Richly Activated Graph Convolutional Network for Robust Skeleton-based Action Recognition

Yi-Fan Song, Zhang Zhang, *Member, IEEE,*
Caifeng Shan, *Senior Member, IEEE,* and Liang Wang, *Fellow, IEEE*

Abstract—**Current methods for skeleton-based human action recognition usually work with complete skeletons. However, in real scenarios, it is inevitable to capture incomplete or noisy skeletons, which could significantly deteriorate the performance of current methods when some informative joints are occluded or disturbed. To improve the robustness of action recognition models, a multi-stream graph convolutional network (GCN) is proposed to explore sufficient discriminative features spreading over all skeleton joints, so that the distributed redundant representation reduces the sensitivity of the action models to non-standard skeletons. Concretely, the backbone GCN is extended by a series of ordered streams which is responsible for learning discriminative features from the joints less activated by preceding streams. Here, the activation degrees of skeleton joints of each GCN stream are measured by the class activation maps (CAM), and only the information from the unactivated joints will be passed to the next stream, by which rich features over all active joints are obtained. Thus, the proposed method is termed richly activated GCN (RA-GCN). Compared to the state-of-the-art (SOTA) methods, the RA-GCN achieves comparable performance on the standard NTU RGB+D 60 and 120 datasets. More crucially, on the synthetic occlusion and jittering datasets, the performance deterioration due to the occluded and disturbed joints can be significantly alleviated by utilizing the proposed RA-GCN.[1]**

Index Terms—**Action Recognition, Skeleton, Activation Map, Graph Convolutional Network, Occlusion, Jittering**

I. INTRODUCTION

Human action recognition has achieved promising progress in recent computer vision researches and plays an increasingly crucial role in many potential applications, such as video surveillance, human-computer interaction, video retrieval and so on [1], [2], [3]. The main purpose of action recognition is to classify human actions from motion data which can be captured as RGB videos [4], [5], depth maps [6], infrared images [7] and 3D skeleton sequences [8], [9], [10].

Traditional action recognition is dominated by RGB video-based methods. These methods usually consider RGB videos as temporal sequences of image frames, and use sequential models such as recurrent neural network (RNN) to exploit temporal information from all the feature maps extracted by convolution neural networks (CNN) for each frame [4]. On the other hand, many researchers utilize 3D CNN to derive useful information directly from the videos [5], which has obtained a comparable performance with the former methods. Nevertheless, both of these two categories extract spatial

structure information from 2D RGB frames, while the spatial configurations of actors are absolutely presented in 3D space. Thus, these RGB-based methods will lose some crucial information due to the intrinsic weakness. Moreover, the RGB videos often contain complex background and illumination variations, which leads to significant performance degradation in practice.

Compared to RGB videos, skeleton-based human action recognition methods reflect a growing prospect, due to its superiority in background adaptability, robustness to light variations and less computational cost. Skeleton data is composed of 2D/3D coordinates of multiple skeleton joints in motion sequences, which can be either collected by multimodal sensors such as Kinect or directly estimated from 2D images by pose estimation methods [11]. Current methods usually deal with skeleton data in two ways. One is to connect these joints into a global vector, then model temporal information by using RNN-based methods [8], [9], [12], [13]. The other way is to treat or expand temporal sequences of joints into 2D images, then utilize CNN-based methods to recognize actions [14], [15], [16], [17]. However, it is difficult to utilize the spatial structure information among skeleton joints effectively with both the RNN and CNN methods, though many researchers propose additional constraints or dedicated network structures to model the spatial structure of skeleton joints. Recently, graph neural networks (GNN), which can explicitly incorporate graphical structure information into the learning of neural network, have made great progress in many fields [18]. Yan et al.[10] firstly propose a spatial temporal graph convolutional network (ST-GCN) to capture the patterns embedded in the spatial configuration as well as the temporal dynamics in skeleton sequences, which achieves a significant improvement in action recognition.

However, current skeleton-based action recognition models still lack of robustness to the noisy or incomplete skeleton data captured in real scenarios. For example, the subjects may be self-occluded by pose variations or occluded by other contextual objects. Fig.1 displays some examples of occluded actions due to other persons in the scenes. The noisy data will deteriorate the performance of the models heavily. Therefore, how to enhance the robustness of skeleton-based action recognition models is still an urgent and challenging problem.

To against various degradations, ensemble learning has proved to be an effective strategy [19], which induces multiple classifiers based on the same or distinct predictive classifiers so that the integration of these individual classifiers could

---

[1]The codes and pretrained models of the preposed RA-GCN are available at http://github.com/yfsong0709/RA-GCNv2.
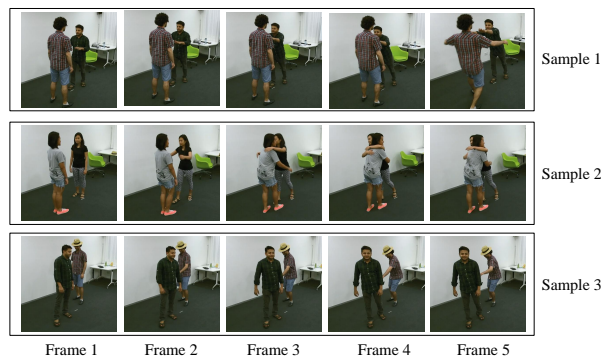
Fig. 1. Some examples of occluded actions. The person behind is partially occluded by the person in front.

enhance the robustness of recognition models. Inspired by the success of ensemble learning, in this paper, we propose a GCN-based multi-stream model, which aims to learn rich discriminative features from skeleton motion sequences, and thereby improve the robustness of the proposed model. The purpose of each stream in our approach is to explore a group of discriminative features over the skeleton joints unactivated by previous streams. The learnt redundant but complementary features over all skeleton joints provide an effective strategy to handle the noisy or incomplete skeleton data. For example, when we recognize the action *throwing*, the most discriminative joints are located on the two arms at the moment of object leaving the hands in the process of *throwing*, while the body swaying as well as the contextual sub-actions of hands can also be used to infer the action of *throwing*.

In order to distinguish the most informative joints for each stream, we introduce a successful technique named class activation maps (CAM) [20] into our model, which initially aims to visualize the activation heatmap in the final CNN layer responsible for visual classification. The activation maps obtained by previous GCN streams are accumulated as a mask matrix to inform the new stream about which joints have been already activated. Then, the new stream will be forced to explore new discriminative features from unactivated joints. Therefore, the proposed method is called richly activated GCN (RA-GCN), where the richly discovered and complementary features will improve the robustness of the model to non-standard skeletons. To the best of our knowledge, this is the first time to employ the CAM technique to enhance the model robustness by expanding the activated skeleton joints, which alleviates the problems of occlusion and jittering in skeleton-based action recognition.

To validate the advantages of the proposed methods, besides the traditional skeleton action datasets, the NTU RGB+D 60 [21] & 120 [22] datasets, we also build four synthetic occlusion datasets, where the joints in the NTU 60 and 120 datasets are partially occluded over both spatial and temporal dimensions, and two synthetic jittering datasets, where some randomly selected joints are disturbed by Gaussian noises. More details of these datasets can be found in Section IV-A. Our experiments on these new datasets demonstrate that the proposed RA-GCN significantly alleviates the performance

deterioration in the case of incomplete or noisy skeleton data.

This work is an extension of an earlier and preliminary version presented in [23]. Compared to our previous work, the modifications and contributions of this paper are summarized as follows:

- In previous work, the activation masks are obtained by a Softmax function in the activation modules, which activates only a few joints for each stream. In contrast, we propose to use a normalization activation function to expand the activated scope, thus the corresponding stream will obtain a better and more interpretable activation map.
- Compared to previous work, we extend the original loss function with a number of additional cross-entropy regularizations on each individual network stream, so that the features can be learnt more effectively.
- The synthetic datasets are extended by more degradation operators, where the occlusion degradation is further divided into four types, including Frame, Part, Block and Random, and two synthetic jittering datasets are newly constructed. More experiments are performed to validate the effectiveness and robustness of the proposed approach in different degradation conditions.

The remainder of this paper is organized as follows: Section II describes recent studies related to our work. Section III introduces several crucial components of the proposed RA-GCN. Extensive experimental results on standard and non-standard datasets are reproted in Section IV, and the conclusion of this paper is given in Section V.

II. RELATED WORK

a) Skeleton-based models: To find a more effective representation of the dynamics of human actions, Johansson [24] utilizes 3D skeleton sequences for action recognition, making an obvious decrease of computational cost as well as a good performance boost. Recently, with the rapid development of deep learning techniques, skeleton-based action recognition methods have attracted increasing attentions. Researchers have proposed various models to improve the performance of action recognition, which can be divided into three major categories. The first category builds the models with convolutional networks. For example, Li et al. [16] propose a CNN-based co-occurrence feature learning framework, which gradually aggregates various levels of contextual information. Kim et al. [15] build a temporal convolutional network to explicitly learn readily interpretable spatio-temporal representations for 3D human action recognition.

Besides, for the second category, researchers concatenate all joints in one frame into a single vector, then use sequential models such as long short-term memory (LSTM) to explore the temporal dynamics. Du et al. [25] design a hierarchical bidirectional RNN to capture rich dependencies between different human body parts. The study in [9] employs a view adaptive LSTM, which enables the network itself adaptive to the most suitable observation viewpoints. Additionally, Song et al. [26] firstly introduce attention modules into skeleton-based action recognition.

Both CNN-based and RNN-based methods are still limited to extract the spatial structure information among skeleton
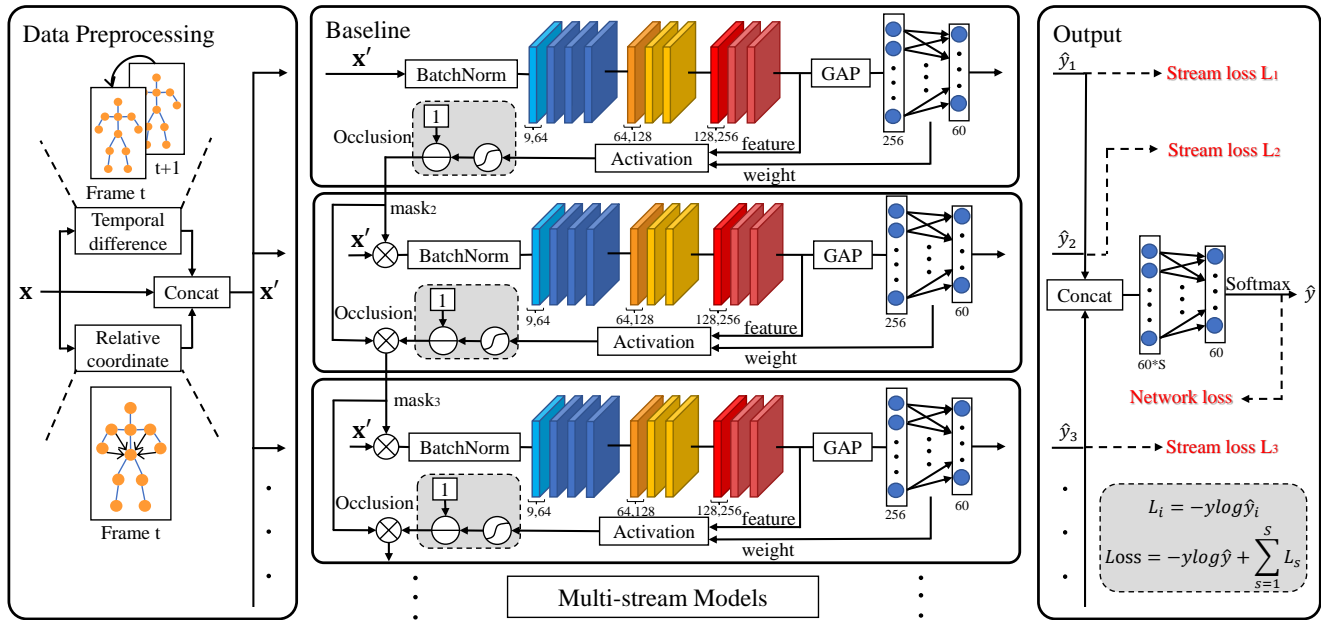
Fig. 2. The pipeline of RA-GCN with three stream networks. Each stream contains one ST-GCN. The two numbers under the ST-GCN layers are the numbers of input and output channels, respectively. Other layers contain the same input and output channels. Both of the fifth and the eighth layers use a temporal stride 2 to reduce the sequence length. **GAP** and **Concat** are global average pooling and concatenation operation, S is the number of streams, and $\otimes$ and $\ominus$ denote element-wise multiplication and subtraction, respectively. This model totally contains three steps. The input sequence $\mathbf{x}$ is firstly transformed into $\mathbf{x}'$ by the data preprocessing module. Secondly, $\mathbf{x}'$ will be sent to each stream after being filtered by a corresponding mask matrix. Finally, the output of each stream will be concatenated to obtain the final class of $\mathbf{x}$.

joints, where the joints of different body parts are connected as a skeleton graph. Instead, in the third category, graph-based methods can be naturally utilized to deal with the skeleton graph, which successfully captures the most informative features for various actions. Si et al. [13] use GNN to model the relationships among five body parts. Yan et al. [10] initially introduce GCN into skeleton-based action recognition, and produce a baseline named ST-GCN for future research. Based on the ST-GCN, many studies achieve continuous improvements on skeleton-based action recognition [27], [28], [29].

b) Occlusion in human action recognition: Occlusion is a prominent challenge in human action recognition. If the skeleton joints are partially occluded, the approaches mentioned above will face a considerable decline of performance. To handle this problem, Wang et al. [30] try to infer occlusion maps from a global SVM classifier, and Weinland et al. [31] propose a local partitioning and hierarchical classification of the 3D Histogram of Oriented Gradients (HOG) descriptor for providing the robustness to both occlusions and view point changes. However, there are few studies addressing the problem of noisy or incomplete data in skeleton joints. In this paper, we propose an approach to exploring rich features over all joints, so as to alleviate the effects of data degradation.

c) Salient Regions Exploration: Similar with our motivation, some previous studies have proposed to explore the salient regions or erase them to exploit complementary information for referring expression grounding [32], [33] or weakly supervised detection tasks [34]. The study [32] proposes a simple yet effective network to prohibit attentions

from spreading to unexpected background regions, in order to promote the quality of object attention. Liu et al. [33] design a novel attention-guided erasing approach to aligning various types of information crossing visual and textual modalities. Moreover, Li et al. [34] provide a framework to dynamically erase the focused area according to the on-line attention maps. However, previous salient regions exploration methods mainly concentrate on object detection or localization tasks in images, while in this work, we exploit the complementary attended skeleton joints for alleviating the occlusion or jittering problems, which is still not considered in previous work.

III. MODEL ARCHITECTURE

In order to enhance the robustness of action recognition models, we propose the RA-GCN to explore sufficient discriminative features from all skeleton joints. The proposed RA-GCN constructs a multi-stream network, where each stream is responsible for extracting features from a group of activated joints. In this way, when the joints activated by the first stream are occluded, the model can also discover discriminative information from the other streams. The overview of RA-GCN is presented in Fig.2. Suppose that V is the number of joints in one skeleton and T is the number of frames in one sequence, the size of input data $\mathbf{x}$ is $C_{in} \times T \times V$, where $C_{in} = 3$ denotes the 3D coordinates of each joint. Note that different skeletons in a multi-agent action are treated as different samples.

The proposed method consists of three main steps. Firstly, in the preprocessing module, for extracting more informative features, the input data $\mathbf{x}$ is transformed into $\mathbf{x}'$, which is subsequently sent to all the GCN streams. Secondly, for
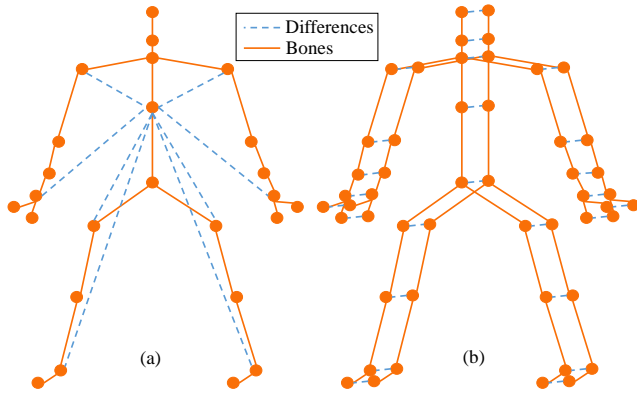
Fig. 3. (Best viewed in color.) The illustration of data preprocessing. (a) is relative joint coordinates. (b) is temporal displacements. The orange solid lines mean the physical connections between neighboring joints (bones) and the blue dash lines denote the differences between two joints.

each stream, the skeleton joints in $\mathbf{x}'$ will be filtered by the element-wise product with a mask matrix, which records the currently unactivated joints. These joints are distinguished by accumulating the activated maps calculated by the activation modules of preceding streams. Here, the mask matrix of each stream is initialized to an all-one matrix with the same shape as $\mathbf{x}'$. After the masking operation, the input data of each stream only contains the joints unactivated by the preceding streams, and passes through a baseline network to obtain a feature representation based on the incomplete skeleton joints. Finally, the features of all streams are concatenated in the output module, and a fully connected layer with Softmax activation function is used to obtain the final class of input $\mathbf{x}$. These three steps will be discussed in details in next sections.

A. Data Preprocessing

Usually, some actions such as *taking off the shoes* and *wearing the shoes* are extremely similar with only spatial features. To tackle this problem, conventional RGB-based methods introduce a sophisticated technique named Optical Flow [35] into their models, for depicting the motion features exactly. Besides, the work in [28] argues that relative coordinates of joints are usually more informative than absolute coordinates. Inspired by this, geometric features such as relative coordinates and motion features such as temporal displacements are applied in our models to increase the discriminative information for action recognition. Therefore, the input data need to be preprocessed before distributing it to all the GCN streams.

The relative coordinates can be recognized as the difference $\dot{\mathbf{x}}_r$ between all joints and the center joint (middle spine) in each frame, which can be seen in Fig.3(a). In this way, all joints are transformed to the relative coordinates, which is more robust to the changing position. Besides, for extracting more informative motion features, we compute $\dot{\mathbf{x}}_t$ by $\mathbf{x}_{t+1} - \mathbf{x}_t$, where $\mathbf{x}_t$ means the feature map of the tth frame, which is shown in Fig.3(b). Then, $\mathbf{x}'$ will be obtained by concatenating $\mathbf{x}$, $\dot{\mathbf{x}}_r$ and $\dot{\mathbf{x}}_t$.
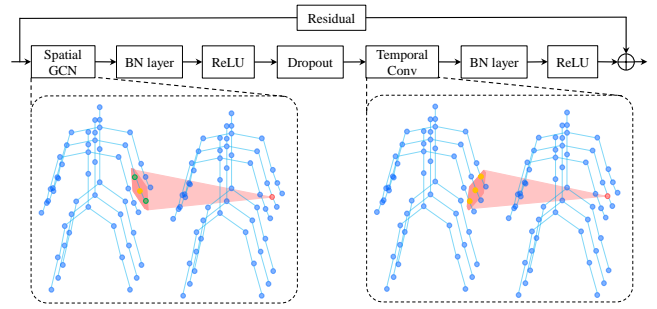

Fig. 4. The structure of an ST-GCN layer, where BN means the BatchNorm layer, and $\oplus$ denotes the element-wise summation operation.

B. Richly Activated GCN

1) Baseline Model: The baseline of our method is the ST-GCN [10], which is composed of ten graph convolutional layers. Yan et al. [10] formulate spatial graph convolutional operation as follows:

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

where f_{in} and f_{out} are the input and output feature maps respectively, v_{ti} denotes the ith joint at the tth frame, which can also be regarded as the root joint in this procedure, $B(v_{ti})$ is the neighbor set of v_{ti}, the normalizing term Z_{ti} is added to balance the contributions of different neighbors, $\mathbf{w}(\cdot)$ is a weight function implemented by several 1×1 Conv layers and $l_{ti}(\cdot)$ means a label function. There are three label functions in [10], but we only choose the distance-based label function in our method, which defines $l_{ti}(v_{tj}) = d(v_{ti}, v_{tj})$. That means the neighbor set $B(v_{ti})$ is divided into several subsets, according to the graph distance between v_{tj} and the root joint v_{ti}. For example, if the joint v_{tj} directly connects with the root joint v_{ti}, then $d(v_{ti}, v_{tj}) = 1$. The joints with the same distance will form a subset and share a learnable weight function $\mathbf{w}(\cdot)$. To implement the spatial graph convolution with the adjacency matrix $\mathbf{A}$, Eq.1 is transformed into:

$$\mathbf{f}_{out} = \sum_{d=0}^{D} \mathbf{W}_d \mathbf{f}_{in}(\Lambda_d^{-\frac{1}{2}} \mathbf{A}_d \Lambda_d^{-\frac{1}{2}} \otimes \mathbf{M}_d), \quad (2)$$

where D is the predefined maximum distance, $\mathbf{A}_d$ denotes the adjacency matrix for distance d, $\Lambda_d^{ii} = \sum_k A_d^{ik} + \alpha$ is the normalized diagonal matrix, A_d^{ik} denotes the element of the ith row and k^{th} column of $\mathbf{A}_d$ and α is set to a small value, e.g., 10^{-4}, to avoid the empty rows in Λ_d. For each adjacency matrix, we accompany it with a learnable matrix $\mathbf{M}_d$, which expresses the importance of all edges in one skeleton.

After the spatial graph convolutional block, a $1 \times L$ convolutional layer is used to extract temporal information of the feature map $\mathbf{f}_{out}$, where L is the temporal window size. Both spatial and temporal convolutional blocks are followed with a BatchNorm layer and a ReLU layer, and the total ST-GCN layer contains a residual connection. Besides, a dropout layer with the drop probability of 0.5 is added between every spatial convolutional block and temporal convolutional block to avoid

overfitting. The structure of one ST-GCN layer is shown in Fig.4.

2) Activation Module: The activation module in the RA-GCN is constructed to distinguish the activated joints of each stream, then guide the learning process of the new stream by accumulating the activated maps of preceding streams. This procedure can be mainly implemented by extending the CAM technique [20] to the field of GCN. The original CAM technique is to localize class-specific image regions in CNNs, and $score_c$ is defined as the scores of all pixels for class c, where the score of each pixel is

$$score_c(x, y) = \sum_k w_k^c f_k(x, y). \quad (3)$$

In this formulation, $f_k(\cdot, \cdot)$ is the feature map before the global average pooling operation, and w_k^c is the weight of the kth channel for class c. In this paper, we replace the coordinate (x, y) in a feature map with the frame number t and the joint number i in a skeleton sequence, by which we are able to locate the activated joints. Here, the class c is selected as the ground truth. We use $score_c^s$ to denote the score map of all joints for the true class and the sth stream. To determine which joints are activated by the corresponding stream, a predefined threshold δ is utilized, and the activation map of map_c^s is calculated by

$$map_c^s = \varepsilon\left(\frac{score_c^s}{max(score_c^s)} - \delta\right) \quad (4)$$

where $\varepsilon(\cdot)$ is the Heaviside step function and $max(\cdot)$ denotes the maximum function. Then, the mask matrix of the sth stream is represented as

$$mask_s = \left(\prod_{i=1}^{s-1} mask_i\right) \otimes (1 - map_c^{s-1}), \quad (5)$$

where $\prod$ denotes the element-wise product of all mask matrices before the sth stream. Specially, the mask matrix of the first stream is an all-one matrix. Finally, the input of the sth stream will be obtained by

$$\mathbf{x}_s = \mathbf{x}' \otimes mask_s, \quad (6)$$

where $\mathbf{x}'$ is the skeleton representation after preprocessing.

Eq.5 and Eq.6 illustrate that the input of the sth stream only consists of the joints which are not activated by previous streams. Thus, the RA-GCN will explore discriminative features from all joints sufficiently.

3) Loss Function: In our previous model [23], the loss function only supervises the total network, which is extended with a number of additional losses for each individual network streams in this paper, so that the feature can be learnt more effectively. Suppose that $\hat{\mathbf{y}}_s \in \mathbb{R}^C$ is the output of the sth stream, $\hat{\mathbf{y}} \in \mathbb{R}^C$ is the output of the whole model and $\mathbf{y} \in \mathbb{R}^C$ is the ground truth, where C is the number of classes. Then the loss function of the proposed RA-GCN is

$$L = -\mathbf{y}log\hat{\mathbf{y}} - \sum_{s=1}^{S} \mathbf{y}log\hat{\mathbf{y}}_s \quad (7)$$

where S is the number of streams.

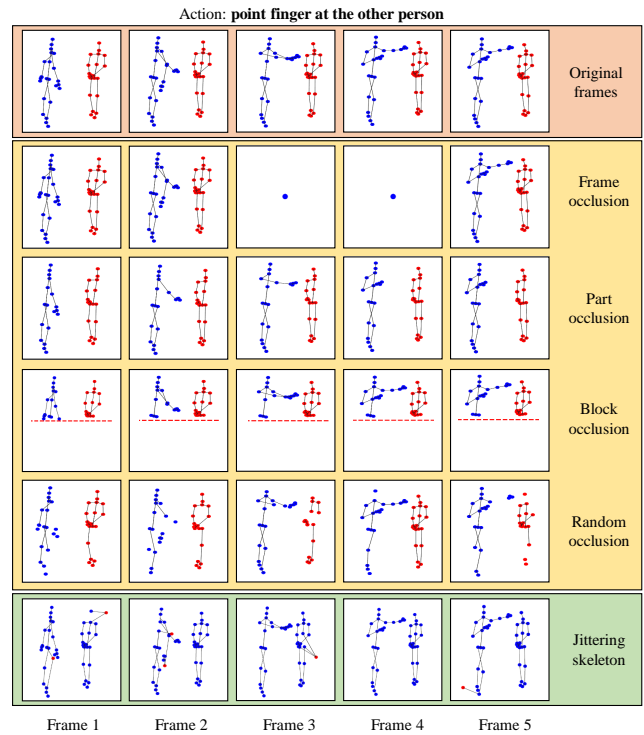Action: **point finger at the other person**

Fig. 5. (Best viewed in color.) The demonstration of the degraded datasets generated by adding various types of noise and occlusion to the standard NTU 60 dataset. The first line is the original frames. The middle lines are the frame occlusion without frames 3 and 4, the part occlusion without left arm, the block occlusion without the joints under the red dash line and the random occlusion with an occluded probability $p = 0.2$, respectively. In addition, the bottom line is an example of the jittering skeletons, where the red joints denote the noisy joints.

IV. EXPERIMENTAL RESULTS

A. Dataset

a) NTU RGB+D 60 [21]: This dataset is a large-scale indoor action recognition dataset, which contains 56880 video samples collected by Microsoft Kinect v2, and consists of 60 action classes performed by 40 subjects. Each video is composed of 25 joints and no more than two skeletons in one frame. The maximum frame number T is set to 300 for simplicity. The authors of this dataset recommend two benchmarks: (1) **cross-subject (CS)** contains 40320 samples for training and 16560 samples for evaluation, by splitting 40 subjects into two groups; (2) **cross-view (CV)** uses cameras 2 and 3 (37920 samples) for training and camera 1 (18960 samples) for evaluation. We follow this convention and report the top-1 recognition rate on both two benchmarks. In addition, according to [12], there are 302 wrong samples that need to be ignored during training and evaluation.

b) NTU RGB+D 120 [22]: This dataset is currently the largest indoor action recognition dataset, which is an extended version of NTU 60 dataset. It contains 114480 videos and consists of 120 classes. Similarly, two benchmarks are suggested: (1) **cross-subject (CSub)** contains 630226 samples for training and 50922 samples for evaluation; (2)**cross-setup (CSet)** contains 54471 videos for training and 59477 videos

for evaluation, which are separated based on the distance and height of their collectors. Note that there are 532 bad samples in this dataset which should be ignored in all experiments.

c) Occlusion dataset: To validate the robustness of our method to incomplete skeletons, we construct a synthetic occlusion dataset based on the CS benchmark of NTU 60 dataset and the CSet benchmark of NTU 120 dataset, where some joints are selected to be occluded (set to zero) over both spatial and temporal dimensions. Note that this operation is executed before data preprocessing, and all the joints related to the occluded joints (with zero energy) are ignored in data preprocessing phase. For example, if the frame $\mathbf{x}_t$ is occluded, then the temporal displacements $\dot{\mathbf{x}}_t = \mathbf{x}_{t+1} - \mathbf{x}_t$ and $\dot{\mathbf{x}}_{t-1} = \mathbf{x}_t - \mathbf{x}_{t-1}$ are both set to zero. This synthetic dataset consists of four cases, which are frame occlusions, part occlusions, block occlusions and random occlusions, respectively. Part occlusion and block occlusion are both used to simulate the real scenarios that occluded by contextual objects. Frame occlusion is designed for the loss of key frames, while random occlusion is for data missing in signal transmission. Some examples on the four types of occlusions are illustrated in Fig.5. Note that all the models in occlusion experiments are trained with standard skeletons, and then tested with incomplete skeletons.

d) Jittering dataset: Skeleton jittering is a common factor that has a big impact on the recognition performance. To claim the robustness of the proposed method to jittering skeletons, we propose a synthetic jittering dataset based on the CS benchmark of NTU 60 dataset and the CSet benchmark of NTU 120 dataset, where the Gaussian noise $N(\mu, \sigma^2)$ is added to some randomly selected joints to simulate the jittering joints. This jittering operation is also executed before data preprocessing module, while the data preprocessing module has no difference with that in standard setting. In this paper, two types of Gaussian noise are used, which are $N(0, 0.1^2)$ and $N(0, 0.05^2)$. In the bottom line of Fig.5, an example of jittering skeletons is displayed, and the red joints denote the noisy joints. Similar with the occlusion dataset, all the models of this dataset are trained with standard skeletons.

B. Implementation Details

In our experiments, some hyper-parameters need to be modified. The initial learning rate is set to 0.1 and divided by 10 every 20 epochs, while the maximum number of iterations is set to 60. The models are learnt by using the stochastic gradient descent (SGD) algorithm with a momentum 0.9 and a weight decay 10^{-4}. In order to avoid overfitting, the probability of the dropout layer between the spatial and temporal blocks is selected as 0.5. The first four ST-GCN layers have 64 channels for output, while the number will be 128 and 256 for the middle three layers and the last three layers. Moreover, at the fifth and eighth layers, the temporal convolutional blocks contain a temporal stride 2, for reducing the computational cost. As to the maximum graph distance D, the temporal window size L and the mask threshold δ, we will discuss about their effects in Section IV-C.

Before training a multi-stream RA-GCN, we need to pre-train a one-stream RA-GCN with preprocessed skeleton data

TABLE I
COMPARISON OF DIFFERENT PARAMETER SETTINGS OF RA-GCN ON THE TWO BENCHMARKS NTU 60 (%)

Model	Parameters	CS	CV
baseline (1s RA-GCN*)	$D = 1, L = 5$	85.2	90.5
	$D = 2, L = 5$	**85.8**	91.6
	$D = 3, L = 5$	**85.8**	92.2
	$D = 1, L = 9$	85.4	91.7
	$D = 2, L = 9$	85.4	92.7
	$D = 3, L = 9$	85.0	**93.1**
2s RA-GCN*	$\delta = 0.1$ $(D = 2, L = 5)$	86.5	–
	$\delta = 0.3$ $(D = 2, L = 5)$	**86.7**	–
	$\delta = 0.5$ $(D = 2, L = 5)$	86.3	–

*: 1s and 2s denote the number of streams

TABLE II
COMPARISON OF DIFFERENT MODEL SETTINGS ON THE CS BENCHMARK OF NTU 60 (%)

Model	Setting	accuracy
2s RA-GCN	w/o activation module	85.5
	w/o pretrained	85.2
	w/ activation function in [23]	85.8
2s RA-GCN	only raw skeleton	76.8
	only relative coordinates	73.4
	only temporal displacements	83.1
1s RA-GCN	–	85.8
2s RA-GCN	–	86.7
3s RA-GCN	–	**87.3**
4s RA-GCN	–	87.2

to get the baseline model, so as to ensure that the first stream of RA-GCN is able to capture the most informative joints. Accordingly, the following streams are forced to seek for other discriminative joints. Additionally, the mask matrix of each stream is initialized to an all-one matrix. Finally, we finetune the RA-GCN model with the setting mentioned above. All the experiments are performed on two TITAN X GPUs.

C. Parameters Setting

In order to train a baseline model, we firstly need to determine the value of two hyper-parameters introduced in Section III-B1, i.e., the D for the maximum distance and the L for the temporal window size. These two hyper-parameters have a great impact on our model, because they control the receptive field of the GCN blocks. To find the optimal values, we evaluate many groups of the two hyper-parameters ($D \in \{1, 2, 3\}$ and $L \in \{3, 5, 7, 9, 11\}$) on the NTU 60 dataset, and some representative experimental results are given in the first part of Tab.I. It is observed that the baseline model achieves the best accuracy when $D = 2$ and $L = 5$ on the CS benchmark. As to the CV benchmark, D and L are optimally set to 3 and 9, respectively. Note that it is not always better to choose bigger D and L, since a bigger receptive field will lead to the over-smoothing problem, and eventually harm the model performance. The experimental results also demonstrate this point. From these experiments, an optimal baseline model is obtained, which will be utilized to construct the multi-stream RA-GCN.

For another hyper-parameter δ mentioned in Section III-B2, we select its value on a two-stream RA-GCN, with the hyper-parameters $D = 2$ and $L = 5$. The δ decides which joints

TABLE III
COMPARISON OF THE SOTA METHODS ON THE TWO BENCHMARKS OF NTU 60 IN ACCURACY (%) AND MODEL SIZE (MILLION)

Model	Year	Param.	Data	CS	CV
DSSCA-SSLM [36]	2017	–	Both	74.9	–
2D-3D-Softargma [37]	2018	–	RGB	85.5	–
Glimpse Clouds [38]	2018	–	RGB	86.6	93.2
H-BRNN [25]	2015	–	Skeleton	59.1	64.0
VA-LSTM [9]	2017	–	Skeleton	79.4	87.6
CNN+Motion+Trans [14]	2017	–	Skeleton	83.2	89.3
3scale ResNet152 [39]	2017	–	Skeleton	85.0	92.3
HCN [16]	2018	–	Skeleton	86.5	91.1
ST-GCN [10]	2018	3.10*	Skeleton	81.5	88.3
DPRL+GCNN [17]	2018	–	Skeleton	83.5	89.8
SR-TSL [13]	2018	19.07*	Skeleton	84.8	92.4
PB-GCN [28]	2018	–	Skeleton	87.5	93.2
AS-GCN [40]	2019	6.99*	Skeleton	86.8	94.2
2s-AGCN [29]	2019	6.94*	Skeleton	88.5	95.1
AGC-LSTM [27]	2019	–	Skeleton	**89.2**	**95.0**
PL-GCN [41]	2020	20.70†	Skeleton	89.2	95.0
NAS-GCN [42]	2020	6.57†	Skeleton	89.4	95.7
preliminary version [23]	2019	6.21	Skeleton	85.9	93.5
baseline (1s RA-GCN)	2020	2.03	Skeleton	85.8	93.1
2s RA-GCN	2020	4.13	Skeleton	86.7	93.4
3s RA-GCN	2020	6.21	Skeleton	**87.3**	**93.6**

*: These results are implemented by ourselves.
†: These results are provided by their authors.

TABLE IV
COMPARISON OF THE SOTA METHODS ON THE TWO BENCHMARKS OF NTU 120 IN ACCURACY (%) AND MODEL SIZE (MILLION)

Model	Year	Param.	Data	CSub	CSet
Soft RNN [43]	2018	–	RGB	36.3	44.9
Pose Evolution Map [44]	2017	–	Both	64.6	66.9
PA-LSTM [21]	2016	–	Skeleton	25.5	26.3
ST-LSTM [8]	2016	–	Skeleton	55.7	57.9
2s attention LSTM [45]	2017	–	Skeleton	61.2	63.3
Skeleton Visualization [46]	2018	–	Skeleton	60.3	63.2
FSNet [47]	2018	–	Skeleton	59.9	62.4
SkeleMotion [48]	2019	–	Skeleton	67.7	66.9
TSRJI [49]	2019	–	Skeleton	67.9	62.8
ST-GCN [10]	2018	3.10*	Skeleton	70.7*	73.2*
SR-TSL [13]	2018	19.07*	Skeleton	74.1*	79.9*
2s-AGCN [29]	2019	6.94*	Skeleton	**82.5***	**84.2***
AS-GCN [40]	2019	6.99*	Skeleton	77.7†	78.9†
GVFE+DH-TCN [50]	2019	–	Skeleton	78.3	79.8
preliminary version [23]	2019	6.25	Skeleton	74.4	79.4
baseline (1s RA-GCN)	2020	2.07	Skeleton	78.2	80.0
2s RA-GCN	2020	4.17	Skeleton	81.0	82.5
3s RA-GCN	2020	6.25	Skeleton	**81.1**	**82.7**

*: These results are implemented by ourselves.
†: These results are reported in [50].

are activated, and hereby controls the number of activated joints for each stream. As seen in the bottom part of Tab.I, the model obtains the best accuracy when δ is set to 0.3 on the CS benchmark.

D. Ablation Studies

The proposed method consists of several fundamental components, e.g., the data preprocessing module, the activation module and so on. In this section, we will analyze the significance of each component. All these experiments are performed by a two-stream RA-GCN with $D = 2$, $L = 5$ and $\delta = 0.3$ on the CS benchmark, and the results are presented in the top line of Tab.II. As we remove the activation module, the accuracy of our model will drop by 1.2%. And the pretrained procedure is also important, without which the performance will have a 1.5% decline. In addition, if the Heaviside step function $\varepsilon(\cdot)$ and the threshold δ are replaced by a Softmax function as shown in our previous model [23], then the accuracy will drop to 85.8%. According to these experimental results, the necessity of each component in our model is validated for boosting the performance of action recognition.

Furthermore, as seen in the middle line, the data preprocessing module brings a huge improvement, without which the performance is significantly deteriorated. Concretely, the temporal displacements of raw skeletons obtain the best performance, but which are still worse than the whole feature concatenated by the original coordinates, relative coordinates, and temporal displacements. That means all components of the data preprocessing module are beneficial to our model.

The bottom line of Tab.II shows the results of the RA-GCN models with different numbers of streams. We will find that when the stream number is more than 3, the accuracy growth will be moderate, which is analyzed in Section IV-H.

E. Experimental Results on Standard Dataset

We compare the performance of RA-GCN against several previous SOTA methods on the NTU RGB+D 60 & 120 datasets. The hyper-parameters are chosen as the optimal value given in Tab.I. Tab.III and Tab.IV display the experimental results of one-stream (1s, baseline), two-stream (2s), three-stream (3s) RA-GCN and the other SOTAs methods, e.g., AGC-LSTM [27] and so on.

a) NTU RGB+D 60: Since Yan et al. [10] introduce GCN into skeleton action recognition, there have been many graph-based methods proposed recently, including our methods. Compared to them, our method is only 1.9% less than AGC-LSTM [27] on the CS benchmark and 1.4% on the CV benchmark, while the two accuracy differences are 1.9% and 1.4% for PL-GCN [41] or 2.1% and 2.1% for NAS-GCN [42]. Although the newly public methods, e.g., PL-GCN and NAS-GCN, have better recognition accuracies than ours, their complexities in model size or training procedures are much higher than the proposed RA-GCN. Compared to ST-GCN [10], which is the first graph-based model for skeleton action recognition, our method outperforms by 5.8% and 5.3%, respectively. With respect to the SOTA methods which is not based on GCN, e.g., VA-LSTM [9] and HCN [16], our method achieves a more significant superiority. The 3s RA-GCN outperforms HCN by 0.8% and 2.5%, and exceeds VA-LSTM nearly 8%.

b) NTU RGB+D 120: On the currently largest indoor action recognition dataset, NTU RGB+D 120, our approach achieves 81.1% for the CSub benchmark and 82.7% for the CSet benchmark. Compared to other GCN-based methods, the proposed method boosts the performance significantly. Concretely, 3s RA-GCN outperforms GVFE+DH-TCN [50] by 2.8% and 2.9% on the two benchmarks, while the non-graph models such as TSRJI [49] are far behind our model. Furthermore, for more convincing, three popular models, i.e., ST-GCN [10], SR-TSL [13] and 2s-AGCN [29], are implemented

TABLE V
EXPERIMENTAL RESULTS (%) WITH FRAME OCCLUSION ON THE CS BENCHMARK OF NTU 60 (TOP) AND THE CSET BENCHMARK OF NTU 120 (BOTTOM)

Frame Occlusion	Number of Occluded Frames					
	0	10	20	30	40	50
ST-GCN [10]	80.7	69.3	57.0	44.5	34.5	24.0
SR-TSL [13]	84.8	70.9	62.6	48.8	41.3	28.8
2s-AGCN [29]	**88.5**	74.8	60.8	49.7	38.2	28.0
preliminary version [23]	85.9	81.9	75.0	**66.3**	54.4	**40.6**
baseline (1s RA-GCN)	85.8	81.6	72.9	61.6	47.9	34.0
2s RA-GCN	86.7	83.0	**76.4**	65.6	53.1	39.5
3s RA-GCN	87.3	**83.9**	**76.4**	66.3	53.2	38.5
difference*	1.5	2.3	3.5	4.7	5.3	4.5
ST-GCN [10]	73.2	60.8	48.8	38.2	27.3	17.4
SR-TSL [13]	79.9	67.4	58.8	50.4	44.7	37.1
2s-AGCN [29]	**84.2**	73.4	56.8	44.2	33.0	23.4
preliminary version [23]	79.4	76.2	69.8	60.4	47.4	33.0
baseline (1s RA-GCN)	80.0	76.1	67.9	57.1	44.0	30.3
2s RA-GCN	82.5	79.1	72.4	**63.3**	**51.2**	36.6
3s RA-GCN	82.7	**79.6**	**72.9**	**63.3**	**51.2**	**36.8**
difference*	2.7	3.5	5.0	6.2	7.2	6.5

*: the difference between 3s RA-GCN and the baseline model

TABLE VI
EXPERIMENTAL RESULTS (%) WITH PART OCCLUSION ON THE CS BENCHMARK OF NTU 60 (TOP) AND THE CSET BENCHMARK OF NTU 120 (BOTTOM)

Part Occlusion	Occluded Part					
	None	1	2	3	4	5
ST-GCN [10]	80.7	71.4	60.5	62.6	77.4	50.2
SR-TSL [13]	84.8	70.6	54.3	48.6	74.3	56.2
2s-AGCN [29]	**88.5**	72.4	55.8	**82.1**	74.1	71.9
preliminary version [23]	85.9	73.4	60.4	73.5	81.8	70.6
baseline (1s RA-GCN)	85.8	69.9	54.0	66.8	82.4	64.9
2s RA-GCN	86.7	**75.9**	**62.1**	69.2	**83.3**	**72.8**
3s RA-GCN	87.3	74.5	59.4	74.2	83.2	72.3
difference*	1.5	4.6	5.4	7.4	0.8	7.4
ST-GCN [10]	73.2	59.7	47.3	52.5	68.5	48.5
SR-TSL [13]	79.9	59.4	50.3	41.2	64.8	55.0
2s-AGCN [29]	**84.2**	62.8	46.6	**77.8**	67.0	60.7
preliminary version [23]	79.4	65.6	51.2	57.3	75.3	64.9
baseline (1s RA-GCN)	80.0	64.0	49.7	50.0	74.7	60.2
2s RA-GCN	82.5	67.4	54.1	56.0	77.6	67.7
3s RA-GCN	82.7	**68.5**	**54.9**	57.5	**79.0**	**69.9**
difference*	2.7	4.5	5.2	7.5	4.3	9.7

*: the difference between 3s RA-GCN and the baseline model

TABLE VII
EXPERIMENTAL RESULTS (%) WITH BLOCK OCCLUSION ON THE CS BENCHMARK OF NTU 60 (TOP) AND THE CSET BENCHMARK OF NTU 120 (BOTTOM)

Block Occlusion	Height Range of the Horizontal Line					
	None	1	2	3	4	5
ST-GCN [10]	80.7	76.4	70.1	60.7	48.4	36.1
SR-TSL [13]	84.8	74.9	69.3	61.1	49.4	36.9
2s-AGCN [29]	**88.5**	79.2	73.6	64.6	53.4	40.2
preliminary version [23]	85.9	81.6	78.6	72.5	62.3	48.2
baseline (1s RA-GCN)	85.8	82.6	78.0	69.5	57.8	43.8
2s RA-GCN	86.7	84.4	**81.2**	**74.0**	**62.9**	**48.7**
3s RA-GCN	87.3	**84.5**	81.0	73.8	62.3	47.6
difference*	1.5	1.8	3.0	4.3	4.5	3.8
ST-GCN [10]	73.2	65.5	56.8	45.0	32.7	22.6
SR-TSL [13]	79.9	68.2	60.1	47.7	36.5	29.0
2s-AGCN [29]	**84.2**	71.7	65.7	56.3	44.6	31.1
preliminary version [23]	79.4	75.7	72.3	63.8	50.9	36.0
baseline (1s RA-GCN)	80.0	74.8	70.0	61.3	48.7	34.7
2s RA-GCN	82.5	79.0	75.2	67.8	55.8	40.6
3s RA-GCN	82.7	**79.7**	**76.2**	**69.1**	**57.3**	**42.2**
difference*	2.7	4.9	6.2	7.8	8.6	7.5

*: the difference between 3s RA-GCN and the baseline model

by ourselves, according to their released codes. Compared with them, our model only falls behind 2s-AGCN by 1.4% and 1.5% on the two benchmarks of NTU 120 dataset.

c) Model complexity: To compare the model complexity and computational cost, we calculate the number of parameters in terms of released codes, or ask the authors for the complexities of some models. All the results shown in Tab.III and Tab.IV, from which our model contains similar amount of parameters with other SOTA methods. For some LSTM-based methods, such as SR-TSL [13], our model only contains nearly 1/3 parameters, due to the efficiency of the GCN technique. Totally, there are only 6.21 million parameters in the 3s RA-GCN, and the inference speed of this model is about 18.7 sequences per second per GPU. Note that there are usually more than 50 frames in an action sequence, thus the training and testing speeds are obviously sufficient for real-time processing.

In general, the proposed RA-GCN can achieve comparable performance with the SOTA methods, because the main purpose of RA-GCN is to discover sufficient redundant representation, while most actions can be recognized by only a few informative joints. However, when these informative joints are occluded or disturbed, the performance of traditional methods will deteriorate significantly.

F. Experimental Results on Occlusion Datasets

In this section, we will analyze the experimental results on the synthetic occlusion datasets based on the CS benchmark of NTU 60 dataset and the CSet benchmark of NTU 120 dataset. There are four types of occlusion given as follows:

a) Frame occlusions: This type of occlusion is constructed to simulate temporal occlusion. We randomly occlude a subsequence in first 100 frames, because the length of most samples is less than 100. The length of the subsequence is set to 10, 20, 30, 40, 50, respectively, and the experimental results are shown in Tab.V. It is observed that the proposed RA-GCN achieves a significant superiority to ST-GCN [10], SR-TSL

[13] and 2s-AGCN [29]. Besides, the difference between 3s RA-GCN and the baseline model shows a rising trend with the increasing number of occluded frames. As to the comparison of different numbers of streams, 3s RA-GCN is a little better than the others.

b) Part occlusions: The part occlusion aims at imitating the cases that some key parts of a person are occluded. The occluded parts 1, 2, 3, 4, 5 denote left arm, right arm, two hands, two legs and torso, respectively. As seen in Tab.VI, there is a huge gap between RA-GCN and other SOTA methods. Compared to the baseline model, 3s RA-GCN obtains large advantages when evaluating without parts 3 and 5. Moreover, 2s RA-GCN has a similar performance with the 3s model.

c) Block occlusions: Usually, the pedestrian would be occluded by contextual objects. For simulating this real scenario, we design the block occlusion experiments, which occlude the joints behind a predefined horizontal line. The

TABLE VIII
EXPERIMENTAL RESULTS (%) WITH RANDOM OCCLUSION ON THE CS BENCHMARK OF NTU 60 (TOP) AND THE CSET BENCHMARK OF NTU 120 (BOTTOM)

Random Occlusion	Occluded Probability					
	0	0.2	0.3	0.4	0.5	0.6
ST-GCN [10]	80.7	12.4	6.6	6.2	4.0	4.2
SR-TSL [13]	84.8	43.0	25.2	12.1	6.0	3.7
2s-AGCN [29]	**88.5**	38.5	22.8	13.4	8.5	6.1
preliminary version [23]	85.9	84.1	81.7	77.2	70.0	57.4
baseline (1s RA-GCN)	85.8	82.4	77.1	72.3	63.8	49.9
2s RA-GCN	86.7	85.2	**83.1**	**79.4**	**73.0**	60.1
3s RA-GCN	87.3	**85.4**	82.9	78.9	71.9	**61.1**
difference*	1.5	3.0	5.8	6.6	8.1	11.2
ST-GCN [10]	73.2	4.1	2.2	1.9	1.6	1.3
SR-TSL [13]	79.9	44.4	27.1	10.5	8.8	5.1
2s-AGCN [29]	**84.2**	20.1	9.4	6.3	4.6	3.7
preliminary version [23]	79.4	77.0	73.9	70.4	**62.6**	42.1
baseline (1s RA-GCN)	80.0	75.1	68.4	57.4	44.7	27.6
2s RA-GCN	82.5	79.7	**76.2**	**71.0**	62.0	**48.7**
3s RA-GCN	82.7	**79.8**	75.6	68.9	58.1	43.7
difference*	2.7	4.7	7.2	11.5	13.4	16.1

*: the difference between 3s RA-GCN and the baseline model

TABLE IX
EXPERIMENTAL RESULTS (%) WITH JITTERING SKELETONS ($\mu = 0$, $\sigma = 0.1$) ON THE CS BENCHMARK OF NTU 60 (TOP) AND THE CSET BENCHMARK OF NTU 120 (BOTTOM)

$\mu = 0$ $\sigma = 0.1$	Jittering Probability					
	0	0.02	0.04	0.06	0.08	0.10
ST-GCN [10]	80.7	66.4	44.1	32.7	13.3	7.0
SR-TSL [13]	84.8	70.4	53.2	41.0	33.9	21.4
2s-AGCN [29]	**88.5**	74.9	60.9	41.9	29.4	20.6
preliminary version [23]	85.9	73.2	59.8	45.3	41.6	34.5
baseline (1s RA-GCN)	85.8	84.1	66.1	34.2	22.2	13.9
2s RA-GCN	86.7	70.0	55.3	48.2	41.5	**36.4**
3s RA-GCN	87.3	**84.2**	**72.4**	**61.6**	**42.4**	28.7
difference*	1.5	0.1	6.3	27.4	20.2	14.8
ST-GCN [10]	73.2	63.4	50.2	33.7	18.6	10.3
SR-TSL [13]	79.9	60.3	50.9	39.2	30.7	19.6
2s-AGCN [29]	**84.2**	42.3	37.9	35.8	31.0	**23.7**
preliminary version [23]	79.4	78.1	**73.4**	**55.2**	26.9	17.2
baseline (1s RA-GCN)	80.0	72.2	35.9	12.6	7.0	5.4
2s RA-GCN	82.5	**79.3**	62.5	38.4	22.2	15.2
3s RA-GCN	82.7	77.8	65.7	47.9	**29.5**	20.5
difference*	2.7	5.6	29.8	35.3	22.5	15.1

*: the difference between 3s RA-GCN and the baseline model

TABLE X
EXPERIMENTAL RESULTS (%) WITH JITTERING SKELETONS ($\mu = 0$, $\sigma = 0.05$) ON THE CS BENCHMARK OF NTU 60 (TOP) AND THE CSET BENCHMARK OF NTU 120 (BOTTOM)

$\mu = 0$ $\sigma = 0.05$	Jittering Probability					
	0	0.02	0.04	0.06	0.08	0.10
ST-GCN [10]	80.7	76.4	65.1	50.2	32.8	19.5
SR-TSL [13]	84.8	69.4	55.3	50.1	46.6	39.2
2s-AGCN [29]	**88.5**	78.9	79.8	76.8	72.6	60.7
preliminary version [23]	85.9	83.8	81.3	75.3	69.2	**61.4**
baseline (1s RA-GCN)	85.8	82.4	77.1	72.3	63.8	49.9
2s RA-GCN	86.7	83.8	77.3	71.6	61.6	58.5
3s RA-GCN	87.3	**87.0**	**84.5**	**81.1**	72.9	**61.4**
difference*	1.5	4.6	7.4	8.8	9.1	11.5
ST-GCN [10]	73.2	70.4	64.0	59.5	44.0	32.1
SR-TSL [13]	79.9	68.2	55.4	47.9	41.3	33.6
2s-AGCN [29]	**84.2**	56.7	49.2	41.9	37.3	32.0
preliminary version [23]	79.4	79.0	78.7	**77.1**	**72.3**	**61.2**
baseline (1s RA-GCN)	80.0	79.4	71.9	46.4	25.2	14.3
2s RA-GCN	82.5	81.9	**79.7**	69.5	44.9	31.8
3s RA-GCN	82.7	**82.0**	79.6	74.2	62.1	48.5
difference*	2.7	2.6	7.7	27.8	36.9	34.2

*: the difference between 3s RA-GCN and the baseline model

height of the horizontal line is set to five ranges, and the fourth line of Fig.5 shows an example of range 3. Tab.VII presents these experimental results. Similar with the above occlusion experimental results, our RA-GCN obtains the best accuracies over most of experiments compared to other models.

d) Random occlusions: During the process of signal transmission, the transmitted data are prone to be lost. Thus, the purpose of random occlusion experiments is to imitate this situation. The occluded probability for every joint is set to 0.2, 0.3, 0.4, 0.5 and 0.6, respectively. The experimental results are displayed in Tab.VIII, from which we can see that RA-GCN extremely alleviates the performance deterioration, while ST-GCN [10], SR-TSL [13] and 2s-AGCN [29] have a rapid performance degradation. This phenomenon is caused mainly because these conventional graph-based models require an integrated graph structure, however, which is entirely destroyed by random occlusions.

e) Failure cases: Specially, we also find that when some important joints, such as right arm, are occluded, some action categories, e.g., *handshaking*, cannot be inferred by other joints. The proposed method will fail in such cases.

G. Experimental Results on Jittering Datasets

To discuss the impact of jittering skeletons, two jittering datasets are designed by adding different Gaussian noises, and the corresponding experimental results are shown in Tab.IX and Tab.X. The jittering probability for every joint is set to 0.02, 0.04, 0.06, 0.08 and 0.10, respectively. It is clearly observed from these tables that, the 3s RA-GCN outperforms other models by a huge gap on NTU 60-based jittering dataset. Moreover, with the increase of jittering probability, the gap between 3s RA-GCN and the baseline is synchronously increasing. Therefore, it is proven that our method is robustness to the current skeletons.

H. Results Analysis

In this section, we will analyze why our model is more robust to noisy or incomplete input data. Fig.6 shows an example of the activated joints. In this figure, the top line shows the activated joints of the baseline model, which is also considered as a 1s RA-GCN. The activated joints in the middle line correspond to a 2s RA-GCN, while the bottom line presents the results of a 3s RA-GCN. For a clearer display, we select five contextual frames from the sequence to represent the action *throwing*, instead of a whole sequence.

From this figure, it is observed that the proposed RA-GCN successfully expands the activation map of the baseline model. The baseline model only captures the most discriminative joints, e.g., the joints in two arms. In contrast, our model not only concentrates on the joints in two arms, but also activates some other discriminative joints, which play an auxiliary role in this action, such as the slightly swaying body and stepping
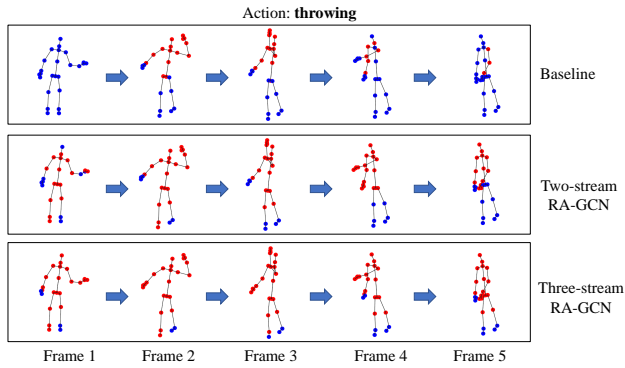
Fig. 6. (Best viewed in color.) An example of activated joints of all streams for the baseline model, the RA-GCN with two streams and three streams. The red points denote the activated joints, while the blue points denote the unactivated joints.

legs. Moreover, with the increase of streams, more activated joints can be discovered accordingly. Therefore, the multi-stream RA-GCN still has a modest recognition performance when a few joints are occluded or disturbed. This will lead to a more robust capability of our model to data degradation than other models.

In addition, it is worth to notice that more streams in the RA-GCN will not always obtain a more accurate model, because the number of discriminative joints in an action category is often limited. In this work, three streams are sufficient to discover these joints, and the experimental results in Section IV-D also demonstrates this point.

V. CONCLUSION

In this paper, to reduce the impact of noisy or incomplete skeletons in action recognition, we have proposed a novel model named RA-GCN for discovering rich features over all skeleton joints, which achieves a much better performance than the baseline model and improves the robustness of the model. With extensive experiments on the NTU RGB+D 60 & 120 datasets, we verify the effectiveness and robustness of our model. For evaluating the model's performance on non-standard skeletons, we construct various synthetic datasets composed of four types of occlusion and two types of jittering. On these synthetic datasets, the proposed RA-GCN outperforms the other SOTA methods, as well as showing a significant improvement than the baseline model. In the future, we will consider to add the attention module within our model, in order to make each stream focus more on informative joints.

ACKNOWLEDGEMENT

REFERENCES

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, p. 16, 2011.

[2] M. Sudha, K. Sriraghav, S. G. Jacob, S. Manisha *et al.*, "Approaches and applications of virtual reality and gesture recognition: A review," vol. 8, no. 4, pp. 1–18, 2017.

[3] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Comput. Vis. Image Understand*, vol. 115, no. 2, pp. 224–241, 2011.

[4] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv:1511.04119*, 2015.

[5] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 4489–4497.

[6] A. Jalal, Y.-H. Kim, Y.-J. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatiotemporal multi-fused features," *Pattern Recognit.*, vol. 61, pp. 295–308, 2017.

[7] A. Akula, A. K. Shah, and R. Ghosh, "Deep learning approach for human action recognition in infrared images," *Cogn. Syst. Res.*, vol. 50, pp. 146–154, 2018.

[8] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 816–833.

[9] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2117–2126.

[10] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI Conf. Artif. Intell.*, 2018.

[11] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 7291–7299.

[12] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 1012–1020.

[13] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, "Skeleton-based action recognition with spatial reasoning and temporal stack learning," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 103–118.

[14] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *IEEE Int. Conf. Multimedia Expo Workshop (ICMEW)*. IEEE, 2017, pp. 597–600.

[15] T. Soo Kim and A. Reiter, "Interpretable 3d human action analysis with temporal convolutional networks," in *IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2017, pp. 20–28.

[16] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *IJCAI Int. Joint Conf. Artif. Intell.*, 2018.

[17] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, "Deep progressive reinforcement learning for skeleton-based action recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 5323–5332.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv:1609.02907*, 2016.

[19] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Inf. Fusion*, vol. 37, pp. 132–156, 2017.

[20] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2921–2929.

[21] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1010–1019.

[22] J. Liu, A. Shahroudy, M. L. Perez, G. Wang, L.-Y. Duan, and A. K. Chichung, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[23] Y.-F. Song, Z. Zhang, and L. Wang, "Richly activated graph convolutional network for action recognition with incomplete skeletons," in *IEEE Int. Conf. Image Process (ICIP)*, 2019.

[24] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception & Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[25] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1110–1118.

[26] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *AAAI Conf. Artif. Intell.*, 2017.

[27] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.

[28] K. Thakkar and P. Narayanan, "Part-based graph convolutional network for action recognition," in *Br. Mach. Vis. Conf. (BMVC)*, 2018.

[29] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.

[30] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *IEEE Int. Conf. Comput. Vis. (ICCV)*. IEEE, 2009, pp. 32–39.

[31] D. Weinland, M. Özuysal, and P. Fua, "Making action recognition robust to occlusions and viewpoint changes," in *Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2010, pp. 635–648.

[32] Q. Hou, P. Jiang, Y. Wei, and M.-M. Cheng, "Self-erasing network for integral object attention," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 549–559.

[33] X. Liu, Z. Wang, J. Shao, X. Wang, and H. Li, "Improving referring expression grounding with cross-modal attention-guided erasing," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.

[34] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.

[35] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 20–36.

[36] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang, "Deep multimodal feature analysis for action recognition in rgb+ d videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1045–1058, 2017.

[37] D. C. Luvizon, D. Picard, and H. Tabia, "2d/3d pose estimation and action recognition using multitask deep learning," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 5137–5146.

[38] F. Baradel, C. Wolf, J. Mille, and G. W. Taylor, "Glimpse clouds: Human activity recognition from unstructured feature points," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 469–478.

[39] B. Li, Y. Dai, X. Cheng, H. Chen, Y. Lin, and M. He, "Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn," in *IEEE Int. Conf. Multimedia Expo Workshop (ICMEW)*, 2017, pp. 597–600.

[40] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 3595–3603.

[41] L. Huang, Y. Huang, W. Ouyang, and L. Wang, "Part-level graph convolutional network for skeleton-based action recognition," in *AAAI Conf. Artif. Intell.*, 2020.

[42] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," in *AAAI Conf. Artif. Intell.*, 2020.

[43] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, J. Lai, and J. Zhang, "Early action prediction by soft regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2568–2583, 2018.

[44] M. Liu and J. Yuan, "Recognizing human actions as the evolution of pose estimation maps," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 1159–1168.

[45] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention lstm networks," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1586–1599, 2017.

[46] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognit.*, 2017.

[47] J. Liu, A. Shahroudy, G. Wang, L. Duan, and A. C. Kot, "Skeleton-based online action prediction using scale selection network," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[48] C. Caetano, J. Sena, F. Brémond, J. A. Dos Santos, and W. R. Schwartz, "Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition," in *IEEE Conf. Adv. Video Signal Based Surveill. (AVSS)*. IEEE, 2019, pp. 1–8.

[49] C. Caetano, F. Brémond, and W. R. Schwartz, "Skeleton image representation for 3d action recognition based on tree structure and reference joints," in *SIBGRAPI Conf. Graph., Patterns Images*. IEEE, 2019, pp. 16–23.

[50] K. Papadopoulos, E. Ghorbel, D. Aouada, and B. Ottersten, "Vertex feature encoding and hierarchical temporal modeling in a spatial-temporal graph convolutional network for action recognition," *arXiv:1912.09745*, 2019.

Yi-Fan Song received his M.S. degree in Zhengzhou University, Zhengzhou, China, in 2018. He is currently working toward the Ph.D. degree in the Center for Research on Intelligent Perception and Computing (CRIPAC), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. His research interests include computer vision, activity recognition, video surveillance, and time series analysis.

Zhang Zhang received the B.S. degree in computer science and technology from Hebei University of Technology, Tianjin, China, in 2002, and the Ph.D. degree in pattern recognition and intelligent systems from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China in 2009. Currently, he is an associate professor at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA). His research interests include activity recognition, video surveillance, and time series analysis. He has published 20s research papers on computer vision and pattern recognition, including IEEE TPAMI, CVPR, and ECCV etc.

Caifeng Shan received the B.Eng. degree from the University of Science and Technology of China (USTC), the M.Eng degree from the Institute of Automation, Chinese Academy of Sciences, and the PhD degree in computer vision from Queen Mary, University of London. His research interests include computer vision, pattern recognition, image and video analysis, machine learning, bio-medical imaging, and related applications. He has authored more than 100 papers and 60 patent applications. He has served as Associate Editor or Guest Editor for many scientific journals including IEEE Transactions on Circuits and Systems for Video Technology and IEEE Journal of Biomedical and Health Informatics. He is Senior Member of IEEE.

Liang Wang received both the BEng and MEng degrees from Anhui University in 1997 and 2000, respectively, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences (CASIA) in 2004. From 2004 to 2010,he was a research assistant at Imperial College London, United Kingdom, and Monash University, Australia, a research fellow at the University of Melbourne, Australia, and a lecturer at the University of Bath, United Kingdom, respectively. Currently, he is a full professor of the Hundred Talents Program at the National Lab of Pattern Recognition, CASIA. His major research interests include machine learning, pattern recognition, and computer vision. He has widely published in highly ranked international journals such as IEEE Transactions on Pattern Analysis and Machine Intelligence and IEEE Transactions on Image Processing, and leading international conferences such as CVPR, ICCV, and ICDM. He is an IEEE Fellow, and an IAPR Fellow.