Hybrid Features for Skeleton-based Action Recognition based on Network Fusion

Abstract

In recent years, the topic of skeleton-based human action recognition has attracted significant attention from researchers and practitioners in graphics, vision, animation, and virtual environments. The most fundamental issue is how to learn an effective and accurate representation from spatio-temporal action sequences towards improved performance, and this paper aims to address the aforementioned challenge. In particular, we design a novel method of hybrid features' extraction based on the construction of multi-stream networks and their organic fusion. First, we train a CNN model to learn CNN-based features with the raw skeleton coordinates and their temporal differences serving as input signals. The attention mechanism is injected into the CNN model to weigh more effective and important information. Then, we employ LSTM to obtain long-term temporal features from action sequences. Finally, we generate the hybrid features by fusing the CNN and LSTM networks, and we classify action types with the hybrid features. The extensive experiments are performed on several large-scale publically-available databases, and promising results demonstrate the efficacy and effectiveness of our proposed framework.

Keywords: Action Recognition; Human Skeleton; Hybrid Features; Multi-stream Neural Network; CNN; LSTM

1 Introduction

Human action recognition is one of the fundamental topics in computer vision. It has a wide range of applications in many areas, such as intelligent video surveillance, sport analysis and human computer interaction. Compared with RGB images or videos data, the skeleton-based action recognition has quite a few advantages. On one hand, skeleton data is a high-level abstraction of human actions and is robust against interference of backgrounds. On the other hand, the data size of skeleton data is extremely small, since it is represented as three-dimensional (3D) coordinates of the major body joints. The key tasks in skeleton-based action recognition are to extract distinguishable spatial temporal features to represent human action sequence and to acquire the high recognition accuracy. To deal with these issues, scholars have done abundant related research. With the successful development of deep learning, the researches have made significant improvement. However, the problems are not yet fully addressed.

Some scholars designed hand-crafted features to represent skeleton sequences, such as covariance matrices of joint trajectories [1], histograms of 3D joint location (HOJ3D) [2] and relative positions of joints [3]. These methods pay more attention to the spatial information. They capture the temporal dynamics through hierarchical structures. To extract more informative temporal dynamics, the recurrent neural networks (RNNs) are adopted to action recognition [4, 5, 6, 7]. The LSTM based-on RNNs can model the long-term contextual information of temporal sequences well. However, RNN-based model tends to emphasize the temporal information [8].

Considering the convolution neural networks (CNN) model is effective for classify images, increasing number of researchers use CNN to learn spatio-temporal features for skeleton se-

quences. Some approaches transform skeleton sequences into images, then they are fed into CNN model for action recognition [9, 10, 11]. [12] propose the spatial temporal graph convolutional networks (ST-GCN) for human action recognition by extending graph neural networks to a spatial-temporal graph model. [13] present a co-occurence feature learning framework based on CNN model. The co-occurence features are learned gradually from point-level features to global features. However, the above research learn global features containing limited local information of skeleton sequences. Moreover, quite a few human actions have characteristic frequency, such as shaking hands, clapping, but these typical methods ignore periodic patterns in the frequency domain.

To overcome the limitations and extract more discriminative information for skeleton-based action sequence, we propose a novel method, as shown in Figure 1), which uses hybrid features to recognize human actions. The hybrid features consist of CNN features and LSTM features. We design the CNN model based on the two-stream framework [14], which contains the raw skeleton position and the temporal difference. After convolution operation for each stream, we aggregate the two outputs to combined feature map. Then the LSTM model is employed to get the long-term temporal features. Since each type of features describes slightly different aspects of the sequence, we fuse various features to acquire more discriminative expression of action sequences. Finally, we achieve the hybrid features by confusing the CNN features and LSTM features, and perform action classification using hybrid features.

The major contributions of this work include:

- We proposed a multi-stream framework that integrates CNN-based features and LSTM-based features. It demonstrates that the hybrid features are more efficient in representation to receive improved performance.
- The attention mechanism is introduced to the CNN model to reallocate the feature maps by computing associations among elements. The improved CNN model can effectively learn more discriminative features.

2 Related work

2.1 Action recognition methods

Early approaches focus on the hand-crafted features to represent the human body for recognizing human action. [15] employs the relative positions of the joints to characterize position feature, motion feature, and overall dynamics feature. Principal Component Analysis (PCA) is applied to obtain EigenJoints representation. [15] proposes an actionlet ensemble model. The pairwise relative positions of each joint with other joints are computed to represent the position features, and Fourier Temporal Pyramid (FTP) is used to represent the temporal dynamics. [16] uses the rotations and translations between various body parts to represent geometric relationships, and the human action sequence is modeled as a curve in the Lie group. However, hand-crafted features can barely effectively represent spatio-temporal information of action sequences.

With the successful development of deep learning based methods in image recognition and Natural Language Processing (NLP), more and more literatures learn skeleton representations by adopting deep learning methods and achieve improved performance. There are mainly three categories: CNN-based methods, RNN-based methods and GNN-based methods.

Recurrent Neural Networks (RNNs) and Long-Short Term Memory (LSTM) networks are used to model temporal information of skeleton sequences [4, 5, 6, 7, 17]. [4] divides the skeleton joints into five sets corresponding to five body parts. Then, the five sets are fed into five LSTMs. [5] proposes a spatio-temporal LSTM framework to model the dynamics and dependency relations in both temporal and spatial domains. CNN-based methods represents the skeleton sequence as a pseudo-image, and then feed it into a CNN to recognize the action class just like image classification [9, 10, 11, 13]. In [11], the skeleton sequences are represented as three gray-scale images encoded from raw data. [13] proposes an end-to-end convolutional co-occurrence features learning framework, which uses CNN to learn point-level features for each joint and then aggregate these features from all joints to obtain co-occurrence features hierar-

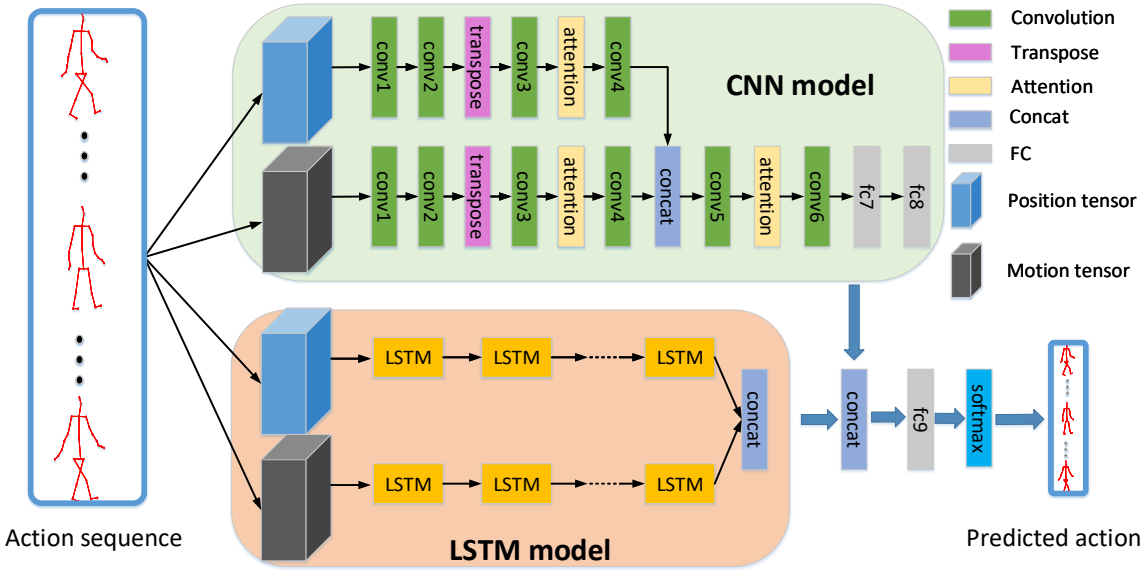

Figure 1: The framework of our proposed method.

chically. The GNN-based methods model the skeleton as a graph with joints and bones as vertexes and edges separately [12, 18, 19, 20].

2.2 Attention mechanism

The attention mechanism in Deep Learning can ignore irrelevant information and focus on important information. In action recognition field, researchers have utilized attention to learn more abundant information feature by focusing on a few joints, parts and frames. To select discriminative spatial information, the attention mechanism is employed to focus on key joints [21]. [17] proposes a spatio-temporal attention model to allocate different attention weights to joints and frames. [22] adopts a residual frequency attention block in the frequency domain to focus on discriminative patterns.

3 Proposed method

3.1 CNN-based features learning

The CNN model used in our frame work can be described as Fiture 1. The raw human skeleton data is a sequence of frames. Each of frames contains a set of joint 3D coordinates. Besides the joint coordinate location, the differences of joints contain temporal movements in a sequence. Given a skeleton sequence, we calculate the Euclidean distance between same joints in adjacent frames. The skeleton sequence in frame t can be represented:

$$\mathbf{S}^t = \{\mathbf{J}_1^t, ..., \mathbf{J}_N^t\}, \tag{1}$$

where N is the number of joint and $\mathbf{J} = (x, y, z)$. The skeleton motion is defined as:

$$\mathbf{M}^t = \mathbf{S}^{t+1} - \mathbf{S}^t. \tag{2}$$

We feed the joints coordinates $\mathbf{S}$ and the skeleton motion $\mathbf{M}$ into the network simultaneously (Figure 1).

We represent a skeleton sequence $\mathbf{X}$ as a $T \times N \times C$ tensor, where C, T, N denote the coordinate dimension, the number of frames, and the number of joints. As illustrated in Figure 1, we feed two types of input into the CNN model. Given a skeleton sequence tensor $\mathbf{X}$, we obtain a skeleton motion tensor $\mathbf{M}$ using Equation 2. We design two branches to accept the skeleton sequence $\mathbf{X}$ and skeleton motion $\mathbf{M}$. Both two branches have the same architecture and different parameters. We fuse two feature maps by concatenation along the channels.

The convolution operation enables interaction between channels of feature map, where features are aggregated from all input channels. Accordingly, we introduce transpose operation into the network. Given a feature maps $\mathbf{X}$ with shape

(C, T, N), we transpose it and the feature maps get new shape (C, N, T), so the frames is moved to channels. As illustrated in Figure 1, we adopt 1×1 and 3×1 kernels in conv1 and conv2, respectively. The kernel size of other convolutions is equal to 3×3. Before conv4 and conv5, we attach two attention layers, which reallocate weights of feature maps. The input and output feature maps have in common shape by the attention layer. After that, the feature maps can contain more temporal information under subsequent convolution layers and fully connected layers. Then the CNN-based features are obtained.

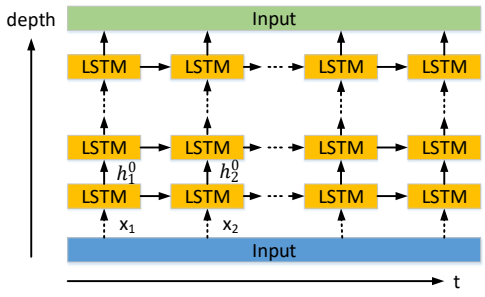

Figure 2: The LSTM module.

3.2 LSTM-based features learning

Generally, Long Short-Term Memory (LSTM) networks have superior performance in NLP field, such as speech recognition, text categorization, machine translation, etc. Considering LSTM can model the long-term contextual information of temporal sequences well, we adopt the two-layer LSTM model to obtain LSTM-based features. For each element in the input sequence, each layer computes the following function:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ii}\mathbf{x}_t + \mathbf{b}_{ii} + \mathbf{W}_{hi}\mathbf{h}_{(t-1)} + \mathbf{b}_{hi}), \quad (3)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{if}\mathbf{x}_t + \mathbf{b}_{if} + \mathbf{W}_{hf}\mathbf{h}_{(t-1)} + \mathbf{b}_{hf}), \quad (4)$$

$$\mathbf{c}_t = \mathbf{f}_t * \mathbf{c}_{(t-1)} + \mathbf{i}_t * tanh(\mathbf{W}_{cx}\mathbf{x} + \mathbf{W}_{ch}\mathbf{h}_{t-1} + \mathbf{b}_c), \quad (5)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{(t-1)} + \mathbf{W}_{oc}\mathbf{c}_{t-1} + \mathbf{b}_o), \quad (6)$$

$$\mathbf{h}_t = \mathbf{o}tanh(\mathbf{c}_t), \quad (7)$$

where $\sigma(\cdot)$ is the sigmoid function, and $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$, $\mathbf{o}_t$ are the input, forget, cell, and output gates respectively and $*$ indicates element-wise product. The structure of LSTM unit is shown in Figure 3.

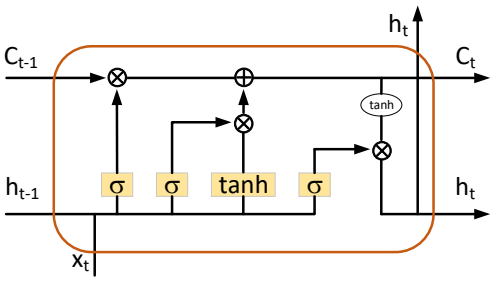

Figure 3: The structure of LSTM unit.

We process two types tensor: the skeleton sequence $\mathbf{X}$ and the skeleton motion $\mathbf{M}$ as inputs in Section 3.1. Consequently, they are fed into two LSTM networks branches shared the same architecture. As shown in Figure 1, through the LSTM networks, we obtain two LSTM-based features, which are concatenated for next stage. As illustrated in Figure 2, x_t is the input of LSTM. There are the configurations of LSTM model for different datasets mentioned in Section 4.2.

3.3 Attention module

In this work, the attention module is employed in CNN model to learning spatial features. We are inspired by self attention proposed in [23], which imported the self-attention mechanism into Generative Adversarial Networks (GAN) framework. The framework generates high-quality images for this reason the self attention module is effective in modeling long-range dependencies.

We utilize the self attention module to our framework for skeleton based human action recognition. Given an action sequence, the significance of joints and frames are diverse. To obtain discriminative representation, we reassign weights of feature maps by building associations among elements with the attention tighter mechanism. Consequently, the learned features contain dependencies between global features.

As illustrated in Figure 4, given a skeleton action sequence, we obtain the feature maps which

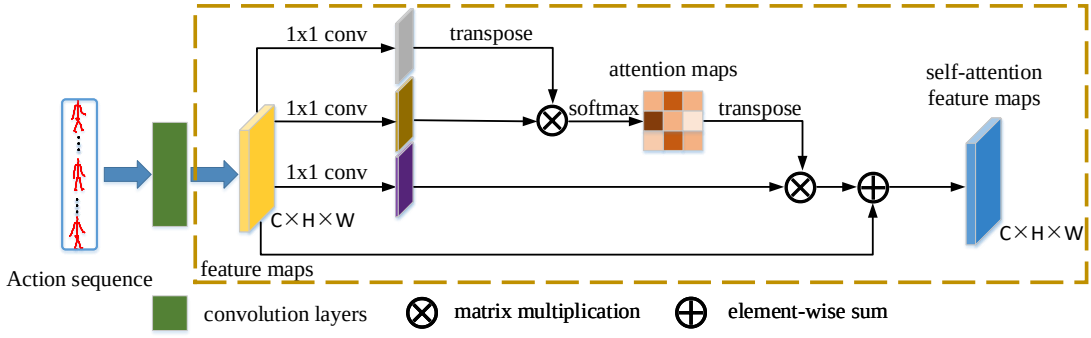

Figure 4: The self attention module.

are the output of convolution layers. Then we will obtain self-attention feature maps with the same shape ($C \times H \times W$) by the attention module. We feed the input feature maps into a convolution layers and generate three new feature maps query, key and value respectively. Then we perform matrix multiplication between query and key with reshape and transpose operations. A softmax layer is employed to obtain the attention maps. Afterwards, we perform another matrix multiplication between value and attention maps to obtain attention feature maps. Finally, we update the original feature maps by element-wise sum operation and obtain the self attention feature maps.

3.4 Proposed network architecture

As shown in Figure 1, the proposed architecture consists of multiple modules. CNN-based feature and LSTM features are obtained through CNN and LSTM model respectively. Then we concatenate them to receive the hybrid features. Afterwards, we use one fully connected layer to represent a skeleton sequence. A softmax layer is added at the end for class prediction. For the recognition task, a softmax function is used to normalize the output of network. The architecture has different structures according to the type of the process such as training and testing.

In the training process, there are three models needed trained, such as CNN, LSTM and the hybrid model. The traing procedure is described in Algorithm 1. Firstly, the submodels are trained with a softmax function as loss function separately. After the models have converged, the softmax layers are discarded into two submodels. We train the entire architecture with the trained parameters in CNN and LSTM models. As the same as submodels, the entire architecture use a softmax function as loss funnction. The probability that a skeleton sequence $\mathbf{X}$ belongs to the $\mathbf{i}^{th}$ class is

$$P(C_i|\mathbf{X}) = \frac{e^{z_i}}{\Sigma_{j=1}^{C} e^{z_i}}, i = 1, 2, ..., C, \quad (8)$$

where $\mathbf{o} = (o_1, o_2, ..., O_c)^T$ is the output of the network, C is the number of classes.

Algorithm 1 The proposed architecture for training.

Input: The skeleton action sequence dataset.
Output: The trained model.

1: Initializing the network parameters.
2: Preprocessing the input data to position tensor $\mathbf{X}$ and motion tensor $\mathbf{M}$.
3: Training the CNN model with $\mathbf{X}$ and $\mathbf{M}$.
4: Training the LSTM model with $\mathbf{X}$ and $\mathbf{M}$.
5: Extracting the CNN features and LSTM features.
6: Training the entire model.
7: Looping until convergence or reach given epochs.
8: Return the trained model of our architecture.

4 Experiments and evaluations

We evaluate the proposed method on two large-scale datasets, i.e. the NTU RGB+ D [7] and the Skeleton-Kinetics [12]. Both of these datasets have been widely used in previous work for

Table 1: Configuration of NTU RGB+D dataset.

Joint	Label	Joint	Label
1	base of the spine	14	left knee
2	middle of the spine	15	left ankle
3	neck	16	left foot
4	head	17	right hip
5	left shoulder	18	right knee
6	left elbow	19	right ankle
7	left wrist	20	right foot
8	left hand	21	spine
9	right shoulder	22	tip of the left hand
10	right elbow	23	left thumb
11	right wrist	24	tip of the right hand
12	right hand	25	right thumb
13	left hip		

skeleton-based action recognition. We work on the two data sets to validate the approach and make a comparison with the state of the art methods.

4.1 Datasets

4.1.1 NTU RGB+D

NTU RGB+D is currently the most widely used skeleton-based action recognition dataset. It contains 56000 skeleton action sequences, each annotated an action. There are 60 classes including single-actor action, e.g., jumping up and two-actor action, e.g., handshaking. We follow the benchmark evaluations in the original paper [7], i.e. Cross-Subject (CS) and Cross-View (CV). In the cross-subject evaluation, the training set contains 40,230 sequences, and validation set contains 16,560 sequences. Each frame contains 25 joints and shown in Figure 5 (left). The corresponding labels of the joints are in Table 1. In the cross-view evaluation, the training set contains 37,920 sequences, and the validation set contains 18,960 sequences. Top-1 accuracy is reported on both the two benchmarks.

4.1.2 Skeleton-Kinetics

The Skeleton-Kinetics is based on Kinetics human action dataset [24] without skeleton data collected from YouTube. There are 400 classes actions in the dataset. The Skeleton-Kinetics [12] are extracted employing the open source toolbox OpenPose [25]. As shown in Figure 5

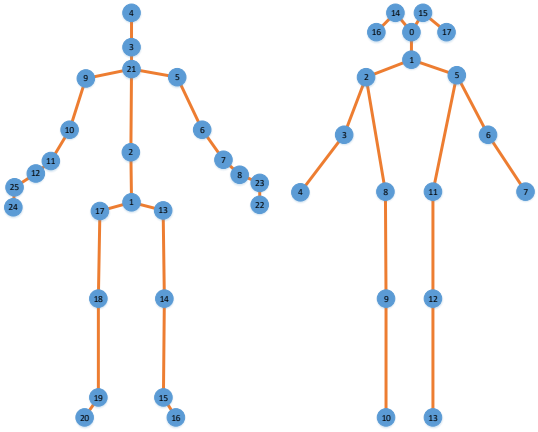

Figure 5: Illustration of the human skeleton graphs in two datasets. The left sketch shows 25 joints of the NTU-RGBD dataset and the figure on the right is Skeleton Kinetics dataset with 18 joints.

(right), the toolbox can estimate 18 joints for each person with 2D coordinates (X, Y) and confidence score C. We represent each joint with a tuple of (X, Y, C). In one frame, only the top-2 persons are selected by the joint confidence. The released data pad every clip to 300 frames. A skeleton sequence with T frames can be represented as a tensor with dimension of $(18 \times 3 \times T)$. Both Top-1 and Top-5 classification accuracies are reported as the recommendation. The dataset provides a training set of 240,000 clips and a validation set of 20,000.

4.2 Implementation details

Our framework is implemented on the Pytorch [26] and trained with the same bath size (32), training epochs (150). The Adam [27] is applied as the optimization algorithm for the network. For the NTU RGB+D dataset, if there are two persons in the sequences, we choose the person with higher value as the main subject. The skeleton sequences are normalized to a fixed length (64) using bilinear interpolation. The learning rate is initialized to 0.0001 and exponentially decayed every 1K steps with a rate of 0.99. In the LSTM model, the number of inputs x is 64. The number of hidden units of LSTM is 200. For the Skeleton-Kinetics, the LSTM has 300 inputs and 200 hidden units.

Table 2: Ablation study on the NTU RGB+D dataset.

Methods	CS(%)	CV(%)
P-LSTM [7]	62.9	70.3
HCN [13]	86.5	91.1
Ours(CNN)	85.6	90.2
Ours(LSTM))	65.2	73.6
Ours(CNN-ATT)	87.1	92.3
Ours(CNN-ATT-LSTM)	**88.0**	**94.5**

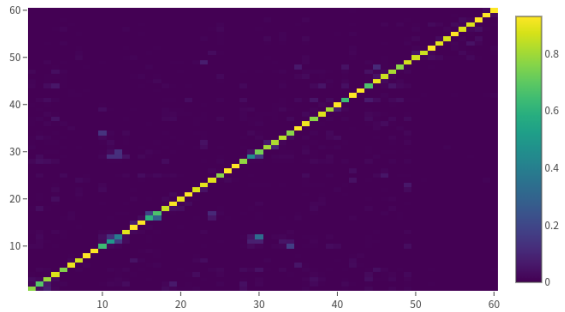

Figure 6: Confusion matrix comparison on the NTU RGB+D dataset.

4.3 Ablation study

In this section, we examine the effectiveness of the proposed hybrid features learning architecture on NTU RGB+D dataset with the benchmark of cross-subject and cross-view. The results of our method are reported in Table 2. First two rows show the accuracies of previous approaches, which are CNN-based method and LSTM-based method, respectively. Here we compare with four baselines, which utilize different types of features to recognize action sequences as follows.

- **CNN**: main CNN network without attention designs.
- **LSTM**: main LSTM network.
- **CNN-ATT**: CNN model with attention module.
- **CNN-ATT-LSTM**: the framework of CNN model with attention and LSTM.

First, we evaluate our CNN method and LSTM method. Table 2 shows validity of baselines with CNN and LSTM models. At the next step, we evaluate CNN-ATT model. Compared to CNN model showed in the third row, CNN-ATT improve the accuracy 1.5% and 2.1% on cross-subject and cross-view, respectively. At the last step, we evaluate the proposed hybrid model (CNN-ATT-LSTM). As shown in Table 2, the hybrid model achieves the highest accuracies 88.0% in cross-subject, and 94.5% in cross-view evaluations. The confusion matrix on the NTU RGB+D dataset is shown in Figure 6.

Table 3: Performance comparison on the NTU RGB+D dataset. CS and CV mean the cross-subject and cross-view respectively.

Methods	CS(%)	CV(%)
Joint [28]	60.2	65.2
P-LSTM [7]	62.9	70.3
HCN [13]	86.5	91.1
VA-LSTM [29]	79.2	87.7
ST-GCN [12]	81.5	88.3
2S-AGCN [30]	88.5	95.1
GCN-NAS [19]	89.4	95.7
DGNN [18]	89.9	96.1
Ours	88.0	94.5

4.4 Comparisons and discussion

To evaluate the performance of our method, we compare it with other skeleton-based action recognition approaches. The compared methods include hand-crafted methods [28], CNN-based methods [13], LSTM-based methods [7, 29, 31] and GCN-based methods [12, 30, 19, 18] on NTU RGB+D and Skeleton-Kinetics datasets. Table 3 and Table 4 show the results on these two datasets respectively. The performance of deep learning based methods is better than hand-crafted based methods. As illustrated in Table 3, our method outperforms hand-crafted based, CNN based and LSTM based methods. The result is similar on Skeleton Kinetics dataset (Table 4). Nevertheless, compared to the GCN-based methods, the results of our model is worse on NTU RGB+D and Skeleton Kinetics datasets. In the GCN-based methods, the human skeleton action sequence is represented as spatial temporal graph instead of pseudo-image. Intuitively, the human skeleton is more like a graph, which

Table 4: Performance comparison on the Skeleton-Kinetics dataset.

Methods	Top-1(%)	Top-5(%)
P-LSTM [7]	16.4	35.3
ST-GCN [12]	30.7	52.8
AS-GCN [20]	34.8	56.5
2S-AGCN [30]	35.1	57.1
DGNN [18]	36.9	59.6
GCN-NAS [19]	37.1	60.1
Ours	30.2	52.4

joints are represented as vertices and bones are represented as edges. These works have combined the joint information and bone information together for skeleton-based action recognition. The spatial and temporal features are obtained simultaneously in these methods.

In our work, we merely consider the information of joints without bones. In spite of the spatial-temporal features we obtain, the features have less information than the features the GCN-based methods obtain. As a result, we report lower recognition accuracies on two datasets, shown as Table 3 and Table 4. Nevertheless, from comparison results, our method is an effective strategy for recognizing human action based on skeleton. The key contribution of our method is coupling different types of features. The model learns discriminative hybrid features with two submodels. In the CNN model, we employ attentional module to reweighs the convolution feature maps. The new feature maps ignore irrelevant information. Moreover, we obtain the temporal features by LSTM model. Then we acquire hybrid features by concatenating two features. The experiment results demonstrate that the hybrid features are high-efficiency representation of skeleton sequence with abundant spatial and temporal information.

5 Conclusion

In this work, we present a hybrid features learning framework for skeleton-based action recognition. The hybrid features consist of CNN-based and LSTM-based features learning by CNN model and LSTM model. Specifically, the skeleton sequence $\mathbf{X}$ and the skeleton motion $\mathbf{M}$ are fed into CNN networks and LSTM networks simultaneously. In addition, we introduce attention module to the CNN layers. Afterwards, we obtain the hybrid features by concatenating two features. We evaluate our method on two large-scale datasets: NTU RGB+D and Skeleton Kinetics. From the experimental results we illustrate that the hybrid features learning framework is an effective strategy. In the future, we will continue combine different types of features and consider the information of bones in skeleton sequences simultaneously. In addition, exploration is recommended into how to extend human action to other scenes, such as predicting people's emotion by coupling human action and facial expression.

References

[1] Mohamed E Hussein, Marwan Torki, Mohammad Abdelaziz Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *In Twenty-Third International Joint Conference on Artificial Intelligence*, volume 13, pages 2466–2472, 2013.

[2] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. View invariant human action recognition using histograms of 3d joints. In *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–27, 2012.

[3] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297, 2012.

[4] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, 2015.

[5] Jun Liu, Amir Shahroudy, Dong Xu, Alex C Kot, and Gang Wang. Skeleton-based action recognition using spatio-

temporal lstm network with trust gates. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3007–3021, 2018.

[6] Hongsong Wang and Liang Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 499–508, 2017.

[7] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016.

[8] Pichao Wang, Wanqing Li, Chuankun Li, and Yonghong Hou. Action recognition based on joint trajectory maps with convolutional neural networks. *Knowledge-Based Systems*, 158:43–53, 2018.

[9] Yong Du, Yun Fu, and Liang Wang. Skeleton based action recognition with convolutional neural network. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 579–583, 2015.

[10] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[11] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 4570–4579, 2017.

[12] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *In Thirty-second AAAI Conference on Artificial Intelligence*, 2018.

[13] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 786–792, 2018.

[14] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[15] Xiaodong Yang and YingLi Tian. Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, 25(1):2–11, 2014.

[16] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *In IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014.

[17] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *In Thirty-first AAAI Conference on Artificial Intelligence*, pages 4263–4270, 2017.

[18] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019.

[19] Wei Peng, Xiaopeng Hong, Haoyu Chen, and Guoying Zhao. Learning graph convolutional network for skeleton-based human action recognition by neural searching. *In Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[20] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 3595–3603, 2019.

[21] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An at-

tention enhanced graph convolutional lstm network for skeleton-based action recognition. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019.

[22] Guyue Hu, Bo Cui, and Shan Yu. Skeleton-based action recognition with synchronous local and non-local spatio-temporal learning and frequency attention. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1216–1221, 2019.

[23] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363, 2019.

[24] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[25] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[28] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 5344–5352, 2015.

[29] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *In IEEE International Conference on Computer Vision*, pages 2117–2126, 2017.

[30] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019.

[31] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *In European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.