Channel Attention and Multi-Scale Graph Neural Networks for Skeleton-Based Action Recognition

Ronghao Dang [a], Chengju Liu [a,b,*], Ming Liu [c] and Qijun Chen [a]

[a] *Department of Control Science and Engineering, Tongji University, Shanghai 201804, China*
E-mails: dangronghao@tongji.edu.cn, qjchen@tongji.edu.cn
[b] *Tongji Artificial Intelligence (Suzhou) Research Institute, Suzhou 215000, China*
E-mail: liuchengju@tongji.edu.cn
[c] *Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong*
E-mail: eelium@ust.hk

Abstract. 3D skeleton data has been widely used in action recognition as the skeleton-based method has achieved good performance in complex dynamic environments. The rise of spatio-temporal graph convolutions has attracted much attention to use graph convolution to extract spatial and temporal features together in the field of skeleton-based action recognition. However, due to the huge difference in the focus of spatial and temporal features, it is difficult to improve the efficiency of extracting the spatiotemporal features. In this paper, we propose a channel attention and multi-scale neural network (CA-MSN) for skeleton-based action recognition with a series of spatio-temporal extraction modules. We exploit the relationship of body joints hierarchically through two modules, i.e., a spatial module which uses the residual GCN network with the channel attention block to extract the high-level spatial features, and a temporal module which uses the multi-scale TCN network to extract the temporal features at different scales. We perform extensive experiments on both the NTU-RGBD60 and NTU-RGBD120 datasets to verify the effectiveness of our network. The comparison results show that our method achieves the state-of-the-art performance with the competitive computing speed. In order to test the application effect of our CA-MSN model, we design a multi-task tandem network consisting of 2D pose estimation, 2D to 3D pose regression and skeleton action recognition model. The end-to-end (RGB video-to-action type) recognition effect is demonstrated. The code is available at https://github.com/Rh-Dang/CA-MSN-action-recognition.git.

Keywords: skeleton, action recognition, channel attention, graph neural networks, multi-scale

1. Introduction

Human action recognition is an important research direction in the field of computer vision. It has wide application scenarios and market value, such as abnormal behavior monitoring [1–3], human-computer interaction [4], etc. In particular, skeleton-based human action recognition methods combined with depth estimation technology [5, 6] have attracted increasing attention from researchers. A skeleton sequence is a kind of abstract human body movement data, which uses joint types, 3D joint coordinates and joint connections to express the movement of various body parts. Compared to RGB videos, skeleton data also has the following advantages. First, the cost of obtaining human skeleton data has become lower with the development of pose estimation technology and depth cameras. Second, the skeleton data can reduce the overfitting problem in network training and the network's coupling

[*] Corresponding author. E-mail: liuchengju@tongji.edu.cn.

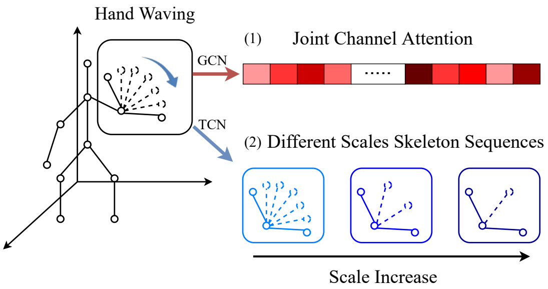

Hand Waving
GCN
TCN
(1) Joint Channel Attention
.....
(2) Different Scales Skeleton Sequences
Scale Increase

Fig. 1. The characteristics of the CA-MSN skeleton action recognition model: (1) Using channel attention GCN to extract joint relationships. The intensity of the color indicates the channel importance; (2) Using multi-scale TCN to model time series.

to subjects' appearances. Third, the skeleton sequence can more intuitively show the movement of various body parts by using the graph topology representation for joints. Fourth, the skeleton data eliminates environmental noise (e.g., background, clothing, brightness) so that neural networks can focus more on modeling human movements and reduce the cost of feature extraction. Furthermore, skeleton-based action recognition can be used as a supplement to RGB-based action recognition, thereby increasing the information richness and improving the overall recognition accuracy. In this work, we focus on skeleton-based action recognition.

There are three basic directions for performing skeleton-based action recognition: based on Recurrent Neural Networks (RNN) [7–14], based on Convolutional Neural Networks (CNN) [15–21], based on Graph Convolutional Networks (GCN) [22–38], and based on two of the above methods [39–47]. RNN-based approaches mainly use models such as LSTM/GRU to model the dynamic changes of the skeleton sequence. However, RNN methods only arrange the joint coordinates into a vector in a certain order and then input it into the recurrent neural network. The important structural information is ignored since the different joint types and connections are not distinguished. CNN-based approaches organize the joint coordinates to a 2D map. The 3D coordinates (x, y, z) are analogous to the three channels (R, G, B) in an image, and the number of frames and joints are analogous to the image length and width. Therefore, the 2D CNN can be used to extract the spatio-temporal combined features from the skeleton. With such a data organization as input, it is difficult to express the topological structure and connection relationship between joints. In recent years, with the rise of graph neural networks, people have gradually discovered the importance and potential of the graph-structured data. GCN-based approaches fully utilize the spatial structured information of the skeleton joints. Further, some methods go beyond the natural skeleton connection and model the implicit connection between joints. Yan et al. [22] first apply GCNs to model skeleton data. They add temporal edges between corresponding joints in consecutive frames and propose a distance-based sampling function to construct a graph convolutional layer. However, recent studies have found that networks which extract spatial and temporal information independently such as SGN [47] DGNN [29] MS-G3D [45], etc. perform better than ST-GCN [22] and 2s-AGCN [28], etc. which use GCN to directly learn spatio-temporal feature representations. Inspired by their works, we utilize GCN and muti-scale TCN to extract spatial and temporal features, respectively, and concatenate the both to use.

Many recent studies have focused on exploring the implicit connection between distant joints, such as the relationship between arm swing and foot swing when walking. The two-stream adaptive graph convolution network (2s-AGCN) [28] and actional-structural graph convolution network (AS-GCN) [27] invented the adaptive graph structure. In this structure, the adjacency matrix is not limited to natural bone connections but adaptively explores each joint's correlation as the dataset changes. However, adaptive graph approaches only focus on exploring the correlation between the spatial joints and do not explicitly model interdependencies between the channels. For actions such as "waving hand", the channels about the body's frontal plane are more important than the channels about the body's median plane.

In terms of time series modeling, researchers mostly use Long Short-Term Memory (LSTM) network or Temporal Convolution Network (TCN). It is challenging for LSTM to learn the long-distance temporal correlations considering problems like losing effective features and vanishing gradient. TCN is the mainstream CNN method for modeling skeleton sequences, but the lack of time series direction and the single scale make it difficult to guarantee the richness and comprehensiveness of the information extracted by TCN.

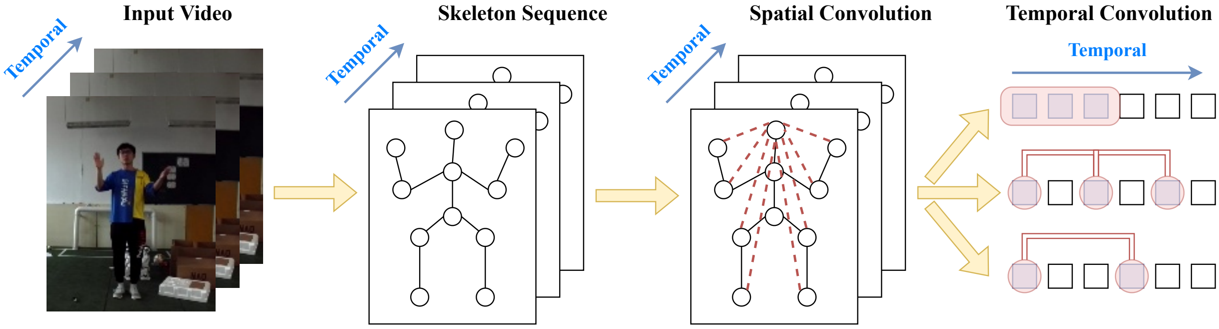


Fig. 2. Model overview. First, the skeleton sequence is extracted from the original video. Then, the information transfer in both spatial and temporal directions is separated.

In this work, we address the above limitations from two aspects (Figure 1). First, we introduce the channel attention mechanism in the GCN module to model interdependencies between the joint channels. Therefore, the GCN spatial feature extraction module can focus on important semantics more efficiently. Second, we use dilated convolutions [48] with different dilation rates to process time series in parallel, and adopt the technique of deep concatenation in [49] to achieve the fusion of different receptive fields. This leads to a more powerful network that can not only model long-term skeleton actions, but also recognize short-term repetitive actions such as clapping hands. We also incorporate the joint type and frame index to the network [47], so that the joint connections and the temporal sequences both have directionality. Besides, the skeleton sequence's dynamics (position/3D coordinates and velocity) are input into the network to make periodic characteristics of some actions are also merged into the input features. In the end, we propose the powerful channel attention and multi-scale neural network, named CA-MSN. We illustrate the overall model architecture in Figure 2.

To verify the effectiveness of the proposed CA-MSN, we conduct extensive experiments on two large-scale datasets: NTU-RGBD60 [9] and NTU-RGBD120 [45]. The experiments have demonstrated that the channel attention GCN and the multi-scale TCN can significantly improve network accuracy. For better application in real life, a multi-task tandem network is designed to realize the complete action recognition process (RGB video, 2D pose, 3D pose, action type). We summarize our main contributions as follows:

- We propose a channel attention mechanism (CA-GCN) for graph convolutional networks, which effectively models the relationship between joint feature channels and improves the spatial feature extractor's performance.
- We propose a multi-scale temporal feature extraction scheme (MS-TCN) for skeleton sequence, so that both long-term continuous actions and short-term repetitive actions can be precisely classified.
- We connect the CA-GCN and MS-TCN modules in series to form a powerful skeleton action recognition network: CA-MSN, which shows the state-of-the-art performance on the NTU-RGBD60 and NTU-RGBD120 datasets.
- We design a complete end-to-end network from RGB videos to actions, so that the effect of our proposed CA-MSN model in the application process can be displayed.

2. Relative works

2.1. 3D Skeleton Action Recognition

With the rise of deep learning and neural networks, end-to-end approaches are more competitive than traditional approaches that use hand-crafted features in the field of 3D skeleton action recognition. Most of the earliest end-to-end approaches use recurrent neural networks such as LSTM/GRU. Du et al. [7] divide the human skeleton into five parts according to the human's physical structure and then separately feed them to five subnets. Zhu et al. [8] take the skeleton as the input at each time slot and introduce a novel regularization scheme to learn the skeleton joints' co-occurrence features. Inspired by the skeleton graphical structure, Liu et al. [10] propose a more powerful tree-structure-based traversal method. After that, CNN approaches gradually emerge and are widely used in 3D skeleton action recognition. Kim et al. [15] and Liu et al. [18] use the CNN characteristics to explicitly learn interpretable action spatio-temporal representations and visualize them. Ke et al. [17] and Le et al. [19] exploit the correlations between the different time periods of a skeleton sequence. In 2018, ST-GCN [22] sets a precedent for using graph neural network methods to process skeleton sequences. After that, the GCN method gradually becomes the mainstream method in the field of skeleton action recognition. Li et al. [27] and Shi et al. [28] make the topology of the graph model can capture implicit joint correlations. Shi et al. [29] also represent the skeleton data as a directed acyclic graph (DAG) based on the dependency between the joints and bones in the natural human body. Peng et al. [50] propose the first automatically designed GCN for skeleton-based action recognition. However, the huge computational complexity brought by the graph convolution network method is challenging to solve. Recently, in order to combine the advantages of RNN, CNN and GCN networks, researchers use GCN module to extract the topological relationship and RNN or CNN to model time series [43, 45–47].

2.2. Attention Mechanism in Computer Vision

The attention mechanism was first produced in NLP and then widely used in the field of computer vision. Its basic idea is to teach the network to ignore the irrelevant information and focus on the key information. After the years of development, the attention mechanism is mainly classified into three types: spatial attention [51, 52], channel attention [53, 54], spatial and channel hybrid attention [55].

In most computer vision problems, only task-related areas need to be concerned, such as the subject in classification tasks. The spatial attention allows the network to focus more on essential spatial areas. Spatial Transformer Network (STN) [51] proposed by Google DeepMind is the most representative spatial attention network. Different from the single-stage STN, Dynamic Capacity Network (DCN) [52] uses two sub-networks: low-capacity network and high-capacity network. Low-capacity network is used to process the entire image and locate the region of interest. High-capacity network refines the region of interest.

For the feature maps in CNN, the modeling of channel dimensions is also crucial. Squeeze-and-Excitation Networks (SENet) [53] learn the importance of each channel and then enhance or suppress different channels according to different inputs. Furthermore, Selective Kernel Networks (SKNet) [54] and other methods combine such channel weighting idea with the multi-branch network structure to improve the network performance.

Convolutional Block Attention Module (CBAM) [55] is a representative network of the spatial and channel attention hybrid mechanisms. The channel dimension utilizes both the max pooling outputs and average pooling outputs with a shared network. Next, the both outputs are merged using element-wise summation. The spatial dimension also uses the max pooling and average pooling to concatenate a feature map, and then uses convolutional layers for learning. In addition, there are many researches related to the attention mechanism, such as residual attention, multi-scale attention, recursive attention, etc.

3. Channel Attention and Multi-Scale Neural Networks

Channel attention-aware spatial modeling and multi-scale temporal modeling are the two main modules of our network. For some simple actions with clear direction, the channel attention mechanism can make the GCN network focus more on channels with rich information. The multi-scale temporal modeling makes the skeleton action recognition network more adaptable to actions with different velocities. A skeleton-based video is a sequence of frames formulated as $F = \{F_1, F_2, \cdots, F_T\}$, where T denotes the length of the video. A skeleton-based frame can be formulated as a directed graph $F_t = \{V_t, \xi_t\}$, where V_t is a set of skeleton joints and ξ_t is a set of directed edges (bones). $V_t = \{v_t^1, v_t^2, \cdots, v_t^K\}$ shows that there are K joints in the skeleton, and the 3D coordinate of the k^{th} joints at the t frame is expressed as v_t^k. The data stream and symbolic representation of the overall model are shown in detail in Figure 3.

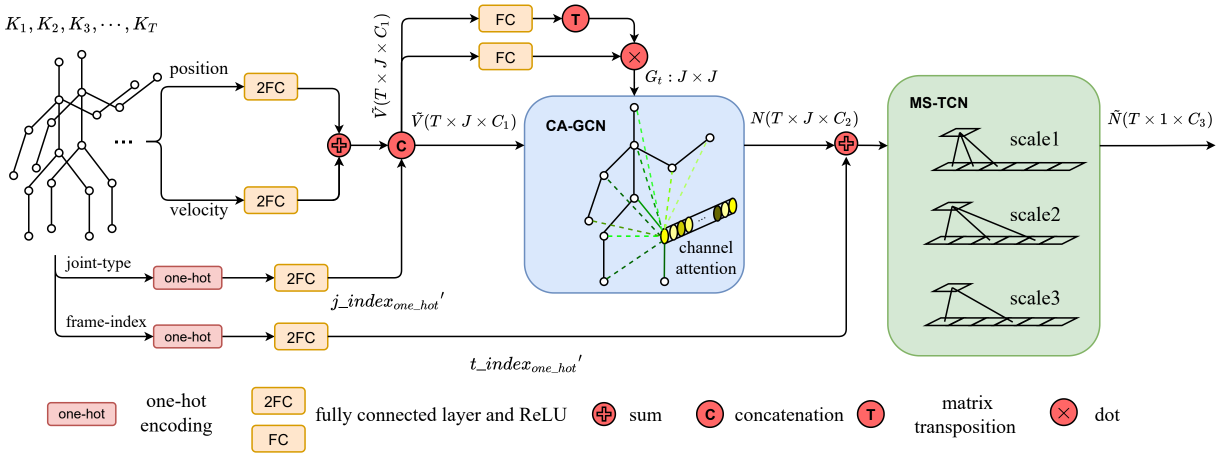


Fig. 3. This figure shows the architecture of the proposed channel attention and multi-scale neural networks (CA-MSN). Input is the addition of position and velocity. Joint-type and frame-index are separately incorporated before CA-GCN and MS-TCN. CA-GCN can learn the spatial relationship between joints and the dependencies between channels. MS-TCN can aggregate features of different temporal scales.

3.1. Spatial Feature Extraction

3.1.1. Encoding Input Features

The 3D coordinate of the skeleton sequence v_t^k is the initial input of the network. But for action categories that have a strong periodicity, v_t^k is cyclical in the time dimension. When predicting a set of time series, it is necessary to ensure that the statistical properties of the time series are invariant to time translation, so we need to stabilize the periodic series. Differencing can smooth out the mean of the time series by removing some of the changing features, and thus remove (or reduce) the trend and periodicity of the time series. In our model, the joint velocity $v_t^k - v_{t-1}^k$ at each frame is used to stabilize the time series of skeleton actions. To impart directionality to the bones, we introduce the joint type JT into the network, which is encoded by the one-hot method. Aggregating the above three inputs can be formulated as:

$$\tilde{v_t^k} = cat({v_t^k}' + (v_t^k - v_{t-1}^k)', JT') \in \mathbb{R}^{C_1} \tag{1}$$

where ${v_t^k}'$, $(v_t^k - v_{t-1}^k)'$ and JT' are all encoded by two fully connected (FC) with ReLU layers before concatenated. C_1 is the dimension of the final input representation. *cat* denotes concatenating features in channel dimension when the other dimensions are equal.

3.1.2. Adjacency Matrix

Since the natural connection between joints cannot fully represent the coupling relationship between skeleton joints during actions, we need to recalculate the adjacency matrix to represent the dynamic correlation weight. There are currently three mainstream methods for obtaining an adjacency matrix:

- `Inner product`: This method directly uses the similarity between the joint features to calculate the connection weight between the joints.
- `Bi-linear form`: This method uses a linear mapping on one of the feature vectors before calculating the inner product.
- `Trainable relevance score`: This method fully uses the neural network layer to learn the adjacency matrix.

The pure inner product method is too simple to explore the potential connection relationship between joints and does not have data-driven characteristics. Therefore, this work uses the adaptive learning method to obtain the adjacency matrix:

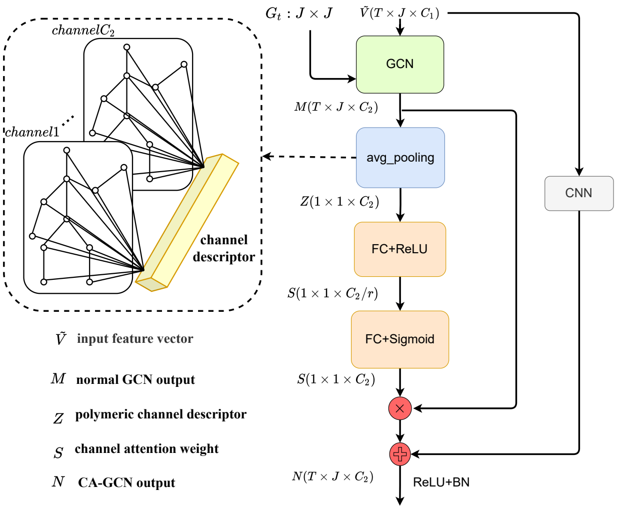

$G_t : J \times J$
$\tilde{V}(T \times J \times C_1)$
GCN
$M(T \times J \times C_2)$
avg_pooling
$Z(1 \times 1 \times C_2)$
FC+ReLU
$S(1 \times 1 \times C_2/r)$
FC+Sigmoid
$S(1 \times 1 \times C_2)$
$N(T \times J \times C_2)$
ReLU+BN
CNN
$channelC_2$
$channel1$
channel descriptor
$\tilde{V}$ input feature vector
M normal GCN output
Z polymeric channel descriptor
S channel attention weight
N CA-GCN output

Fig. 4. CA-GCN block can model the potential relationship among joint channel features by adding a channel feature learning module after graph convolution. Before learning the channel weights, average pooling is used to extract the channel descriptor. The recalculated output is obtained by multiplying each channel by the corresponding weight.

$$G_t(i, j) = softmax((W_1 \tilde{v_t^i} + b_1)^T \cdot (W_2 \tilde{v_t^j} + b_2)) \tag{2}$$

We first use the FC layer to set learnable parameters $W_1 \in \mathbb{R}^{C_1 \times 2C_1}$, $W_2 \in \mathbb{R}^{C_1 \times 2C_1}$ for the process of calculating the adjacency matrix. Then, the inner product is used to calculate the connection weight between every two joints. SoftMax is implemented using degree matrix D, $G_t = D^{-1/2} C_t D^{-1/2}$ normalizes C_t, so that the total weight of each joint to other joints is 1.

3.1.3. Channel Attention GCN

Using the normalized adjacency matrix G_t and the encoded joint features $\tilde{V}_t$, combined with the graph convolutional network (GCN), messages can be transferred between joints:

$$M_t = G_t \tilde{V}_t W_g \tag{3}$$

where $W_g \in \mathbb{R}^{C_1 \times C_2}$ contains the learnable parameters in the GCN. We can get $M = \{M_1, M_2, \cdots, M_T\}$ by performing the above formula on each frame. However, the traditional GCN only models the spatial relationship between joints, and the interdependencies between channels are not fully explored. To solve this problem, we propose the CA-GCN block which contains the channel attention module, as shown in Figure 4.

Firstly, we squeeze the global spatial and temporal information into a channel descriptor using global average pooling:

$$z = F_{sq}(M) = \frac{1}{T \times K} \sum_{i=1}^{T} \sum_{j=1}^{K} M_i^j \tag{4}$$

where $z \in \mathbb{R}^{C_2}$ contains the feature representation of each channel. After that, we use the adaptive recalibration method in SENet [53] to excite the channel features. The excitation module has two advantages: first it is flexible and computationally small, second it can learn a non-mutually-exclusive relationship:

$$s = F_{ex}(z, W) = \sigma(W_3 \delta(W_4 z)) \tag{5}$$

where δ and σ refers to the ReLU [56] and Sigmoid function, $W_3 \in \mathbb{R}^{\frac{C_2}{r} \times C_2}$ and $W_4 \in \mathbb{R}^{C_2 \times \frac{C_2}{r}}$ are trainable parameters. In order to limit the computational complexity, we only use two FC layers around the non-linearity to

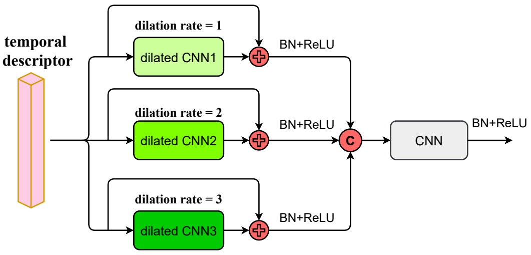


Fig. 5. The structures of MS-TCN unit. Compared with a single-scale temporal series classifier, the MS-TCN can robust to actions of different temporal scales.

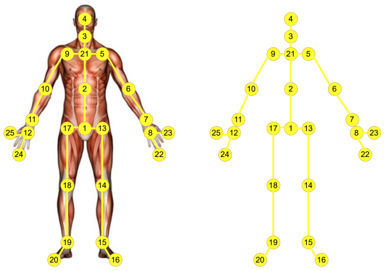


Fig. 6. Illustration of the human skeleton graphs on NTU-RGBD dataset.

model the channel attention, i.e. a dimensionality-reduction layer with reduction ratio r and then a dimensionality-increasing layer returning to the input channel dimension. We can adjust r to balance the accuracy and the computation complexity. Finally, the weight of each channel trained is multiplied by the skeleton feature M after the node massages exchange:

$$m_c{}' = F_{scale}(s_c, m_c) = s_c m_c \tag{6}$$

where $M' = \{m_1{}', m_2{}', \cdots, m_{C_2}{}'\}$, $m_c \in \mathbb{R}^{T\times K}$ is every channel's feature map, s_c is the channel's weight. Finally, add the residual connection:

$$N = M_t{}' + \tilde{V}_t W_v \tag{7}$$

W_v upgrade the original input features $\tilde{V}_t$ to be the same dimension as $M_t{}'$. We superimpose multiple channel attention GCN modules to model the spatial relationship between joints and the interdependencies relationship between channels.

3.2. Multi-Scale TCN

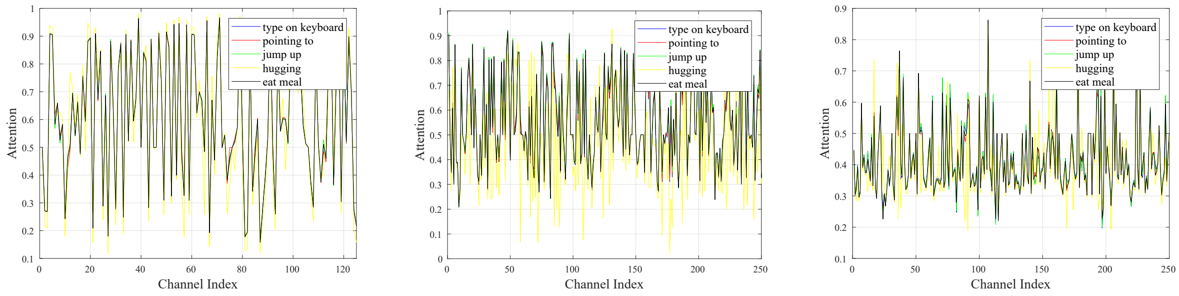
To extract the skeleton sequence features of different temporal scales, we use multi-scale TCN as shown in Figure 5 to model the time series. First, we merge the one-hot encoded frame index: $n_t^k = n_t^k + FI' \in \mathbb{R}^{C_2}$. The encoding approach of the frame index (FI) is the same as the joint type (JT) in section 3.1.1. After Max-Pooling the spatial dimension, we get $N \in \mathbb{R}^{T\times 1\times C_2}$. The calculation of each branch is as the follow:

$$\tilde{N_g} = A_g(\delta(A_g(N))) + N \tag{8}$$

where A_g is a dilated convolution layer, g is the dilation rate. We set different g for different branches to obtain different sizes of receptive fields without increasing the number of parameters. It is worth noting that each branch has a residual connection. Next, the different scale features learned by the branches with different receptive fields are aggregated $\tilde{N} = cat(\tilde{N_1}, \tilde{N_2}, \tilde{N_3}) \in \mathbb{R}^{T\times 1\times 3C_2}$. After the concatenation, we use a pointwise convolutional layer to fuse the features of the three different scale branches to C_3 dimensions. Finally, the Max-Pooling layer aggregates all frames' feature representations followed by a fully connected layer and then a Softmax layer to make the prediction.

4. Experiments

To prove the effectiveness of our method, we conduct extensive experiments on two skeleton-based action recognition datasets: NTU-RGBD60 [9] and NTU-RGBD120 [57]. We first perform exhaustive ablation studies to verify the efficiency and capacity of our proposed channel attention graph convolutional networks (CA-GCN) and multi-scale temporal convolutional networks (MS-TCN) on the NTU-RGBD60 dataset. Finally, the network is evaluated on NTU-RGBD60 and NTU-RGBD120 datasets to compare with the other state-of-the-art approaches.



(a) CA_GCN_1

(b) CA_GCN_2

(c) CA_GCN_3

Fig. 7. Activations induced by excitation operator in CA-GCN modules of different depths and different classes. Each set of activations is named according to the following scheme: CA_GCN_blockID.

4.1. Datasets

NTU-RGBD60 Dataset (NTU60) [9]. This dataset consists of 56,880 action samples, including each sample's RGB video, depth map sequence, 3D skeleton data and infrared video. This dataset utilizes 3 Microsoft Kinect v2 RGB-D cameras to capture RGB and depth videos simultaneously. The RGB video resolution is 1920×1080, and the depth map and infrared video are both 512×424. The 3D skeleton data contains the 3D positions of 25 main body joints per frame, as shown in Figure 6. Skeleton tracking technology establishes the coordinates of various joints by processing the depth data. There are two cross-validation methods, Cross Subject (CS) and Cross View (CV). For CS setting, half of the 40 subjects are used for training and the rest for testing. For CV setting, two of three cameras are used for training and the rest for testing.

NTU-RGBD120 Dataset (NTU120) [57]. This dataset is an extension of the NTU-RGBD60 action recognition dataset with a total of 114480 action samples. The data structure is similar to NTU-RGBD60, while it expands the 60 action categories in NTU-RGBD60 to 120 action categories performed by 106 subjects. There are two cross-validation methods, Cross Subject (C-Subject) and Cross Setup (C-Setup). For C-Subject setting, half of the 106 subjects are used for training and the rest for testing. For C-Setup setting, half of the setups are used for training and the rest for testing.

All models are trained with the same batch size (64), learning schedule (adam with an learning rate as 0.001 and reduced by 10 in epoch 60, 100, 120), and training epoch (140) with the Pytorch framework on a workstation with an AMD Ryzen 9 3900XT CPU, an NVIDIA RTX 3090 GPU, and 64 GB of ECC RAM. Before the experiments, we preprocess the initial skeleton sequences. Similar to [29], in order to eliminate the falsely detected body skeleton, we first determine that the body energy is the summation of the skeleton's standard deviation across each channel. Then we choose the top two skeletons with the most energy. According to [10], we split the raw video into 20 clips and randomly select a frame in each clip to compose a sequence with 20 frames. Finally, we center the skeleton in every frame to eliminate the influence of subject's position.

4.2. Ablation Study

In this section, we examine the effectiveness of the proposed channel attention GCN block, multi-scale TCN block and their related components. Moreover, in order to have a deeper understanding of the channel attention in skeleton action recognition, we deeply analyze the internal attention distribution and the effect on different actions. To ensure no serious over-fitting and under-fitting problems, the dropout rate is adjusted accordingly in each experiment.

Table 1
Verify the difference between introducing semantic information in different ways

Model	#Params ($\times 10^6$)	Accuracy	
		CS (%)	CV (%)
w/o JT w/o FI	1.3	87.3	92.9
add JT w/o FI	1.32	87.7	93.3
cat JT w/o FI	1.36	88.7	94.0
w/o JT add FI	1.32	87.9	93.4
w/o JT cat FI	1.45	88.1	93.3
cat JT add FI	1.38	89.2	94.3

4.2.1. Semantic of Joint Type and Frame Index

In section 3, we introduce the joint type (JT) and the frame index (FI) into the network by adding and concatenating, separately. The both different processing ways are based on the following analysis and experiments. Above all, to our knowledge the joint type is more important than the frame index. Because the MS-TCN has implicitly encoded the order of the sequence which is strengthened by the FI. Therefore, adding FI into the network is sufficient for this purpose and save the amount of calculation. In contrast, if there is no JT in the GCN network, it is completely impossible to distinguish the joint type, which is crucial for the overall network. The input concatenated with the JT can have richer semantic information and stronger expression ability. The experiments in Table 1 verify the above conclusions. w/o denotes without this semantic information, add denotes this semantic information is added into the network, cat denotes this semantic information is concatenated into the network.

For the joint type, adding into the network brings the performance improvement of 0.4% and 0.4% in the accuracy of the CS and CV settings, concatenating into the network brings the performance improvement of 1.4% and 1.1% in the accuracy of the CS and CV settings. It is obvious that when the difference in the number of parameters is only 0.04M, the effect of concatenation method is much better than add method, so we choose the concatenation method to fuse the joint type and features. For the FI, adding into the network brings the performance improvement of 0.6% and 0.5% in the accuracy of the CS and CV settings, concatenating into the network brings the performance improvement of 0.8% and 0.4% in the accuracy of the CS and CV settings. The concatenation method increases the 0.13M parameters compared with the add method while the accuracy is almost not improved, so we choose the add method to fuse the frame index and features.

4.2.2. Effectiveness of Channel Attention GCN

As shown in Table 2, we use three series GCNs without graph channel attention mechanism as the baseline for spatial modeling. By adding the graph channel attention mechanism to GCN at different depths, it can be shown that how to maximize the graph channel attention's performance with the fewer number of parameters. For example, All CA-GCN means that all three GCN modules use graph channel attention, and First CA-GCN means that only the first GCN module uses graph channel attention. Comparing the results of experiments, it shows that only adding graph channel attention to the first GCN module can not only use a smaller number of parameters, but also have the same accuracy as all three GCNs using graph channel attention. The results inspire us that simply superimposing graph channel attention cannot continuously improve the performance of the overall network. Only when channel interdependencies need to be considered, the graph channel attention can work.

4.2.3. Visualization of Channel Attention Output

In order to explain the results of the experiments in Table 2 and understand the role of the excitation operator in CA-GCN more clearly, in this section, we visualize the channel activation distribution of different network depths and different action classes.

Above all, we observe the difference in channel activation between different actions. Specifically, we sample five categories from the NTU-RGBD60 dataset: type on a keyboard, point to, jump up, hug and eat meal. We select 64 samples from the five action classes to generate their channel activation weights, and then average these activations of each channel to plot Figure 7.

Table 2

Effectiveness of CA-GCN and MS-TCN on the NTU-RGBD60 dataset

Model	#Params ($\times 10^6$)	Accuracy	
		CS (%)	CV (%)
Baseline	0.69	87.9	93.6
+ All CA-GCN	0.99	88.6	94.3
+ First CA-GCN	0.72	88.6	94.3
+ Second CA-GCN	0.82	88.5	94.3
+ Third CA-GCN	0.82	88.5	94.5
+ MS-TCN	1.35	88.7	93.9
+ First CA-GCN + MS-TCN	1.38	89.2	94.3

We make the following two observations about the role of channel attention. Firstly, in the five classes of actions, it can be found that there is a clear difference between the two-player action (hugging) and the other four single-player actions in the line chart. Visualization results show that the channel attention mechanism can clearly distinguish single-player actions and multi-player actions. Secondly, the channel activations of the four single-player actions are basically the same. Similar to the findings of different pictures in [54], the earlier layer features in different single-player skeleton actions are usually general. Only the deeper level features can effectively distinguish the single-player skeleton actions.

Next, we compare the output of the excitation operator in different depth CA-GCN. We find that as the feature level deepens, the discrimination of channel activation between different channels decreases (for all categories), which is very obvious in CA_GCN_3 (the last block of spatial feature extraction). This result proves that channel attention has a lower effect on channel recalibration in the GCN module close to the global pooling layer than in earlier modules. The explanation also indirectly proves the experimental results that using channel attention in the first GCN module better than using it in the latter GCN module in Table 2.

4.2.4. *Effectiveness of Multi-Scale TCN*

We use the multi-scale TCN to model the time series. The number and type of branches have a great impact on the final modeling effect. Table 3 compares the effects of the combined models with different scale branches. D1 means 1×3 convolution with $dilation\,rate = 1$, D2 means 1×3 convolution with $dilation\,rate = 2$, and D3 means 1×3 convolution with $dilation\,rate = 3$. C means pointwise convolution, MP means 1×3 maximum pooling, C-MP means pointwise convolution first, and then 1×3 maximum pooling. The spatial module uses the channel attention mechanism in the first GCN. We explore that the experimental results are not always positively correlated with the scale richness contained in branches. For example, adding C or C-MP to D1+D2+D3 can make the over-fitting problem prominent, thereby reducing the accuracy of the test. To sum up the above, we aggregate the three scale branches of $dialation\,rate = 1, 2, 3$ to model the time series.

4.2.5. *Attention Influence of Different Actions*

Our graph channel attention approach has different effects for different skeleton actions. In Figure 8, we select the top-six action categories with the most significant effects after using graph channel attention to show the accuracy. These actions can be divided into two categories according to the difference in motion dimensions. The first action

Table 3

Effects of different scales branch combination

Model	#Params ($\times 10^6$)	Accuracy	
		CS (%)	CV (%)
D1	0.72	88.6	94.3
D1+D2	1.05	88.4	93.6
D1+D2+D3*	1.38	89.4	94.5
D1+D2+D3+C	1.58	88.6	93.9
D1+D2+D3+C-MP	1.58	88.1	93.2

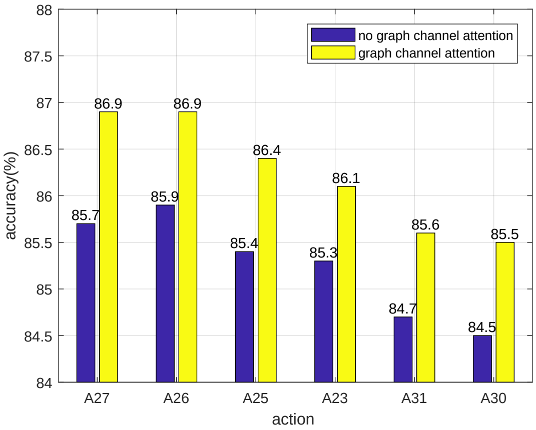


Fig. 8. The top-six actions with the greatest accuracy improvement after using the graph channel attention: A27: jump up A26: hopping A25: reach into pocket A23: hand waving A31: point to something A30: type on a keyboard.

type has clear directionality. For example, jumping up and hopping are global movements that are completely perpendicular to the ground; reaching into a pocket and pointing to something are local movements with clear directionality. The second action type has a fixed surface to move. For example, while waving hand, the arm only moves in the frontal plane of the human body; while typing on a keyboard, both hands' movement is basically limited to the surface of the keyboard. On the contrary, channel attention has a weak effect on brushing teeth, falling down, taking a photo, wielding knife and et al. In summary, the actions mentioned above which are more sensitive to graph channel attention have significant differences in the information richness of different dimensions and joints.

4.2.6. Hyperparameter Selection

After determining the network architecture, the choice of hyperparameters is also critical to the model performance. There are 4 FC layers in our network to encoding features. The FC layers which encode the joint position features, the joint velocity features and the joint type have 64 nodes. The FC layer which encodes the frame index has 256 nodes. In Table 4, we explore the effect of reduction ratio in the graph channel attention module. From the experimental data in Table 4, it can be found that the reduction ratio and the parameters are inversely proportional. In general, with the increase of reduction ratio, the accuracy decays more faster. We can use an appropriate reduction ratio to balance the accuracy and the computation complexity. In this paper, the reduction ratio is set to 1 for achieving the best performance. The dropout is used only in MS-TCN, the dropout rate is set to 0.3.

4.3. Comparisons to the State-of-the-art

In Table 5, we compare various typical methods such as: RNN-based [7, 9–12], CNN-based [15, 17], GCN-based [22, 27, 28], mixed methods-based [41, 47] with our skeleton action recognition model CA-MSN (Figure 3) on the NTU-RGBD60 dataset. In Table 6, we compare the results of our model on the NTU-RGBD120 dataset with other methods to prove the effectiveness of our method for fine-grained motions and object-related actions.

Table 4
Graph channel attention with different reduction ratios

Model	#Params ($\times 10^6$)	Accuracy	
		CS (%)	CV (%)
w/o CA	0.69	87.9	93.6
W CA r=1*	0.99	88.6	94.3
W CA r=2	0.84	88.3	94.1
W CA r=4	0.77	87.8	93.6

Table 5

Classification accuracy comparison against state-of-the-art methods on the NTU-RGBD60 Skeleton dataset

Method	CS (%)	CV (%)
HBRNN-L [7]	59.1	64
Part-Aware LSTM [9]	62.9	70.3
ST-LSTM+Trust Gate [10]	69.2	77.7
STA-LSTM [11]	73.4	81.2
VA-LSTM [12]	79.4	87.6
TCN [15]	74.3	83.1
Clips+CNN+MTLN [17]	79.6	84.8
ST-GCN [22]	81.5	88.3
AS-GCN [27]	86.8	94.2
2s-AGCN [28]	88.5	95.1
SR-TSL [41]	84.8	92.4
SGN [47]	89	94.5
CA-MSN	89.4	94.5

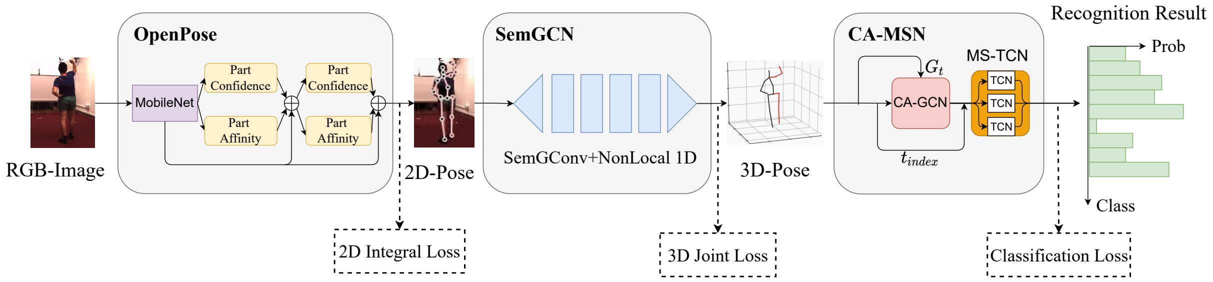


Fig. 9. This figure shows the series network architecture of pose estimation and action recognition. The whole model has three parts: OpenPose, SemGCN and CA-MSN. Each sub-network uses the loss of its own task for training.

Some pure LSTM methods in Table 5, such as Part-Aware LSTM [9] and STA-LSTM [11], have an lower accuracy by about 20% compared with our method. The accuracy of the typical CNN methods, such as Clips+CNN+MTLN [17] and RotClips+MTCNN [58] in Table 5 and Table 6 have not reached the advanced level which can be applied. CA-MSN outperforms the typical spatio-temporal GCN method ST-GCN [22] in Table 5 by 7.9% in the accuracy for CS setting. The above results show that simply using a certain method to model both the temporal and spatial characteristics of skeleton sequences is limited and cannot fully explore the potential spatial and temporal dependencies.

Table 6

Classification accuracy comparison against state-of-the-art methods on the NTU-RGBD120 Skeleton dataset

Method	C-Subject (%)	C-Setup (%)
Part-Aware LSTM [9]	25.5	26.3
Clips+CNN+MTLN [17]	58.4	57.9
RotClips+MTCNN [58]	62.2	61.8
SkeleMotion [20]	62.9	63
TSRJI [21]	67.9	59.7
SGN [47]	79.2	81.5
CA-MSN	79.5	81.8

Compared with the mixed model of multiple methods, such as SR-TSL [43] and SGN [47], our CA-MSN has more in-depth exploration of the potential relationship between the frames and the channels of skeleton actions. In Table 5, CA-MSN brings the performance improvement of 4.2% and 2.1% in the accuracy of the CS and CV settings than SR-TSL. Notably, our method is the first to integrate the channel attention mechanism into the graph network for skeleton action recognition. The accuracy comparison results also verify the effectiveness of our method.

5. Application

In the above research, we propose an advanced skeleton-based action recognition model CA-MSN. However, in actual applications, the input is usually RGB videos, while the input of CA-MSN model is 3D skeletons. In this chapter, we will explore how to extract human 3D skeleton sequences from raw videos and classify them with the CA-MSN model in series. As shown in Figure 9, the OpenPose method is firstly used to extract 2D skeleton from the original RGB image. Subsequently, the semantic graph convolution network (SemGCN) learns the potential relationship between 2D skeleton sequences and 3D skeleton sequences to predict the 3D pose. Finally, the 3D pose is used as the input of the CA-MSN to obtain the final action classification result. It is worth noting that the three networks in Figure 9 are trained separately and then used in series.

Human3.6M [59]. The motions in the dataset are performed by 11 professional actors, including 5 females and 6 males. We choose 7 subjects (3 females and 4 males) for training and 4 subjects (2 females and 2 males) for testing. To ensure authenticity, the subjects are dressed in their regular clothing, rather than special motion-capture outfits. The dataset consists of 3.6 million different human poses collected with 4 digital cameras. This dataset has complete RGB video - 2D pose - 3D pose data, and is widely used in the study of predicting action categories from RGB videos. So this paper uses the Human3.6M dataset to conduct experiments on our series networks.

5.1. 3D Skeleton Sequence Extraction

There are two mainstream methods for extracting 3D skeletons from monocular RGB images. The first method uses the deep learning model to establish an end-to-end mapping from monocular RGB images to 3D coordinates, but the features that need to be learned are too complex for a single model. The second method needs two steps. The first step is to get 2D skeleton using 2D pose estimation model. The second step is to regress the identified 2D skeleton to predict the 3D skeleton using the prior knowledge of the dataset. Although the end-to-end regression method is simple to operate, the accuracy is difficult to guarantee due to the complexity of feature mapping. So we use the two-step method. The 2D pose estimation is implemented using OpenPose proposed by Cao et al. [60] of Carnegie Mellon University (CMU) in 2017. The mapping encoding from 2D to 3D poses is managed by SemGCN proposed by Zhao et al. [61] in 2019. There are two reasons for choosing the SemGCN network. Above all, this method uses graph convolution networks to process joint coordinates, which is consistent with our CA-MSN network. This consistency facilitates subsequent tandem deployment. In addition, due to the serial deployment of three networks, it is difficult to guarantee the speed of calculation, so reducing the amount of calculation in each step is important. The SemGCN we choose has an order of magnitude smaller model size than other algorithms. Figure 10 shows the results of the 3D skeleton extraction experiment using the Human3.6M dataset. The first line is the extracted 2D skeleton; the second line is the 3D skeleton obtained by the 2D to 3D pose regression. We select some representative frames in several continuous actions for visualization. Comparing the predicted 3D skeleton sequence with the ground truth, the error is within an acceptable range. The observation confirms that it is feasible to predict the action using 3D skeleton sequences which are regressed from their corresponding 2D projections.

Table 7

Comparison with video-based models in terms of parameters, calculations, FPS (Measured on NVIDIA GTX2080Ti GPU), and accuracy

Method	#Params ($\times 10^6$)	#FLOPs ($\times 10^9$)	#FPS	Accuracy (%)
X3D-M [62]	3.76	4,73	174	83.0
Gate-Shift [63]	10.5	16.45	98	86.8
TSM [64]	48.6	98	25	84.7
GST [65]	21	29.5	58	86.3
O-S-C (Ours)	10+0.43+1.38=11.81	12.45+0.73+1.92=15.54	64	87.5

5.2. Joint Deployment

In actual applications, multiple models need to work together to realize the action recognition function. The end-to-end (RGB video-to-action type) network is obtained by serially deploying the three models of OpenPose, SemGCN and CA-MSN. Figure 10 shows the test results of the joint deployment. We intercept video clips of 3 different actions (eating, greeting and taking photos) as test samples, and visualize the RGB image + 2D posture and 3D posture. In the process of testing the overall model, the three submodels bring huge cumulative errors. There are two connection points in the series network. The first is the 2D pose output by OpenPose as the input of SemGCN, and the second is the 3D pose output by SemGCN as the input of CA-MSN. Our improved OpenPose achieves the single-person AP of 91.3% based on the Human3.6M dataset when the OKS threshold is 0.5. Therefore, 8.7% of the 2D poses output by OpenPose have a large deviation from the ground truth 2D poses in Human3.6M. The actual 2D pose input to SemGCN is significantly different from the standard input. Similarly, the actual 3D pose input to CA-MSN is also significantly different from the standard input. Errors in the two stages cause that the SemGCN and CA-MSN models tested in our series network perform far worse than tested with standard data. To solve this problem, OpenPose is first pretrained on the coco dataset, and then fine-tuned on the Human3.6M dataset. SemGCN uses the prediction results of the OpenPose model fine-tuned on the Human3.6M dataset as the training input, and still the 3D skeleton ground truth as the label value. The skeleton action recognition model CA-MSN also needs to be retrained on the Human3.6M dataset. After the above training mode changes, it can be ensured that the domain relationships of the three models are relatively close. The serious influence caused by the cumulative errors of the series models is prevented.

5.3. Comparison with Video-Based Action Recognition

This chapter combines the skeleton action recognition method with the pose estimation to form an integrated network. The network uses the skeleton output of the pose estimation to classify the actions in the video. Through the application, it shows that compared with video-based recognition methods, skeleton action recognition has more steps and cumulative error problems. As shown in Table 7, we select several of the advanced video-based action recognition methods in recent years to compare with our skeleton-based action recognition method in terms of parameters, calculations, speed and accuracy. O-S-C denotes our OpenPose-SemGCN-CA-MSN series network. In terms of accuracy, our method outperforms the RGB video-based methods with the same computational complexity. Because video-based methods are easily disturbed by visual features such as background and clothing, they are less robust than skeleton-based methods. In terms of computational complexity, our approach with 11.81M parameters and 15.54GFLOPs calculation has no obvious advantage. Because the 2D pose estimation occupies most of the computing resources (84.7% in parameters, 80.1% in FLOPs). In future work, reducing the weight of the 2D pose estimation method will make the skeleton-based action recognition method full of potential in computational speed. Notably, our series network achieves 5.53% fewer GFLOPs than the Gate-Shift method, but a 40.81% drop in FPS. This illustrates that series network has lower computational efficiency with similar computational complexity. Making the series network more holistic is a direction to solve this problem.

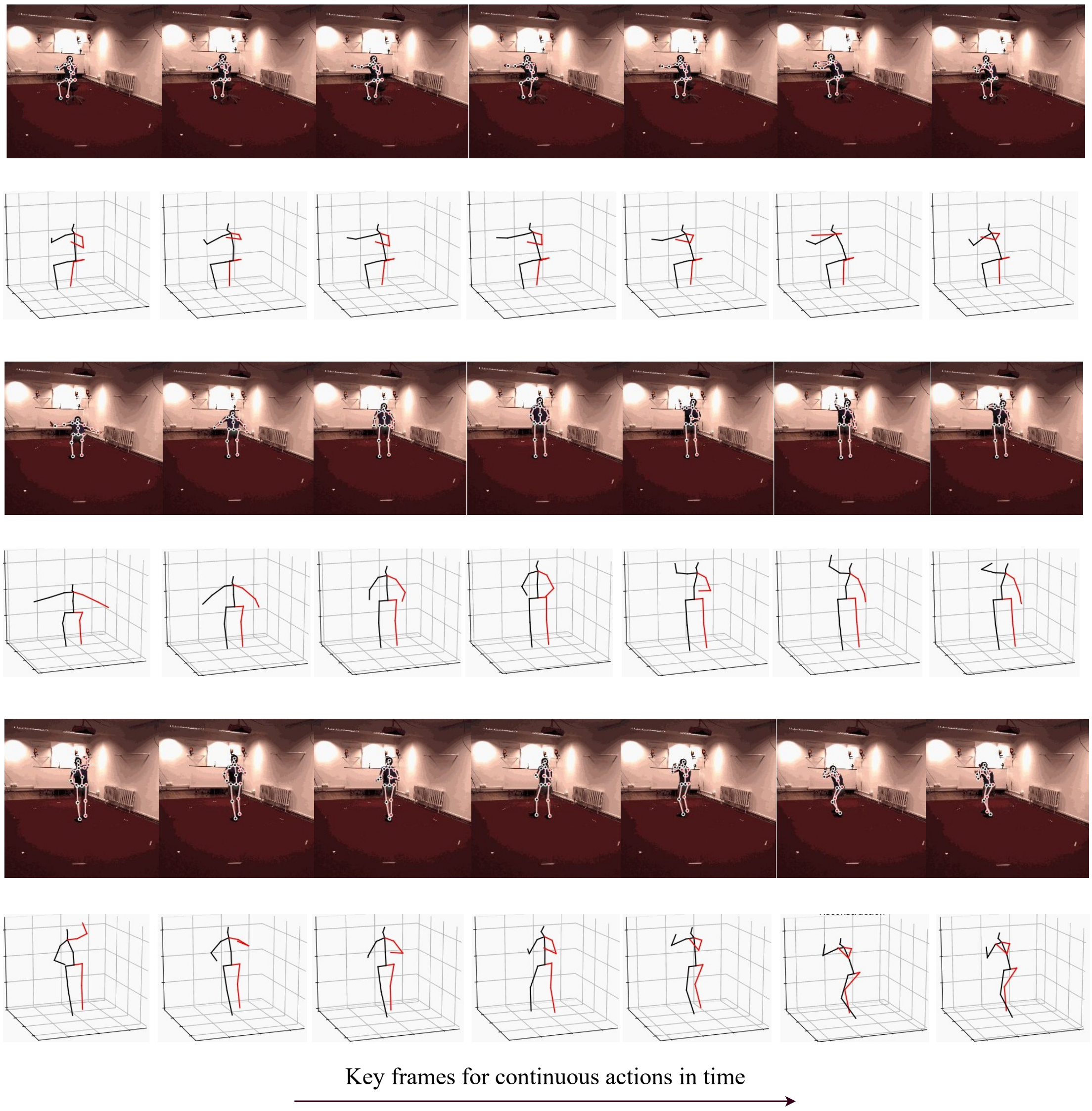


Fig. 10. OpenPose-SemGCN-CA-MSN joint deployment test. The actions from top to bottom are eating, greeting and taking pictures. The first line of each action is RGB image+2D Pose. The second line of each action is 3D Pose.

5.4. Viewpoint Invariance

Viewpoint invariance is indispensable for the practical application of action recognition algorithms. Recognition networks generally use pictures from several fixed views during training, but in practical applications, human actions may be observed from many different views. In Table 8, we compare the viewpoint invariance differences of video, 2D skeleton, and 3D skeleton-based methods using Cross View settings. 3-TrainV denotes three of four views are used for training, and 1-TestV denotes one of four views are used for testing. When using three views for training and one view for testing, the 3D skeleton-based method achieves the best performance of 90.4%, the video-based method achieves 84.6%, and the 2D skeleton-based method performs the worst, only 75.2%. In the more difficult task of using 2 views for training and 2 views for testing, the 3D skeleton-based method drops by 6.3%, the video-based method drops by 8.7%, and the 2D skeleton-based method drops by 20%. In conclusion, 3D skeleton method > video method > 2D skeleton method in terms of the viewpoint invariance. The results of 2D skeleton-based method

Table 8

With Cross View setting, compare the robustness of the three methods (video-based, 2D skeleton-based, 3D skeleton-based) to perspective changes

Dataset Segmentation	Method	#Params ($\times 10^6$)	#FLOPs ($\times 10^9$)	#FPS	Accuracy (%)
3-TrainV 1-TestV	X3D-M	3.76	4.73	174	84.6
	O-C	11.38	14.37	72	75.2
	O-S-C	11.81	15.54	64	90.4
2-TrainV 2-TestV	X3D-M	3.76	4.73	174	75.9
	O-C	11.38	14.37	72	54.2
	O-S-C	11.81	15.54	64	84.1

are almost catastrophic. The main reason is that it is difficult to learn the 3D spatial motion relationship between joints directly from 2D poses. And because the 2D-3D pose regression method has a small number of parameters, the 2D skeleton-based method has a weak advantage in computational complexity. Based on the above analysis, the 3D skeleton-based method has obvious advantages in view invariance.

6. Conclusion

In this work, we propose a new time-space series network based on channel attention GCN and multi-scale TCN to improve the accuracy of skeleton action recognition. We explore the characteristics of the channel attention mechanism in the GCN network while extracting skeleton joint features. In addition, the processing method of multi-scale dilated convolution makes the temporal receptive field more abundant and positively affects actions of different cycle periods. Furthermore, We use the global maximum pooling for each frame of joint features to connect the spatial module and the temporal module. The specialization of feature extraction can improve the efficiency of modeling feature representation. The final model exceeds the current state-of-the-art performance on two large-scale datasets: NTU-RGBD60 and NTU-RGBD120. In the end, we design a OpenPose-SemGCN-CA-MSN network to realize the end-to-end (RGB video-to-action type) application. From the application, we discover that it is difficult to train the sub-models separately and then test the whole model in series. Therefore, the focus of future research will be how to design a multi-task framework for jointly estimating 2D or 3D human poses from color images and classifying human actions from video sequences. Meanwhile, the interpretability of GCN will also play a critical role in skeleton action recognition.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant U1713211, Grant 62073245, and Grant 61733013; in part by Pudong New Area Science and Technology Development Fund under Grant PKX2019-R18; and in part by Jiangsu Key Research and Development Project under Grant BE2020101.

References

[1] U. Gaur, Y. Zhu, B. Song and A. Roy-Chowdhury, A "string of feature graphs" model for recognition of complex activities in natural videos, in: *2011 International Conference on Computer Vision*, IEEE, 2011, pp. 2595–2602.

[2] O.P. Popoola and K. Wang, Video-based abnormal human behavior recognition—A review, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42**(6) (2012), 865–878.

[3] N.C. Tay, T. Connie, T.S. Ong, K.O.M. Goh and P.S. Teh, A robust abnormal behavior detection method using convolutional neural network, in: *Computational Science and Technology*, Springer, 2019, pp. 37–47.

[4] Z. Duric, W.D. Gray, R. Heishman, F. Li, A. Rosenfeld, M.J. Schoelles, C. Schunn and H. Wechsler, Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction, *Proceedings of the IEEE* **90**(7) (2002), 1272–1289.

[5] C. Godard, O. Mac Aodha and G.J. Brostow, Unsupervised monocular depth estimation with left-right consistency, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 270–279.

[6] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari and N. Navab, Deeper depth prediction with fully convolutional residual networks, in: *2016 Fourth international conference on 3D vision (3DV)*, IEEE, 2016, pp. 239–248.

[7] Y. Du, W. Wang and L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1110–1118.

[8] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen and X. Xie, Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30, 2016.

[9] A. Shahroudy, J. Liu, T.-T. Ng and G. Wang, Ntu rgb+ d: A large scale dataset for 3d human activity analysis, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1010–1019.

[10] J. Liu, A. Shahroudy, D. Xu and G. Wang, Spatio-temporal lstm with trust gates for 3d human action recognition, in: *European conference on computer vision*, Springer, 2016, pp. 816–833.

[11] 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'17 Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 4263–4270.

[12] 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: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2117–2126.

[13] I. Lee, D. Kim, S. Kang and S. Lee, Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks, in: *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1012–1020.

[14] H. Wang and L. Wang, Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 499–508.

[15] T.S. Kim and A. Reiter, Interpretable 3d human action analysis with temporal convolutional networks, in: *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, IEEE, 2017, pp. 1623–1631.

[16] C. Li, Q. Zhong, D. Xie and S. Pu, Skeleton-based action recognition with convolutional neural networks, in: *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, 2017, pp. 597–600.

[17] Q. Ke, M. Bennamoun, S. An, F. Sohel and F. Boussaid, A new representation of skeleton sequences for 3d action recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3288–3297.

[18] M. Liu, H. Liu and C. Chen, Enhanced skeleton visualization for view invariant human action recognition, *Pattern Recognition* **68** (2017), 346–362.

[19] T.M. Le, N. Inoue and K. Shinoda, A fine-to-coarse convolutional neural network for 3D human action recognition, *arXiv preprint arXiv:1805.11790* (2018).

[20] 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: *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 2019, pp. 1–8.

[21] C. Caetano, F. Brémond and W.R. Schwartz, Skeleton image representation for 3d action recognition based on tree structure and reference joints, in: *2019 32nd SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, IEEE, 2019, pp. 16–23.

[22] S. Yan, Y. Xiong and D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: *AAAI*, 2018, pp. 7444–7452.

[23] C. Li, Z. Cui, W. Zheng, C. Xu and J. Yang, Spatio-Temporal Graph Convolution for Skeleton Based Action Recognition, in: *AAAI*, 2018, pp. 3482–3489.

[24] K.C. Thakkar and P.J. Narayanan, Part-based Graph Convolutional Network for Action Recognition., in: *BMVC*, 2018, p. 270.

[25] Y.-F. Song, Z. Zhang and L. Wang, Richly activated graph convolutional network for action recognition with incomplete skeletons, in: *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 1–5.

[26] Y.-H. Wen, L. Gao, H. Fu, F.-L. Zhang and S. Xia, Graph CNNs with motif and variable temporal block for skeleton-based action recognition, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 2019, pp. 8989–8996.

[27] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang and Q. Tian, Actional-structural graph convolutional networks for skeleton-based action recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3595–3603.

[28] L. Shi, Y. Zhang, J. Cheng and H. Lu, Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12026–12035.

[29] L. Shi, Y. Zhang, J. Cheng and H. Lu, Skeleton-based action recognition with directed graph neural networks, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7912–7921.

[30] C. Wu, X.-J. Wu and J. Kittler, Spatial residual layer and dense connection block enhanced spatial temporal graph convolutional network for skeleton-based action recognition, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[31] C. Plizzari, M. Cannici and M. Matteucci, Spatial Temporal Transformer Network for Skeleton-Based Action Recognition., *ICPR Workshops (3)* (2020), 694–701.

[32] J. Cai, N. Jiang, X. Han, K. Jia and J. Lu, JOLO-GCN: Mining Joint-Centered Light-Weight Information for Skeleton-Based Action Recognition, in: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2735–2744.

[33] W. Peng, J. Shi, Z. Xia and G. Zhao, Mix dimension in poincaré geometry for 3d skeleton-based action recognition, in: *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1432–1440.

[34] F. Ye, S. Pu, Q. Zhong, C. Li, D. Xie and H. Tang, Dynamic GCN: Context-enriched Topology Learning for Skeleton-based Action Recognition, in: *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 55–63.

[35] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng and H. Lu, Skeleton-based action recognition with shift graph convolutional network, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 183–192.

[36] H. Xia and X. Gao, Multi-Scale Mixed Dense Graph Convolution Network for Skeleton-Based Action Recognition, *IEEE Access* **9** (2021), 36475–36484.

[37] Y.-F. Song, Z. Zhang, C. Shan and L. Wang, Stronger, Faster and More Explainable: A Graph Convolutional Baseline for Skeleton-based Action Recognition, in: *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1625–1633.

[38] X. Zhang, C. Xu and D. Tao, Context aware graph convolution for skeleton-based action recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14333–14342.

[39] C. Li, P. Wang, S. Wang, Y. Hou and W. Li, Skeleton-based action recognition using LSTM and CNN, in: *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, 2017, pp. 585–590.

[40] C. Xie, C. Li, B. Zhang, C. Chen, J. Han and J. Liu, Memory Attention Networks for Skeleton-based Action Recognition., in: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018, pp. 1639–1645.

[41] C. Si, W. Chen, W. Wang, L. Wang and T. Tan, 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*, 2019, pp. 1227–1236.

[42] R. Zhao, K. Wang, H. Su and Q. Ji, Bayesian graph convolution LSTM for skeleton based action recognition, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6882–6892.

[43] C. Si, Y. Jing, W. Wang, L. Wang and T. Tan, Skeleton-based action recognition with spatial reasoning and temporal stack learning, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 103–118.

[44] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue and N. Zheng, View adaptive neural networks for high performance skeleton-based human action recognition, *IEEE transactions on pattern analysis and machine intelligence* **41**(8) (2019), 1963–1978.

[45] Z. Liu, H. Zhang, Z. Chen, Z. Wang and W. Ouyang, Disentangling and unifying graph convolutions for skeleton-based action recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 143–152.

[46] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang and Q. Tian, Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 214–223.

[47] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue and N. Zheng, Semantics-guided neural networks for efficient skeleton-based human action recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1112–1121.

[48] F. Yu and V. Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, in: *ICLR 2016 : International Conference on Learning Representations 2016*, 2016.

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, Going deeper with convolutions, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[50] W. Peng, X. Hong, H. Chen and G. Zhao, Learning graph convolutional network for skeleton-based human action recognition by neural searching, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 2020, pp. 2669–2676.

[51] M. Jaderberg, K. Simonyan, A. Zisserman and k. kavukcuoglu, Spatial Transformer Networks, in: *Advances in Neural Information Processing Systems*, Vol. 28, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama and R. Garnett, eds, Curran Associates, Inc., 2015.

[52] A. Almahairi, N. Ballas, T. Cooijmans, Y. Zheng, H. Larochelle and A. Courville, Dynamic capacity networks, in: *International Conference on Machine Learning*, PMLR, 2016, pp. 2549–2558.

[53] J. Hu, L. Shen and G. Sun, Squeeze-and-excitation networks, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[54] X. Li, W. Wang, X. Hu and J. Yang, Selective kernel networks, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 510–519.

[55] S. Woo, J. Park, J.-Y. Lee and I.S. Kweon, Cbam: Convolutional block attention module, in: *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[56] V. Nair and G.E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, in: *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 807–814.

[57] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan and A.C. Kot, Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding, *IEEE transactions on pattern analysis and machine intelligence* **42**(10) (2019), 2684–2701.

[58] Q. Ke, M. Bennamoun, S. An, F. Sohel and F. Boussaid, Learning clip representations for skeleton-based 3d action recognition, *IEEE Transactions on Image Processing* **27**(6) (2018), 2842–2855.

[59] C. Ionescu, D. Papava, V. Olaru and C. Sminchisescu, Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments, *IEEE transactions on pattern analysis and machine intelligence* **36**(7) (2013), 1325–1339.

[60] Z. Cao, T. Simon, S.-E. Wei and Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.

[61] L. Zhao, X. Peng, Y. Tian, M. Kapadia and D.N. Metaxas, Semantic graph convolutional networks for 3d human pose regression, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3425–3435.

[62] C. Feichtenhofer, X3d: Expanding architectures for efficient video recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 203–213.

[63] S. Sudhakaran, S. Escalera and O. Lanz, Gate-shift networks for video action recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1102–1111.

[64] J. Lin, C. Gan and S. Han, Tsm: Temporal shift module for efficient video understanding, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7083–7093.

[65] C. Luo and A.L. Yuille, Grouped spatial-temporal aggregation for efficient action recognition, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5512–5521.

[66] J.-F. Hu, W.-S. Zheng, J. Lai and J. Zhang, Jointly learning heterogeneous features for RGB-D activity recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5344–5352.

[67] X. Gao, W. Hu, J. Tang, J. Liu and Z. Guo, Optimized skeleton-based action recognition via sparsified graph regression, in: *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 601–610.

[68] 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 Transactions on Image Processing* **27**(4) (2017), 1586–1599.

[69] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao and N. Zheng, Adding attentiveness to the neurons in recurrent neural networks, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 135–151.