Pattern Recognition 107 (2020) 107511

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Skeleton-based action recognition with hierarchical spatial reasoning and temporal stack learning network

Chenyang Si[a,b,c,1], Ya Jing[a,b,c,1], Wei Wang[a,b,c,*], Liang Wang[a,b,c], Tieniu Tan[a,b,c]

[a] University of Chinese Academy of Sciences (UCAS), China
[b] Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition (NLPR), China
[c] Institute of Automation, Chinese Academy of Sciences (CASIA), China

ARTICLE INFO

Article history:
Received 5 December 2019
Revised 10 June 2020
Accepted 17 June 2020
Available online 19 June 2020

Keywords:
Skeleton-based action recognition
Hierarchical spatial reasoning
Temporal stack learning
Clip-based incremental loss

ABSTRACT

Skeleton-based action recognition aims to recognize human actions by exploring the inherent characteristics from the given skeleton sequences and has attracted far more attention due to its great important potentials in practical applications. Previous methods have illustrated that learning discriminative spatial and temporal features from the skeleton sequences is a crucial factor to recognize human actions. Nevertheless, how to model spatio-temporal evolutions is still a challenging problem. In this work, we propose a novel model with hierarchical spatial reasoning and temporal stack learning network (HSR-TSL) to explore the discriminative spatial and temporal features for human action recognition, which consists of a hierarchical spatial reasoning network (HSRN) and a temporal stack learning network (TSLN). Specifically, the HSRN employs a hierarchical residual graph neural network to capture two-level spatial features: intra spatial information of each part and body-level structural information between each part. The TSLN models the detailed temporal dynamics of skeleton sequences by a composition of multiple skip-clip LSTMs. During training, we develop a clip-based incremental loss to effectively optimize the model. We perform extensive experiments on five challenging benchmarks to verify the effectiveness of each component of our model. The comparison results illustrate that our approach significantly boosts the performances for skeleton-based action recognition.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Human action recognition is a fundamental and challenging time series classification task in computer vision research. This task involves exploring the motion characteristics of human action from given videos to predict human action classes. In addition, human action recognition has gained more and more attention due to its important role in many applications, such as intelligent video surveillance, video retrieval, human-computer interaction and game control [1,2]. For action recognition, how to analyze human motion information and understand its temporal characteristics is a challenging problem.

Recently, many advanced algorithms [3–5] have been proposed to learn and extract effective spatio-temporal features from RGB videos, where spatial appearance and temporal optical flow from RGB videos generally are applied to model the motion dynamics, such as two-stream convolutional networks in [3]. However, the spatial appearance only contains 2D information that is hard to capture all the motion information, and the optical flow generally needs high computing costs. Compared to RGB videos, 3D skeleton data can represent the body structure with a set of 3D coordinate positions, and it is not affected by background clutter, illumination changes and appearance variation. Therefore, the more effective and discriminative representation about human action can be easily learned from 3D skeleton data. Moreover, Johansson et al. [6] have explained that 3D skeleton sequences can effectively represent the dynamics of human actions. Besides, the skeleton sequences can be obtained by the Microsoft Kinect [7] and the advanced human pose estimation algorithms [8]. Considering the advantages of 3D skeleton data, we focus on skeleton-based action recognition in this work.

Over the years, skeleton-based human action recognition has attracted more and more attention [9–11]. As a time series classification task, recurrent neural networks (RNNs) performing a strong power in learning the temporal dependencies are naturally ap-

[*] Corresponding author at: University of Chinese Academy of Sciences (UCAS), China.

E-mail addresses: chenyang.si@cripac.ia.ac.cn (C. Si), ya.jing@cripac.ia.ac.cn (Y. Jing), wangwei@nlpr.ia.ac.cn (W. Wang), wangliang@nlpr.ia.ac.cn (L. Wang), tnt@nlpr.ia.ac.cn (T. Tan).

[1] Equal contribution

https://doi.org/10.1016/j.patcog.2020.107511
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

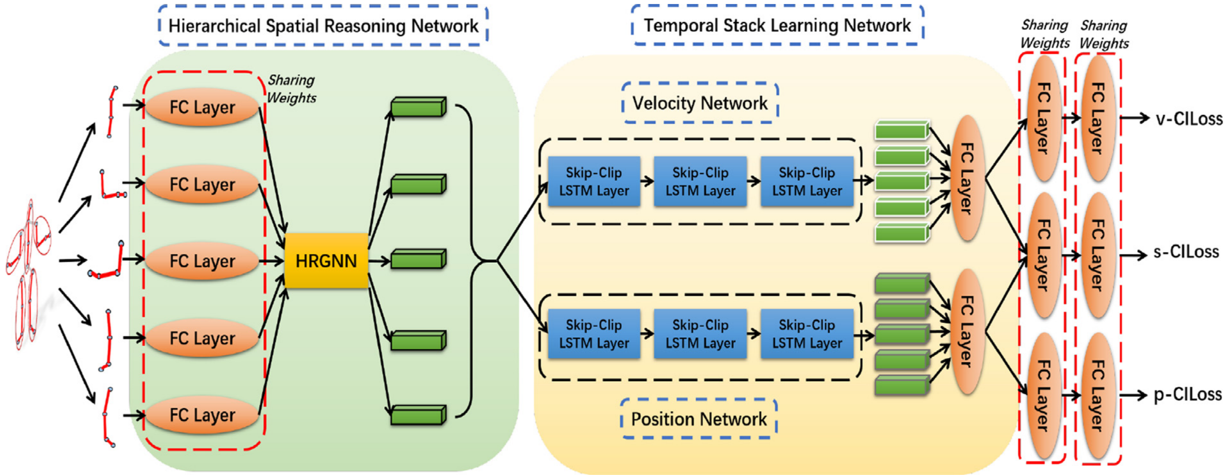


Fig. 1. The overall pipeline of our model which contains a hierarchical spatial reasoning network and a temporal stack learning network. In the hierarchical spatial reasoning network, a hierarchical residual graph neural network (HRGNN) is used to capture the body-level structural information between each part and the intra spatial relationships of joints in each part. The temporal stack learning network can model the detailed temporal dynamics for the skeleton sequence. During training, the proposed model is efficiently optimized with the clip-based incremental losses (CIloss).

plied to model the temporal dynamics of skeleton sequences in many previous works. For example, a hierarchical RNN [10] is proposed to learn motion representations from skeleton sequences. Shahroudy et al. [12] introduce a part-aware LSTM network to further improve the performance of the LSTM framework. To model the discriminative features, a spatial-temporal attention model [11] based on LSTM is proposed to focus on discriminative joints and pay different attentions to different frames.

These methods have achieved a great improvement in performance of action recognition, but how to explore spatial structure features and temporal characteristics is still a challenging problem. As we known, human behavior is accomplished in coordination with each part of the body. For example, walking requires legs to walk, and it also needs the swing of arms to coordinate the body balance. In addition, each body part contains several joints. Therefore, there are two-level important cues to recognize human actions: the body-level structural information between each part and the intra spatial relationships of joints in each part. Besides spatial information, the temporal dynamics characteristics of human actions play another significant role in human action recognition. Most methods generally utilize RNNs to directly model the overall temporal dynamics of skeleton sequences. And the hidden representation of the final RNN is used to recognize the actions. However, the last hidden representation cannot completely contain the detailed temporal dynamics for long-term sequences.

In this work, to solve the above challenges, we propose a novel model with hierarchical spatial reasoning and temporal stack learning (HSR-TSL) for skeleton-based action recognition. The proposed HSR-TSL is an end-to-end deep network architecture, which contains a hierarchical spatial reasoning network (HSRN) and a temporal stack learning network (TSLN). The overall pipeline of HSR-TSL is shown in Fig. 1. For spatial structures of the skeleton, a hierarchical spatial reasoning network is proposed to capture the high-level spatial structural features within each frame. As discussed above, human body structure contains two-level spatial information, i.e., the body-level structural information between each part and the intra spatial relationships of joints in each part. We apply a hierarchical residual graph neural network (HRGNN) to model the two-level spatial structural features. Specifically, the human body is firstly decomposed into different parts, e.g., two arms, two legs and one trunk. An intra-parts residual graph network (intra-RGNN) explores the spatial structural relationship between joints of each part. Another graph network termed as inter-parts residual graph network (inter-RGNN) captures the body-level structural information between each part. Moreover, the joint features in intra-RGNN will be aggregated into the corresponding part nodes of inter-RGNN. Therefore, the representation of each node in inter-RGNN contains not only the body-level structural information between each part but also the intra spatial relationships of joints in each part. For temporal dynamics of sequences, we propose a temporal stack learning network to model the detailed temporal dynamics of the sequences, which consists of three skip-clip LSTMs. Given a long-term sequence, it is divided into multiple clips. The short-term temporal information of each clip is modeled with an LSTM layer that shared among the clips in a skip-clip LSTM layer. When feeding a clip into shared LSTM, the hidden state of shared LSTM is initialized with the sum of the final hidden state of all previous clips, which can inherit previous dynamics to maintain the dependency between clips. Note that each part sequence is processed by TSLN. As shown in Fig. 1, the aggregation of all part features is used to predict human action classes. Besides, we propose a clip-based incremental loss to further improve the ability of stack learning, which can also effectively solve the problem of long-term sequence optimization. Experimental results show that the proposed HSR-TSL speeds up the model convergence and improves the performance.

The main contributions of this paper are summarized as follows:

1. We propose a hierarchical spatial reasoning network for each skeleton frame, which can effectively capture the body-level structural information between each part and the intra spatial relationships of joints in each part with a hierarchical residual graph neural network.
2. We propose a temporal stack learning network to model the detailed temporal dynamics of skeleton sequences by a composition of multiple skip-clip LSTMs.
3. We perform extensive experiments on five challenging benchmarks to verify the effectiveness of each component of our model. The comparison results illustrate that our approach significantly boosts the performances for skeleton-based action recognition.

It should be noted that this paper is an extension of the preliminary conference paper [13]. The present work mainly adds to the preliminary version in several significant ways. First, we introduce a novel graph structure termed as hierarchical residual graph neural network that is capable of exploiting not only the body-level structural information between each part but also the intra spatial relationships of joints in each part. Experimentally, we demonstrate that it can obviously improve the performances in comparison to the previously proposed residual graph neural network that focuses on the body-level structural information between each part. Second, instead of aggregating all part features before TSLN, we apply TSLN to process each part sequence to learn their unique temporal features then aggregate all part features to predict human action classes. Third, we perform more rich experiments on five challenging benchmarks to further analyse the effectiveness of our model. The comparison results illustrate that our approach significantly boosts the performances for skeleton-based action recognition.

2. Related work

2.1. Skeleton-Based action recognition

Recently, amounts of works have been proposed for skeleton-based action recognition [14–23]. For the traditional approaches [16,18,24–26], they represent human motion by designing various hand-crafted features from skeleton sequences, such as relative 3D geometry between all pairs of body parts [27].

Recently, deep learning has also been applied to this task due to its wide success. Due to the strong ability of Convolutional Neural Networks (CNNs) for learning hierarchical representation, some methods [17,19,20,28–31,31–36] exploit CNNs to recognize human actions. For example, Du et al. [17] and Li et al. [29] represent the skeleton sequence as an image, where the value of 3D coordinates is corresponding to three channels of an image, the joints of each frame are represented as columns and the frame sequences are arranged in rows. Considering the powerful capability of capturing the dynamics of sequences for the Recurrent Neural Networks (RNNs), most of methods utilize RNNs for this task [10–12,37–43]. Du et al. [10] first propose an end-to-end hierarchical RNN for skeleton-based action recognition. Zhang et al. [38] exploit a view adaptive model with LSTM architecture, which enables the network to adapt to the most suitable observation viewpoints from end to end. The skeleton data can naturally be represented as the graph-structured data. Therefore, graph-based approaches [13,44–48] are popularly adopted for skeleton-based action recognition, such as ST-GCN [45], AGC-LSTM [46]. The most similar works to ours are [40] and [46]. Lee et al. [40] proposes an ensemble temporal sliding LSTM (TS-LSTM) network for skeleton-based action recognition. They utilize an ensemble of multi-term temporal sliding LSTM networks to capture short-term, medium-term, long-term temporal dependencies and even spatial skeleton pose dependency. Unlike the ensemble of multi-term temporal sliding LSTM networks, we propose a simple temporal stack learning network to effectively learn the detailed temporal dynamics of skeleton sequences. Si et al. [46] introduces a graph convolutional LSTM to capture discriminative spatiotemporal features. In [46], each node of graph denotes the body part, which only considers the body-level structural information between each part but neglects the intra spatial relationships of joints in each part. In this paper, we design a hierarchical spatial reasoning network, that can leverage not only the body-level structural information but also the intra spatial relationships of joints in each part with a hierarchical residual graph neural network.

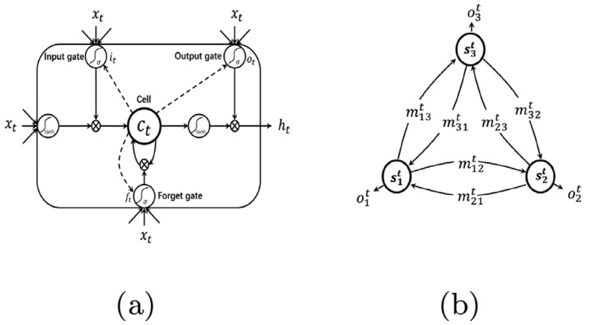


Fig. 2. (a) shows the structure of an LSTM neuron. (b) shows a graph neural network with 3 nodes.

2.2. Graph-based models

Due to the effective representation for the graph structure data, graph-based models have received a lot of attention and are applied to represent various datasets, such as web link data, social network, human skeleton, etc. According to the way of updating the state of the node, existing graph models can be categorized into two architectures. The first framework is to apply Convolutional Neural Networks to graph, namely graph convolutional network (GCN), which improves the traditional convolution network on graph. Henaff et al. [49], Duvenaud et al. [50] utilize the CNNs in the spectral domain relying on the graph Laplacian. LeCun [51], Niepert et al. [52] apply the convolution directly on the graph nodes and their neighbors, which construct the graph filters on the spatial domain. The other framework is the combination of graph and recurrent neural network, namely graph neural network (GNN), which utilizes the recurrent neural networks to every node of the graph. Scarselli et al. [53] proposes to recurrently update the hidden state of each node of the graph. In this paper, a hierarchical residual graph neural network is utilized to model the body-level structural information between each part and the intra spatial relationships of joints in each part.

3. Overview

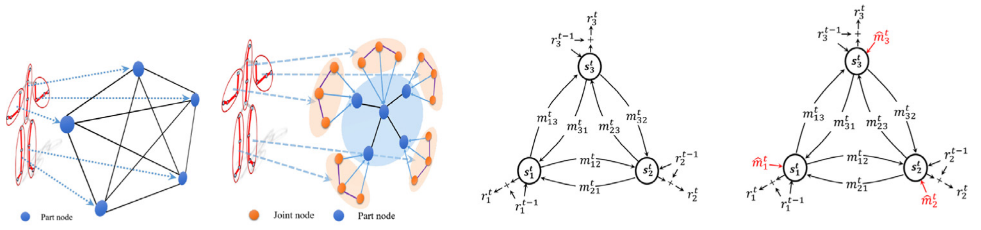
In this section, we briefly review the Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and the Graph Neural Network (GNN)., which is utilized in our framework.

3.1. RNN and LSTM

In this section, we briefly review the Recurrent Neural Network(RNN) and the Long Short-Term Memory (LSTM). RNN is a powerful model to capture the dynamics of sequences via cycles in the network of nodes. However, the standard RNN is difficultly optimized for the long-term sequence tasks due to the vanishing and exploding gradient problems. Hochreiter et al. [54] propose an advanced RNN architecture of Long Short-Term Memory (LSTM) to overcome the vanishing and exploding gradient problems. As shown in Fig. 2a, an LSTM neuron contains an input gate $\boldsymbol{i}_t$, a forget gate $\boldsymbol{f}_t$, an output gate $\boldsymbol{o}_t$ and a cell $\boldsymbol{c}_t$, which can promote the ability to learn long-term dependencies.

3.2. Graph neural network

Graph Neural Network (GNN) is a powerful model to deal with a more general class of graphs, which is introduced in [53] as a generalization of recursive neural networks. Particularly, a GNN can be defined as an ordered pair $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. For the GNN shown in Fig. 2b, the input vector of each node $v \in \mathcal{V}$ is based on the information contained in the neighborhood of node v, and the hidden state of each node is updated recurrently. Specifically, at time step t, the received messages of a node are calculated with the hidden states of its neighbors. Then, the received messages and previous states



(a) RGNN ([13]) (b) HRGNN (Ours) (c) intra-RGNN (d) inter-RGNN

Fig. 3. (a) shows the corresponding relationships (RGNN) between human part and graph node in the preliminary conference paper [13]. (b) illustrates the architecture of proposed HRGNN and the corresponding relationships with human skeleton. (c) is the architecture of intra-RGNN that is the same as RGNN. (d) is the architecture of inter-RGNN that has an additional input $\widehat{\boldsymbol{m}}_k^t$ to update the node hidden state, where $\widehat{\boldsymbol{m}}_k^t$ is the aggregated information from intra-RGNN.

are utilized to update its hidden state. Finally, the outputs can be computed with the hidden states of the nodes. The GNN formulation at time step t is defined as follows:

$$\boldsymbol{m}_i^t = f_m\left(\{\boldsymbol{s}_{\hat{i}}^{t-1} | \hat{i} \in \{1, \ldots, |\Omega_{v_i}|\}\right) \tag{1}$$

$$\boldsymbol{s}_i^t = f_s\left(\boldsymbol{m}_i^t, \boldsymbol{s}_i^{t-1}\right) \tag{2}$$

$$\boldsymbol{o}_i^t = f_o\left(\boldsymbol{s}_i^t\right) \tag{3}$$

where $\boldsymbol{s}_i^t$ is the hidden state of the ith ($i \in \{1, \ldots, |\mathcal{V}|\}$) node. The set of nodes Ω_{v_i} stands for the neighbors of node v_i. $\boldsymbol{m}_i^t$ is the sum of all the messages that the neighbors Ω_{v_i} send to node v_i. f_m is the function to compute the incoming messages. f_s is the function that expresses the state of a node and f_o is the function to produce the output $\boldsymbol{o}_i^t$. Similar to RNNs, these functions are the learned neural networks and are shared among different time steps.

4. Model architecture

In this paper, we propose an effective model for skeleton-based action recognition, which contains a hierarchical spatial reasoning network and a temporal stack learning network. The overall pipeline of our model is shown in Fig. 1. In this section, we will introduce these networks in detail.

4.1. Hierarchical spatial reasoning network

Rich inherent structures of the human body that are involved in action recognition task, motivate us to design an effective architecture to model the high-level spatial structural information within each frame. As we known, human behavior is accomplished in coordination with each part of the body. And the body can be decomposed into multi parts, e.g. two arms, two legs and one trunk, which express the knowledge of human body configuration. In the preliminary conference paper [13], the spatial structure information is exploited with a proposed residual graph neural network (RGNN), where each node of the RGNN denotes the body part and the input of each node is the concatenation of all joint coordinates in each part (shown in Fig. 3a). Si et al. [13] only considers the body-level structural information between each part but neglects the intra spatial relationships of joints in each part.

In order to overcome the above problem, we propose a novel network termed as hierarchical spatial reasoning network (HSRN), which can leverage not only the body-level structural information but also the intra spatial relationships of joints in each part with a hierarchical residual graph neural network (HRGNN). Specifically, the HSRN encodes the coordinate vectors via two steps (shown in Fig. 1) to capture the high-level spatial features of skeleton structural information. First, the preliminary encoding process maps the coordinate vector of each joint into the feature space with a linear layer that is shared among different joints. Second, all joint features are fed into the proposed hierarchical residual graph neural network (HRGNN) to model the structural relationships of intra-parts and inter-parts. As illustrated in Fig. 3b, the HRGNN contains an intra-parts residual graph neural network (intra-RGNN) and an inter-parts residual graph neural network (inter-RGNN). The intra-RGNN and the inter-RGNN are essentially two residual graph neural networks, but there are two differences: 1) Each node of the intra-RGNN denotes a joint in a body part and the intra-RGNN explores the spatial structural relationship between joints of each part. Moreover, the intra-RGNN is shared among different parts. For the inter-RGNN, its nodes correspond to the human body parts and it captures the body-level structural information between each part. 2) Fig. 3c shows the architecture of intra-RGNN that is the same as RGNN. In order to enrich the inter-structure representation of part node, we aggregate the joint features in intra-RGNN into the corresponding part nodes of inter-RGNN. Hence the inter-RGNN shown in Fig. 3d has an additional input $\widehat{\boldsymbol{m}}_k^t$ to update the node hidden state, where $\widehat{\boldsymbol{m}}_k^t$ is the aggregated information from intra-RGNN.

Formally, there is a RGNN with K nodes that correspond to the joints in intra-RGNN or the parts in inter-RGNN. We use $\boldsymbol{r}_k^t \in \mathcal{R}^t$ to denote the relation feature vector of each node at time step t, where $k \in \{1, \ldots, K\}$ and $\mathcal{R}^t = \{\boldsymbol{r}_1^t, \ldots, \boldsymbol{r}_K^t\}$. And $\boldsymbol{r}_k^t$ represents the spatial structural relationship of the joint (part) k with other joints (parts). For the intra-RGNN, the $\boldsymbol{r}_k^t$ is initialized with the individual joint feature in each part. For the inter-RGNN, we aggregate the joint features of corresponding part to initialize the $\boldsymbol{r}_k^t$. We use $\boldsymbol{m}_{ik}^t$ to denote the received message of node k from node i at time step t, where $i \in \{1, \ldots, K\}$. Furthermore, the received messages $\boldsymbol{m}_k^t$ of node k from all the neighbors Ω_{v_k} at time step t are defined as follows:

$$\boldsymbol{m}_k^t = \sum_{i \in \Omega_{v_k}} \boldsymbol{m}_{ik}^t = \sum_{i \in \Omega_{v_k}} \boldsymbol{W}_m \boldsymbol{s}_i^{t-1} + \boldsymbol{b}_m \tag{4}$$

where $\boldsymbol{s}_i^{t-1}$ is the state of node i at time step $t-1$, and a shared linear layer of weights $\boldsymbol{W}_m$ and biases $\boldsymbol{b}_m$ will be used to compute the messages for all nodes. After aggregating the messages, the messages are used to update the node hidden state:

$$\boldsymbol{s}_k^t = f_{lstm}\left(\boldsymbol{r}_k^{t-1}, \boldsymbol{m}_k^t, \boldsymbol{s}_k^{t-1}\right) \tag{5}$$

where $f_{lstm}(\cdot)$ denotes the LSTM cell function. Note that the inter-RGNN has an additional input $\widehat{\boldsymbol{m}}_k^t$ to update the node hidden state. Therefore, the updating function of the node hidden state in the inter-RGNN can be defined as follows:

$$\boldsymbol{s}_k^t = f_{lstm}\left(\boldsymbol{r}_k^{t-1}, \boldsymbol{m}_k^t, \boldsymbol{s}_k^{t-1}, \widehat{\boldsymbol{m}}_k^t\right) \tag{6}$$

where the $\widehat{\boldsymbol{m}}_k^t$ is obtained with a linear layer mapping the concatenation of the joint structural features of corresponding part k.

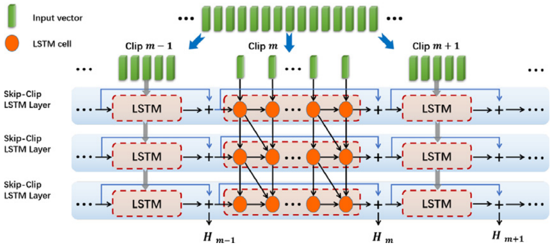


Fig. 4. The architecture of three skip-clip LSTM layers.

Then, we calculate the relation representation $\boldsymbol{r}_k^t$ at time step t via:

$$\boldsymbol{r}_k^t = \boldsymbol{r}_k^{t-1} + \boldsymbol{s}_k^t \tag{7}$$

The residual design of Eq. (7) aims to add the relationship features between each node based on the individual features, so that the representations contain the fusion of both features. Because the joint features in intra-RGNN will be aggregated into the corresponding part nodes of inter-RGNN. The representation of each node in inter-RGNN contains not only the body-level structural information between each part but also the intra spatial relationships of joints in each part. After the HRGNN is updated T times, we extract node-level output of inter-RGNN as the spatial structural relationships $\boldsymbol{r}_k^T$ of each part within each frame, which will be fed into the TSLN to learn the temporal dynamics for each part.

4.2. Temporal stack learning network

Temporal information is another important cue for this task. Rich temporal dynamics drives us to design an effective network to learn the discriminative features of various action. We propose a temporal stack learning network further focuses on modeling detailed temporal dynamics. Instead of aggregating all part features before TSLN in [13], we apply a shared TSLN to process each part feature sequence to learn their unique temporal features then aggregate all part features to predict human action classes (shown in Fig. 1). The TSLN is a two stream architecture, containing a position network and a velocity network. These two networks have the same architecture, which is composed of three skip-clip LSTM layers (shown in Fig. 4).

In TSLN, to capture the detailed temporal information, the long-term sequence can be decomposed into multiple continuous clips. Specifically, a skeleton sequence with N frames is divided into M clips at intervals of d frames. With the hierarchical spatial reasoning network, we can gain the spatial structural features $\{R_1, R_2, \dots, R_M\}$ of body parts, where $R_m = \{\boldsymbol{r}_{md+1}, \boldsymbol{r}_{md+2}, \dots, \boldsymbol{r}_{(m+1)d}\}$ is the set of part features of clip m, and $\boldsymbol{r}_n$ denotes the high-level spatial structural feature of a body part in the skeleton frame $n, n \in \{1, \dots, N\}$. Then, the spatial structural features $\{R_1, R_2, \dots, R_M\}$ are fed into the position network of TSLN. In addition, the inputs of velocity network are the temporal differences $\{V_1, V_2, \dots, V_M\}$ of the spatial features between two consecutive frames, where $V_m = \{\boldsymbol{v}_{md+1}, \boldsymbol{v}_{md+2}, \dots, \boldsymbol{v}_{(m+1)d}\}$. $\boldsymbol{v}_n = \boldsymbol{r}_n - \boldsymbol{r}_{n-1}$ denotes the temporal difference of spatial features of a body part in the skeleton frame n.

Skip-Clip LSTM Layer In the skip-clip LSTM layer, a shared LSTM layer among the continuous clips is used to learn the temporal information of each clip (see Fig. 4). For example, we feed the spatial features of continuous skeleton frames in the clip m into the shared LSTM to capture the short-term temporal dynamics for position network:

$$\boldsymbol{h}_m^{'} = f_{LSTM}(R_m) = f_{LSTM}\left(\{\boldsymbol{r}_{md+1}, \boldsymbol{r}_{md+2}, \dots, \boldsymbol{r}_{(m+1)d}\}\right) \tag{8}$$

where $\boldsymbol{h}_m^{'}$ is the last hidden state of shared LSTM for the clip m, $f_{LSTM}(\cdot)$ denotes the shared LSTM in the skip-clip LSTM layer. It should note that the inputs of LSTM cell between the first skip-clip LSTM layer and the other layers are different. As shown in Fig. 4, the input $\boldsymbol{x}_t^l$ of LSTM cell for the l $(l \geq 2)$ layer at time step t is the concatenation of $\boldsymbol{h}_{t-1}^{l-1}$ and $\boldsymbol{h}_t^{l-1}$, *i.e.*, $\boldsymbol{x}_t^l = concat(\boldsymbol{h}_{t-1}^{l-1}, \boldsymbol{h}_t^{l-1})$, where $\boldsymbol{h}_t^{l-1}$ is the hidden state of the $l-1$ LSTM layer at time step t. This can make the network gain more dependency between two adjacent frames

To aggregate all the detailed temporal dynamics of the mth clip and all previous clips to represent the long-term sequence, we calculate the representation of clip dynamics as follows:

$$\boldsymbol{H}_m = \boldsymbol{H}_{m-1} + \boldsymbol{h}_m^{'} = \sum_{i=1}^{m} \boldsymbol{h}_i^{'} \tag{9}$$

where $\boldsymbol{H}_{m-1}$ and $\boldsymbol{H}_m$ denote the representations of clip $m-1$ and m, respectively. When feeding the clip m into the shared LSTM layer, we initialize the initial hidden state $\boldsymbol{h}_m^0$ of the shared LSTM with the $\boldsymbol{H}_{m-1}$, such that $\boldsymbol{h}_m^0 = \boldsymbol{H}_{m-1}$, which can inherit previous dynamics to learn the short-term dynamics of the mth clip to maintain the dependency between clips. Therefore, it is effective for LSTM layer to learn the temporal dynamics of the short-term clip based on the temporal information of previous clips. And the larger m is the richer temporal dynamics $\boldsymbol{H}_m$ contains.

Learning the Classier After processing each part sequence with a shared TSLN, we can get the local representation $\boldsymbol{H}_{km}$, where $\boldsymbol{H}_{km}$ is the feature of clip m of the part k. Then, the global representation $\widehat{\boldsymbol{H}}_m$ of clip m can be calculated with a linear layer:

$$\widehat{\boldsymbol{H}}_m = F_g(concat(\boldsymbol{H}_{1m}, \dots, \boldsymbol{H}_{Km})) \tag{10}$$

where F_g is the linear layer. Finally, two linear layers F_o are used to compute the scores for C classes:

$$\boldsymbol{O}_m = F_o\left(\widehat{\boldsymbol{H}}_m\right) \tag{11}$$

where $\boldsymbol{O}_m$ is the score of clip m and $\boldsymbol{O}_m = (o_{m1}, o_{m2}, \dots, o_{mC})$. And the output is fed to a softmax classifier to predict the probability being the ith class:

$$\hat{y}_{mi} = \frac{e^{o_{mi}}}{\sum_{j=1}^{C} e^{o_{mj}}}, i = 1, \dots, C \tag{12}$$

where $\hat{y}_{mi}$ indicates the probability that the clip m is predicted as the ith class. And $\hat{\boldsymbol{y}}_m = \left(\hat{y}_{m1}, \dots, \hat{y}_{mC}\right)$ denotes the probability vector of clip m.

With the TSLN, we can extract the dynamic representations $\widehat{\boldsymbol{H}}_m^p$ and $\widehat{\boldsymbol{H}}_m^v$ from the position and velocity for the clip m, respectively. Moreover, the fusion representation $\widehat{\boldsymbol{H}}_m^s$ can be calculated with the sum of $\widehat{\boldsymbol{H}}_m^p$ and $\widehat{\boldsymbol{H}}_m^v$. Hence, according to the clip dynamic representations ($\widehat{\boldsymbol{H}}_m^p$, $\widehat{\boldsymbol{H}}_m^v$ and $\widehat{\boldsymbol{H}}_m^s$), the probability vectors ($\hat{\boldsymbol{y}}_m^p$, $\hat{\boldsymbol{y}}_m^v$ and $\hat{\boldsymbol{y}}_m^s$) can be predicted from the network.

Following [13], the clip based incremental losses are used to optimize the model for a skeleton sequence:

$$\mathcal{L}_p = -\sum_{m=1}^{M} \frac{m}{M} \sum_{i=1}^{C} y_i log \hat{y}_{mi}^p \tag{13}$$

$$\mathcal{L}_v = -\sum_{m=1}^{M} \frac{m}{M} \sum_{i=1}^{C} y_i log \hat{y}_{mi}^v \tag{14}$$

$$\mathcal{L}_s = -\sum_{m=1}^{M} \frac{m}{M} \sum_{i=1}^{C} y_i log \hat{y}_{mi}^s \tag{15}$$

where $\boldsymbol{y} = (y_1, \dots, y_C)$ denotes the groundtruth label. The greater the coefficient $\frac{m}{M}$ is, the richer temporal information the clip contains. The network can benefit from clip-based incremental losses, which will promote the ability of modeling the detailed temporal dynamics for long-term skeleton sequences. Finally, the training loss of our model is defined as follows:

$$\mathcal{L} = \mathcal{L}_s + \alpha_1 \mathcal{L}_p + \alpha_2 \mathcal{L}_v \tag{16}$$

where α_1 and α_2 are the hyper-parameters that are set to 1 to ensure equivalent importance of two loss terms during training.

Due to the mechanisms of skip-clip LSTM (see the Eq. (9)), the representation $\widehat{\boldsymbol{H}}_m^s$ of clip M aggregates all the detailed temporal dynamics of the continuous clips from the position sequences and velocity sequences. In the testing process, we only use the probability vector $\hat{\boldsymbol{y}}_M^s$ to predict the class of the skeleton sequence.

5. Experiments

5.1. Datasets and experimental settings

5.1.1. NTU RGB+D dataset (NTU) [12]

This dataset has 56,880 video samples and is the current largest action recognition dataset with joints annotations that are collected by Microsoft Kinect v2. It contains 60 action classes in total. These actions are performed by 40 distinct subjects. These video samples are collected with three cameras simultaneously in different horizontal views. The joints annotations consist of 3D locations of 25 major body joints. We follow the two standard evaluation protocols proposed in [12] to verify our model: Cross-Subject and Cross-View. For Cross-Subject evaluation, the 40 subjects are split into training and testing groups. Each group consists of 20 subjects. For Cross-View evaluation, all the samples of cameras 2 and 3 are used for training while the samples of camera 1 are used for testing.

5.1.2. SYSU 3D Human-object interaction dataset (SYSU) [55]

This dataset contains 480 video samples with 12 action classes. These actions are performed by 40 subjects. There are 20 joints for each subject in the 3D skeleton sequences.

5.1.3. Northwestern-UCLA dataset (N-UCLA) [56]

This dataset contains 1494 video clips covering 10 action categories. It is captured by three Kinect cameras simultaneously from a variety of viewpoints. Each action sample contains RGBD and human skeleton data performed by 10 different subjects. Each subject has 20 joints. The evaluation protocol is the same as in [56]. We use samples from the first two cameras as training data, and the samples from the other camera as the testing dataset.

5.1.4. UTD Multimodal human action dataset (UTD-MHAD) [22]

This dataset is collected using a Microsoft Kinect sensor and a wearable inertial sensor in an indoor environment. The dataset contains 27 actions performed by 8 subjects (4 females and 4 males). Each subject repeats each action 4 times. It includes 861 data sequences. For the skeleton sequences, each frame contains 20 skeleton joints.

5.1.5. UWA3D multiview activity II dataset (UWA3D) [57]

This dataset has 1075 video samples. It consists of 30 actions performed by 10 subjects. Each action is observed from different views: front view (v1), left side view (V2), right side view (V3) and top view (V4). This dataset is challenging because of varying viewpoints, self-occlusion and high similarity among activities.

5.1.6. Experimental settings

In all our experiments, we set the hidden state dimension of RGNN to 256. For the NTU dataset, the human body is decomposed into $K = 8$ parts: two arms, two hands, two legs, one trunk and one head. For the SYSU dataset, N-UCLA dataset, the UTD-MHAD dataset and the UWA3D dataset, there are $K = 5$ parts: two arms, two legs, and one trunk. The neuron size of LSTM cell in the skip-clip LSTM layer is 512. During training, we randomly select N frames to make a new sequence. During testing, we randomly select N frames with three times to create 3 sequences and the mean score is used to predict the class. The frame number N of skeleton sequence for the NTU, SYSU, N-UCLA, UTD-MHAD and UWA3D dataset are 100, 100, 50, 50, and 50 respectively. The learning rate, initiated with 0.0001, is reduced by multiplying it by 0.1 every 30 epochs. The network is optimized using the ADAM optimizer [58]. Dropout with a probability of 0.5 is utilized to alleviate overfitting during training.

5.2. Ablation study

5.2.1. Effectiveness of the HSRN and TSLN

To validate the effectiveness of the proposed HSRN and TSLN for representing spatial and temporal features, we conduct experiments with different key model components on the five datasets. The comparison results are shown in Table 1.

FC+LSTM For this model, the coordinate vectors of each body part are encoded with the linear layer and three LSTM layers are used to model the sequence dynamics. It is also a two stream network to learn the temporal dynamics from position and velocity.

HSRN+LSTM Compared with FC+LSTM, this model uses hierarchical spatial reasoning network to capture the high-level spatial structural features of skeleton sequences within each frame.

FC+TSLN Compared with FC+LSTM, the temporal stack learning network replaces three LSTM layers to learn the detailed sequence dynamics for skeleton sequences.

HSR-TSL (Position) Compared with our proposed model, the temporal stack learning network of this model only contains the position network.

HSR-TSL (Velocity) Compared with our proposed model, the temporal stack learning network of this model only contains the velocity network.

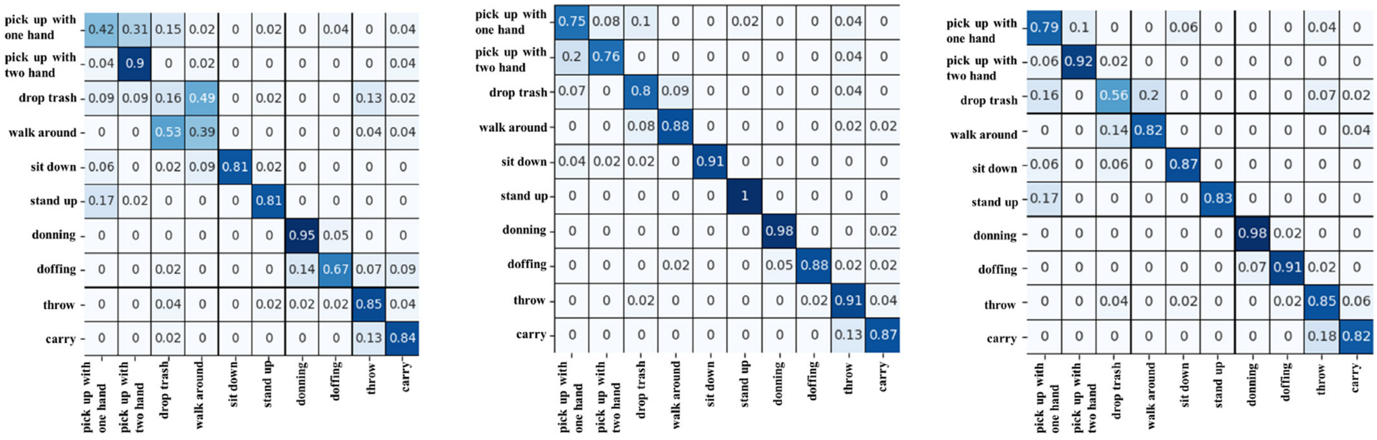
HSR-TSL It denotes our proposed model.

View the effectiveness of the HSRN for skeleton-based action recognition. Compared with *FC+LSTM* and *HSRN+LSTM*, *FC+TSLN* and *HSR-TSL* in Table 1, we can find that the HSRN consistently achieves substantial improvements. For example, *HSRN+LSTM* increases the performance by 4.9% on NTU dataset for cross-view evaluation and 19.4% on N-UCLA dataset due to replacing FC with HSRN. Unlike FC encoding the concatenation of joint coordinates, the proposed HSRN can leverage not only the body-level structural information but also the intra spatial relationships of joints in each part with a hierarchical residual graph neural network. Furthermore, we analyze the classification results with confusion matrix on the N-UCLA dataset. Fig. 5a and b show the confusion matrices of *FC+LSTM* and *HSRN+LSTM*, respectively. As shown in Fig. 5a, *FC+LSTM* is very easy to misclassify the similar actions. For example, 31% samples of “pick up with one hand” are miscategorized to “pick up with two hands”. The reason for this is that *FC+LSTM* neglects the spatial structural information so that it only focuses on the process of “pick up”. Nevertheless, the *HSRN+LSTM* can classify these similar actions to some degree by using HSRN to explore spatial characteristics. The comparison results validate the importance of spatial structural information for skeleton-based action recognition and the effectiveness of the HSRN on modeling spatial features.

View the effectiveness of the TSLN for skeleton-based action recognition. Due to the powerful ability to capture the dynamics of sequences via cycles for RNN or LSTM, many methods apply RNN or LSTM to extract discriminative temporal features from skeleton sequence, such as [10–12,38–40,59]. In general, these methods

Table 1
The comparison results of analyzing the effectiveness of the HSRN and TSLN in accuracy (%). CS and CV denotes cross-subject and cross-view, respectively.

Methods	NTU		SYSU		N-UCLA	UTD-MHAD	UWA3D
	CS	CV	Setting-1	Setting-2			
FC + LSTM	76.3	84.5	50.3	50.7	67.9	77.2	66.5
HSRN + LSTM	80.8	89.4	66.0	65.7	87.3	84.4	68.9
FC + TSLN	84.7	92.1	79.2	78.8	83.4	83.9	77.0
HSR-TSL(Position)	83.5	90.5	77.0	77.8	88.2	83.3	64.6
HSR-TSL(Velocity)	84.5	92.2	72.7	71.8	91.4	85.1	68.6
HSR-TSL (Ours)	**87.7**	**94.4**	**82.5**	**82.8**	**94.8**	**94.4**	**77.9**



	pick up with one hand	pick up with two hand	drop trash	walk around	sit down	stand up	donning	doffing	throw	carry
pick up with one hand	0.42	0.31	0.15	0.02	0	0.02	0	0.04	0	0.04
pick up with two hand	0.04	0.9	0	0.02	0	0	0	0	0	0.04
drop trash	0.09	0.09	0.16	0.49	0	0.02	0	0	0.13	0.02
walk around	0	0	0.53	0.39	0	0	0	0	0.04	0.04
sit down	0.06	0	0.02	0.09	0.81	0.02	0	0	0	0
stand up	0.17	0.02	0	0	0	0.81	0	0	0	0
donning	0	0	0	0	0	0	0.95	0.05	0	0
doffing	0	0	0.02	0	0	0	0.14	0.67	0.07	0.09
throw	0	0	0.04	0	0	0.02	0.02	0.02	0.85	0.04
carry	0	0	0.02	0	0	0	0	0	0.13	0.84

(a) FC+LSTM

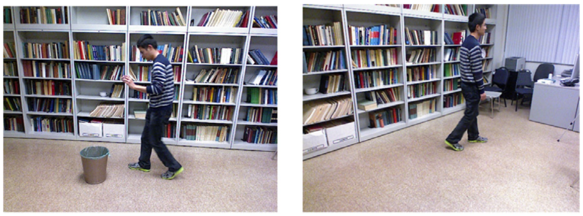
	pick up with one hand	pick up with two hand	drop trash	walk around	sit down	stand up	donning	doffing	throw	carry
pick up with one hand	0.75	0.08	0.1	0	0	0.02	0	0	0.04	0
pick up with two hand	0.2	0.76	0	0	0	0	0	0	0.04	0
drop trash	0.07	0	0.8	0.09	0	0	0	0	0.04	0
walk around	0	0	0.08	0.88	0	0	0	0	0.02	0.02
sit down	0.04	0.02	0.02	0	0.91	0	0	0	0	0
stand up	0	0	0	0	0	1	0	0	0	0
donning	0	0	0	0	0	0	0.98	0	0	0.02
doffing	0	0	0	0.02	0	0	0.05	0.88	0.02	0.02
throw	0	0	0.02	0	0	0	0	0.02	0.91	0.04
carry	0	0	0	0	0	0	0	0	0.13	0.87

(b) HSRN+LSTM

	pick up with one hand	pick up with two hand	drop trash	walk around	sit down	stand up	donning	doffing	throw	carry
pick up with one hand	0.79	0.1	0	0	0.06	0	0	0	0.04	0
pick up with two hand	0.06	0.92	0.02	0	0	0	0	0	0	0
drop trash	0.16	0	0.56	0.2	0	0	0	0	0.07	0.02
walk around	0	0	0.14	0.82	0	0	0	0	0	0.04
sit down	0.06	0	0.06	0	0.87	0	0	0	0	0
stand up	0.17	0	0	0	0	0.83	0	0	0	0
donning	0	0	0	0	0	0	0.98	0.02	0	0
doffing	0	0	0	0	0	0	0.07	0.91	0.02	0
throw	0	0	0.04	0	0.02	0	0	0.02	0.85	0.06
carry	0	0	0	0	0	0	0	0	0.18	0.82

(c) FC+TSLN

Fig. 5. Confusion matrix comparison on the N-UCLA dataset. (a), (b), (c) show the confusion matrices of the FC+LSTM, HSRN+LSTM and FC+TSLN, respectively.



(a) drop trash (b) walk around

Fig. 6. The frames of "drop trash" and "walk around". (a) "drop trash", (b) "walk around".

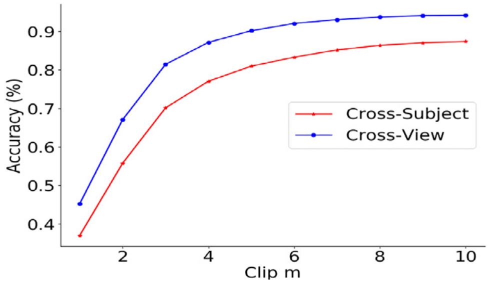


Fig. 7. The accuracy of the increasing clips on the testing set of NTU RGB+D dataset.

directly process the entire skeleton sequences to learn temporal dynamics and the last hidden representation of RNN or LSTM is used to recognize the actions. However, for long-term sequences, such as the sequences with more 100 frames on NUT dataset, the last hidden state may represent the temporal dynamics that occur quite a lot in the sequences, so that it may ignore the detailed or instantaneous behavior features which may express the true action. For example, Fig. 6 shows the action "drop trash" and "walk around" on the N-UCLA dataset. The human walks all the time in the sequence of action "drop trash" and the behavior of "drop trash" only lasts for a very short time. As shown in Fig 5a, it is very confusing for *FC+LSTM* to recognize "drop trash" and "walk around", the reason of which is that *FC+LSTM* focuses on the features of "walk" and ignores the detailed temporal features of "drop trash". We propose a temporal stack learning network to further focus on modeling detailed temporal dynamics. The results in Fig. 5c illustrate that the *FC+TSLN* has better discriminatory performances. Furthermore, in Table 1, we can observe *FC+TSLN* can obviously improve the recognition performances on the long-term sequence datasets.

We also show the process of temporal stack learning in Fig. 7. With the increase of *m*, the much richer temporal information is contained in the representation of a sequence. And the network can consider more temporal dynamics of the details to recognize human action, so as to improve the accuracy. Furthermore, the two stream architecture of temporal stack learning network is effective to learn the temporal dynamics from the velocity sequence and position sequence. The persuasive experimental results verify the effectiveness of the TSLN on exploring the detailed temporal dynamics for long-term sequences.

Considering the importance of spatial structure information and the detailed temporal dynamics for skeleton-based action recognition, the proposed *HSR-TSL* captures discriminative spatio-temporal features to predict the action class. Therefore, compared with *FC+LSTM, HSRN+LSTM* and *FC+TSLN*, the *HSR-TSL* greatly increases the performances on all datasets.

Table 2
The comparison with the preliminary work [13] on NTU dataset in accuracy (%).

HRGNN	CS	CV
SRN + LSTM [13]	78.7	87.3
FC + TSLN [13]	83.8	91.6
SR-TSL [13]	84.8	92.4
HSRN + LSTM	80.8	89.4
FC + TSLN	84.7	92.1
HSR-TSL (Ours)	87.7	94.4

Comparison with the preliminary work [13]. In this work, we introduce a novel graph structure termed as hierarchical residual graph neural network that is capable of learning not only the body-level structural information between each part but also the intra spatial relationship of joint in each part. As shown in Table 2, *HSRN + LSTM* achieves better results than *SRN + LSTM* [13] that only focuses on the body-level structural information between each part. This demonstrates that the proposed extension of HSRN is effective to learn special information. Furthermore, instead of aggregating all part features before TSLN, we apply TSLN to process each part sequence to learn their unique temporal features then aggregate all part features to predict human action classes. It can be seen that, compared with *FC + TSLN* of [13], *FC + TSLN* of this work further improves the performances. With the extensions of spatial structure and temporal dynamic learning networks, our *HSR-TSL* significantly outperforms the *SR-TSL* [13].

5.2.2. Influence of HRGNN

HRGNN is proposed to model the structural relationships of intra-parts and inter-parts. In order to analyze the influence of the HRGNN designs, such as intra-RGNN, inter-RGNN and the residual design of RGNN, we conduct experiments on NTU, SYSU, N-UCLA, UTD-MHAD and UWA3D datasets in Table 3. We compare our model with three baselines as follows:

HSR-TSL w/o inter In *HSR-TSL w/o inter*, the HRGNN removes inter-RGNN and only contains intra-RGNN to capture spatial structural information.

HSR-TSL w/o intra In *HSR-TSL w/o intra*, the HRGNN removes intra-RGNN and only contains inter-RGNN to capture spatial structural information.

HSR-TSL w/o Res In *HSR-TSL w/o Res*, the HRGNN dose not contain the residual design, which means that the relation representation $\boldsymbol{r}_k^t$ equals the node state $\boldsymbol{s}_k^t$ and is not calculated by Eq. (7).

Table 3 shows the results of the proposed HSR-TSL and the three baseline models. When only considering the spatial structural relationship between joints of each part with the intra-RGNN, we can see that the performances of *HSR-TSL* are better than that of *HSR-TSL w/o inter*. Similarly, *HSR-TSL* performs better than *HSR-TSL w/o intra* that only explores the body-level structural information between each part with the inter-RGNN. The comparison results demonstrate that the proposed HRGNN is an effective framework to extract the intra spatial information of each part and body-level structural information between each part. In addition, we analyze the necessity of the residual design of HRGNN. As shown in Table 3, compared with *HSR-TSL w/o Res* without the residual design, *HSR-TSL* obviously improves the performances due to the residual design of HRGNN. The reason is that graph neural network focuses on capturing the relationship features between each node so that the node hidden state may weaken the original node representation. The residual design of HRGNN aims to add the relationship features between each joint/part based on their individual features, so that the representations contain the fusion of both features.

5.2.3. Influence of skip-clip LSTM and clip-based incremental loss

For temporal features, the proposed TSLN focuses on leveraging detailed temporal dynamics with three skip-clip LSTM layers. The clip-based incremental loss is proposed to further promote the ability of modeling the detailed temporal dynamics for long-term skeleton sequences. In order to demonstrate the effectiveness of the Skip-Clip LSTM and clip-based incremental loss, we conduct experiments to compare our model with two baselines as follow:

HSR-TSL w/o Skip In *HSR-TSL w/o Skip*, the TSLN uses the standard LSTM without the skip connections to learn temporal features.

HSR-TSL w/o ClLoss In *HSR-TSL w/o ClLoss*, we use standard cross-entropy loss as the training loss instead of the clip-based incremental loss.

The comparison results are shown in Table 3. Compared with *HSR-TSL w/o Skip, HSR-TSL* has significantly performance improvements on all datasets due to the skip-clip LSTM. The long-term sequences contain various temporal dependencies. The proposed skip-clip LSTM can effectively capture the short-term temporal dy-

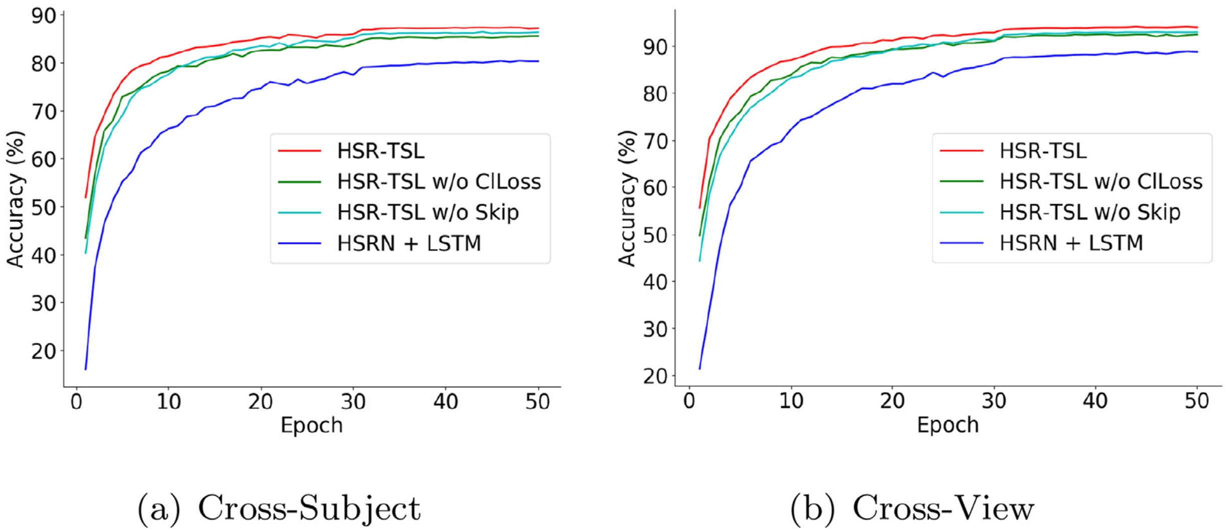


Fig. 8. The accuracy of the baselines and our model on the testing set of NTU RGB+D dataset during learning phase. (a) and (b) show the comparison results for cross-subject evaluation, and cross-view evaluation, respectively.

Table 3
The comparison results on five datasets in accuracy (%). We compare the performances of several variants and our proposed model to verify the effectiveness of some components, such as HRGNN, skip-clip LSTM and clip-based incremental loss.

Methods	NTU		SYSU		N-UCLA	UTD-MHAD	UWA3D
	CS	CV	Setting-1	Setting-2			
HSR-TSL w/o inter	86.5	93.4	80.6	81.0	91.8	85.6	77.3
HSR-TSL w/o intra	86.3	93.4	80.0	80.7	89.2	86.7	77.2
HSR-TSL w/o Res	85.7	93.0	74.5	77.0	86.4	75.4	65.4
HSR-TSL w/o Skip	86.4	93.0	77.7	78.6	91.6	85.4	77.3
HSR-TSL w/o ClLoss	85.6	92.4	80.5	80.4	93.1	91.6	77.1
HSR-TSL (Ours)	**87.7**	**94.4**	**82.5**	**82.8**	**94.8**	**94.4**	**77.9**

Table 4
Comparison results with different time steps for the HRGNN on NTU dataset in accuracy (%).

HRGNN	Cross-Subject	Cross-View
$T = 1$	85.9	93.2
$T = 2$	86.7	93.7
$T = 3$	87.1	94.0
$T = 4$	87.4	**94.4**
$T = 5$	**87.7**	94.2
$T = 6$	87.2	93.9

Table 5
Comparison results with different the length d of clips on NTU dataset in accuracy (%).

TSLN	Cross-Subject	Cross-View
$d = 2$	84.6	93.1
$d = 4$	86.8	94.0
$d = 6$	87.1	94.3
$d = 8$	87.4	**94.4**
$d = 10$	**87.7**	**94.4**
$d = 15$	87.2	94.1
$d = 20$	86.8	93.4

Table 6
Comparison results with different weights (α_1, α_2) of training losses on NTU dataset in accuracy (%).

α_1, α_2	Cross-Subject	Cross-View
0.5, 0.8	87.3	94.2
0.8, 0.5	87.4	94.2
0.5, 1.0	87.4	94.2
1.0, 0.5	87.5	94.3
1.0, 1.0	87.7	94.4

Table 7
The comparison results on NTU RGB+D dataset with Cross-Subject and Cross-View settings in accuracy (%).

Methods	Cross-Subject	Cross-View
Lie Group [27]	50.1	52.8
Dynamic Skeletons [55]	60.2	65.2
HBRNN-L [10]	59.1	64.0
Part-aware LSTM [12]	62.9	70.3
Trust Gate ST-LSTM [59]	69.2	77.7
Two-stream RNN [39]	71.3	79.5
STA-LSTM [11]	73.4	81.2
Ensemble TS-LSTM [40]	74.6	81.3
Visualization CNN [31]	76.0	82.6
VA-LSTM [38]	79.4	87.6
ST-GCN [45]	81.5	88.3
HCN [32]	86.5	91.1
PB-GCN [44]	87.5	93.2
AGC-LSTM [46] (Part)	87.5	93.8
JSR + JMR + BSR + BMR [36]	85.6	92.0
SR-TSL [13]	84.8	92.4
HSR-TSL (Ours)	**87.7**	**94.4**

Table 8
The comparison results on SYSU dataset in accuracy (%).

Methods	Setting-1	Setting-2
LAFF [60]	-	54.2
Dynamic Skeletons [55]	75.5	76.9
ST-LSTM + Trust Gate [42]	-	76.5
VA-LSTM [38]	76.9	77.5
BNN [47]	-	82.0
LGN [34]	-	**83.3**
SR-TSL [13]	80.7	81.9
HSR-TSL (Ours)	**82.5**	**82.8**

namics and the long-term dependencies between two adjacent clips. In addition, we can observe that *HSR-TSL* achieves better performances compared with *HSR-TSL w/o ClLoss* on both datasets. Fig. 8 shows the accuracy of our model and the baselines on the testing set of NTU RGB+D dataset during learning phase. We can find that the skip-clip LSTM and clip-based incremental loss can speed up convergence and obviously improve the performance.

5.2.4. Influence of hyper-parameters

We explore the effects of three important hyper-parameters: the time step T of the HRGNN, the length d of clips and the weights (α_1, α_2, α_3) of training losses. The comparison results are shown in Tables 4–6. For the time step T, we can find that the performance increases by a small amount when increasing T, and saturates soon. We think that the body-level structural information and the intra spatial relationships of joints in each part can be learned quickly by the HSRN. For the length d of clips, with the increase of d, the performance is significantly improved and then decreased. The reason of decreased is that learning short-term dynamic does not require too many frames for each clip. For the weights (α_1, α_2) of training losses, it can be seen that the effect of weights on performance is very small. This illustrates that our proposed model is robust and effective for skeleton-based action recognition.

5.3. Comparisons to other state-of-the-art approaches

In this section, we compare the performance of our proposed model against several state-of-the-art approaches on the NTU, SYSU, N-UCLA, UTD-MHAD and UWA3D datasets in Tables 7–11, respectively.

Table 9
The comparison results on N-UCLA dataset in accuracy (%).

Methods	Accuracy
Lie group [27]	74.2
Actionlet ensemble [15]	76.0
HBRNN-L [10]	78.5
Visualization CNN [31]	86.1
Ensemble TS-LSTM [40]	89.2
AGC-LSTM [46] (Part)	90.0
Clips+MTCNN [33]	93.4
Skeleton Recovery [35]	94.4
JSR [36]	90.9
JMR [36]	84.4
BSR [36]	89.4
BMR [36]	87.4
JSR + JMR + BSR + BMR [36]	**95.0**
SR-TSL [13]	89.3
HSR-TSL (Ours)	**94.8**

Table 10
The comparisons between our proposed method and state-of-the-art methods on UTD-MHAD dataset in accuracy (%).

Methods	Sensor	Accuracy
Cov3DJ [26]	Kinect	85.6
Kinect [22]	Kinect	66.1
Inertial [22]	Inertial	67.2
Kinect&Inertial [22]	Kinect + Inertial	79.1
JTM [30]	Kinect	85.8
Optical Spectra [61]	Kinect	87.0
3DHOT-MBC [23]	Kinect	84.4
JDM [28]	Kinect	88.1
BNN [47]	Kinect	92.1
JSR [36]	Kinect	91.9
JMR [36]	Kinect	92.3
BSR [36]	Kinect	92.8
BMR [36]	Kinect	93.3
JSR + JMR + BSR + BMR [36]	Kinect	**98.4**
SR-TSL [13]	Kinect	93.1
HSR-TSL (Ours)	Kinect	**94.4**

5.3.1. NTU Dataset

Following the two standard evaluation protocols in [12], we compare the proposed HSR-TSL with the previous state-of-the-art methods, such as some traditional approaches based on hand-crafted features [27,55], RNN-based approaches [10–12,38–40,59], CNN-based approaches [31,32] and graph-based models [44,45]. Tables 7 shows the comparison results on the large-scale NTU dataset. We can observe that our proposed method achieves the best performances of 87.7% and 94.4% for cross-subject evaluation and cross-view evaluation, respectively. Compared with VA-LSTM [38] that is the current best RNN-based method for action recognition, our results are about 8.3% and 6.8% better than VA-LSTM on the NTU dataset. HCN [32] is the state-of-the-art CNN model. Our performances significantly outperform the HCN [32] by about 1.2% and 3.3% for cross-subject evaluation and cross-view evaluation, respectively. In this work, we use a hierarchical spatial reasoning network to capture the spatial structural features by a HRGNN. For graph-based methods, the proposed HSR-TSL outperforms ST-GCN [45], PB-GCN [44], AGC-LSTM [46] on this dataset. Moreover, compared with the preliminary work (SR-TSL) [13], our HSR-TSL consistently achieves substantial improvements.

5.3.2. SYSU Dataset

We follow the standard evaluation protocols in [55] for this dataset. In the first setting (setting-1), for each activity class, half of the samples are used for training and the rest for testing. In the second setting (setting-2), half of subjects are used to train the model and the rest for testing. For each setting, there is 30-fold cross validation. The averaged results of the two evaluation protocols are shown in Table 8. With hierarchical spatial reasoning and temporal stack learning network, our method achieves competitive results with 82.5% and 82.8% in setting-1 evaluation and setting-2 evaluation respectively, while the latest method [34] outperforms ours by 0.5% in setting-2 evaluation. They need transform skeletons to images inputs to the CNN by the skeleton visualization preprocessing. Hence, the computational cost of [34] is much higher than our model. Moreover, the CNN networks they leverage are significantly more complex than our model.

5.3.3. N-UCLA Dataset

We follow the standard evaluation protocol in [56]. As shown in Table 9, our performances significantly outperform the state-of-the-art methods [33,35,46]. Ensemble TS-LSTM [40] as the most similar work to ours employs multiple Temporal Sliding LSTM to extract short-term, medium-term and long-term temporal dynamics respectively, which has similar functionality to our temporal stack learning network. However, our method outperforms Ensemble TS-LSTM [40] by 2.6%. Li et al. [36] uses a four-stream CNNs to extract features from Joint-Shape Representation (JSR), Joint-Motion Representation (JMR), Bone-Shape Representation (BSR) and Bone-Motion Representation (BMR), respectively. Although the high computational cost for the transformation of four Representations (JSR, JMR, BSR and BMR), we can see that our method achieves better results than each individual Representation. And compared with the JSR + JMR + BSR + BMR [36], our light model also achieves a competitive result.

5.3.4. UTD-MHAD Dataset

We follow the cross-subject protocol proposed by Chen et al. [22] to evaluate the performance, where the samples of subjects 1, 3, 5, 7 are used for training and subjects 2,4,6,8 are used for testing. The results are shown in Table 10. Compared with the previous state-of-the-art methods [23,28,47], our HSR-TSL outperforms them by a large margin. Compared with [36], our method achieves better results than each individual Representation (JSR[36], JMR[36], BSR[36] and BMR[36]), while JSR + JMR + BSR + BMR [36] outperforms ours by about 4%. In JSR + JMR + BSR + BMR [36], its four-stream CNNs network is significantly more complex than our model and has higher computation cost. Therefore, our light model can achieve competitive results for this task.

Table 11
The comparison results on UWA3D dataset in accuracy (%).

Training views	V1 & V2		V1 & V3		V1 & V4		V2 & V3		V2 & V4		V3 & V4		Average
Testing views	V3	V4	V2	V4	V2	V3	V1	V4	V1	V3	V1	V2	
HOJ3D [25]	15.3	28.2	17.3	27.0	14.6	13.4	15.0	12.9	22.1	13.5	20.3	12.7	17.7
AE [15]	45.0	40.4	35.1	36.9	34.7	36.0	49.5	29.3	57.1	35.4	49.0	29.3	39.8
LARP [27]	49.4	42.8	34.6	39.7	38.1	44.8	53.3	33.5	53.6	41.2	56.7	32.6	43.4
ESV (Synthesized + Pre-trained) [31]	72.3	76.3	64.7	75.5	63.5	74.0	83.1	75.1	82.4	71.1	83.5	63.5	73.8
SR-TSL [13]	68.9	80.7	67.3	79.9	71.6	72.5	77.6	78.7	81.5	67.3	71.7	66.9	74.6
HSR-TSL (Ours)	**74.9**	**81.1**	**72.8**	**80.7**	**74.0**	**72.5**	**83.5**	**80.7**	**82.0**	**72.1**	**83.5**	**76.8**	**77.9**

5.3.5. *UWA3D dataset*

We follow the standard evaluation protocol in [57] on UWA3D dataset. This dataset is observed from 4 views. It contains 12 kinds of evaluation partitions. For each partition with the samples of 3 views, two views are used as training data and the samples from the remaining view are used as testing data. Table 11 shows the comparison results on UWA3D dataset. Although this dataset is challenging due to varying viewpoints, our method outperforms ESV [31] by 4.1%.

6. Conclusions

We present a novel model with hierarchical spatial reasoning and temporal stack learning for long-term skeleton based action recognition. The proposed hierarchical spatial reasoning network can effectively capture the body-level structural information between each part and the intra spatial relationships of joints in each part with a hierarchical residual graph neural network, while the temporal stack learning network can model the detailed temporal dynamics of skeleton sequences. A clip-based incremental loss is employed to further improve the ability of stack learning, which provides an effective way to solve long-term sequence optimization. With extensive experiments on five challenging benchmarks, we verify the contributions and demonstrate the effectiveness of our model for the skeleton based action recognition. The proposed method, though having enabled unprecedented achievements, ignores the co-occurrence relationship between spatial and temporal domains. The effective coupling of spatial features into temporal representations is an important subject that is worth exploring in the future work. Moreover, inspired by the success of learning two-level spatial features for skeleton-based action recognition, learning the multi-scale temporal features is also an interesting topic to enhance the discriminating of temporal representations. From a viewpoint of the computational efficiency of graph neural network, designing a simple yet effective graph-based networks is very beneficial for practical application of this task.

Acknowledgements

This work is jointly supported by National Key Research and Development Program of China (2016YFB1001000), National Natural Science Foundation of China (61420106015, 61976214, 61721004), Shandong Provincial Key Research and Development Program (Major Scientific and Technological Innovation Project) (NO.2019JZZY010119).

References

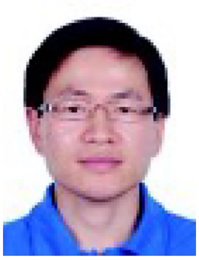
[1] R. Poppe, A survey on vision-based human action recognition, Image Vis. Comput. (2010) 976–990.
[2] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Comput. Vis. Image Underst. (2011) 224–241.
[3] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, NIPS, 2014.
[4] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L.V. Gool, Temporal segment networks: towards good practices for deep action recognition, ECCV, 2016.
[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Slarge-scale video classification with convolutional neural networks, CVPR, 2014.
[6] G. Johansson, Visual perception of biological motion and a model for its analysis, Percept. Psychophys. (1973) 201–211.
[7] Z. Zhang, Microsoft kinect sensor and its effect, IEEE Multimedia (2012) 4–10.
[8] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, CVPR, 2017.
[9] J.K. Aggarwal, L. Xia, Human activity recognition from 3d data: a review, Pattern Recognit. Lett. (2014) 70–80.
[10] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, CVPR, 2015.
[11] S. Song, C. Lan, J. Xing, W. Zeng, J. Liu, An end-to-end spatio-temporal attention model for human action recognition from skeleton data, AAAI, 2017.
[12] A. Shahroudy, J. Liu, T.-T. Ng, G. Wang, Ntu rgb+d: a large scale dataset for 3d human activity analysis, CVPR, 2016.
[13] C. Si, Y. Jing, W. Wang, L. Wang, T. Tan, Skeleton-based action recognition with spatial reasoning and temporal stack learning, ECCV, 2018.
[14] H. Rahmani, A. Mian, Learning a non-linear knowledge transfer model for cross-view action recognition, CVPR, 2015.
[15] J. Wang, Z. Liu, Y. Wu, J. Yuan, Learning actionlet ensemble for 3d human action recognition, IEEE TPAMI (2014) 914–927.
[16] R. Vemulapalli, R. Chellapa, Rolling rotations for recognizing human actions from 3d skeletal data, CVPR, 2016.
[17] Y. Du, Y. Fu, L. Wang, Skeleton based action recognition with convolutional neural network, ACPR. IEEE, 2015.
[18] M.E. Hussein, M. Torki, M.A. Gowayyed, M. El-Saban, Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations, IJCAI, 2013.
[19] Q. Ke, M. Bennamoun, S. An, F. Sohel, F. Boussaid, A new representation of skeleton sequences for 3d action recognition, CVPR, 2017.
[20] T.S. Kim, A. Reiter, Interpretable 3d human action analysis with temporal convolutional networks, CVPR Workshops, 2017.
[21] X. Cai, W. Zhou, L. Wu, J. Luo, H. Li, Effective active skeleton representation for low latency human action recognition, IEEE Trans Multimedia (2016) 141–154.
[22] C. Chen, R. Jafari, N. Kehtarnavaz, UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor, ICIP, 2015.
[23] B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han, L. Shao, Action recognition using 3d histograms of texture and a multi-class boosting classifier, IEEE TIP (2017) 4648–4660.
[24] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, CVPR, 2012.
[25] X. Lu, C.-C. Chen, J.K. Aggarwal, View invariant human action recognition using histograms of 3d joints, CVPRW, 2012.
[26] M.E. Hussein, M. Torki, M.A. Gowayyed, M. El-Saban, Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations, IJCAI, 2013.
[27] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3d skeletons as points in a lie group, CVPR, 2014.
[28] C. Li, Y. Hou, P. Wang, W. Li, Joint distance maps based action recognition with convolutional neural networks, IEEE Signal Process. Lett. (2017) 624–628.
[29] B. Li, Y. Dai, X. Cheng, H. Chen, Y. Lin, M. He, Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN, in: IEEE International Conference on Multimedia & Expo Workshops, 2017.
[30] P. Wang, Z. Li, Y. Hou, W. Li, Action recognition based on joint trajectory maps using convolutional neural networks, ACM MM, 2016.
[31] M. Liu, H. Liu, C. Chen, Enhanced skeleton visualization for view invariant human action recognition, Pattern Recognit. (2017) 346–362.
[32] C. Li, Q. Zhong, D. Xie, S. Pu, Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation, IJCAI, 2018.
[33] Q. Ke, M. Bennamoun, S. An, F. Sohel, F. Boussaid, Learning clip representations for skeleton-based 3d action recognition, IEEE TIP (2018) 2842–2855.
[34] Q. Ke, M. Bennamoun, H. Rahmani, S. An, F. Sohel, F. Boussaid, Learning latent global network for skeleton-based action prediction, IEEE TIP (2019) 959–970.
[35] Q. Nie, J. Wang, X. Wang, Y. Liu, View-invariant human action recognition based on a 3d bio-constrained skeleton model, IEEE TIP (2019) 3959–3972.
[36] Y. Li, R. Xia, X. Liu, Learning shape and motion representations for view invariant skeleton-based action recognition, Pattern Recognit. (2020) 107293.
[37] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks, AAAI, 2016.
[38] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive recurrent neural networks for high performance human action recognition from skeleton data, ICCV, 2017.
[39] H. Wang, L. Wang, Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks, CVPR, 2017.
[40] I. Lee, D. Kim, S. Kang, S. Lee, Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks, ICCV, 2017.
[41] W. Li, L. Wen, M.-C. Chang, S.N. Lim, S. Lyu, Adaptive RNN tree for large-scale human action recognition, ICCV, 2017.
[42] J. Liu, A. Shahroudy, D. Xu, A.C. Kot, G. Wang, Skeleton-based action recognition using spatio-temporal LSTM network with trust gates, IEEE TPAMI (2018) 3009–3021.
[43] C. Fu, W. Pei, Q. Cao, C. Zhang, Y. Zhao, X. Shen, Y.-W. Tai, Non-local recurrent neural memory for supervised sequence modeling, ICCV, 2019.
[44] K. Thakkar, P.J. Narayanan, Part-based graph convolutional network for action recognition, BMVC, 2018.
[45] S. Yan, Y. Xiong, D. Lin, xiaoou Tang, Spatial temporal graph convolutional networks for skeleton-based action recognition, AAAI, 2018.
[46] C. Si, W. Chen, W. Wang, L. Wang, T. Tan, An attention enhanced graph convolutional LSTM network for skeleton-based action recognition, CVPR, 2019.
[47] Z. Rui, W. Kang, S. Hui, J. Qiang, Bayesian graph convolution LSTM for skeleton based action recognition, ICCV, 2019.
[48] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, Q. Tian, Actional-structural graph convolutional networks for skeleton-based action recognition, CVPR, 2019.
[49] M. Henaff, J. Bruna, Y. LeCun, Deep convolutional networks on graph-structured data, in: arXiv:1506.05163, 2015.

[50] D.K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, R.P. Adams, Convolutional networks on graphs for learning molecular fingerprints, NIPS, 2015.
[51] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, ICLR, 2014.
[52] M. Niepert, M. Ahmed, K. Kutzkov, Learning convolutional neural networks for graphs, ICML, 2016.
[53] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. (2009) 61–80.
[54] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. (1997) 1735–1780.
[55] J.-F. Hu, W.-S. Zheng, J. Lai, J. Zhang, Jointly learning heterogeneous features for RGB-D activity recognition, CVPR, 2015.
[56] J. Wang, X. Nie, Y. Xia, Y. Wu, S.-C. Zhu, Cross-view action modeling, learning, and recognition, CVPR, 2014.
[57] H. Rahmani, A. Mahmood, D. Huynh, A. Mian, Histogram of oriented principal components for cross-view action recognition, IEEE TPAMI (2016) 2430–2443.
[58] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, ICLR, 2015.
[59] J. Liu, A. Shahroudy, D. Xu, G. Wang, Spatio-temporal LSTM with trust gates for 3d human action recognition, ECCV, 2016.
[60] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, J. Lai, Real-time RGB-D activity prediction by soft regression, ECCV, 2016.
[61] Y. Hou, Z. Li, P. Wang, W. Li, Skeleton optical spectra-based action recognition using convolutional neural networks, IEEE Trans. Circuits Syst. Video Technol. (2016) 807–811.

Chenyang Si received the B.S. degree from Zhengzhou University in 2016. He is currently pursuing the Ph.D. degree at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. His research interests include action recognition, person retrieval, and deep learning.

Ya Jing received the B.S. degree in Department of Automation from Beihang University in 2016. She is currently pursuing the Ph.D. degree at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. Her research interests include situation recognition, person retrieval, action recognition and deep learning.

Wei Wang received the B.E. degree in Department of Automation from Wuhan University in 2005, and Ph.D. degree in School of Information Science and Engineering at the Graduate University of Chinese Academy of Sciences (GUCAS) in 2011. Since July 2011, Dr. Wang has joined NLPR as an assistant professor. His research interests focus on computer vision, pattern recognition and machine learning, particularly on the computational modeling of visual attention, deep learning and multimodal data analysis. He has published more than ten papers in the leading international conferences such as CVPR and ICCV.



Liang Wang (SM'09) received both the B.S. and M.S. degrees from Anhui University in 1997 and 2000 respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), in 2004. From 2004 to 2010, he has been working as 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 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 at highly-ranked international journals such as IEEE TPAMI and IEEE TIP, and leading international conferences such as CVPR, ICCV and ICDM. He is an associate editor of IEEE Transactions on SMC-B. He is currently an IAPR Fellow and Senior Member of IEEE.

Tieniu Tan received his B.Sc. degree in electronic engineering from Xi'an Jiaotong University, China, in 1984, and his M.Sc. and Ph.D. degrees in electronic engineering from Imperial College London, U.K., in 1986 and 1989, respectively. He is currently a professor in the Center for Research on Intelligent Perception and Computing, NLPR, CASIA, China. He has published more than 450 research papers in refereed international journals and conferences in the areas of image processing, computer vision and pattern recognition, and has authored or edited 11 books. His research interests include biometrics, image and video understanding, information hiding, and information forensics. He is a fellow of the CAS, the TWAS, the BAS, the IEEE, the IAPR, the UK Royal Academy of Engineering, and the Past President of IEEE Biometrics Council.