Hindawi
Complexity
Volume 2020, Article ID 8717942, 10 pages
https://doi.org/10.1155/2020/8717942

Research Article

Research on Discriminative Skeleton-Based Action Recognition in Spatiotemporal Fusion and Human-Robot Interaction

Qiubo Zhong,[1,2] **Caiming Zheng**,[1] **and Haoxiang Zhang**[1]

[1]*Robotics Institute, Ningbo University of Technology, Ningbo 315211, China*
[2]*State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710054, China*

Correspondence should be addressed to Qiubo Zhong; zhongqiubo@nbut.edu.cn

Received 18 June 2020; Revised 25 July 2020; Accepted 29 July 2020; Published 25 August 2020

Guest Editor: Hang Su

A novel posture motion-based spatiotemporal fused graph convolutional network (PM-STGCN) is presented for skeleton-based action recognition. Existing methods on skeleton-based action recognition focus on independently calculating the joint information in single frame and motion information of joints between adjacent frames from the human body skeleton structure and then combine the classification results. However, that does not take into consideration of the complicated temporal and spatial relationship of the human body action sequence, so they are not very efficient in distinguishing similar actions. In this work, we enhance the ability of distinguishing similar actions by focusing on spatiotemporal fusion and adaptive feature extraction for high discrimination information. Firstly, the local posture motion-based attention (LPM-TAM) module is proposed for the purpose of suppressing the skeleton sequence data with a low amount of motion in the temporal domain, and the representation of motion posture features is concentrated. Besides, the local posture motion-based channel attention module (LPM-CAM) is introduced to make use of the strongly discriminative representation between different action classes of similarity. Finally, the posture motion-based spatiotemporal fusion (PM-STF) module is constructed which fuses the spatiotemporal skeleton data by filtering out the low-information sequence and enhances the posture motion features adaptively with high discrimination. Extensive experiments have been conducted, and the results demonstrate that the proposed model is superior to the commonly used action recognition methods. The designed human-robot interaction system based on action recognition has competitive performance compared with the speech interaction system.

1. Introduction

With the development of artificial intelligence technology, human-robot interaction technology has become a research hotspot. Compared with speech and image signals, vision-based human-robot interaction technology is more stable, and it attracts a lot of research interest. The key to human-centered visual interaction technology is to understand human activities [1] and human social behaviors [2]. Therefore, action recognition plays an important role in the field of human-robot interaction [3]. The two main approaches of human action recognition are RGB-based and skeleton-based. The RGB-based method makes full use of the image data and can obtain higher performance in the recognition rate. However, this method usually needs to process every pixel in the image to extract features. Therefore, high-cost computing resources are required and real-time processing can hardly be achieved. It is also vulnerable to poor lighting conditions and background noise. In the skeleton sequence method, the 2D or 3D coordinates are expressed as human joint positions. Due to the limited number of joints in the human skeleton, only a few dozen, some modest computing resources would be enough for real-time applications. It is also robust to dynamic environments and complex backgrounds. Many widely available devices are suitable for extracting human skeleton features, such as Microsoft Kinect, OpenPose [4], and CPN [5].

The conventional deep learning-based methods convert the skeleton sequence as a set of joint vector sequences, input them to RNNs [6], or extract features by feeding 2D

pseudoimages representing skeleton sequences into CNNs [7], and then predict the action classes. However, neither the joint vector sequence nor the 2D pseudoimage can represent the correlation between human joints effectively. Recently, graph convolutional neural networks (GCNs) have extended the convolution operation from 2D image structure to graph structure and have shown good performance in many applications. Yan et al. [8] used GCNs for the first time in skeleton-based action recognition and proposed a spatial-temporal graph model. Subsequently, the methods for optimizing spatial feature extraction were proposed. Yang et al. [9] presented a finite-time convergence adaptive fuzzy control method for a dual-arm robot with an unknown number of kinematics and dynamics. Shi et al. [10] used adaptive graph convolutional layer and attention mechanism to increase the flexibility of the model, first-order joint information, second-order bone information, and motion information as inputs to construct multistream networks. Liu et al. [11] proposed multiscale aggregation across spatial and temporal dimensions effectively to eliminate the importance of neighbor nodes for long-range modeling. Yang et al. proposed a personalized variable gain control with tremor attenuation for robot teleoperation [12] and used adaptive parameter estimation and control design for robot manipulators with finite-time convergence [13]. Peng et al. [14] used high-order representations of skeleton adjacency graphs and dynamic graph modeling mechanisms to find implicit joint correlations. Obinata and Yamamto [15] modeled the spatiotemporal graph by adding extra edges on the interframe to extract the relevant features of the human joints. However, all these methods ignore the fusion of posture motion and skeleton joint features in the temporal domain.

In the existing research work, the spatial information and motion information of the spatiotemporal graph are not fused to achieve end-to-end training effectively. The proposed novel posture motion-based spatiotemporal graph convolution networks (PM-STFGCNs) use the posture motion-based spatiotemporal fusion (PM-STF) module to perform feature fusion of motion and skeleton representation in the spatiotemporal domain for enhancing skeleton features adaptively. The defined local posture motion-based attention module (LPM-TAM) is used to constrain the disturbance information in the temporal domain and learn the representation of motion posture. The introduced local posture motion-based channel attention module (LPM-CAM) is employed to learn the strong discrimination representation between similar action classes in order to enhance the ability to distinguish fuzzy action. Extensive experiments have been performed on two large-scale skeleton datasets. Compared with common methods, the proposed method can further improve the recognition performance which combines with the method of optimizing the spatial graph convolution only. In addition, a human action recognition interactive system was designed to compare with speech interaction.

The main contributions of our methods are the following:

(1) A novel local posture motion-based attention module (LPM-TAM) filters out low motion information in the temporal domain that helps to improve the ability of relevance motion feature extraction

(2) Local posture motion-based channel attention module (LPM-CAM) is employed to enhance the ability to distinguish similar actions for learning the strong discriminative representation adaptively between different action classes

(3) The posture motion-based spatiotemporal fusion (PM-STF) module is used, which integrates LPM-TAM and LPM-CAM to effectively fuse the spatiotemporal feature information and extract high-discriminative feature for improving the ability to distinguish similar actions

(4) The effectiveness of the proposed method has been verified through extensive experiments, compared with other common methods to evaluate the competitiveness of the proposed method and applied in humanoid robots successfully to verify that action interaction is better than speech interaction

2. Related Work

2.1. Spatial Graph Convolution Networks. The spatial-temporal graph convolutional neural network [8] represents the connection relationship of the joints with the self-connected identity matrix I and the adjacency matrix A^S. In the case of a single frame, the convolution operation of the spatial dimension is performed as follows:

$$f_{\text{out}} = \sum_{k=1}^{K} \left(\Lambda_k^{-1/2} A_k^S \Lambda_k^{-1/2} \otimes A_S\right) f_{\text{in}} M_k, \tag{1}$$

where $f_{\text{in}} \in \mathbb{R}^{C_{\text{in}} \times T \times N}$ is the feature map with input dimension of (C_{in}, T, N) tensor, N is the number of joints, A_k^S is the $N \times N$ adjacency-like matrix, $A_k^{S^{i,j}} = 1$ denotes the vertex v_i in the subset of the vertex v_j, and $\Lambda_k^{ii} = \sum_j (A_k^{S^{ij}}) + \lambda$ is the normalized diagonal matrix, where $\lambda = 0.001$. K represents the numbers of different subsets in spatial dimension based on spatial distance partition strategies. There are three different subsets, namely, $K = 3$. A_0^S represents the connection of the vertex itself, A_1^S represents the connection of the centripetal subset, and A_2^S represents the connection of the centrifugal subset. $M_k \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times 1 \times 1}$ is 1×1 convolution weight. A_S is a spatial attention feature map of the $N \times N$ dimension, which denotes the importance of each joint. $\otimes$ is the multiplication of the corresponding elements of the matrix, which means that it can only affect the vertices connected to the current target.

2.2. Temporal Graph Convolutional Networks. The literature [10] proposed a temporal attention module, and the attention coefficient is calculated as follows:

$$\mathbf{A}_T = \sigma\left(\theta\left(\text{AvgPool}\left(\mathbf{f}_{\text{in}}\right)\right)\right), \tag{2}$$

where $\mathbf{f}_{\text{in}} \in \mathbb{R}^{C_{\text{in}} \times T \times N}$ is an input feature map. AvgPool is an average pooling operation. θ is a 1×1 convolution operation, and the weight matrix $\mathbf{M}_\theta \in \mathbb{R}^{1 \times C_{\text{in}} \times S}$, where S is the size of the convolution kernel. σ refers to the Sigmoid activation function. The attention feature map $\mathbf{A}_T \in \mathbb{R}^{1 \times T \times 1}$, which denotes the importance of the skeleton graph at a temporal dimension, and T refers to the length in time.

The literature [8] defined the temporal graph convolution based on a simple strategy. In equation (1), they use the kernel size $\Theta \times 1$ in the temporal dimension to perform graph convolution. Therefore, the sampling area on the vertex v_{ti} is $S_a(v_{ti}) = \{v_{qi} \mid |q - t| \le |\Theta/2|\}$, where Θ is the kernel size in temporal dimension, which is set to 9 in [8].

3. Posture Motion-Based Spatiotemporal Fusion Graph Convolution

3.1. Posture Motion Representation. The posture motion represents the motion information of the corresponding joint in a series of consecutive frames, for example, the j_{th} joint of the given frame $u+1$, i.e., $v_{u+1,j} = (x_{u+1,j}, y_{u+1,j}, z_{u+1,j})$ and the j_{th} joint of frame u, i.e., $v_{uj} = (x_{uj}, y_{uj}, z_{uj})$, which posture motion is represented as $\tau_{u,j} = (x_{u+1,j} - x_{u,j}, y_{u+1,j} - y_{u,j}, z_{u+1,j} - z_{u,j})$. $\tau_{u,j}$ is the posture motion representation of the j_{th} joint of frame u.

3.2. Local Posture Motion-Based Temporal Attention Module. A novel local posture motion-based temporal attention module (LMP-TAM) is proposed for suppressing a large amount of disturbance information in the temporal dimension. As shown in Figure 1, the posture motion feature map of each vertex in the spatiotemporal graph is calculated as follows:

$$\mathbf{\Omega} = \sum_{t=1}^{T} \theta_t\left(\varepsilon_m\left(\mathbf{f}_{\text{in}}^{t}, \mathbf{f}_{\text{in}}^{t-1}\right)\right), \tag{3}$$

where $\mathbf{f}_{\text{in}}^{t} \in \mathbb{R}^{C_{\text{in}} \times 1 \times N}$ is the input feature map at time t. $\theta_t \in \mathbb{R}^{C_{\text{in}}/2 \times C_{\text{in}} \times 1 \times 1}$ is the convolution weight matrix of 1×1. ε_m extracted the posture motion representation from the input feature map. $\mathbf{\Omega} \in \mathbb{R}^{C_{\text{in}}/2 \times T \times N}$ is the posture motion feature map, where the channel of the feature map is half of the input channel.

Human motion is body movements, which involve part or all of the limbs. The attention map Φ_T of skeleton sequence in the spatiotemporal graph is represented by the attention Φ_L of local limbs in the temporal dimension. The importance of local limb in temporal dimension is determined by motion information in the local perception domain D. $D = \{d_0, d_1, d_2, d_3, d_4\}$, where d_0 denotes left hand, d_1 denotes right hand, d_2 denotes left leg, d_3 denotes right leg, and d_4 denotes other limb parts. $\Phi_L \in \mathbb{R}^{1 \times T \times \eta}$, where η refers to the number of limbs being denoted, and η has been set 5 in this work. The temporal attention of local posture motion-based is calculated as follows:

$$\mathbf{\Phi}_T = \sigma(\text{AvgPool}(\text{MaxPool}(\Gamma \otimes \mathbf{\Omega}))), \tag{4}$$

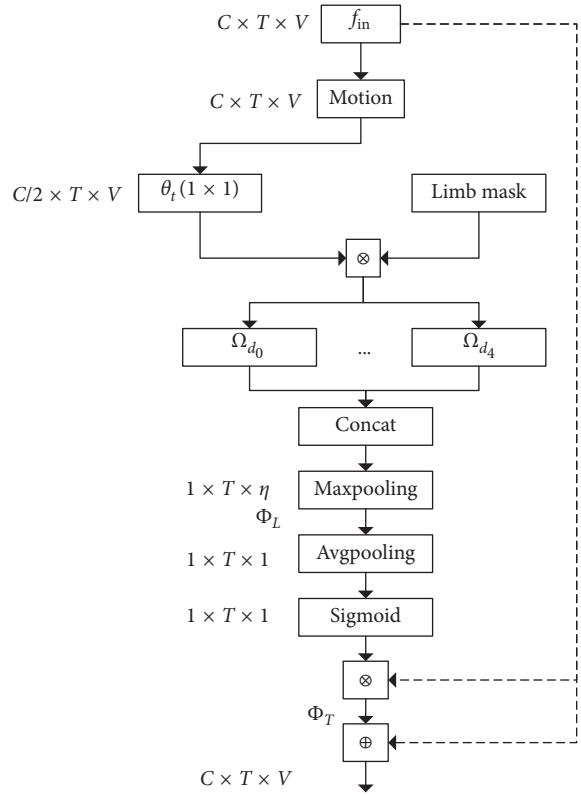

Figure 1: Local posture motion-based temporal attention (LMP-TAM) module.

where $\Phi_T \in \mathbb{R}^{1 \times T \times 1}$ refers to the importance of each frame of the spatiotemporal graph with a time length of T. $\mathbf{\Omega}$ is the motion feature map of joints, and $\Gamma \in \mathbb{R}^{1 \times 1 \times V}$ is the local limb mask set D. $\Gamma \otimes \mathbf{\Omega}$ is the attention Φ_L based on local limbs in the temporal dimension. $\otimes$ is the multiplication of the corresponding elements of the matrices. σ is the sigmoid activation function. The final output is as follows:

$$\mathbf{f}_{\text{out}} = \mathbf{f}_{\text{in}} \otimes \mathbf{\Phi}_T \otimes \mathbf{f}_{\text{in}}. \tag{5}$$

The input feature map is multiplied by the attention feature map Φ_T in a residual manner to calculate adaptive feature enhancement, and $\otimes$ refers to the addition of corresponding matrix elements.

3.3. Local Posture Motion-Based Channel Attention Module. The local posture motion-based channel attention module (LPM-CAM) has been proposed to improve the ability to learn the strong discrimination representation between different postures. As shown in Figure 2, the input includes the posture motion feature map $\mathbf{\Omega} \in \mathbb{R}^{C \times T \times V}$ extracted based on the local posture motion-based temporal attention module and generated temporal attention, which multiplied of each other to obtain the spatial-temporal graph after attention. The temporal sequence action segment with rich action semantic information is paid more attention. The channel attention coefficient is calculated as follows:

$$\mathbf{\Phi}_C = \sigma\left(\text{AvgPool}\left(\text{MaxPol}\left(\rho\left(\Gamma \otimes \left(\mathbf{\Omega} \otimes \mathbf{\Phi}_T\right)\right)\right)\right)\right). \tag{6}$$

The motion feature map after attention is denoted as $\mathbf{\Omega} \otimes \Phi_T$, which is decomposed into several local limbs in the local perception field to represent local posture movement. The action sequence with important semantic on the spatial-temporal graph has been screened out by the temporal attention module, and the channel attention selects the strong discriminative representations between different posture movements for action recognition. ρ is marked as ReLu nonlinear activation function. Concat refers to concatenate the local limb feature map:

$$\mathbf{f}_{\text{out}} = \mathbf{f}_{\text{in}} \otimes \mathbf{\Phi}_C \otimes \mathbf{f}_{\text{in}}. \tag{7}$$

The input feature map is multiplied by the channel attention feature map in way of residual connection to achieve adaptive feature enhancement.

3.4. Posture Motion-Based Spatiotemporal Fusion. In order to fuse skeleton joints information and motion features to achieve an end-to-end learning manner, the posture motion-based spatiotemporal fusion module (PM-STF) is proposed to fuse spatial and temporal features and enhance the discriminative feature adaptively. The output of temporal convolution module at the i_{th} vertex of frame u is

$$f_{\text{out}}(v_i) = f_{\text{in}}(v_i) + \sum_{v_j \in S_a(v_i)} \frac{1}{Z_{ui}(v_j)} \tau(v_j) \cdot \delta\big(\gamma_j(v_j)\big). \tag{8}$$

This is different from formula (1), and the input is a posture motion feature map extracted from the spatiotemporal graph and adopts a residual connection to enhance the motion feature. $\tau(v_j)$ is the posture motion feature of the neighborhood vertex v_j. δ is the weighting function. $\gamma_j(v_j)$ refers to the mapping label of the subset of the neighborhood vertex v_j, which is divided into three subsets S'_{a0}, S'_{a1}, and S'_{a2} based on the spatial distance partition strategy.

To implement the PM-STF, equation (8) is transformed into

$$\mathbf{f}_{\text{out}} = \mathbf{f}_{\text{in}} \oplus \sum_{k=1}^{K} \big(\mathbf{\Lambda}_k^{-1/2} \mathbf{A}_k^S \mathbf{\Lambda}_k^{-1/2} \otimes \mathbf{A}_S\big)(\mathbf{M}_{\mathbf{m}}\mathbf{\Omega})\mathbf{M}_k, \tag{9}$$

where $\mathbf{\Omega} \in \mathbb{R}^{C_{\text{in}}/2 \times T \times N}$ is posture motion feature map and $\mathbf{M}_m \in \mathbb{R}^{C_{\text{in}} \times C_{\text{in}}/2 \times 1 \times 1}$ is a 1×1 convolution weight matrix, increasing the channel of the same posture motion feature map as input channel. $\mathrm{M}_k \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times 1 \times 1}$ is a 1×1 convolution weight vector. $\mathbf{A}_S \in \mathbb{R}^{1 \times T \times N}$ is a spatial attention map which is used to distinguish the importance of vertices. $\otimes$ refers to the multiplication of the corresponding elements of matrices. $\mathbf{A}_k^S$ is adjacency-like matrix, and

$$\mathbf{A}_k^{S^{i,j}} = \begin{cases} 1, & \text{if vertex } v_j \text{ in the subset } S'_{ak} \text{ of vertex } v_i, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

3.5. Implementation of PM-STFGCN. The implementation of our module is combined with the model of optimizing the spatial graph convolution only, such as ST-GCN and 2s-AGCN. Taking ST-GCN as an example, shown in Figure 3, the implementation of our module PMSTF-GCN is added between S-GCN and T-GCN. Each layer of PMSTF-GCN contains LPM-TAM, LPM-CAM, and PM-STF. S-GCN and T-GCN are named as the spatial graph convolution layer and temporal graph convolution layer of the original model. GAP is a global average pooling layer, and FCN is marked as a fully connected network layer. Finally, a spatiotemporal fusion graph convolution block is constructed. The overall architecture of the network consists of several STFGCN blocks. The batch normalization layer is added to the skeleton data input to normalize the input data. Finally, the global average pooling layer is implemented to pool the feature graphs to the same size, and the followed layer is a SoftMax classifier to obtain the prediction.

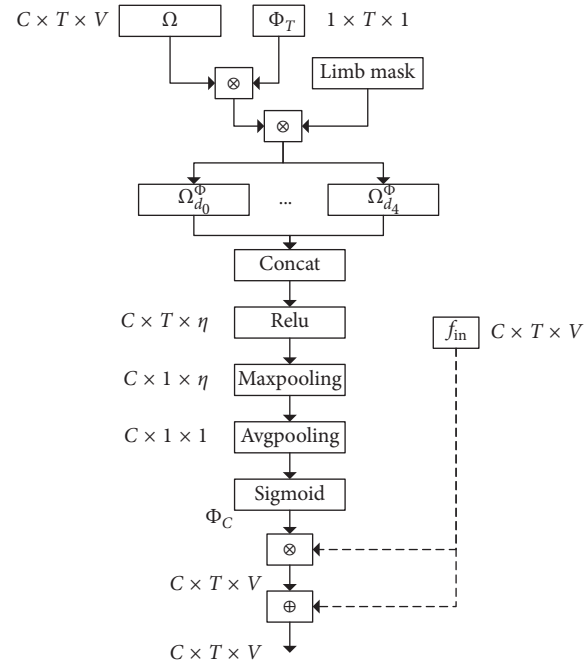

Figure 2: Local posture motion-based channel attention module (LMP-CAM).

3.6. Implementation in Human-Robot Interaction. The presented action recognition schemes were applied on a real system, which consists of a Pepper robot and an external Kinect v2 depth camera. The implementation in human-robot interaction was performed as follows (Algorithm 1).

4. Experiments

4.1. Datasets

4.1.1. NTU-RGB + D. NTU-RGB + D [16] is the largest and most widely used multimodality dataset for skeleton-based action recognition. Each action segment was performed by 40 volunteers aged 10 to 35 and captured by three camera sensors at the same height but from different horizontal

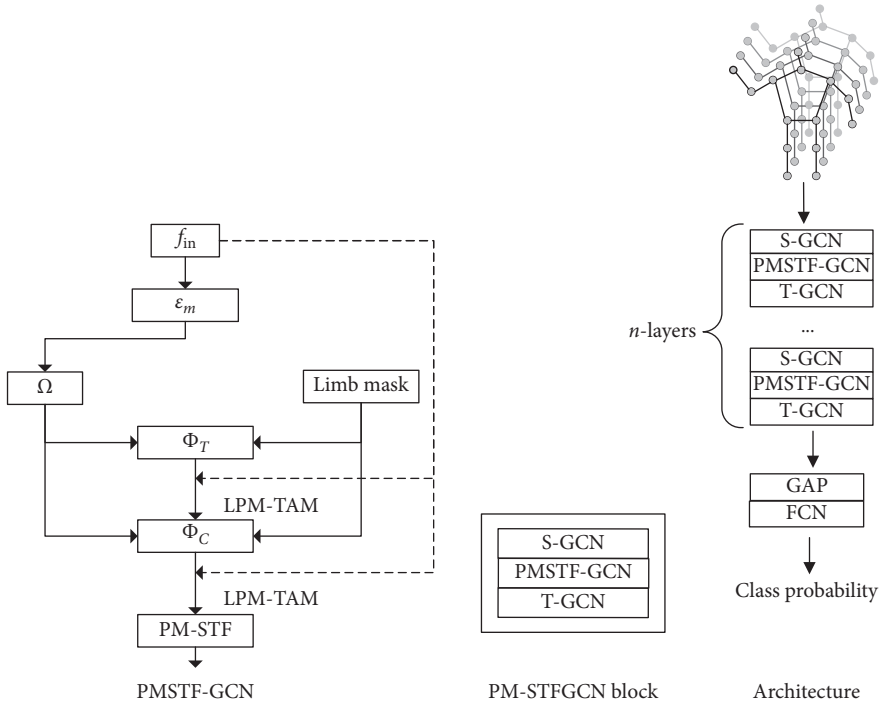

Figure 3: ST-GCN based spatiotemporal fusion graph convolution neural networks.

(1) Initialize action recognition network f
(2) Initialize threshold $\alpha = 0.75$
(3) **while**
(4) obtain the skeleton data series D from Kinect v2
(5) $C_{class}, P_{probability} \longleftarrow \arg\max_{probability}(f(D))$
(6) **if** $P_{probability} \geq \alpha$
(7) digital instruction $I \longleftarrow C_{class}$
(8) pass the instruction I to Pepper robot
(9) **end**

Algorithm 1: Human-robot action interaction.

angles: −45°, 0°, and 45°. The human skeleton has 25 joint points, and the number of skeletons in each video is no more than 2. It contains 60 action classes and 56880 video clips. There are two kinds of training benchmarks, and [16] recommends as follows: cross-subject (CS) and cross-view (CV). In cross-subject (CS) benchmarks, the training dataset contains 40320 action samples, and the testing dataset contains 16560 action samples. In cross-view (CV) benchmarks, the training dataset contains 37920 action samples taken by camera sensors 2 and 3, and the testing dataset contains 18960 action samples taken by camera sensors 1. In the following experiments, we test the top-1 accuracy on two benchmark datasets.

4.1.2. Kinetics-Skeleton. Kinetics-Skeleton [17] is a large dataset for skeleton-based action recognition. Kinetic contains 300000 action video clips, and a total of 400 classes [8] used the publicly available OpenPose toolbox [4] to estimate the pose of 18 joints in each fragment frame. There are a total of 300 frames for each action video frame. According to the average joint confidence, two people are selected as multi-person clips in each frame. The training dataset contains 240000 video clips, and the testing dataset contains 20000 video clips. We make use of the training dataset and then perform experiments to verify the accuracy of top-1 and top-5 on the testing dataset.

4.2. Ablation Study. The effectiveness of the proposed method has been verified over two large skeleton datasets in Kinetics-Skeleton and NTU-RGB + D. The local posture motion-based attention module (LPM-TAM), local posture motion-based channel module (LPM-TAM), and posture motion-based spatiotemporal fusion (PM-STF) module are represented by PM-STFGCN. Two sets of comparisons are

made between ST-GCN [8] and ST-GCN + PM-STFGCN, and between 2s-AGCN [18] and 2s-AGCN + PM-STFGCN. The results show that the performance has been improved over the original models and verified the effectiveness of LPM-TAM, LPM-CAM, and PM-STF.

As shown in Tables 1 and 2 for ST-GCN [8] and ST-GCN + PM-STFGCN, PM-STFGCN improves the top-1 accuracy of the CS and CV benchmarks by 4.2% and 1.6%, respectively, and the accuracy of the top-1 and top-5 of the Kinetics-Skeleton dataset by 2.5% and 1.9%, respectively. For 2s-AGCN [18] and 2s-AGCN + PM-STFGCN, PM-STFGCN improves the top-1 accuracy of CS and CV benchmarks by 3.3% and 1.3%, respectively, and the accuracy of the top-1 and top-5 of the Kinetics-Skeleton dataset has been improved by 1.4% and 2.0%, respectively. 2s-AGCN + PM-STFG performed best on NTU-RGB + D and Kinetics-Skeleton datasets.

4.2.1. Attention Module. Experiments were also performed to verify the local posture motion-based temporal attention module (LPM-TAM) and channel attention module (LPM-CAM). In the ST-GCN [8] and 2s-AGCN [18] networks, only the LPM-TAM or LPM-CAM is added to the convolutional layer of the spatial-temporal graph. The results are shown in Tables 1 and 2. Compared with ST-GCN, the LPM-TAM module improves the top-1 accuracy of the CS and CV benchmarks by 2.6% and 0.6%, respectively, and the accuracy of top-1 and top-5 of Kinetics-Skeleton by 1.0% and 0.9%, respectively. The LPM-CAM module improved the top-1 accuracy of the CS and CV benchmarks by 3.0% and 0.7%, respectively, and the accuracy of top-1 and top-5 of Kinetics-Skeleton by 1.4% and 1.1%, respectively. Compared with 2s-AGCN, the LPM-TAM module improves the top-1 accuracy of the CS and CV benchmarks by 1.7% and 0.5%, respectively, and the accuracy of the top-1 and top-5 of Kinetics-Skeleton by 0.4% and 0.9, respectively. The LPM-CAM module improves the top-1 accuracy of the CS and CV benchmarks by 2.1% and 0.7%, respectively, and the accuracy of the top-1 and top-5 of Kinetics-Skeleton by 0.5% and 1.1%, respectively. The temporal and channel attention module improved the recognition performance than the original model which verifies the effectiveness of the feasibility of the attention modules.

4.2.2. Spatiotemporal Fusion Module. Experiments were also carried out on the posture motion-based spatiotemporal fusion (PM-STF) module. In the ST-GCN [8] and 2s-AGCN [18] networks, only the PM-STF is added to the convolutional layer of the spatial-temporal graph. The results are shown in Tables 1 and 2. Compared with ST-GCN, the PM-STF module improved the top-1 accuracy of CS and CV benchmarks by 3.2% and 0.9%, respectively, and the accuracy of top-1 and top-5 of Kinetics-Skeleton by 1.6% and 1.2%, respectively. Compared with 2s-AGCN, the PM-STF module improves the top-1 accuracy of the CS and CV benchmarks by 2.5% and 0.7%, respectively, and the accuracy of the top-1 and top-5 of the dataset Kinetics-Skeleton by 0.8% and 1.3%, respectively. Compared with the original model, the spatiotemporal fusion module has a greater contribution to the improvement of the recognition performance which verifies the effectiveness and necessity of spatiotemporal fusion.

Table 1: Ablation study on the benchmark of NTU-RGB + D.

Methods	CS (%)	CV (%)
ST-GCN [8]	81.5	88.3
2s-AGCN [18]	88.6	95.2
ST-GCN + LPM-TAM (ours)	84.1	88.9
2s-AGCN + LPM-TAM (ours)	90.3	95.7
ST-GCN + LPM-CAM (ours)	84.5	89.0
2s-AGCN + LPM-CAM (ours)	90.7	95.9
ST-GCN + PM-STF (ours)	84.7	89.2
2s-AGCN + PM-STF (ours)	91.1	95.9
ST-GCN + PM-STFGCN (ours)	85.7	89.9
2s-AGCN + PM-STFGCN (ours)	91.9	96.5

Table 2: Ablation study on the skeleton-based dataset Kinetics-Skeleton.

Methods	Top-1 (%)	Top-5 (%)
ST-GCN [8]	32.5	54.9
2s-AGCN [18]	36.7	59.8
ST-GCN + LPM-TAM (ours)	33.5	55.8
2s-AGCN + LPM-TAM (ours)	37.1	60.7
ST-GCN + LPM-CAM (ours)	33.9	56.0
2s-AGCN + LPM-CAM (ours)	37.2	60.9
ST-GCN + PM-STF (ours)	34.1	56.1
2s-AGCN + PM-STF (ours)	37.5	61.1
ST-GCN + PM-STFGCN (ours)	35.0	56.8
2s-AGCN + PM-STFGCN (ours)	38.1	61.8

4.3. Comparison with State-of-the-Art Schemes. The proposed method is compared with some of the state-of-the-art schemes, and the results are shown in Tables 3 and 4. Among them, 2s-AGCN + PM-STFGCN achieved very good performance on CS and CV. On the Kinetics-Skeleton dataset, the accuracy of top-1 and top-5 of 2s-AGCN + PM-STFGCN also showed decent performance.

4.4. Human-Robot Interaction Demonstration. To further evaluate the robustness of the proposed action recognition schemes to distinguish similar action classes, action recognition is applied to a real system that consists of a Pepper robot and an external Kinect v2. As shown in Table 5, there is a correspondence between action semantics and interactive action.

The designed correspondence between action semantic and interactive activities ranges from partial and limb movements of hands to whole-body movements with more complexity. For example, waving the hand, touching the ear, holding the head with hands, and applauding are all hand movements. Among them, the hand movements of the first three movements are related to the head with high similarity. Also, squatting, sitting

Table 3: Comparison of CS and CV benchmarks with state-of-the-art schemes.

Methods	CS (%)	CV (%)
STA-LSTM [6]	73.4	81.2
ST-GCN [8]	81.5	88.3
CNN-based [7]	83.2	89.3
GCN-NAS [14]	89.4	95.7
MS-AAGCN [10]	90.0	96.2
2s-AGCN + PM-STFGCN (ours)	91.9	96.5

Table 4: Comparison with Kinetics-Skeleton dataset with state-of-the-art schemes.

Methods	Top-1 (%)	Top-5 (%)
ST-GCN [8]	30.7	52.8
GCN-NAS [14]	37.1	60.1
MS-AAGCN [10]	37.4	60.6
2s-AGCN + PM-STFGCN (ours)	38.1	61.8

Table 5: Correspondence between action semantic and interactive action.

Action semantics	Interactive action
Wave the hand	Raise the hand
Touch the ear	Nod the head
Hold the head	Look left
Applaud	Look right
Squat	Sit down
Sit down	Wake up
Jump up	Stand up

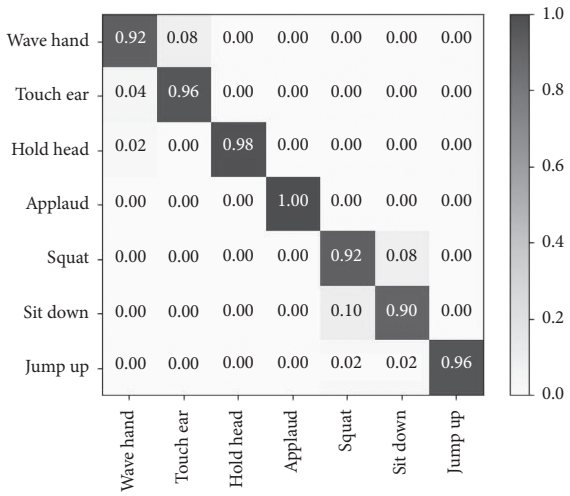

Figure 4: The confusion matrix of action interaction with the Pepper robot.

down, and jumping involve movements of the whole body with high similarity.

As shown in Figure 4, the measurement results are obtained in this work after conducting 50 experiments. Among them, each similar action has a high recognition accuracy which means our method can effectively distinguish each different action. Each action sequence can be seen as a combination of many steps. For example, waving the hand can be divided into two steps: first, raise your right hand to above the head; second, swing the hand around the head. Similarly, a video can be decomposed into multiple frames of images.

4.4.1. Strong Discrimination Analysis. As shown in Figures 5–8, there are examples of human-robot interaction with similar actions. A period of time action sequence has been calculated, and the classification result with the highest probability is selected as the recognized result. Skeleton sequence with low motion information can be filtered out well by LPM-TAM, which helps to identify the process from raising hands to the head and swinging, and more purposefully recognize interactive actions. The main action of touching the ear is the process of raising the hand to the ear. Compared with waving the hand, the main difference is the movement of the hand swinging near the head. The characteristics of strong discrimination have been paid more attention by LPM-CAM to constraint similar movement processes, such as the process of raising the hand which serves as the basis for action recognition. The action of holding the head with both hands is similar to touching the ears. However, the main difference is that holding the head with both hands is the movement of the left and right hands, while touching the ears is the movement of the limbs with one hand. The main difference between similar movements in the local limb area can be captured by LPM-CAM effectively that enables the proposed method to extract stronger and discrimination representation. The human-robot interaction experiments verified that similar action did not affect the recognition result at all and has a strong discrimination of similar actions.

4.4.2. Comparison with Speech Interaction. In this work, the two indicators of accuracy and real-time performance are compared with speech interaction. The accurate times of these interaction methods were recorded 50 times to verify the reliability of the action interaction. Figure 9 shows the confusion matrix of the Pepper robot speech interaction recognition. In the testing phase, it only needs to speak out the corresponding action, such as wave the hand or touch the ear. The recognition result is regarded as "jumping" if the result of speech interaction has not been recognized within the specified test time. The recognition result of speech interaction is easily affected by external noise and distance which cause recognition errors, or no recognition results. From the experimental results, the average recognition rate of action interaction and speech interaction is 95.7% and 94.8%, respectively. Compared with speech interaction, our scheme has highly competitive which verifies the reliability of the action interaction in the recognition effect.

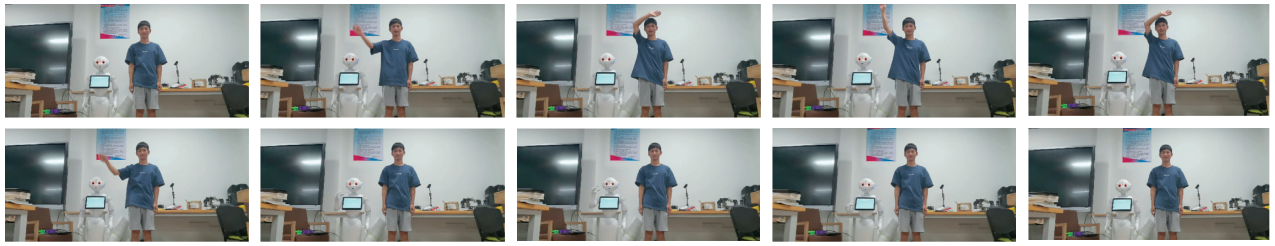

Figure 5: Action interaction with waving the hand as an example.

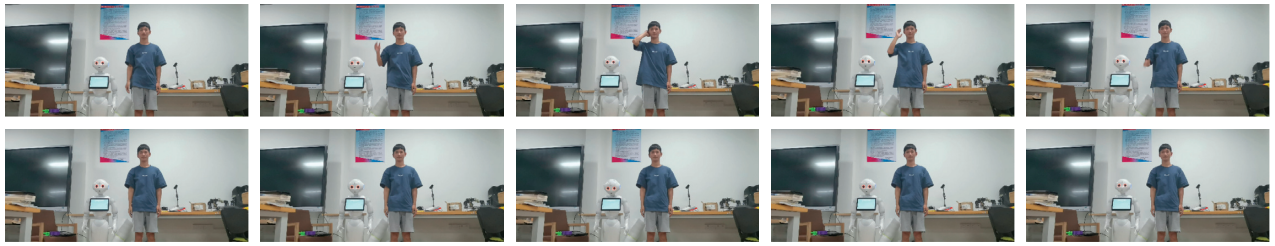

Figure 6: Action interaction with touching the ear as an example.

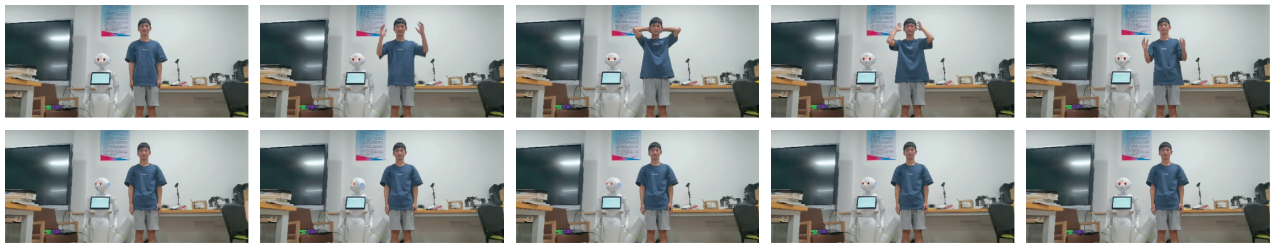

Figure 7: Action interaction with holding hands as an example.

Figure 8: Action interaction with applause as an example.

4.4.3. Comparison of Response Time. As shown in Figure 10, comparison with the response time of speech and action interaction shows the average time of the 10 test results. Due to the different durations of each action, using the same time segment as inputs will cause fluctuations of response time. We try to do a few more experiments to eliminate the differences among the action response time. The results show that the response time of action interaction is shorter than speech interaction because of the robustness to external environment noise. The average response time of speech and action interaction is 2.05 s and 1.86 s, respectively. Compared with speech interaction, the proposed scheme reduced the responding time by 0.19 s in real-time. The main reason is that video frames within a certain time range are used for recognition and shorter processing time for the action recognition network.

In conclusion, through the experimental comparison of two human-robot interaction ways, the action recognition has its advantages: it is not affected by environmental noise or spatial distance; it provides better real-time response during the interaction.

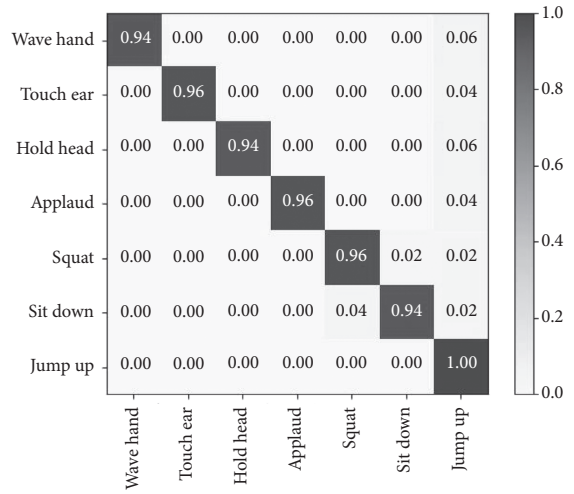

Figure 9: The confusion matrix of speech interaction with the Pepper robot.

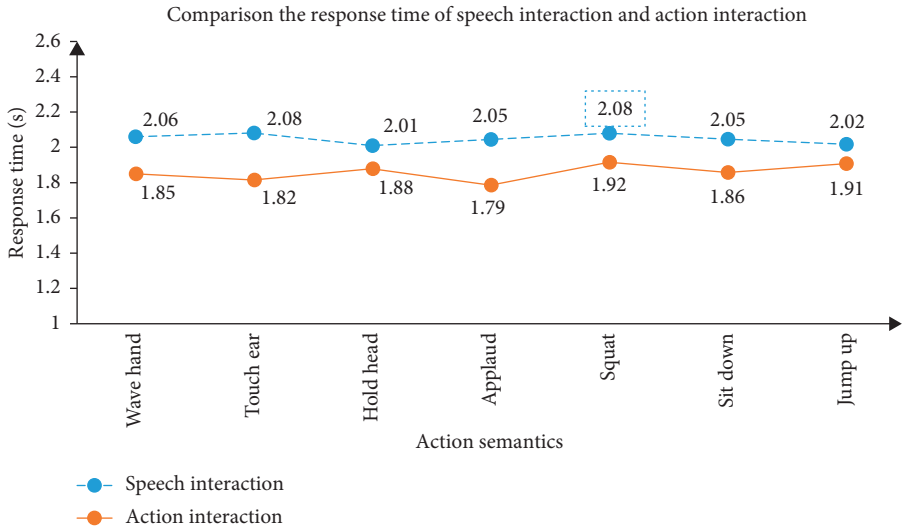

Figure 10: Comparison of the response time of speech and action interaction.

5. Conclusion

Previous works in the literature mostly make use of modeling of motion information and skeleton joint information independently, which cannot fully express the relationship between them. The posture motion-based spatiotemporal fusion graph convolution network (PM-STFGCN) is presented to fuse temporal and spatial features and enhance the posture motion features adaptively with high discrimination. A novel local posture motion-based temporal attention (LPM-TAM) module is introduced to suppress the disturbance information with low motion in the temporal domain efficiently and fully learn the representation of the posture motion. The local posture motion-based channel attention module (LPM-CAM) is proposed for the purpose of learning strong discrimination representation between different motion postures which improved the ability to discriminate action classes, and the posture motion-based spatio-temporal fusion module (PM-STF) is adopted to fuse the motion feature and skeleton representation effectively. Extensive experiments were performed on two large skeleton datasets, and the constructed scheme shows substantial improvement over some other methods. The proposed action recognition interaction system has a competitive performance in accuracy and response time compared with speech interaction.

Data Availability

The data used to support the findings of this study are included with the supplementary information files.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This material is based upon work funded by the State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University Foundation of China, under grant no. sklms2019011, "13th Five-Year Plan" Talent Training Project of Higher Education in Zhejiang Province under grant no. jg20190487, and Research Project of Educational Science Planning in Zhejiang Province under grant no. 2020SCG090.

References

[1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," 2014.

[2] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese, "Social scene understanding: end-to-end multi-person action localization and collective activity recognition," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[3] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3551–3558, Sydney, Australia, December 2013.

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[5] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[6] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3d human action recognition," *Computer Vision-ECCV 2016*, vol. 9907, pp. 816–833, 2016.

[7] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops ICMEW*, Hong Kong, China, July 2017.

[8] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," 2018, http://arxiv.org/abs/1801.07455v2.

[9] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, and C.-Y. Su, "Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 574–588, 2019.

[10] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," 2019, https://arxiv.org/abs/1912.06971.

[11] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions fo skeleton-based action recognition," 2020, https://arxiv.org/abs/2003.14111.

[12] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8112–8123, 2018.

[13] C. Yang, J. Luo, Y. Pan, Z. Liu, and C.-Y. Su, "Personalized variable gain control with tremor attenuation for robot teleoperation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 10, pp. 1759–1770, 2018.

[14] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 3, pp. 2669–2676, 2020.

[15] Y. Obinata and T. Yamamto, "Temporal extension module for skeleton-based action recognition," 2020, https://arxiv.org/abs/2003.08951.

[16] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB + D: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1010–1019, Las Vegas, NV, USA, June 2016.

[17] W. Kay, J. Carreira, K. Simonyan et al., "The kinetics human action video dataset," 2017, https://arxiv.org/abs/1705.06950.

[18] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019.