symmetry

MDPI

Article

Relation Selective Graph Convolutional Network for Skeleton-Based Action Recognition

Wenjie Yang [1,2,3], Jianlin Zhang [2,3,*], Jingju Cai [2,3] and Zhiyong Xu [2,3]

1 Key Laboratory of Optical Engineering, Chinese Academy of Sciences, Chengdu 610209, China; yangwenjie17@mails.ucas.ac.cn
2 Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610209, China; xueman1999@163.com (J.C.); xuzhiyong@ioe.ac.cn (Z.X.)
3 School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: jlin@ioe.ac.cn

Abstract: Graph convolutional networks (GCNs) have made significant progress in the skeletal action recognition task. However, the graphs constructed by these methods are too densely connected, and the same graphs are used repeatedly among channels. Redundant connections will blur the useful interdependencies of joints, and the overly repetitive graphs among channels cannot handle changes in joint relations between different actions. In this work, we propose a novel relation selective graph convolutional network (RS-GCN). We also design a trainable relation selection mechanism. It encourages the model to choose solid edges to work and build a stable and sparse topology of joints. The channel-wise graph convolution and multiscale temporal convolution are proposed to strengthening the model's representative power. Furthermore, we introduce an asymmetrical module named the spatial-temporal attention module for more stable context modeling. Combining those changes, our model achieves state-of-the-art performance on three public benchmarks, namely NTU-RGB+D, NTU-RGB+D 120, and Northwestern-UCLA.

Keywords: human skeleton; action recognition; graph convolutional networks

Citation: Yang, W.; Zhang, J.; Cai, J.; Xu, Z. Relation Selective Graph Convolutional Network for Skeleton-Based Action Recognition. *Symmetry* **2021**, *13*, 2275. https://doi.org/10.3390/sym13122275

Academic Editor: Jan Awrejcewicz

Received: 13 October 2021
Accepted: 22 November 2021
Published: 30 November 2021

1. Introduction

Action recognition is an essential computer vision topic. There are broad practical applications like intelligent monitoring, human-computer interaction, video summary, and healthy caring. For example, ref. [1] design a rehabilitation system based on a customizable exergame protocol to prevent falls in the elderly population. In recent years, skeleton-based human action recognition has attracted increasing attention from researchers. The skeleton data is the coordinate sequence of each key point of the joints when the action occurs, which has a small data size and robustness to variations of viewpoints, appearances, and environmental change. Compared with the video, the skeleton sequence is a compact and high-level description of actions.

In the literature, the methods for skeleton-based action recognition can be classified as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and graph convolutional neural networks (GCNs). The CNN-based and RNN-based methods tend to represent the skeleton data with pseudo-image or vector sequences. However, the skeleton sequence is not the regular Euclidean structure data. It inherently has the structure information of the human body, which the vector or pseudo-image cannot fully express. To better model the topology information in skeleton sequences, graph convolutional networks are first introduced by [2]. They encode the skeleton sequence by the spatial-temporal graph . They build the adjacency matrix for the joints on the basis of the physical human body structure, making impressive progress. But the fixed adjacency matrix makes their model unable to capture some critical correlation between joints that are not included in

the human physical structure. Thus 2s-AGCN (two-stream adaptive graph convolutional network) [3] proposes the adaptive adjacency matrix, which allows the generation of edges between arbitrary joints. After that, many methods follow the idea and utilize adaptive adjacency matrices to extract topology information.

Due to the excellent use of human body structure information and the continuous enhancement of model's representative power, GCN-based methods have made significant progress. However, there are still limitations that exist in those methods. Firstly, the dense adjacency matrices may generate many redundant connections. These connections do not help extract discriminant features and may even disturb the capture of detailed local information on the spatial dimension. Secondly, the correlation of the joints in different actions is diverse. Those methods use the same adjacency matrices in every neuron of a layer, which seriously limits the model's ability to capture topology in skeleton sequences of different actions.

To address such issues, we propose a trainable relation selection mechanism, which can help the model choose the most informative connection of the graph in a trainable way. Therefore, the adjacency matrix can be sparsed, and the nodes can focus on the most important neighbors. Then we propose the channel-wise graph convolution (CWG) and multiscale temporal convolution (MTC) to strengthen the model's representative power. The CWG assigns adjacency matrices for channels, and diverse relations are built in different channels. The MTC has several branches, and each has its kernel size for assembling information from various temporal receptive fields. Furthermore, we introduce the spatial-temporal attention module (STAM) to enhance the model's ability to capture context relations. Incorporating these improvements, we built a novel model called relation selective graph convolutional networks (RS-GCN). In the experiments, our proposed approach outperforms state-of-the-art methods on three large-scale skeleton action benchmarks: NTU-RGB+D [4], NTU-RGB+D 120 [5], and Northwesten-UCLA [6]. The main contributions of this work are summarized as follows:

- We design a relation selection mechanism that helps the model choose the most helpful connections of the graph. It allows the model to generate sparse adjacency matrices and avoid redundant information transfer between nodes;
- We propose the channel-wise graph convolution and the multiscale temporal convolution. Those two operations significantly enhance the model's adaptability to different actions and different speeds of motion;
- We introduce a spatial-temporal attention module that has a symmetrical structure, making our model more sensitive to complex context relations;
- We integrate the components, forming a novel model named RS-GCN. In experiments, our model outperforms the state-of-the-art method on all three public skeleton-based action recognition datasets, which illustrates its superiority.

2. Related Works

In dealing with the skeletal action recognition problem, researchers have attempt many approaches. Early methods usually rely on the hand-craft features like a histogram of 3D joints [7], histogram of oriented displacements (HOD) [8], and relative 3D rotations between various body parts [9].

With the extensive application of deep learning technology, researchers begin to address this task through convolutional neural networks (CNNs) or recurrent neural networks (RNNs). The CNN-based methods usually encode the skeleton sequence into a pseudo-image and use the CNNs as the classifier [10–13]. The RNN-based methods view the skeleton data as the sequences of vectors, and various recurrent neurons are applied to model the temporal evolution [14–16]. However, as we all know, the connections between graph structure data nodes are complex and changeable. Neither convolutional neural networks nor recurrent neural networks fully express the human body's structural information because of the grid-structured representation of features.

On the contrary, graph convolutional networks (GCNs) can effectively capture and represent this structural information using the adjacency matrix. The work of [2] first introduced the graph convolution into the task and has gained an exciting improvement, attracting extensive attention from researchers. Many methods solve the skeleton-based action recognition problem using GCNs. Ref. [3] propose adaptive graph convolution, and it can generate the connection between two arbitrary nodes to capture the relations of joints not connected by the natural human skeleton. Ref. [17] propose a multi-stream framework, each stream only focuses joints not activated by the previous streams, for more discriminative features. Ref. [18] design a graph edge convolutional neural network to explore the beneficial information in bones for skeleton-based action recognition. Ref. [19] decompose graph convolution into feature learning components that evolve the features of each graph vertex to learn the latent graph topologies. Ref. [20] propose a part-level graph convolutional network to capture the part-level information of skeletons. Ref. [21] design a split-transform-merge strategy in GCNs for skeleton sequence processing. Ref. [22] refine the pose before recognizing the action of the skeleton to reduce the effect of pose mistakes. Ref. [23] use two scales of graphs to explicitly capture relations among body-joints and body-parts. Ref. [24] guide GCNs to perceive significant variations across local movements by a tri-attention module. Ref. [25] attempt to fix the shortcomings of isolated temporal information in spatial temporal graph convolutional networks by a two-stream network called RNXt-GCN. Ref. [26] propose a graph pooling method, named Tripool, for a lower computational cost and large reception field.

No matter how much progress has been made in these GCN-based methods, there is still some room for improvement. Many of those methods focus on modeling the relation in nodes by generating excessive graph edges. However, some of those are redundant, and do not help extract discriminant features and may even disturb the capture of detailed local information on the spatial dimension. Moreover, those GCN-based methods usually share the same adjacency matrices among all neurons of a layer, which is low-efficient to capture topology in different actions. In this work, we deploy channel-wise adjacency matrices in the GCNs to strengthen the model's ability to capture more abundant relations. Furthermore, we propose a trainable edge-selective mechanism. It chooses the connections really needed for action representation and reduces unnecessary ones. Therefore, our model can focus more attention on potential discriminative parts in the skeleton.

3. Methods and Materials

3.1. Pipeline Overview

The pipeline of our framework is depicted in Figure 1 , which consists of 8 GCN blocks, 2 STAMs, and a fully connected layer. Every GCN block contains a CWG and an MTC for the spatial and temporal dimension, respectively. The CWG deploys the different adjacency matrices for channels so that they can develop their topology independently. The MTC can integrate the information from various temporal receptive fields, enhancing the model's adaptability to the speed of actions. STAMs are embedded behind the fifth and seventh GCN blocks, and they help our model collect the most informative features along spatial and temporal dimensions.

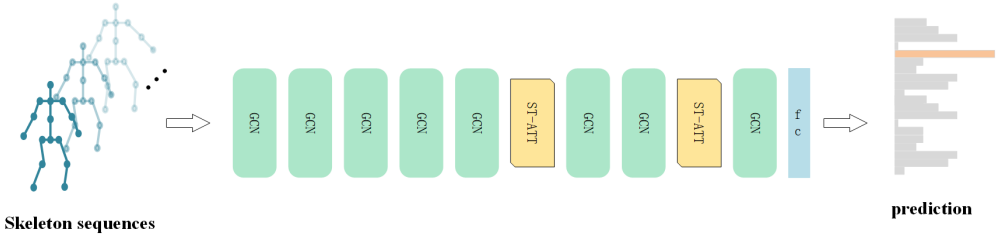

Figure 1. The pipeline of proposed RS-GCN (relation selective graph convolutional network).

3.2. Channel-Wise Graph Convolution

The channel-wise graph convolution (CWG) is proposed to build different correlation in channels. The feature map of the network can be viewed as a $C \times T \times N$ tensor, where N is the number of nodes, T is the temporal length, and C is the number of channels. Before the introduction of CWG, we briefly review the definition of channel-shared graph convolution in our baseline (2s-AGCN [3]), which can be formulated as:

$$\mathbf{f}_{out} = \sum_k \mathbf{W}_k \mathbf{f}_{in} (\mathbf{A}_k + \mathbf{B}_k + \mathbf{C}_k) \tag{1}$$

where $\mathbf{f}_{in} \in \mathbb{R}^{C_{in} \times T \times N}$ is the input feature; $\mathbf{f}_{out} \in \mathbb{R}^{C_{out} \times T \times N}$ is the output feature; $\mathbf{W}_k \in \mathbb{R}^{C_{out} \times C_{in}}$ is the weight vector of the 1×1 convolution operation; and $\mathbf{A}_k \in \mathbb{R}^{N \times N}$ is the original normalized adjacency matrix defined by the skeleton's natural structure. $\mathbf{B}_k \in \mathbb{R}^{N \times N}$ is an adaptive matrix that is optimized with the other parameters; and $\mathbf{C}_k \in \mathbb{R}^{N \times N}$ is a self-attention matrix to model relation in every two nodes.

In this work, we only use the $\mathbf{B}_k$ initialized by $\mathbf{A}_k$ as the adjacency matrix of CWG. It is inspired by [19], they believe that $\mathbf{B}_k$ has higher freedom to learn and represents the skeleton's topology to improve network performance. Moreover, we extend it by channel dimension to get our channel-wise adjacency matrix $\mathbf{G}_k \in \mathbb{R}^{C \times N \times N}$, then Equation (1) can be rewritten as:

$$\mathbf{f}_{out} = \sum_k \mathbf{W}_k ([\mathbf{f}_{in}^1 \mathbf{G}_k^1, \mathbf{f}_{in}^2 \mathbf{G}_k^2, \cdots, \mathbf{f}_{in}^C \mathbf{G}_k^C]) \tag{2}$$

where $[\cdot]$ denotes the concatenation of tensors.

The diagram of the channel-shared and channel-wise adjacent matrix is shown in Figure 2. The left half of the figure is the channel-shared graph convolution used in the previous skeleton-based action recognition approach. We can see that the same adjacency matrices are used in different channels. As we all know, the feature map in each channel is diverse, as are their topological relations. It is not reasonable to apply the same adjacency matrix to model dissimilar topological relations. The right half of the figure is the channel-wise graph convolution (CWG). For each channel, we assign an independent adjacency matrix to establish the topology with its feature map. In this way, the model's ability to model spatial correlation is greatly improved. As the channel-share graph convolution also broadcasts the adjacency matrix to all channels, our method does not bring the additional computational cost of the inference.

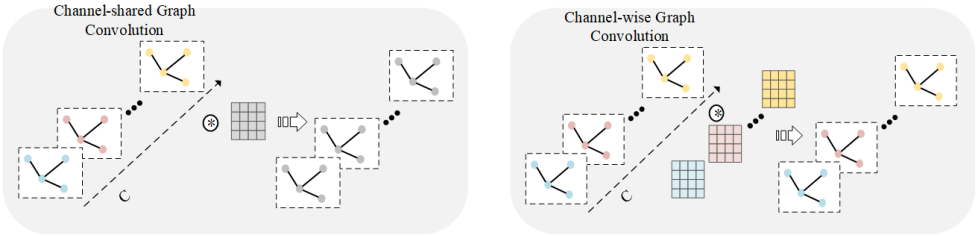

Figure 2. Diagram of the channel-shared graph convolution and the channel-wise graph convolution.

3.3. Relation Selection Mechanism

As mentioned in Section 3.2, we deploy the learnable adjacency matrices in our model, which can add connections between two arbitrary nodes of the graph. It allows the generation of edges between highly correlated points that are not connected by the physical structure of the human skeleton. However, its ability to reduce edges is much weaker than its ability to increase them. The edge of the graph eliminates only when its corresponding element in the adjacency matrix is precisely 0. Eventually, the adjacency matrix produces very dense connections, creating complex and confusing relations among the nodes, which is detrimental to the extraction of discriminative features. Therefore, we propose a relation selection mechanism to pick out meaningful connections in the graph.

Specifically, we set a small threshold δ. Only the edge corresponding to the element larger than the threshold in the adjacent matrix will take effect in the graph convolution operation. It can be formulated as:

$$\hat{\mathbf{G}}(c,i,j) = \begin{cases} \mathbf{G}(c,i,j), & \mathbf{G}(c,i,j) < \delta, \\ 0 \quad , & \mathbf{G}(c,i,j) \geq \delta. \end{cases} \tag{3}$$

where c, i, j is the index of the adjacency matrix elements at three different axes. Then we can represent our CWG as:

$$\mathbf{f}_{out} = \sum_k \mathbf{W}_k([\mathbf{f}_{in}^1 \hat{\mathbf{G}}_k^1, \mathbf{f}_{in}^2 \hat{\mathbf{G}}_k^2, \cdots, \mathbf{f}_{in}^C \hat{\mathbf{G}}_k^C]). \tag{4}$$

Unfortunately, after thresholding the adjacency matrix, we will face two new problems. The first problem is related to initialization. As we initialize using a pre-defined adjacency matrix of the human skeleton, thresholding may prevent generating new edges. Specifically, the new edges are initialized to a minimum value below the threshold, and the threshold filters them out. Therefore, the gradients are always zeros, and the model can not generate new edges. The second problem is about training. The edges that are deleted by thresholding are permanently deactivated. If the deletions are incorrect, the model can not reactivate these edges in later training. To address those issues, we design a reactivation loss:

$$\mathbf{L}_R = -\alpha \sum_{k,c,i,j} \min(\mathbf{G}_k(c,i,j) - \delta, 0) \tag{5}$$

where α is the coefficient to control the speed of reactivation, blowing the threshold, the deactivated edges slowly grow until it reaches the threshold. Then, the beneficial edges will have a chance to grow continually, and the others will be affected by the gradient and fall below the threshold again. The reactivation loss $\mathbf{L}_R$ is added to our loss function $\mathbf{L}_{total}$ during all of the training periods:

$$\mathbf{L}_{total} = \mathbf{L}_{cross\_entropy} + \mathbf{L}_R. \tag{6}$$

In this manner, we construct the sparse adjacency matrices, which focus on the most valuable relations. Redundant edges that impair model performance are removed in a learnable way, and the necessity of edges to be validated repeatedly throughout the training process.

3.4. Multiscale Temporal Convolution

There is a difference in the speed with which people perform actions. Moreover, the starting frames of each skeleton sequence are not uniform, and their discriminative clues may span a great deal of time, even in different time ranges. The ideal skeleton-based action recognition model needs to be adaptable to these changes in the position of velocity and time.

Therefore, the MTC is proposed. The main idea is to divide the channels into several groups, each corresponding to a scale, and then assemble the information from different scales. As shown in Figure 3, we split the input feature into four groups and feed them into four reciprocal symmetric branches, respectively. Each branch applies 2D convolution with kernel size different from others, from 3×1 to 9×1. The outputs of all branches are concatenated, then combined by a 1×1 convolution. The operation of MTC can be represented as:

$$\mathbf{F}_{out} = \mathbf{W}[Conv2D(\mathbf{F}_{in}^1, k_1), Conv2D(\mathbf{F}_{in}^2, k_2), \cdots, Conv2D(\mathbf{F}_{in}^B, k_B)] \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{C_{out} \times C_{in}}$ is the weight matrix of the 1×1 convolution; $[\cdot]$ denotes the concatenation of tensors; $Conv2D(\cdot)$ means the traditional convolution operation; k_i is the kernel of branch i; and B is the number of branches.

Compared with the fixed 9×1 temporal convolutional kernel size deployed in the baseline (2s-agcn [3]), the proposed MTC has fewer parameters. It can also capture the multiscale temporal representation and enable our model to adapt to different action speeds.

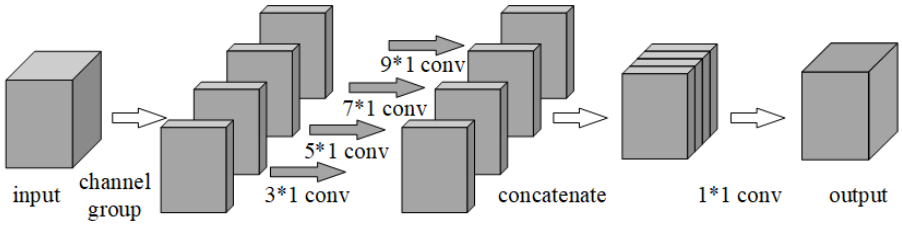

Figure 3. Diagram of the multiscale temporal convolution.

3.5. Spatial-Temporal Attention Module

Since the adjacency matrices are learnable in the spatial dimension, the graph convolution is global. Nonetheless, it generates fixed correlations, which may be unreliable because it can not adjust according to the data. Instead, the self-attention operation can build correlations according to the input, giving it more universality. In the temporal dimension, the convolution operation is still processing a local neighborhood, and the global dependencies of the frames are neglected. Therefore, we propose the symmetry spatial-temporal attention module to model the context relation both in space and time. In the module, the part of spatial attention and temporal attention have the symmetry structure.

We can represent the input feature and output feature as a series of vector: $\mathbf{F}_{in} = \{x_1, x_2, \cdots, x_V\}$, $\mathbf{F}_{out} = \{y_1, y_2, \cdots, y_V\}$, then $x_i \in \mathbb{R}^{C \times T}$, $y_j \in \mathbb{R}^{C \times T}$. The spatial attention unit can be formulated by:

$$y_i = \sum_j softmax(f(\theta(x_i)^T, \phi(x_j)))g(x_j). \tag{8}$$

Here $\theta(x_i) = W_\theta(x_i)$, $\phi(x_j) = W_\phi(x_j)$, and $g(x_j) = W_g(x_j)$ are three linear embedding, f is a pairwise function , which produces the similarity scalar between i and j. We defined the angle-based similarity function f as:

$$f(u, v) = 1/arccos(\frac{uv}{|u| \cdot |v|}). \tag{9}$$

When the angle between u and v is smaller, the two vectors are more correlated. Therefore, we use the reciprocal of the angle computed by the arccos function to measure similarity between vectors. We limit the range of the arccos function to $[-1,1]$.

Likewise, if we unfold the input feature and output feature as: $\mathbf{F}_{in} = \{x_1, x_2, \cdots, x_T\}$, $\mathbf{F}_{out} = \{y_1, y_2, \cdots, y_T\}$, then $x_i \in \mathbb{R}^{C \times V}$, $y_j \in \mathbb{R}^{C \times V}$. The temporal attention unit can also be formulated by Equation (8). Therefore, the temporal attention unit is symmetrical with the spatial attention unit. The temporal attention unit can capture the temporal context, and the spatial attention unit is used to model the spatial context. We first deploy the two units separately to form the temporal attention module (TAM) and the spatial attention module (SAM). Collaborating the two units, we then design two different structures of STAM, namely STAM-A and STAM-B. As shown in Figure 4, the temporal attention unit and spatial attention unit are connected in series or parallel, respectively. Residual connections are added to preserve the local features. A ReLU activation function filters the output. The combination of temporal attention unit and spatial attention unit can establish the temporal and spatial interdependencies in a data-driven manner, which is helpful for skeletal action recognition. Structures of the four attention modules mentioned above are shown in Figure 4. We will compare their performance in Section 4.

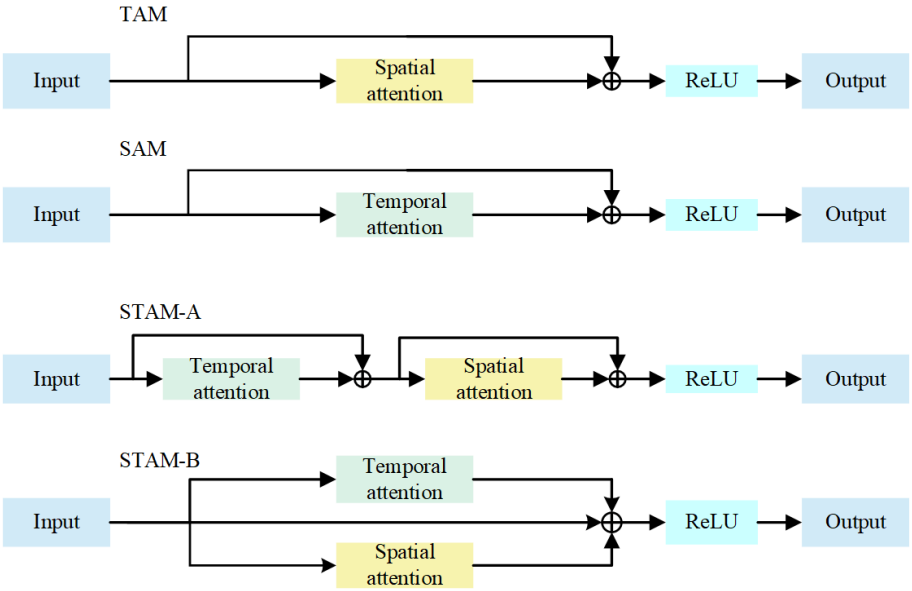

Figure 4. Four different attention modules

3.6. Datasets

We use three public action recognition datasets in our experiments, namely NTU-RGB+D [4], NTU-RGB+D 120 [5], and Northwesten-UCLA [6].

NTU-RGB+D is an indoor action recognition dataset that has been widely used. It has 56,880 action clips in 60 classes. The 3D coordinates of the subject's 25 joints captured by Kinect depth sensors are provided. Each action is performed by 40 subjects. Three cameras capture actions with horizontal angles at -45°, 0° , and 45°. The dataset has two benchmark protocols: (1) Cross-subject (CS): samples are split into two parts, 40,320 samples for the training set and 16,560 samples for the test set, according to the subjects. (2) Cross-view (CV): Samples are split into two parts, 37,920 samples for the training set and 18,960 samples for the test set, according to the camera angle. Following the two protocols, we evaluated our model with the top-1 accuracy on both benchmarks. For each benchmark, the top-1 accuracy is reported.

NTU-RGB+D 120 is an extended version of the NTU RGB+D. Therefore, it is the largest indoor action recognition dataset currently. The dataset contains 114,480 videos in 120 classes. Each action is performed by 106 subjects. The author capture the videos from 155 viewpoints. Similarly, the dataset has two benchmark protocols: (1) Cross-subject (CS): samples are split according to subjects. The training set contains 63,026 samples, while the test set contains 50,922 samples. (2) Cross-setting (CE): Samples are split according to camera setting. The training set contains 54,471 samples, while the test set contains 59,477 samples. Following the two protocols, we evaluated our model with the top-1 accuracy on both benchmarks. For each benchmark, the top-1 accuracy is reported.

Northwesten-UCLA is a multi-modality action recognition dataset, which provides RGB data and depth video data. For each action, the author captures the data from three different viewpoints. Therefore, three Kinect cameras correspond to the viewpoints. The dataset has 1494 video clips in 10 action classes. Each action is performed by 10 actors. The training set contains samples from the first two cameras, and the testing set has the samples from the other camera. In this work, we only use the skeleton data for recognition action and report the top-1 accuracy.

4. Experimental Results

In this section, we introduce the results and details of our experiments in the datasets introduced in Section 3.6. To verify the effectiveness of the components of the RS-GCN, we

first perform exhaustive ablation studies on NTU-RGB+D. Then, we compare the proposed model with other state-of-the-art methods on all three datasets.

4.1. Implementation Details

We conduct all our experiments with the PyTorch deep learning framework. We train our model for 60 epoches totally. We use the stochastic gradient descent (SGD) optimizer with the momentum of 0.9 for training, and the batch size is set to 32. The initial learning rate is set to 0.1 and decays with a factor of 0.1 at the 40th and 50th epoch. We set cross-entropy as the objective function. The weight decay is set to 0.0002. For NTU-RGB+D, and NTU-RGB+D 120, the relation selective threshold delta is set to 0.1, the coefficient alpha is set to 2.2×10^{-5}. For Northwestern-UCLA, the relation selective threshold delta is set to 0.01, and the coefficient alpha is set to 2.1×10^{-5}. We adopt the multi-stream fusion strategy in [27], and we compute the weighted average of each stream's output as the final prediction.

4.2. Ablation Study

This section investigates the contributions of different components in the proposed RS-GCN, the effect of the relation selection mechanism, and the necessity of multi-stream inputs. We conduct experiments on the NTU-RGB+D dataset under the CV benchmark and only the joint-stream as the input.

4.2.1. Network Architectures

We first verify the necessity of the proposed components. We manually delete the proposed component from RS-GCN to form the variants. We compare the performance of those variants and the original RS-GCN on the NTU-RGB+D dataset, and the result is shown in Table 1. This table shows that all STAM, CWG, MTC, and relation selection mechanism benefit action recognition. Deleting any one of the components will harm the performance. The variant without the relation selection mechanism get the lowest accuracy, dropping by 0.6% compared to the original RS-GCN, verifying the effectiveness of the relation selection mechanism. As the components retained in the variants still improve performance, these variants are only marginally less accurate than RS-GCN. By collaborating all those components, RS-GCN gets the best accuracy, and outperforms the baseline [3] by 1.2%.

Table 1. Performance comparison of RS-GCN and its variants. w/o X means the variants deleting the X module.

Method	Accuracy (%)
baseline	93.72
w/o STAM	94.63
w/o CWG	94.46
w/o MTC	94.64
w/o relation selection mechanism	94.33
RS-GCN	94.93

4.2.2. Comparison of Attention Modules

Here we compare the performance of the proposed model with different attention modules in Table 2. As we can see from the table, the TAM and SAM both get a low accuracy. Due to the complementary relationship between temporal attention and spatial attention, STAM-A and STAM-B have a clear improvement compared with the SAM and TAM. The accuracy of STAM-A is slightly higher than that of STAM-B because connecting in a series may be more beneficial for the integration of spatial-temporal information than parallel.

Table 2. Comparison of proposed model with different attention modules. w/X means the model using the X module.

Method	Accuracy (%)
w TAM	94.61
w SAM	94.44
w STMA-A	94.93
w STMA-B	94.69

4.2.3. Necessity of Multi-Stream Inputs

We adopt the multi-stream fusion strategy in [27], which includes four streams. The joint stream's input is the original skeleton coordinates. The bone stream's input is the differential of adjacent joints' coordinates in the skeleton. The joint motion stream's and bone motion stream's inputs are the time differential of the joint stream's input and bone stream's input, respectively. The final output is obtained by the weighted average of each stream's prediction. We tested the performance of four streams and two combinations between them. Here, J, B, J-M, and B-M denote the joint stream, bone stream, joint motion stream, and bone motion stream. Moreover, 2s means the combination of joint stream and bone stream, and 4s means combining all four streams. The result is shown in Table 3. We can find in the table that the combination of the multi-streams method outperforms the single-stream methods.

Table 3. Comparison of the proposed model with different inputs.

Method	Accuracy (%)
J-RS-GCN	94.93
B-RS-GCN	94.79
JM-RS-GCN	92.97
BM-RS-GCN	92.95
2s-RS-GCN	95.92
4s-RS-GCN	96.45

4.2.4. Visualization of Adjacency Matrices

We visualized the adjacent matrix in the baseline and RS-GCN, respectively, and the results are shown in Figure 5. It can be seen that the adjacent matrices in the baseline are excessively dense, which may make redundant and chaotic connections between nodes. It prevents the model from selecting the information effectively. The adjacent matrices in RS-GCN are sparse, and their non-zero values are only distributed on a few crucial connections. The sparsed adjacency matrics allow the node to select the most solid relations and focus on the most informative adjacent nodes. Besides, the adjacency matrix in different channels shows a clear distinction. We can infer that every channel evaluates and selects the connections that are most suitable for its features. It provides enough adaptability for the model's aggregation of features.

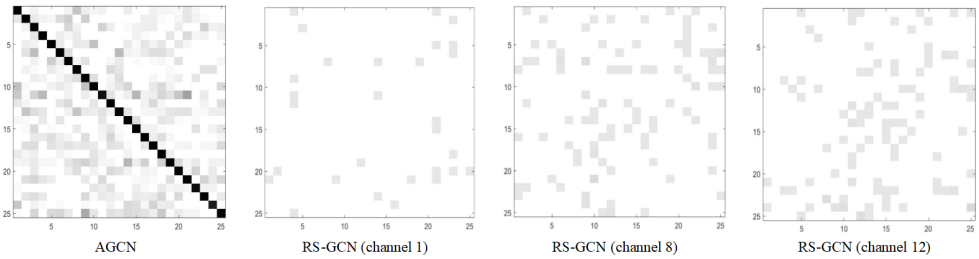

Figure 5. Visualization of adjacency matrices for the baseline (AGCN) and our proposed RS-GCN.

4.3. Comparison with the State-of-the-Art

In this section, we compare the proposed model with the state-of-the-art methods on the NTU-RGB+D dataset, NTU-RGB+D 120, and Northwesten-UCLA dataset. The methods used for comparison mainly include three types: CNN-based method [11,28–31], RNN-based method [4,14,15,32,33], and GCN-based method [2,3,17,27,34,35]. The results are shown in Table 4–6, respectively. Our proposed model outperforms the baseline with a large margin, and it almost achieves the state-of-the-art performance on all three datasets, which verifies the superiority of our model.

Table 4. Comparisons with the state-of-the-art methods on the NTU-RGB+D dataset.

Method	CS (%)	CV (%)
Deep LSTM [4]	60.7	67.3
ST-LSTM [32]	69.2	77.7
Ensemble TS-LSTM [14]	74.6	81.3
VA-LSTM [33]	79.2	87.7
GCA-LSTM [15]	77.1	85.1
TCN [28]	74.3	83.1
Clips + CNN + MTLN [29]	79.6	84.8
Synthesized CNN [30]	80.0	87.2
3scale ResNet 152 [11]	84.6	90.9
HCN [31]	86.5	91.1
SLnL-rFA [36]	89.1	94.9
ST-GCN [2]	81.5	88.3
RA-GCN [17]	85.9	93.5
2s-AGCN (baseline) [3]	88.5	95.1
2s-SDGCN [34]	89.6	95.7
MS-AAGCN [27]	90.0	96.2
BPLHM [18]	85.4	91.1
Pose-refinement GCN [18]	85.2	91.7
2s-FGCN [35]	90.2	96.3
TA-GCN [24]	89.9	96.3
sym-GCN [24]	90.1	96.4
RNXt-GCN [25]	**91.4**	95.8
4s-RS-GCN (ours)	91.3	**96.5**

Table 5. Comparisons with the state-of-the-art methods on the NTU-120 RGB+D dataset.

Method	CS (%)	CE (%)
ST-LSTM [32]	25.5	26.3
GCA-LSTM [15]	61.2	63.3
ST-GCN [2]	72.4	71.3
RA-GCN [17]	81.1	82.7
2s-FGCN [35]	85.4	87.4
RNXt-GCN [25]	83.9	87.6
Tripool [37]	80.1	82.8
4s-RS-GCN (ours)	**87.2**	**88.6**

Table 6. Comparisons with the state-of-the-art methods on the Northwesten-UCLA dataset.

Method	Top1 (%)
HBRNN-L [38]	78.5
Synthesized-pre-trained [30]	86.1
Ensemble-TS-LSTM [14]	89.2
2s-AGC-LSTM	93.3
RNXt-GCN [25]	76.9
4s-RS-GCN (ours)	**94.8**

5. Discussion and Conclusions

In this work, we propose a trainable relation selection mechanism. It helps our model choose the most informative connection of the graph and sparse the adjacency matrices. Therefore, the nodes can focus on the most important neighbors. In addition, we propose channel-wise graph convolution (CWG) and multiscale temporal convolution (MTC) to strengthen the model's representative power. Furthermore, we introduce the spatial-temporal attention module (STAM) to enhance the model's ability to capture context relations. Incorporating these improvements, we built a novel model called relation selective graph convolutional networks (RS-GCN). Comprehensive experiments on three public datasets show our model's overwhelming performance compared to the state-of-the-art approaches, proving the effectiveness of our model. However, there are still some issues that need further investigation. The CWG and MTC also bring some extra computational cost, and when there is much noise in the skeleton sequence, the selection mechanism is prone to remove some desired connections. Therefore, in our future work, we will look for ways to enhance the model's representation ability with less computational consumption and improve our relation selection mechanism to make it more robust to noise.

Author Contributions: Methodology, W.Y.; software, W.Y.; writing—original draft, W.Y.; writing—review and editing, W.Y. and J.Z.; supervision, J.Z., J.C. and Z.X.; funding acquisition, Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Code of RS-GCN: https://github.com/kraus-yang/RS-GCN (accessed on 20 November 2021). Datasets: NTU-RGB+D/NTU-RGB+D: https://github.com/shahroudy/NTURGB-D(accessed on 17 April 2021); Northwesten-UCLA: https://wangjiangb.github.io/my_data.html (accessed on 30 April 2021).

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

References

1. Palestra, G.; Rebiai, M.; Courtial, E.; Koutsouris, D. Evaluation of a Rehabilitation System for the Elderly in a Day Care Center. *Information* **2019**, *10*, 3. [CrossRef]
2. Yan, S.; Xiong, Y.; Lin, D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018 Volume 32.
3. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; IEEE: Long Beach, CA, USA, 2019; pp. 12018–12027. [CrossRef]
4. 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 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1010–1019.
5. Liu, J.; Shahroudy, A.; Perez, M.; Wang, G.; Duan, L.Y.; Kot, A.C. NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2684–2701.

6. Wang, J.; Nie, X.; Xia, Y.; Wu, Y.; Zhu, S.C. Cross-View Action Modeling, Learning, and Recognition. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 2649–2656.
7. Xia, L.; Chen, C.C.; Aggarwal, J.K. View invariant human action recognition using histograms of 3D joints. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; pp. 20–27. [CrossRef]
8. Gowayyed, M.A.; Torki, M.; Hussein, M.E.; El-Saban, M. Histogram of Oriented Displacements (HOD): Describing Trajectories of Human Joints for Action Recognition. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; p. 8.
9. Vemulapalli, R.; Chellappa, R. Rolling Rotations for Recognizing Human Actions from 3D Skeletal Data. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4471–4479. [CrossRef]
10. Du, Y.; Fu, Y.; Wang, L. Skeleton based action recognition with convolutional neural network. In Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 579–583. [CrossRef]
11. 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 2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW), Hong Kong, China, 10–14 July 2017; pp. 601–604. [CrossRef]
12. Rahmani, H.; Bennamoun, M. Learning Action Recognition Model from Depth and Skeleton Videos. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5833–5842. [CrossRef]
13. Caetano, C.; Brémond, F.; Schwartz, W.R. Skeleton Image Representation for 3D Action Recognition Based on Tree Structure and Reference Joints. In Proceedings of the 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Rio de Janeiro, Brazil, 28–30 October 2019; pp. 16–23.
14. Lee, I.; Kim, D.; Kang, S.; Lee, S. Ensemble Deep Learning for Skeleton-Based Action Recognition Using Temporal Sliding LSTM Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1012–1020. [CrossRef]
15. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3671–3680.
16. Liu, H.; Tu, J.; Liu, M.; Ding, R. Learning Explicit Shape and Motion Evolution Maps for Skeleton-Based Human Action Recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 1333–1337. [CrossRef]
17. Song, Y.F.; Zhang, Z.; Wang, L. Richly Activated Graph Convolutional Network for Action Recognition with Incomplete Skeletons. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1–5. [CrossRef]
18. Zhang, X.; Xu, C.; Tian, X.; Tao, D. Graph Edge Convolutional Neural Networks for Skeleton-Based Action Recognition. *IEEE Trans. Neural Networks Learn. Syst.* **2020**, *31*, 3047–3060.
19. Zhu, G.; Zhang, L.; Li, H.; Shen, P.; Shah, S.A.A.; Bennamoun, M. Topology-learnable graph convolution for skeleton-based action recognition. *Pattern Recognit. Lett.* **2020**, *135*, 286–292. [CrossRef]
20. Huang, L.; Huang, Y.; Ouyang, W.; Wang, L. Part-Level Graph Convolutional Network for Skeleton-Based Action Recognition. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 11045–11052. Number: 07. [CrossRef]
21. Huang, Z.; Shen, X.; Tian, X.; Li, H.; Huang, J.; Hua, X.S. Spatio-Temporal Inception Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 26–28 May 2020; pp. 2122–2130. [CrossRef]
22. Li, S.; Yi, J.; Farha, Y.A.; Gall, J. Pose Refinement Graph Convolutional Network for Skeleton-based Action Recognition. *arXiv* **2020**, arXiv:2010.07367.
23. Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; Tian, Q. Symbiotic Graph Neural Networks for 3D Skeleton-based Human Action Recognition and Motion Prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]
24. Li, X.; Zhai, W.; Cao, Y. A tri-attention enhanced graph convolutional network for skeleton-based action recognition. *IET Comput. Vis.* **2021**, *15*, 110–121. [CrossRef]
25. Liu, S.; Bai, X.; Fang, M.; Li, L.; Hung, C.C. Mixed graph convolution and residual transformation network for skeleton-based action recognition. *Appl. Intell.* **2021**, 1–12.
26. Peng, W.; Shi, J.; Zhao, G. Spatial Temporal Graph Deconvolutional Network for Skeleton-Based Human Action Recognition. *IEEE Signal Process. Lett.* **2021**, *28*, 244–248.
27. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Skeleton-Based Action Recognition With Directed Graph Neural Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7904–7913. [CrossRef]
28. 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; pp. 1623–1631. [CrossRef]

29. Ke, Q.; Bennamoun, M.; An, S.; Sohel, F.; Boussaid, F. A New Representation of Skeleton Sequences for 3D Action Recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4570–4579. [CrossRef]
30. Liu, M.; Liu, H.; Chen, C. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognit.* **2017**, *68*, 346–362. [CrossRef]
31. 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, Stockholm, Sweden, 13–19 July 2018; pp. 786–792. [CrossRef]
32. 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.
33. 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]
34. Wu, C.; Wu, X.J.; Kittler, J. Spatial Residual Layer and Dense Connection Block Enhanced Spatial Temporal Graph Convolutional Network for Skeleton-Based Action Recognition. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 27 October–2 November 2019; pp. 1740–1748. [CrossRef]
35. Yang, H.; Yan, D.; Zhang, L.; Li, D.; Sun, Y.; You, S.; Maybank, S.J. Feedback Graph Convolutional Network for Skeleton-based Action Recognition. *arXiv* **2020**, arXiv:2003.07564.
36. Hu, G.; Cui, B.; Yu, S. Skeleton-Based Action Recognition with Synchronous Local and Non-Local Spatio-Temporal Learning and Frequency Attention. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 1216–1221. [CrossRef]
37. Peng, W.; Hong, X.; Zhao, G. Tripool: Graph triplet pooling for 3D skeleton-based action recognition. *Pattern Recognit.* **2021**, *115*, 107921. [CrossRef]
38. Du, Y.; Wang, W.; Wang, L. Hierarchical recurrent neural network for skeleton based action recognition. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1110–1118. [CrossRef]