Advancing Multi-actor Graph Convolutions for Skeleton-Based Action Recognition

Yiqun Zhang[1(✉)], Zhenyu Qin[2], Yang Liu[2], Tom Gedeon[3], and Wu Song[4]

[1] Australian National University, Canberra, Australia
admin@yiqun.io
[2] Seeing Machines, Canberra, Australia
[3] Curtin University, Perth, Australia
Tom.Gedeon@curtin.edu.au
[4] Rongcheng Cloud-Intelligence Co., Ltd., Rongcheng, China

Abstract. Human skeleton motion recognition, notable for its lightweight, interference-resistant, and resource-saving properties, plays a crucial role in human motion recognition and has found widespread applications. The common approach to capture motion features from human skeleton videos involves extracting skeleton features temporally or spatially using Graph Convolution Networks (GCN) or their improved variants. Nevertheless, existing extraction methods encounter two primary limitations: variability in the number of actors involved in an action and disconnected subgraphs representing multiple actors' actions, resulting in a loss of inter-subgraph features. To overcome these challenges, we propose Human Mirror and Human Link strategies, which replicate diverse human data to fill and interlink multiple subgraphs. Empirically, our proposed methods applied to the NTU RGB+D 120 dataset significantly enhanced the performance of the base model MSG3D, demonstrating the effectiveness of our approach in handling multi-actor scenarios.

Keywords: Skeleton-Based Action Recognition · Graph Convolution Networks · Human Link · Human Mirror · Multi-Actor Interaction · Subgraph Unification

1 Introduction

It is a long-standing problem of computer vision to accurately and promptly recognize human actions [1]. Advances of this problem can lead to improvements in many applications such as human-robot interaction [2], sports analysis [3], and smart health-care services [4]. We study skeleton-based action recognition, a subfield of action recognition emerged recently thanks to the ready availability of human pose estimation algorithms and devices [5–7]. Currently, skeleton-based action recognition is swiftly accumulating attention due to numerous advantages

M. Clayton et al. (Eds.): INTETAIN 2023, LNICST 560, pp. 76–95, 2024.
https://doi.org/10.1007/978-3-031-55722-4_7

of skeletonized human representations such as being concise and free from environmental noises [7]. Taking conciseness as an example, it consumes more than 1 GB to store a minute's RGB videos, whereas it merely costs 100 KB to store the skeleton sequence of the same length according to our empirical studies.

Many skeleton sequences involve more than one participant for displaying interactive actions such as *shaking hands* and *punching others* [8]. When tackling these multi-participant skeletonized representations, a common algorithmic pattern is described as follows [9–12]. (1) Each person's skeleton representation is separately processed to extract spatial and temporal patterns; (2) the extracted features of different participants are fused (such as addition or concatenation) to form the overall representation of the action; (3) the fused feature is further transformed to recognize the interactive action. That is, the action's interactive clues are not taken into consideration until the end of the processing pipeline. On the other hand, interactions between the participants contain abundant informative cues for accurately classifying the actions.

Recognizing human actions, especially involving multiple participants, remains a challenging endeavor in computer vision. While skeleton-based action recognition offers a streamlined approach, many algorithms overlook the rich interplay between participants. This oversight can lead to missed nuances crucial for accurate categorization. Our work introduces mechanisms to bridge this gap, harnessing the potential of interpersonal relationships.

We aim to facilitate existing skeleton-based action recognizers to explicitly exploit the interactive relationships between participants. To this end, we propose the **human link** mechanism. Specifically, instead of processing each person's skeleton individually, we create a giant skeleton containing multiple participants' skeletons. Within this giant skeleton, we link together the joints that correspond to the same body part (such as head and left/right hand). These additional links explicitly enable skeleton-based action recognizers to capture the interactive patterns between participants.

To extend human link to be also applicable to the actions that involve only a single participant, we further propose the **human mirror** mechanism. Specifically, human mirror creates a copy of the person's skeletonized representation and treats the copied skeleton as performing the interactive action with the original skeleton. This newly generated skeleton copy grants the feasibility of human link. Additionally, our experimental results also reveal accuracy improvement when merely implementing human mirror to skeleton-based action recognizers. Our further studies imply that the enhanced performance is due to the reduction of distribution shift between the input skeletons of single and multiple participants.

We summarize our contributions as follows:

1. We propose the **human link** method that facilitates existing skeleton-based action recognizers to explicitly capture the interactive relationships between participants as cues for more accurate action recognition.
2. We present the **human mirror** approach that can not only improve action recognition accuracy for single-participant actions, but also enables human link to be applicable for the non-interactive actions.

3. Both the proposed human link and human mirror are widely compatible with existing skeleton-based action recognizers. Human mirror can independently enhance a recognizer's accuracy. Moreover, applying the two methods in conjunction can further improve the performance. We experimentally demonstrate that both human link and mirror consistently boost the accuracy of numerous recent models.
4. Equipped with the conjunction of human link and mirror, a simple model is competitive with more complex models in accuracy and consumes substantially fewer parameters and less inference time. When combining the two components with a more complicated network, we achieve new state-of-the-art accuracy.

After this introduction, we delve into the relevant literature, discussing *Skeleton-Based Action Recognition*, *Neural Nets on Graphs*, and *Multi-Scale Graph Convolutions* in Sect. 2. Section 3 introduces our proposed methods, where we detail the notations used and provide in-depth discussions on **Human Mirror** and **Human Link**. In Sect. 4, we detail our experimental setups, describe the datasets, preprocessing techniques, and present a thorough evaluation of our methods, including component studies to dissect their individual contributions. We conclude the paper in Sect. 5 with our concluding remarks and directions for future work.

2 Related Work

2.1 Skeleton-Based Action Recognition

In the early time, the features of the entire sequence were encoded into a metric space for skeleton-based action recognition [13]. Since these pioneering models did not explicitly leverage the co-occurrences between frames, they insufficiently extract patterns along the action trajectories, producing low recognizing accuracy. Subsequently, convolutional neural networks (CNNs) were revived and soon applied to skeleton-based action recognition. CNNs bolster capturing dynamical clues by convolving along the temporal direction, prompting higher accuracy for recognizing skeleton-based actions [6,14,15]. However, CNNs are not designed to specifically model a skeleton's structural information, thus leaving rich topological features of a skeleton under-exploited.

Graph convolutional networks (GCNs) were later introduced to model the internal topological interconnections by representing a skeleton as a graph, where nodes and edges represent joints and bones respectively. Spatial Temporal Graph Convolutional Networks (ST-GCN) integrated hierarchical spatial features with multiple graph convolutional layers and obtained temporal patterns with convolution [9]. Then, Li et al. designed AS-GCN [16] to supplant the fixed skeleton graph with a learnable adjacency matrix, improving the flexibility of extracting structural information and delivering a boost to the recognition accuracy. In later work, Si et al. proposed An Attention Enhanced Graph Convolutional LSTM (AGC-LSTM) to aggregate graph convolutional operations into Long

Short Term Memory (LSTM)'s internal gate operations for extracting long-range temporal cues of the skeleton motion trajectories [17]. Subsequently, the Two-Stream Adaptive Graph Convolutional Networks (2 s-AGCN) model additionally utilized bone features apart from joint features and was equipped with learnable residual masks to improve the flexibility of aggregating topological features [18]. Recently, many novel techniques, such as graph-based dropout [10], 3D graph convolutions [19], and shift-convolutions [7], have been incorporated into graph networks for skeleton-based action recognition.

2.2 Neural Nets on Graphs

In order to extract information and features from graph structure, many studies have explored graph structure through neural networks [20–23]. With the introduction of Graph Neural Networks (GNNs) [24,25], researchers have applied the convolution method of pictures to graph structures, and the extraction efficiency of individual graph structures has been greatly increased. GNNs incorporate the graph structure into a neural network, allowing the network to process and exchange information between nodes based on the graph topology to accomplish information fusion and information extraction. The nodes in the graph are updated layer by layer through a multilayer GNN network. Each node will fuse and exchange its own data with neighboring nodes after convolution operation, so that different nodes can obtain global information to a little extent. The GNN network is then pooled by an aggregation function such as Average Pooling, and finally by an activation function, which greatly facilitates the study of graph structure.

2.3 Multi-scale Graph Convolutions

However, the skeleton structure is a more complex graph structure, the skeleton structure is generally larger and longer distance, the general way of exchanging information with neighboring nodes can not meet the needs of the skeleton graph structure. At the same time, human action is an overall behavior, local information exchange and feature fusion are not enough to judge human action. Therefore, in order to meet the needs of human skeleton action recognition, researchers have proposed many methods to solve this problem [19,26,27]. Firstly, we can cross the simple neighbor relationship to exchange the data with the nodes at a longer distance, and we can capture the information at a longer distance through a higher order polynomial of the adjacency matrix, and give the larger weight to the nodes at a closer distance. In this way, we can learn the information at a closer distance as well as at a longer distance, and fuse the features of the whole human skeleton graph structure.

3 Method

3.1 Baseline Model

The foundation of our work is built upon the MS-G3D model introduced in [19]. MS-G3D stands out for its ability to effectively handle spatial-temporal graphs in skeleton-based action recognition, particularly excelling in unbiased long-range joint relationship modeling under multi-scale operators and ensuring seamless cross-spacetime information flow. The modifications and enhancements we propose in the subsequent sections are specifically tailored for this baseline. Apart from the alterations and improvements we introduce, all other components and configurations of the model remain consistent with their original design as delineated in the baseline. This approach ensures a transparent assessment, enabling any observed performance variation to be directly linked to our introduced methods, facilitating a precise comparison between the original MS-G3D and its modified counterpart.

3.2 Notations

We use graphs to represent the human skeleton. The graph is denoted as $\mathcal{G}$. There are a set of vertices $\mathcal{V}$, a set of edges $\mathcal{E}$. The set of vertices, $\mathcal{V}$, represents the various joints in the human skeleton, such as the elbows, knees, hips, and so on. Each vertex, therefore, corresponds to a specific joint location at a given frame in the video or motion capture data. The set of edges, $\mathcal{E}$, captures the anatomical relationships between these joints. Each edge in $\mathcal{E}$ denotes a physical connection or bond between two joints, resembling the skeletal structure of the human body. For instance, an edge might exist between the shoulder and elbow joints, representing the upper arm bone. It is worth noting that the configuration of $\mathcal{E}$ is based on the natural topology of the human skeleton, ensuring that our graph representation, $\mathcal{G}$, is anatomically consistent. So we know:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$

$\mathcal{V}$ is a vertices set. There are N vertices in $\mathcal{V}$. We use v_i to represent the i-th vertex. So we know:

$$\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$$

$\mathcal{E}$ is a edge set. We use adjacency matrices $\mathbf{A}$ to represent these edges. This matrix is $N \times N$, so we know $\mathbf{A} \in \mathbb{R}^{N \times N}$. We use $a_{i,j}$ to represent the elements of row i and column j in the adjacency matrix. In addition, $a_{i,j}$ indicates the connection state between v_i and v_j. If $a_{i,j}$ is 1, it indicates that there is a connection between v_i and v_j in $\mathcal{G}$; if $a_{i,j}$ is 0, it indicates that there is no connection.

In the graph, there are some vertices around any vertices v_i, which we call neighbors whose set is denoted by $\mathcal{N}(v_i)$. We use $\mathcal{N}_1(v_i)$ to represent the first-order neighbor, $\mathcal{N}_1(v_i)$, $\mathcal{N}_1(v_i) = \{v_j | a_{i,j} \neq 0\}$. For k-order neighbor $\mathcal{N}_k(v_i)$, we

can know $\mathcal{N}_k(v_i) = \{v_j | \exists v_h \in \mathcal{N}_{k-1}(v_i), a_{h,j} \neq 0 \ and \ v_j \notin \mathcal{N}_1(v_i) \cup \mathcal{N}_2(v_i) \cup \cdots \cup \mathcal{N}_{k-1}(v_i)\}$.

$\mathcal{X}$ is a feature set of a action. We use a tensor $\mathbf{X}$ to represent it. Each action consists of T frames. Each frame consists of M human skeleton graphs. Each graph has V vertices and each vertice has C features. This tensor is $T \times M \times V \times C$, so we know

$$\mathbf{X} \in \mathbb{R}^{T \times M \times V \times C}$$

The i-th frame in an action is represented by f_i, and the i-th graph in a frame is represented by g_i. x_i represents the set of features of f_i. So we know $f_i \in \mathbb{R}^{M \times V \times C}$. $x_{i,j}$ represents the set of features of g_j in t_i. So we know $x_{i,j} \in \mathbb{R}^{V \times C}$. $x_{i,j,k}$ represents the set of features of v_k in g_j and t_i. So we know $x_{i,j,k} \in \mathbb{R}^C$.

3.3 Human Mirror

In the skeleton data, for different actions, the number of people completing the action is also different. Some actions can be completed by only one person. Such as *Eat Meal*, *Brushing Teeth* and *Brushing Hair*. So for any frame f_i, $x_{i,0}$ represents the data of the first actor and $x_{i,1}$ represents the data of the second actor. In this action, $x_{i,1}$ is all 0. For this type of data, we call it *Single Person Action*.

But some actions need two people to complete. Such as *Hugging Other Person*, *Handshaking* and *Walking Towards Each Other*. In such data, both $x_{i,0}$ and $x_{i,1}$ are not all 0. For this type of data, we call it *Double Person Action*.

This way different data types are involved in the neural network as training data. For *Single Person Action*, the neural network only needs to learn the features of one person in space. But for *Double Person Action*, the neural network not only needs to learn the features of each person in space, but also needs to learn the relationship between two people. For *Single Person Action*, $x_{i,1}$ is blank, so the neural network will receive the influence of these data and it is difficult to balance between the two data.

To solve this problem, we use **Human Mirror** to solve the impact of different types of data on neural networks. **Human Mirror** is a processing method for *Single Person Action*, which aims to eliminate the unbalanced gap between *Single Person Action* and *Double Person Action*. **Human Mirror** is to check whether the action is *Single Person Action* by $x_{i,1}$. If $x_{i,1}$ is blank, we copy the action of the $x_{i,0}$ to $x_{i,1}$. In this way, both $x_{i,0}$ and $x_{i,1}$ are not blank. The neural network surface will train two different data in the same way to avoid the negative impact of blank data.

For a tensor $\mathbf{T}$, we define a function $B(\mathbf{T})$. This function checks if all the values in the tensor $\mathbf{T}$ are 0, if they are 0, then the result is 0, otherwise the result is 1.

$$B(\mathbf{T}) = \begin{cases} 0 & all \ values \ are \ 0 \\ 1 & otherwise \end{cases} \tag{1}$$

Based on this function, we can calculate if a graph is empty. In this way, we can determine whether a action is *Single Person Action* or *Double Person Action*. We calculate $B(x_{i,1})$, if the result is 0, we can know this action is *Single Person Action*, otherwise *Double Person Action*. We check each $B(x_{i,1}), 1 <= i <= N$. When $B(x_{i,1}) = 0$, we assign a new value to $B(x_{i,1})$ so that let $B(x_{i,1}) := B(x_{i,0})$, otherwise it remains unchanged (Fig. 1).

$$B(x_{i,1}) = \begin{cases} B(x_{i,1}) & B(x_{i,1}) = 1 \\ B(x_{i,0}) & B(x_{i,1}) = 0 \end{cases} \tag{2}$$

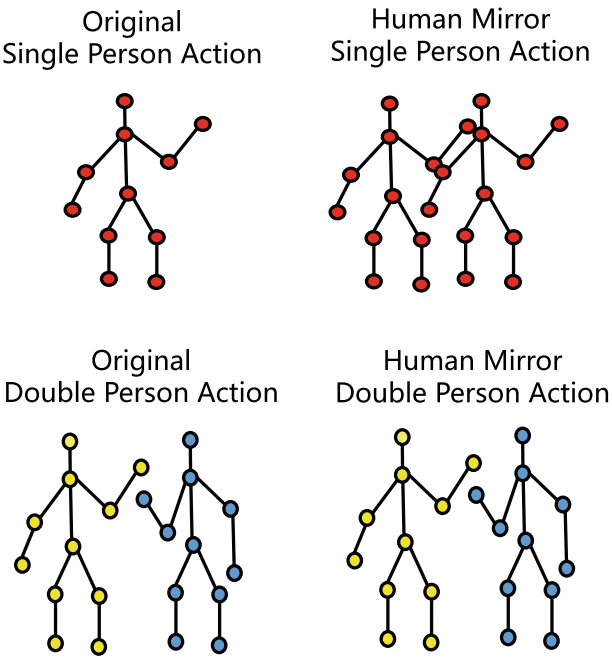


Fig. 1. For *Single Person Action* and *Double Person Action*, Skeleton without **Human Mirror** and with **Human Mirror**

3.4 Human Link

In each frame f_i, g_0 and g_1 are two independent subgraphs without connection. In the traditional strategy [17–19,28], two subgraphs are input into neural network independently, and the temporal and spatial features of the graph are extracted through GCN [29] and other networks. In this way, $x_{i,0}$ and $x_{i,1}$ obtain a feature respectively. Traditional strategy takes the average value of them, and then input the results into the classification neural network for classification.

In this strategy, although the neural network can extract enough features from spatial and temporal scales, the features cannot be exchanged between two individuals. Some actions are performed by two people together, so fusing the features of two people can help the neural network to learn the features better. Also, averaging the features of $x_{i,0}$ and $x_{i,1}$ causes loss of information, which is not a good feature fusion method. These two problems of traditional methods result in inadequate extraction of human features between two individuals.

To solve this problem, we use **Human Link**. **Human Link** optimizes the graph structure by stitching two human skeleton graphs together to form a large human skeleton graph. In this way, two subgraphs consisting of N nodes each are transformed into a super human graph with $2N$ nodes. By connecting two characters together through edges, the GCN [29] network can fuse and extract the features of different characters through the edges, solving the drawback that two people cannot communicate with each other. In addition, by fusing two subgraphs into one, the extracted features are the whole contents of the graph, which avoids the information loss caused by averaging.

The way we connect two people is to connect the corresponding points of two graphs. There are graph g_1 and g_2 in each frame f_i, we replace them with a larger graph g'. To represent this new graph, we need to use the adjacency matrix $\mathbf{A}'$. The larger graph is a splice of g_1 and g_2, so it has $2N$ vertices. So we know

$$\mathbf{A}' \in \mathbb{R}^{2N \times 2N}$$

For g_1 and g_2, their adjacency matrices are $\mathbb{A}_1$ and $\mathbb{A}_2$. In addition, we define the identity matrix $\mathbf{I}_N \in \mathbb{R}^{N \times N}$. So we know:

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A}_1 & \mathbf{I}_N \\ \mathbf{I}_N & \mathbf{A}_2 \end{bmatrix} \tag{3}$$

For the large graph g', tensor $\mathbf{X}'$ is its feature tensor. There is only 1 human large skeleton graph in each frame. So this tensor is $T \times 2V \times C$, so we know

$$\mathbf{X}' \in \mathbb{R}^{T \times 2V \times C}$$

We use the larger graph g' obtained by **Human Link** to input it into the neural network for learning. The neural network can learn not only the temporal and spatial features, but also the relationship features between two people, and can input it into the classifier to get the results without taking the average value (Fig. 2 and 3).

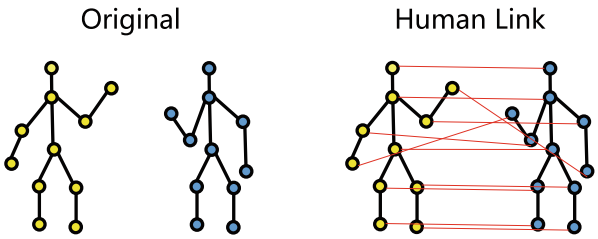


Fig. 2. Skeleton without **Human Link** and with **Human Link**

4 Experiments

4.1 Datasets

NTU RGB+D 60 and NTU RGB+D 120
NTU RGB+D 60 [30] is the skeleton action recognition data set, with a total of 60 different actions (such as drinking water, brushing hair, drop, etc.). The movements were performed by 40 actors between the ages of 10 and 35, with a total of 56880 movement samples recorded. Dataset captured by Microsoft Kinect 2.0 camera, each with RGB video, depth map sequences, 3D skeleton data and infrared video. In order to obtain more adequate data, three different cameras were used for the dataset, all three with the same height and horizontal angles of 0, -45 and $+45°$, and each actor had to perform the action twice (once facing the left camera, then once facing the right camera). In addition, our cameras will have different setups, which differ in terms of height and distance from the actor. The dataset use 17 different setups depending on the height and distance.

We used the 3D skeleton data from the dataset, the node information of the skeleton was obtained by the skeleton tracking technique in the Kinect camera, 25 key nodes were recorded on each human body, and these data could determine the 3D coordinates of human hands, head, body and legs in space for each frame. Depending on the action, each action is performed by 1–2 people (50 kinds of movements for a single person to complete, 10 kinds of movements by a pair to complete). Therefore, each frame contains 25 or 50 nodes, and an action consists of many frames, which are combined together in a temporal order to represent an action (Fig. 4).

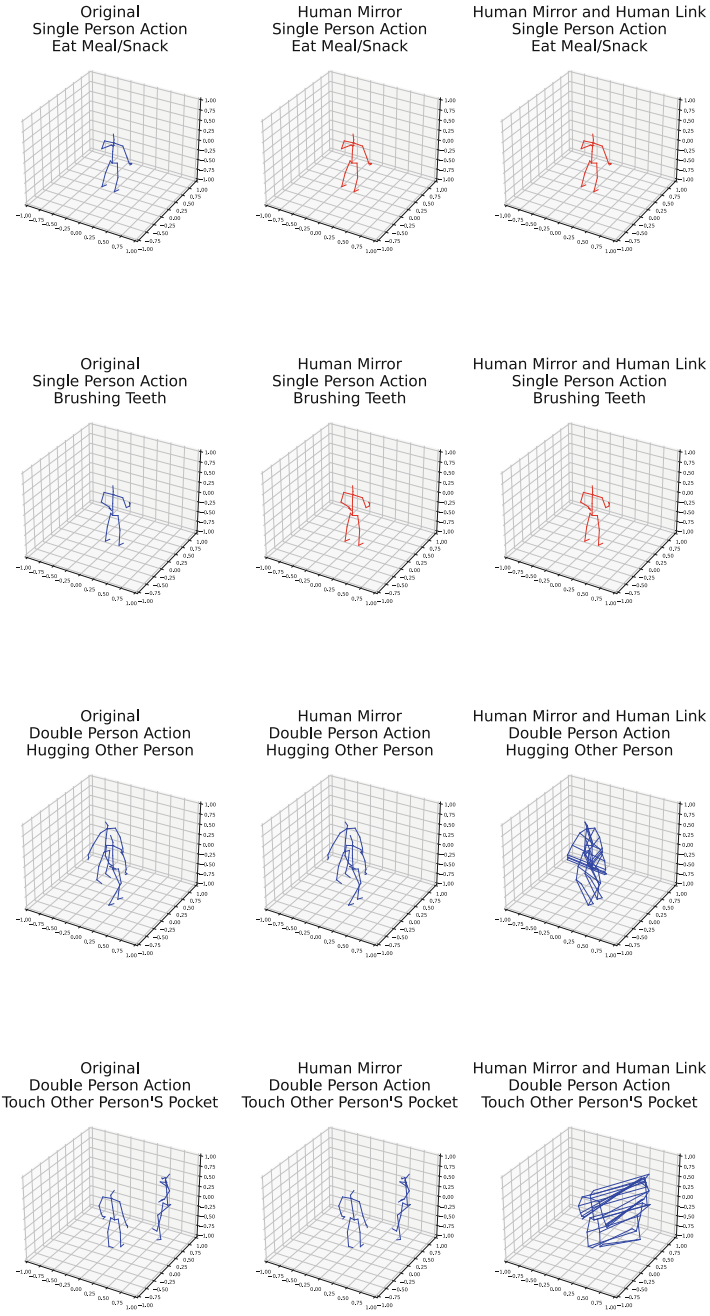


Fig. 3. Comparison figure of different action that original, with **Human Mirror** and **Human Link**. A row represents an action. The left column is without **Human Mirror**, the middle column is with **Human Mirror** and the right column is with **Human Mirror** and **Human Link**. Red represents the coincidence of two blue skeletons. (Color figure online)

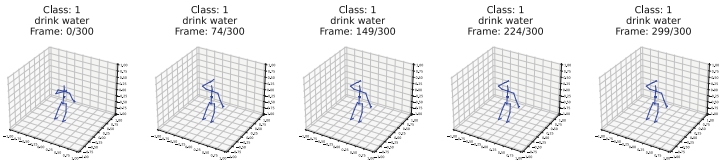


Fig. 4. A skeleton action

In our experiments, we need to divide the data set into a training set and a test set, and there are two ways to do this in NTU60.

- **Cross-Subject.** The actions in the data set are completed by 40 people, we take the actions completed by 20 people as the training set (id is 1, 2, 4, 5, 8, 9, 13–19, 25, 27, 28, 31, 34, 35 and 38), and the rest are divided into the test set. In this way, we will get a training set of 40320 samples and a test set of 16560 samples.
- **Cross-View.** Take the samples collected by camera 1 as the test set, and the samples collected by camera 2 and camera 3 as the training set. So, we will get a training set of 37920 samples and a test set of 18960 samples.

NTU RGB+D 120 [8] continues the style of NTU RGB+D 60 and is the extended content of NTU RGB+D 60. The data set is expanded from 56880 action samples of 60 different actions to 114480 action samples of 120 different actions, from 40 actors to 106 actors, and the camera setting is also increased to 32 kinds. The division of training set and test set is also different.

- **Cross-Subject.** This method continues the method of NTU RGB+D 60, which divides the actions completed by 53 actors into training sets and the rest into test sets. In this way, we will get a training set of 63026 samples and a test set of 59477 50919.
- **Cross-Setup.** We use this method to replace the cross view method used in NTU RGB+D 60. We select 16 camera settings as the training set and the rest as the test set In this way, we will get a training set of 54468 samples and a test set of 59477 samples.

Because NTU RGB+D 120 has more samples and more comprehensive acquisition parameters, the stability of experiments using it is much higher than that of NTU RGB+D 60, and more rigorous and comprehensive results can be obtained, so we will use NTU RGB+D 120 as the data set in our experiments.

4.2 Dataset Setup

- **Dataset Preprocessing.** The data in the dataset needs to be pre-processed in order to be fed into the training program more efficiently. The length of the actions in the dataset is not the same, and in order to be able to input to the neural network for training, we select the first 300 frames for training.

For skeleton sequences beyond 300 frames, we select the first 300 frames and discard the content after that. And for skeleton sequences with less than 300 frames, we use 0 to make up these frames. Making up and discarding are the standard practice in the field. Among the 300 frames selected, one keyframe is selected every three frames, so that we get 100 keyframes. We use these keyframes for training, which greatly accelerates the training speed and memory requirements while extracting sufficient information.
- **Dataset Setup.** In order to experiment, this program adopts the **Cross-Set** setting, in which 54468 data of 16 camera settings are used for training and 59477 data are used for testing. We will get a training set of 37920 samples and a test set of 18960 samples.

4.3 Training and Test Setup

- **Batch Size:** 128
- **Initial Learning Rate:** 0.05
- **Learning Rate Strategy:** Reduce the learning rate to $\frac{1}{10}$ of the original in the 28th, 36th, 44th and 52nd epoch.
- **Epoch:** 60
- **Optimizer:** Adam [31]

4.4 Data Format

For a skeleton action data, we use a graph $\mathcal{G}$ and a feature set $\mathcal{X}$ to represent each action. For all actions, their graph $\mathcal{G}$ are the same, we use adjacency matrix to represent it, which describes the graph structure of human body. For each different action, their feature set $\mathcal{X}$ is different. We use a tensor $\mathbf{X}$ to describe its feature set $\mathcal{X}$. For a skeleton action, it is a graph video, so there are many frames in it, so each frame is represented by a sub-tensor.

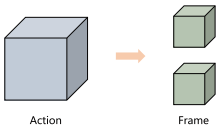
For each frame, there are at most 2 people to complete an action, so we need two graphs to represent the action of each frame. For the feature of each graph, we use a sub-tensor.

For each graph, there are 25 vertices, so we need 25 sub-tensor to represent the vertices of each graph.

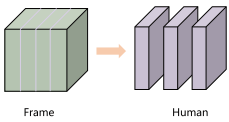
For each vertex, there are 3 features, each feature represents the coordinates of vertices in space. So we need 3 real numbers to represent the features of each vertex.

So, for an action tensor $\mathbf{X}$, it has four dimensions, namely, the number of frames (T), the number of people (M), vertices (V) and features (C). So we know (Fig. 5):

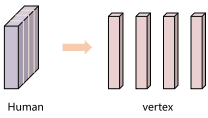
$$\mathbf{X} \in \mathbb{R}^{T \times M \times V \times C}$$



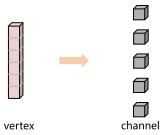
(a) An action consists of many frames, each of which is a sub tensor.



(b) A frame consists of many graph, each of which is a sub tensor.



(c) An graph consists of many vertices, each of which is a sub tensor.



(d) An vertex consists of many features, each of which is a real number.

Fig. 5. .

4.5 Data Preprocessing

In order to improve the training efficiency, [32] method is used to preprocess the data. After pretreatment, the method can save training time and GPU memory grately while keeping the accuracy almost unchanged.

Random Choose. By selectively reducing the frame count, the computational load diminishes, subsequently improving training efficiency. Such reduction, while ensuring that the major patterns are retained, ensures that the model doesn't get overburdened with excessive information. Assume there are T frames in a skeleton action feature tensor x, so we know:

$$x = \{f_1, f_2, \ldots, f_T\}$$

First we have to get the random integer T', which is $0.5T <= T' <= T$. We get a new skeleton action feature tensor x' that x' is the first T' frames of x.

$$x' = \{f_1, f_2, \ldots, f_{T'}\}$$

Then we get a integer T'' and $T'' = \lceil 0.2T \rceil$. We use interpolate to map T' frames in x' to T'' frames:

$$\{f'_1, f'_2, \ldots, f'_{T''}\} = interpolate(\{f_1, f_2, \ldots, f_{T'}\})$$

So we get x'' and

$$x'' = \{f'_1, f'_2, \ldots, f'_{T''}\}$$

We use x'' instead of x as the feature tensor.

Rotation. The introduction of random rotations seeks to enhance the model's generalization capabilities. Real-world data often comes with variability in orientations. By training the model with data in diverse angles, we aim to make the model more robust and adaptive to varied real-world scenarios, thereby improving its predictive capabilities across multiple angles. We randomly rotate all skeletons by an angle θ.

We first get three random values x, y, z while $-0.3 < x, y, z < 0.3$. This represents the rotation angle of the three dimensions. So we can get tensor $\mathbf{r}_x$, $\mathbf{r}_y$, $\mathbf{r}_z$:

$$\mathbf{r}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(x) & sin(x) \\ 0 & -sin(x) & cos(x) \end{bmatrix} \tag{4}$$

$$\mathbf{r}_y = \begin{bmatrix} cos(y) & 0 & -sin(y) \\ 0 & 1 & 0 \\ sin(y) & 0 & cos(x) \end{bmatrix} \tag{5}$$

$$\mathbf{r}_z = \begin{bmatrix} 1 & sin(z) & 0 \\ -sin(z) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

We can get tensor $\mathbf{r}$.

$$\mathbf{r} = \mathbf{r}_x \times \mathbf{r}_y \times \mathbf{r}_z$$

For a frame i, a graph j, we have $x_{i,j}$. For rotation, we use $x'_{i,j}$ instead of $x_{i,j}$:

$$x'_{i,j} = \mathbf{r} \times x_{i,j}$$

Result. We can see the comparison results with and without data preprocessing in Table 1.

Table 1. Experimental results of training time and GPU memory on Dual NVIDIA RTX 3090 while keeping the accuracy almost unchanged.

Use data preprocessing	Batch Size	GPU Memory	Training Time per Epoch
No	32	45 GiB	24 min 30 s
Yes	32	13 GiB	5 min 40 s
Yes	128	44 GiB	5 min 20 s

4.6 MS-G3D

The method we proposed has the goal to improve the relationship between human graph in the human skeleton based human action recognition, so we need a model to experiment. MS-G3D [19] is a skeleton based human action recognition model.

The most important thing of the human skeleton graph is to fuse the skeleton features, so that the neural network can make classification according to the overall features. In the traditional human skeleton action recognition model, there are two problems:

- The existence of cyclic walks on undirected graphs means that edge weights will be biased towards closer nodes against further nodes.
- Traditional methods hinder the exchange of skeleton models in time and space, and can not capture the complex regional space.

Aiming at these two problems, the author of [19] puts forward **MS-G3D** model, which solves the above problems from two aspects.

- A new multi-scale aggregation scheme is proposed to solve the biased weighting problem.
- The author of [19] proposes **G3D**, a new unified convolution model of Spatial-Temporal graph.

Thanks to the above two contributions, **MS-G3D** model surpasses the traditional graph convolution neural network and performs well in the skeleton based human action behavior recognition model. We experimented with our contribution based on this model.

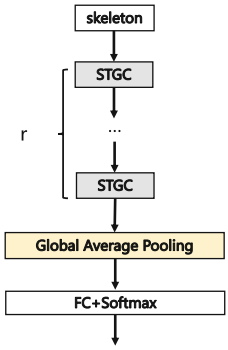
Overall Architecture. At the high level, **MS-G3D** is composed of many stacked Spatial-Temporal graph convolution (**STGC**) blocks, which can extract features from the skeleton. Then the feature is input into a global average pooling layer, and finally passes through the full connection layer and softmax classifier.

STGC Block. STGC block is the main extractor of the model, and the skeleton features first pass through two paths.

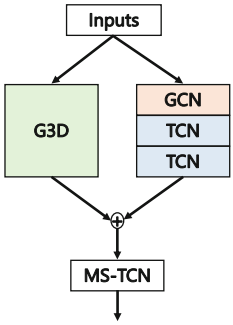
- G3D
- GCN, TCN, TCN

After feature extraction of different paths, the features of the two paths are fused, and then input through TCN.

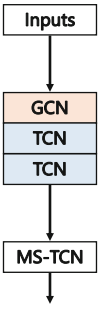
Simplified STGC Block and MS-G3D-Simplified. In order to test our method from multiple dimensions, we propose simplified block based on **STGC** block. Simplified **STGC** block does not go through **G3D**. The **MS-G3D** model composed of simplified stgc block is called **MS-G3D-simplified**.



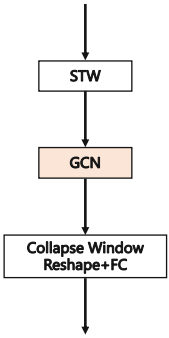
(a) Overall Architecture



(b) STGC Block



(c) Simplified STGC Block



(d) G3D

Fig. 6. .

G3D. This model is to solve the problem that local aggregators (such as GCN and TCN) are weakened in spatiotemporal propagation. The author of [19] uses the method of allowing cross Spatial-Temporal connections to make time and space converge uniformly. The skeleton features are processed by **Sliding Temporal Window**, **GCN** and **Collapse Window Reshape + FC**, and finally the features of cross Spatial-Temporal aggregation are obtained (Fig. 6).

4.7 Component Studies

To investigate the effectiveness of **Human Mirror** and **Human Link**, we analyze the individual components in the above dataset and data set settings. Performance is reported as Top 1 classification accuracy on the Cross-Set setting of NTU RGB+D 120 using only the joint data. In all experiments, **Human Link**

must be used with **Human Mirror**, so *with Human Link* is *with Human Link and Human Mirror* (Table 2).

Table 2. Classification accuracy comparison against state-of-the- art methods on the NTU RGB+D 120 Skeleton dataset.

Method	NTU RGB+D 120(%)
ST-LSTM	57.9
GCA-LSTM	63.3
RotClips+MTCNN	61.8
Body Pose Evolution Map	66.9
2s-AGCN	84.9
MS-G3D-Simplified	85.89
MS-G3D	86.46
MS-G3D-Simplified with Human Mirror	86.07
MS-G3D-Simplified with Human Link	86.13
MS-G3D with Human Mirror	86.69
MS-G3D with Human Link	86.75

From the above experimental results, **Human Mirror** can contribute to neural network learning, and using it helps to improve the ability of neural network to learn in balance, an ability that previous neural network strategies do not have. Then **Human Link** based on **Human Mirror**, can help the neural net to process the feature relationship between action emitters and thus understand the action better. The two approaches proposed in the paper have significant improvement on the accuracy rate and help the neural network performance significantly.

5 Concluding Remarks

5.1 Conclusion

In this study, we propose two optimization strategies: **Human Mirror** and **Human Link**.

Human Mirror eliminates the negative impact of a single person on the neural network in the data set. Under the different data of one action sender and two action senders, the neural network can not use the same strategy to process the data of the two modes, which has a negative impact on the accuracy.

Human Link solves that the neural network can only extract the features of spatial scale and time scale and loss of information caused by averaging in the traditional way. Through the feature extraction of human body scale, the understanding of action by neural network is strengthened.

5.2 Future Work

For **Human Mirror**, the current method is to copy the data from $x_{i,0}$ to $x_{i,1}$, so that the coordinates of the two are exactly the same, but the positions of the two people are not the same for the action performed by the two people, if we can translate the copied $x_{i,1}$ by learning the position interval of the two people, it may be more beneficial for the neural network to learn the two cases in unison.

For **Human Link**, the current method is to connect all the corresponding points of $x_{i,0}$ to $x_{i,1}$, which may not be the best method, we can use other connection methods, such as connecting only a few key nodes instead of all the nodes, or adding some buffer nodes to the connected nodes to increase the connection distance between two people, so that GCN [29] can give priority to extracting features within its own range instead of over-fusing the features between two people.

References

1. Li, Y., et al.: TEA: temporal excitation and aggregation for action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 909–918 (2020)
2. Fanello, S.R., et al.: Keep it simple and sparse: real-time action recognition. J. Mach. Learn. Res. **14**, 2617–2640 (2013)
3. Tran, D., et al.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6450–6459 (2018)
4. Saggese, A., et al.: Learning skeleton representations for human action recognition. Pattern Recogn. Lett. **118**, 23–31 (2019)
5. 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, pp. 1110–1118 (2015)
6. Ke, Q., et al.: A new representation of skeleton sequences for 3d action recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3288–3297 (2017)
7. Cheng, K., et al.: Extremely lightweight skeleton-based action recognition with ShiftGCN++. IEEE Trans. Image Process. **30**, 7333–7348 (2021)
8. Liu, J., et al.: NTU RGB+ D 120: a large-scale benchmark for 3d human activity understanding. IEEE Tran. Pattern Anal. Mach. Intell. **42**(10), 2684–2701 (2019)
9. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
10. Cheng, K., et al.: Decoupling GCN with DropGraph module for skeleton-based action recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020, Part XXIV. LNCS, vol. 12369, pp. 536–553. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58586-0_32
11. Zhang, P., et al.: Semantics-guided neural networks for efficient skeleton based human action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1112–1121 (2020)

12. Shi, L., et al.: Skeleton-based action recognition with directed graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7912–7921 (2019)
13. 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, pp. 588–595 (2014)
14. Wang, L., Koniusz, P., Huynh, D.: Hallucinating IDT descriptors and I3D optical flow features for action recognition with CNNs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019). https://doi.org/10.1109/ICCV.2019.00879
15. Liu, M., Liu, H., Chen, C.: Enhanced skeleton visualization for view invariant human action recognition. Pattern Recogn. **68**, 346–362 (2017)
16. Li, M., et al.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3595–3603 (2019)
17. Si, C., et al.: An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1227–1236 (2019)
18. Shi, L., et al.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12026–12035 (2019)
19. Liu, Z., et al.: Disentangling and unifying graph convolutions for skeleton based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 143–152 (2020)
20. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
21. Cortes, C., et al.: Advances in neural information processing systems 28. In: NIPS 2015 (2015)
22. Bruna, J., et al.: Spectral networks and locally connected networks on graphs. In: arXiv preprint arXiv:1312.6203 (2013)
23. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
24. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: arXiv preprint arXiv:1609.02907 (2016)
25. Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. Appl. Comput. Harmon. Anal. **30**(2), 129–150 (2011)
26. Wan, S., et al.: Multiscale dynamic graph convolutional network for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **58**(5), 3162–3177 (2019)
27. Dang, L., et al.: MSR-GCN: multi-scale residual graph convolution networks for human motion prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 11467–11476 (2021)
28. Zhang, Y., et al.: STST: spatial-temporal specialized transformer for skeleton-based action recognition. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 3229–3237 (2021)
29. Veličković, P., et al.: Graph attention networks. In: arXiv preprint arXiv:1710.10903 (2017)
30. Shahroudy, A., et al.: NTU RGB+ D: a large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1010–1019 (2016)

31. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: arXiv preprint arXiv:1412.6980 (2014)
32. Chen, Y., et al.: Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13359–13368 (2021)