convolution can directly process video data. The two-stream convolution network uses two channels to process optical flow fields and image feature maps. The optical flow fields and image feature maps provide temporal and spatial information, respectively. Image-based methods require pixel-level computation, which makes them quite complex. In addition, such methods use the entire image as an input to the

system, which can cause interference from massive background information. Therefore, image-based methods are more difficult to analyze behavior patterns from skeleton data.

The skeleton-based method is based on pose estimation algorithms. The pose estimation algorithm enables skeleton-based methods to focus only on action recognition rather than image processing. Some methods use LSTM networks to train classifiers that directly process skeleton sequences [8], but the most novel methods are based on the GCN model, which has the advantage of reconstructing skeleton sequences into a set of graph structured data and analyzing the features as a whole. The first introduction of graph convolution in action recognition was proposed by ST-GCN [1]. The ST-GCN model converts skeleton sequences into spatiotemporal graphs, and customizes graph convolutional neural networks for action recognition.

ST-GCN has achieved astonishing performance and pioneered new methods for solving behavior recognition tasks. Inspired by ST-GCN, many excellent methods have emerged subsequently. Cheng proposed Shift-GCN [9], which consists of shift graph operations and lightweight graph convolution units. Shift-GCN has more flexible receptive fields on both spatial and temporal graphs. Cheng designed a drop graph module and migrated the decoupling aggregation mechanism from the CNN model [10]. The drop graph module solves the common overfitting problem in GCN models. The decoupling aggregation mechanism boosts the modeling ability of graphs with lower hardware costs. PB-GCN divides the skeleton graph into four subgraphs [11], which share the convolution process of the graph. Experiments have shown that the recognition performance of these subgraphs is superior to that of the entire graph. Wang et al. proposes Temporal-Channel Aggregation Graph Convolutional Networks (TCA-GCN) [12] to learn spatial and temporal topologies dynamically and efficiently, and aggregate topological features in different temporal and channel dimensions for skeleton-based action recognition. Xing proposed an Improved Spatial Temporal Graph Convolutional Network (IST-GCN) model in [13], where three modules are designed to aim at the situation when critical joints or frames of the skeleton sequence are occluded or disrupted. To capture the relativity of the various joints among the frames, Zhang et al. presented a new sequence segmentation attention network (SSAN) [14], where the successive frames are encoded in each of the segments that make up the skeleton sequence and a self-attention block is provided to record the associated information among various joints in successive frames. Chen et al. [15] proposed a more comprehensive two-stream GCN architecture containing the vertex-domain graph convolution and the spectral graph convolution based on Graph Fourier Transform (GFT). This structure has been proved to have the ability to reduce the action misalignment for certain actions.

Although these GCN-based improvement methods have achieved satisfactory performance, there are still some unresolved issues in action recognition and there is room for further improvement. For example, the current method is to generate an adjacency matrix containing the connection states of the human body's physical structure. Therefore, in each recognition process, the connection relationship between key points is constant, but in some specific tasks, such as recognizing the action "clapping", the connection between the left and right hands is equally crucial as the connection between key points. A constant adjacency matrix cannot help the model handle the synergistic relationships between disconnected key points in physical space. In addition, the analysis of time information also needs improvement. The existing method models lack the ability to extract high-quality information from multiple frames of data. Therefore, the model proposed in this paper will improve the overall model's ability from both temporal and spatial perspectives.

3. Methods

Our proposed AMG-TDA-GCN enhances the model's processing capabilities in both spatial and temporal domains, respectively. In the spatial domain, the AMG module pre-analyzes the node set at the beginning of the task and generates an adaptive adjacency matrix that is more efficient than the physical adjacency matrix. In the temporal domain,

the TDA mechanism identifies and weighs high-quality frames in video sequences that are crucial for recognition.

3.1. AMG Module

The AMG module helps models analyze spatial domain information. Its structure is shown in Figure 1, which consists of three convolutional (Conv) layers. Each convolutional layer is followed by a Softmax layer and an Offset Leaky ReLU (OLR) layer as the activation function. To avoid overfitting issues caused by modules, a residual connection is included in the outermost layer of the network.

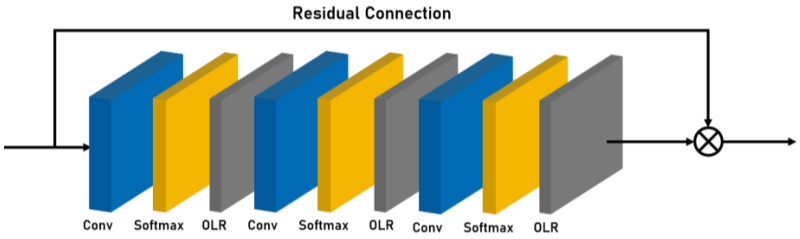


Figure 1. The structure of AMG module.

The reason for choosing three convolutional layers as the AMG module backbone is because that each spatial temporal graph contains thousands of nodes. If the main structure of the network uses multi-layer perceptron, modeling each layer requires thousands of neurons. As the depth of the network increases, the waste of computing resources will be staggering.

The reason for choosing the Softmax layer and OLR layer as the activation functions after the convolutional layer is to optimize the distribution of data in the adjacency matrix. In early testing, we found that if the Sigmoid function was used as the activation function for normalization, the variance of the adaptive adjacency matrix elements would be extremely small. This trend causes the adjacency matrix to lose its ability to extract local information. In extreme cases, such as when each graph node is directly connected to all other nodes, the entire GCN model will degenerate into a fully connected graph and lose its topology. This means that the GCN model will degenerate into a simple multi-layer perceptron. To avoid this problem, our model uses a combination of Softmax layer and OLR layer to constrain the generation process of adjacency matrix.

The Softmax layer is an activation function that maps output values to (0, 1), which is common in modern neural network models. The mathematical definition of the Softmax function can be formulated as:

$$f(Z_J) = \frac{e^{Z_J}}{\sum_{k}^{K} e^{Z_k}} \tag{1}$$

where J, k is the index of a node in graph data and K is the set of nodes in a graph. Softmax function ensures that the sum of the outputs is always equal to 1. Although the Softmax layer can constrain the accumulated sum of the input, it cannot solve the problem of small output variance. This problem can be solved by passing the feature values to the OLR layer.

Before giving the definition of the OLR layer, it is necessary to introduce the ReLU function. The ReLU function is the most common activation function in deep learning, and its mathematical expression is very simple:

$$F(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{2}$$

But ReLU has a fatal drawback, which is the generation of "dead neurons". This is mainly caused by the large negative gradient flowing through the network, which leads to

the large negative weight of ReLU neurons. The output of this neuron during feedforward and backpropagation is always equal to zero, so the weight of this neuron is never updated and is considered dead forever.

In some ordinary neural networks, the number of neurons is so large that the death of some neurons is acceptable. But in the adaptive adjacency matrix, this means that once a connection is abandoned by the module, there is no possibility to be reconnected again. To solve this problem, we propose an OLR activation function based on the ReLU function. The definition of OLR is as follows:

$$F(x) = \begin{cases} x \times 10^{-a}, & x < L \\ x, & x \leq L \end{cases} \tag{3}$$

Here, L is activation threshold of a neuron and a is a gradient coefficient. There are two differences between OLR layers and the ReLU function:

1. When x is less than the threshold, the output value is equal to $x \times 10^{-a}$ instead of directly specifying it as 0. This results in neurons with low weights having almost no impact on the results during graph convolution, but still obtaining gradients during backpropagation. Therefore, the opportunity to be selected into the connection relationship will not be lost in subsequent calculations.
2. Translate the activation threshold of the joint point to L instead of fixing it at the coordinate origin. By adjusting the threshold of OLR to L, the effective number of connections at a joint point can be constrained below $1/L$ after data normalization.

The AMG module compresses input node information through three convolutional layers and constrains the number of active connections generated by the combination of the Softmax layer and OLR layer, ensuring that the adaptive adjacency matrix can effectively assist in mining spatial domain information.

After implementing the adaptive adjacency matrix generation network, we also improved the structure of GCN units, thereby expanding the GCN calculation process of the adaptive adjacency matrix. The improved structure is depicted in Figure 2.

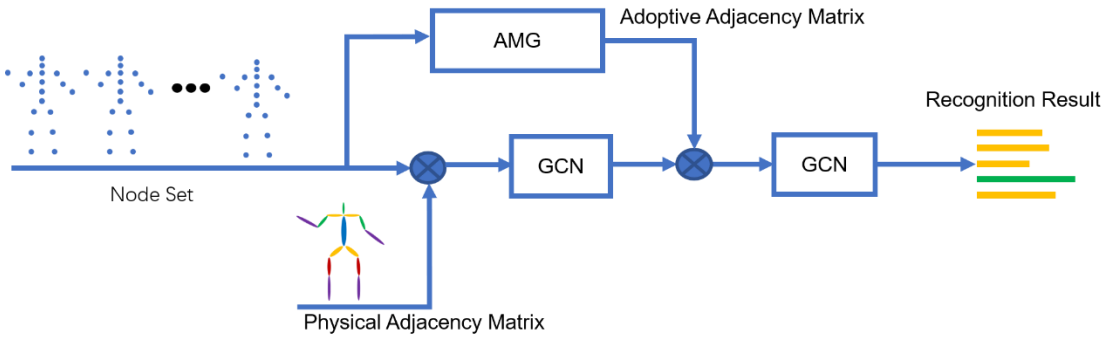


Figure 2. Improved GCN Unit with AMG.

Here, each GCN unit contains two independent graph convolution operations. The first is graph convolution based on the physical adjacency matrix, which provides basic motion information extraction; the second is graph convolution based on the adaptive adjacency matrix, which provides the extraction of synergistic relationships between disconnected points in physical space. This two-stage graph convolutional unit structure helps the model better understand spatial domain information.

By assembling nine convolutional layers of different sizes and adding the BN (Batch Normalization) layer, GP (Global Pooling) layer and FC (Fully Connected) layer during the process, the AMG-GCN model was obtained, as shown in Figure 3.

Due to the fact that different nodes share weights during the calculation process, in order to ensure the same scale of input data, a BN layer is used at the network entrance to normalize the data of the input nodes. The main part of the network consists of nine graph

convolutional units (GCNs). The three parameters in the GCN unit in Figure 3 represent the size of the input convolution channel, the number of output convolution channels and the convolution step size, respectively. After passing through all the graph convolutional layers, the GP layer compresses the data to a manageable size. Then, the FC layer finally outputs a recognition vector, with each element corresponding to the recognition score of each action.

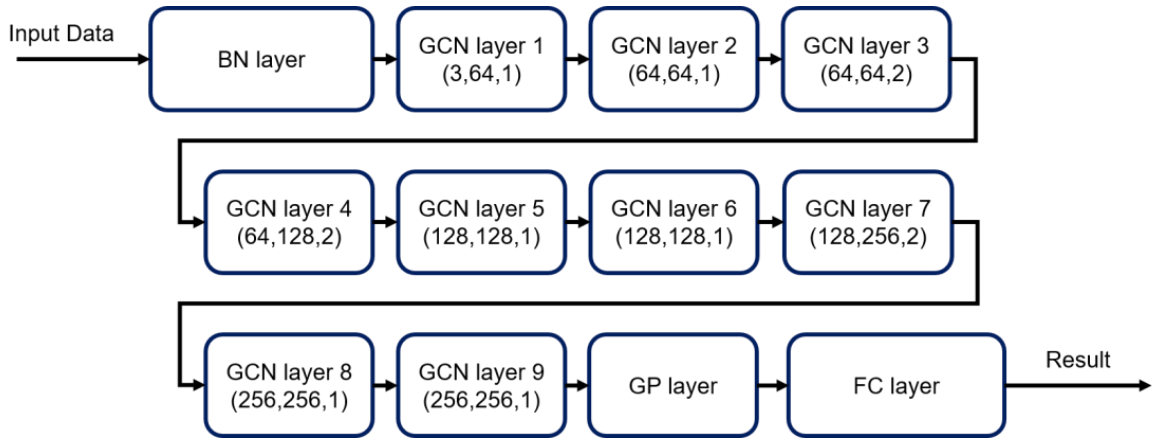


Figure 3. AMG-GCN model.

3.2. TDA Mechanism

In a general spatiotemporal GCN, the graph convolutional unit reads temporal information through temporal connections. This temporal connection form is intuitive, simple and computationally convenient, but it makes the contribution of different time frames to the results indistinguishable, which also brings some drawbacks.

1. Unable to remove frames without information. In a set of video sequences, at certain frames after the start and end of an action, the character is in a stationary state or is not even included in these frames. This means that these frames do not contain character motion information. If the duration of the action is short, then such uninformative frames will increase the workload of recognition and cause disturbance to the recognition results.
2. Unable to distinguish the contribution of different motion frames to the results. In some non-continuous actions, the contribution of different time frames to the results is very different. For example, in the "play badminton" behavior, the action of racket swing is obviously more important to the final behavior judgment than the action of running on the court. Here, simple temporal domain connections cannot effectively capture motion frames that are significant for recognition accuracy.

Due to the aforementioned shortcomings of temporal domain connections, introducing attention mechanisms into graph convolution models can enhance the model's temporal information extraction ability, thereby improving the accuracy of the model in non-continuous action recognition. Based on this idea, this subsection proposes the TDA (Time Domain Attention) module, which can allocate weights to different motion frames, thereby implementing the temporal attention mechanism in the graph convolution process.

In the field of computer vision, the most common attention module is the SE-block module proposed by Hu [16]. The structure of the TDA module proposed in this paper is optimized based on the SE-block module. The SE-block is a channel weighted attention module based on CNN, which includes two parts: compression and excitation. The core idea of SE-block is to compress channel information through global average pooling, and then generate channel weights according to the compressed information. TDA also adopts this compression and excitation structural design.

The structure of the TDA mechanism is shown in Figure 4. Here, F_a and F_m represents the average and maximum pooling operation, respectively. $F_{ex}(W)$ represents the excitation function with parameter matrix W, which is a network parameter learned through back-propagation during the training process. In the beginning, the four-dimensional spatial temporal graph is converted into a 3D feature matrix after the reshape operation. Then, data compression is performed using average pooling and maximum pooling through two channels, respectively. After pooling, the 3D feature matrix data is compressed into two $T \times 1$ vectors. After the activation function, two excited vectors S_1 and S_2 can be obtained, respectively. Finally, the maximum pooling operation is used to fuse the two excited vectors and merge them into the final frame attention vector.

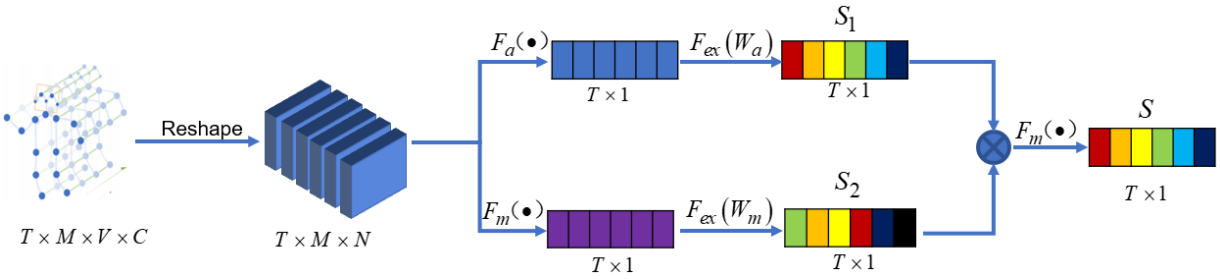


Figure 4. The TDA mechanism.

In the TDA model, the fusion method of the two-channel vector is dimension expansion. The commonly used fusion method in attention mechanism is weighted multiplication, but it is not suitable for the TDA mechanism. Different from convolutional feature map, the node value in the skeleton graph is the coordinate position of the node. So, the weighted multiplication of the node is equivalent to the translation of the node along the coordinate axis, which will bring a data offset problem. Dimension expansion adds importance dimension to every node, avoiding the offset problem during fusion.

Unlike the single channel global average pooling method of the SE-block, TDA adopts a two-channel model that includes two pooling methods to achieve data compression. The two-channel pooling model can avoid the problem of continuous feature attenuation that may occur during backward propagation in a single channel average pooling model, thus enabling a more comprehensive collection of frame information and achieving finer inter-frame differentiation.

The modeling of AMG-TDA-GCN is shown in Figure 5. Firstly, the TDA mechanism obtains attention vectors by computing spatial temporal graph. Secondly, the attention vector is fused with the original graph data to generate a time-domain attention graph. Finally, the time-domain attention graph is used as an input for the AMG-GCN model, and the final recognition result is obtained through computation.

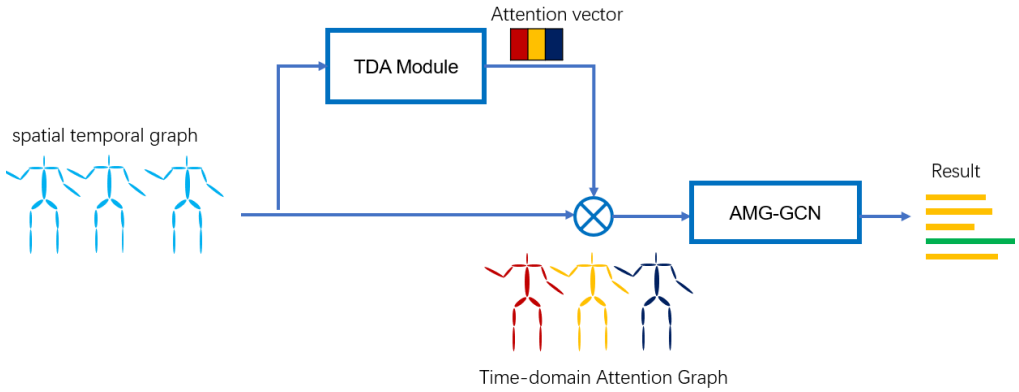


Figure 5. The modeling of AMG-TDA-GCN.

4. Experimental Results and Discussions

After completing the modeling of AMG-TDA-GCN, this section mainly discusses a series of experiments conducted on the action recognition dataset NTU-RGB+D to verify the effectiveness of the improved modules and the overall model proposed in this paper.

4.1. Dataset and Implementation

The NTU-RGB+D dataset [3] is currently the most comprehensive dataset in the field of action recognition. This dataset consists of 56,880 samples, including a total of 60 behavioral categories collected from 40 subjects. These behaviors include 40 classes of daily behaviors (such as eating, drinking and reading), 9 classes of health-related behaviors (such as falling and sneezing), and 11 classes of multi-person interaction behaviors (such as hitting and hugging). This dataset is rich in samples and diverse in types, making it very suitable for training and verifying the effectiveness of behavior recognition models.

Before conducting action recognition training, in order to eliminate the impact of the human body's position in the image on the algorithm, the key point data in the dataset is first subjected to normalization preprocessing operations. During the model training process, data augmentation methods were also used to enhance the robustness of the model. So-called data augmentation is a method of improving model accuracy by expanding the diversity of training data through algorithms. In order to increase the recognition accuracy of the behavior recognition model when facing videos from different angles, the experiment rotated the key point set in the dataset and conducted additional training.

In model training, the training round is set to 60 epochs and the batch size is set to 64. Each graph convolutional layer is set with a dropout coefficient of 0.3 to alleviate overfitting.

In the process of selecting the learning rate, it must be noted that if the learning rate is too small, the convergence speed of the model will be greatly reduced, but if the learning rate is too large, the model accuracy will be poor. In order to balance the accuracy and training speed of the model, the learning rate decay algorithm [17] is used during the training process, which adopts a higher learning rate in the early stage of training. As training progresses, the learning rate will continue to decline. In our experiments, the initial learning rate is set to 0.01, and for every 20 epoch training sessions, it will decrease by 10%.

4.2. Experimental Demonstration of AMG Module

Due to the large scale and random morphology of the adaptive adjacency matrix, its performance was analyzed in the experiment by counting the number of connections and connection states of the adaptive adjacency matrix.

The number of "1" elements in the adjacency matrix is directly related to the neighborhood size of a single node. If the neighborhood of a node is too large, it means that each node is directly connected to a large number of other nodes, which will affect the comprehension of local information by graph convolution. In extreme cases, if the adjacency matrix is a full "1" matrix, it means that every node in the graph is directly connected to all other nodes. At this point, the GCN will degenerate to a Multilayer Perceptron and the graph convolution calculation is meaningless, which will have a significantly negative impact on the recognition ability of the model. In order to demonstrate the constraint effect of the OLR layer on the node neighborhood, the experiment analyzes the node neighborhood size of the adjacency matrix generated by the AMG module in the test set, and the results are summarized in Table 1.

In Table 1, the maximum neighborhood size of a single node is 5, which means that a single node is directly connected to up to five other nodes, indicating that the OLR layer successfully constrains the neighborhood size of a single node, thereby effectively preventing the generated adjacency matrix from degenerating into a full "1" matrix. Meanwhile, the number of nodes with a neighborhood size of 3 in Table 1 is the highest, accounting for 42.52% of the total number of nodes. This data distribution is also similar to the physical adjacency matrix. Therefore, it can be concluded that the number of connections in the adjacency matrix is in a reasonable state under the constraints of the OLR layer.

Table 1. Statistics of Neighborhood Size Nodes.

Neighborhood Size	Node Number	Node Ratio
1	56,068	13.60%
2	88,397	21.45%
3	175,227	42.52%
4	85,246	20.68%
5	7212	1.75%
6	0	0%

Due to the large scale and complex connection of the original adaptive adjacency matrix, it is difficult to intuitively express the connection state represented by it. In the experiment, all nodes of the human body are divided into six parts: head, body, left arm, right arm, left leg and right leg. The connection relationship of the adaptive adjacency matrix is compressed into a 6×6 matrix, which is beneficial to observe the generation state of the adjacency matrix.

In our experiment, action propensity matrix P is used to represent the connection state of adaptive adjacency matrix with its definition as follows:

$$G_S = \sum_{j=1}^{N} \frac{A_s^{(j)}}{N} \tag{4}$$

$$G_c = \sum_{i=1}^{M} \frac{A_c^{(i)}}{M} \tag{5}$$

$$P_c = \frac{G_c - G_S}{G_S} \tag{6}$$

Here, A_s and A_c represent the adjacency matrix of a specific action c and all actions, respectively. G_S and G_c represent the adjacency matrix expectation generated by AMG module when dealing with a specific action c and general action, respectively. M and N represent the number of all types of actions and of action c, respectively. Lastly, P_c represents the degree of tendency between different parts during the generation of adjacency matrices.

After generating the action propensity matrix P_c, the elements in the matrix are marked with different colors according to their values. The larger the value, the darker the red color, indicating that the model has a higher tendency to generate this connection when dealing with action c.

The propensity matrix of "clapping" is shown in Figure 6. Here, the tendency values between the left arm and body, between the right arm and body and between the left arm and right arm are 0.082, 0.141 and 0.237, respectively, which are significantly higher than those between other parts. This indicates that the model pays more attention on the connections between the left arm, right arm and body when recognizing the behavior of "clapping". This is because the synergistic relationship between the left and right arms has a significant impact on the results in the recognition process of the "clapping" behavior. Therefore, in the adaptive adjacency matrix generated for the "clapping" behavior, the AMG module tends to generate more connections between both hands and body. Reflected in the propensity matrix, it is manifested as the tendency values between the left and right arms and the body being much greater than others. This proves that the adaptive AMG network can indeed generate adjacency matrices that are more conducive to mining synergistic information by pre-analyzing key point features during the recognition process of "clapping" action.

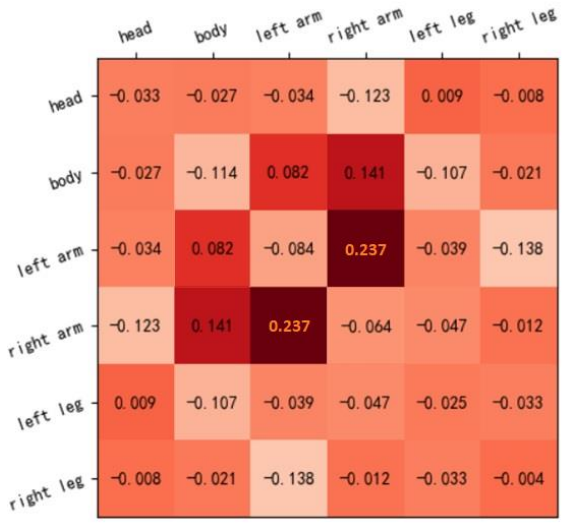
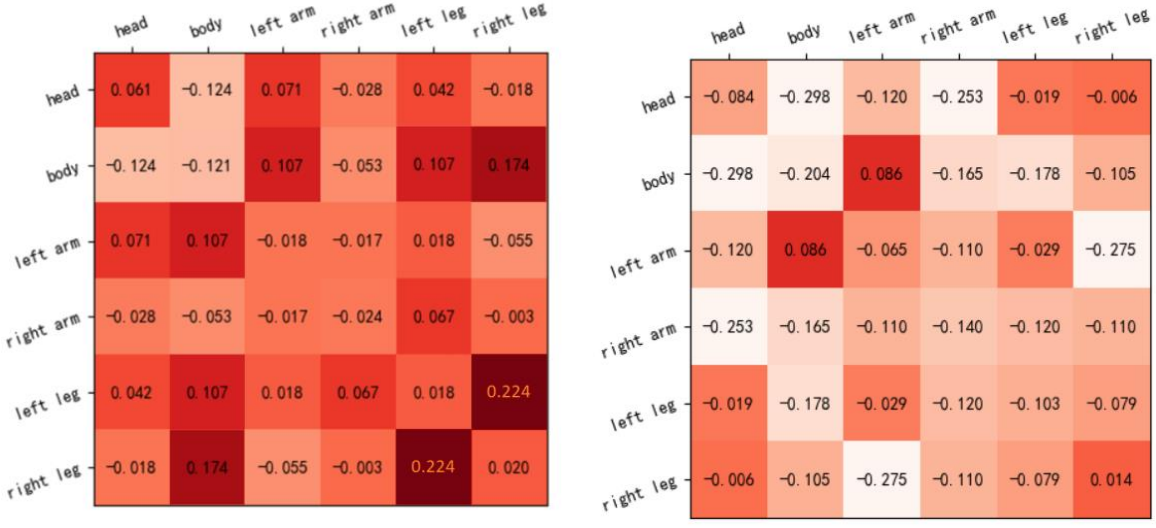


Figure 6. Propensity matrix of "clapping".

Figure 7 compares and analyzes the propensity matrix between "writing" and "jumping" behaviors. Compared to the "writing" behavior, the behavior tendency matrix of the "jumping" action is significantly darker in color, indicating that the tendency values in the behavior tendency matrix of the "jumping" action are significantly higher than those in the "writing" behavior. This shows that the adjacency matrix generated by the AMG module for "jumping" actions tends to contain more connection information, while the "writing" action is the opposite. This is because "jumping" is a relatively large action compared to "writing", with significant displacement of key points in the body, and the synergistic relationship between various parts has a significant impact on the recognition results.



(**a**) (**b**)

Figure 7. Comparison on propensity matrices between "writing" and "jumping" behaviors: (**a**) Propensity matrix of "jumping" behavior; (**b**) Propensity matrix of "writing" behavior.

Therefore, the adaptive adjacency matrix for the "jumping" behavior contains a larger number of connections. In recognition of the "writing" behavior, the decisive information comes more from the synergistic information between the left and right arms and the body, while the movement of other parts is relatively small, so the number of connections in the adaptive adjacency matrix is relatively small.

Table 2 provides a comparison of the AMG-GCN model with and without OLR layers. Here, Cross Subject (CS) and Cross View (CV) are two different divisions of the test set. CS refers to dividing the training set and test set based on different characters, where none of the characters that appear in the test set have appeared in the training set. CV is the division of training and testing sets based on the different cameras.

Table 2. Model comparison with and without OLR layer.

Model	CS	CV
ST-GCN	81.5%	88.3%
AMG-GCN without OLR layers	62.3%	69.1%
AMG-GCN	82.8%	89.1%

According to the experimental results in Table 2, the recognition effect of model without OLR is obviously poor. After extracting and analyzing the generation state of the adaptive adjacency matrix, we found that the neighborhood size of nodes in the graph data is too large, resulting in poor model extraction of local information. The AMG-GCN model with added constraints has a 1.3% and 0.8% improvement in accuracy compared with the original ST-GCN model, respectively.

4.3. Experimental Demonstration of TDA Mechanism

In the field of machine learning, confusion matrices are often used to demonstrate the model's ability to misclassify different types of actions. Here, confusion matrices are used to analyze the impact of attention mechanisms in different action classifications. The element values in the conventional confusion matrix represent the number of classifications, but due to the non-uniformity of the dataset, the number of actions of different types varies greatly, so it cannot directly reflect the classification situation of the model. Therefore, in our experiment, the elements in the confusion matrix are changed to the percentage of classification results in the total number of classifications. In addition, due to the fact that the NTU-RGB+D dataset has two evaluation indicators (CS and CV), the elements of the confusion matrix are set as the fusion result of the two evaluation indicators in the experiment.

Since the dataset contains a total of 60 action categories, the confusion matrix heatmap contains a total of 3600 elements. Such a huge amount of data is not suitable for detailed analysis of the performance of a single action category. Therefore, six behaviors are selected to demonstrate the effect of the TDA mechanism. These six behaviors can be clearly divided into two categories, sustained behaviors and non-sustained behaviors. Sustained behaviors are "reading", "writing" and "calling", which are similar at the beginning and end of the action. Non-sustained behaviors are "jumping", "kicking" and "fall", in which the human postures vary greatly at different stages.

The confusion matrices of AMG-GCN model and TDA-AMG-GCN model are generated for these six behaviors, as shown in Figure 8. In these experimental results, the recognition accuracy of non-sustained behaviors increased by 2 to 5 percentage. Since the different stages of non-sustained actions are quite different, the TDA mechanism can help the model to distinguish high-quality frames. However, the recognition accuracy of continuous actions does not increase significantly.

At the end of this section, we compared the TDA-AMG-GCN model proposed in this paper with state-of-the-art behavior recognition methods. Table 3 summarizes the performance of ten models on the NTU-RGB+D dataset. It can be seen that the TDA-AMG-GCN model proposed in this paper has considerable competitiveness in recognition accuracy.

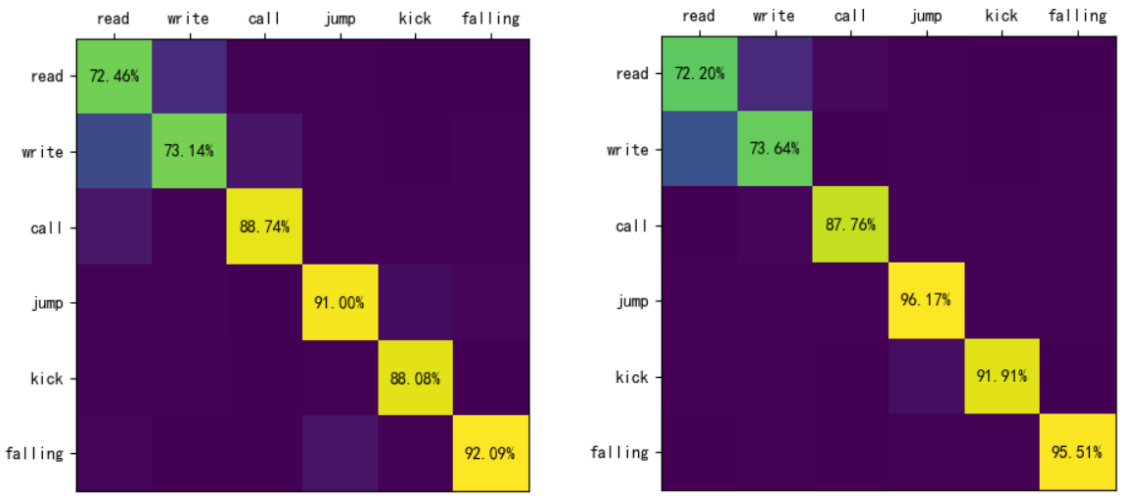


Figure 8. Confusion Matrices of six behaviors: (**a**) AMG-GCN model confusion matrix without TDA mechanism; (**b**) AMG-GCN model confusion matrix with TDA mechanism.

Table 3. Comparison of TDA-AMG-GCN model with the state-of-the-art behavior recognition methods.

Method	CS	CV
Deep LSTM [3]	60.7%	67.3%
ST-LSTM [18]	69.2%	77.7%
VA-LSTM [19]	79.2%	87.7%
GCA-LSTM [20]	77.1%	85.1%
TCN [21]	74.3%	83.1%
HCN [22]	86.5%	91.1%
Two Stream CNN [7]	83.2%	89.3%
DPRL [23]	83.5%	89.8%
ST-GCN [1]	81.5%	88.3%
TDA-AMG-GCN (ours)	84.5%	89.8%

5. Conclusions

In this paper, we propose a GCN model with the Adjacency Matrix Generation (AMG) module and Time-Domain Attention (TDA) mechanism. The AMG module helps to pre-analyze skeleton data and generate an adjacency matrix suitable for the current task before recognition starts. The adjacency matrix generated by the AMG module is conducive to handling the collaborative relationships between skeleton node sets. The TDA mechanism generates weight vectors for time frames, allocating more computational resources to those potential key frames during the action recognition process. The introduction of TDA mechanism can improve the detection precision of the model in non-sustained action classification. An AMG module and TDA mechanism are added to the original GCN model to achieve a more excellent AMG-TDA-GCN model. The effectiveness of the AMG-TDA-GCN model is validated through a series of experiments.

Nevertheless, there are still some issues that need further research, such as the occlusion of key points, which greatly affects the reliability of recognition algorithms. In subsequent research, if the model can infer occluded key points, the performance of the behavior recognition algorithm will be further improved.

Author Contributions: Conceptualization, R.Y. and J.N.; methodology, R.Y.; software, J.N.; validation, Y.X., Y.W. and L.Q.; writing—original draft preparation, J.N.; writing—review and editing, R.Y.; visualization, J.N.; supervision, R.Y.; funding acquisition, R.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant numbers 61773266.

Data Availability Statement: Not applicable.

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

References

1. Yan, S.; Xiong, Y.; Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 7444–7452.
2. Graves, A. Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 1735–1780.
3. Shahroudy, A.; Liu, J.; Ng, 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 1010–1019.
4. Lu, W.-L.; Little, J.J. Simultaneous tracking and action recognition using the PCA-HOG descriptor. In Proceedings of the 3rd Canadian Conference on Computer and Robot Vision (CRV'06), Quebec, ON, Canada, 7–9 June 2006.
5. Thurau, C.; Hlavác, V. Pose primitive based human action recognition in videos or still images. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008.
6. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
7. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2014; pp. 568–576.
8. Liu, J.; Wang, G.; Duan, L.-Y.; Abdiyeva, K.; Kot, A.C. Skeleton-based human action recognition with global context-aware attention LSTM networks. *IEEE Trans. Image Process.* **2018**, *27*, 1586–1599. [CrossRef] [PubMed]
9. 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 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
10. Cheng, K.; Zhang, Y.; Cao, C.; Shi, L.; Cheng, J.; Lu, H. Decoupling GCN with DropGraph module for skeleton-based action recognition. In Proceedings of the 2020 European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
11. Thakkar, K.; Narayanan, P.J. Part-based graph convolutional network for action recognition. *arXiv* **2018**, arXiv:1809.04983.
12. Wang, S.; Zhang, Y.; Zhao, M.; Qi, H.; Wang, K.; Wei, F.; Jiang, Y. Skeleton-based action recognition via temporal-channel aggregation. *arXiv* **2022**, arXiv:2205.15936.
13. Xing, Y.; Zhu, J.; Li, Y.; Huang, J.; Song, J. An improved spatial temporal graph convolutional network for robust skeleton-based action recognition. *Appl. Intell.* **2023**, *53*, 4592–4608. [CrossRef]
14. Zhang, Y.J.; Cai, H.B. Sequence Segmentation Attention Network for Skeleton-Based Action Recognition. *Electronics* **2023**, *12*, 1549. [CrossRef]
15. Chen, S.; Xu, K.; Mi, Z.J.; Jiang, X.H.; Sun, T.F. Dual-domain graph convolutional networks for skeleton-based action recognition. *Mach. Learn.* **2022**, *111*, 2381–2406. [CrossRef]
16. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]
17. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
18. Liu, J.; Shahroudy, A.; Xu, D.; Kot, A.C.; Wang, G. Skeleton-based action recognition using spatio-temporal LSTM network with trust gates. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 3007–3021. [CrossRef] [PubMed]
19. Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J.; Zheng, N. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1963–1978. [CrossRef] [PubMed]
20. Liu, J.; Wang, G.; Hu, P.; Duan, L.-Y.; Kot, A.C. Global context-aware attention LSTM networks for 3D action recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
21. Kim, T.S.; Reiter, A. Interpretable 3D human action analysis with temporal convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017.
22. Li, C.; Zhong, Q.; Xie, D.; Pu, S. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. *arXiv* **2018**, arXiv:1804.06055.
23. Tang, Y.; Tian, Y.; Lu, J.; Li, P.; Zhou, J. Deep progressive reinforcement learning for skeleton-based action recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.