IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 August 2023, accepted 20 August 2023, date of publication 25 August 2023, date of current version 31 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3308685

RESEARCH ARTICLE

A Progressive Decoding Strategy for Action Recognition

XIANGTAO ZHAO[1,2], ZHIHAN LI[1,2], YINHUA LIU[1,2,3], AND YUXIAO XIA[1,2]
[1]School of Automation, Qingdao University, Qingdao 266071, China
[2]Institute of Future, Qingdao University, Qingdao 266071, China
[3]Shandong Key Laboratory of Industrial Control Technology, Qingdao 266071, China

Corresponding author: Yinhua Liu (liuyinhua@qdu.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2020YFB1313600.

ABSTRACT In recent years, significant progress has been made in modeling temporal sequences and spatial structures in skeleton-based human action recognition. However, existing methods rely on explicit modeling of the inherent structure of the human body, which may result in reduced joint saliency and poor interpretability due to the sparsity of skeleton data and the relative smoothness of convolutions. This paper proposes a feature matching method based on the progressive decoding strategy. As human movement is a chain process, the strategy progressively decodes human pose features from the center to the periphery, using multi-level graph filters to obtain multi-frequency hierarchical graph features. Then the adaptive convolution kernels are constructed to match the local similarities between graph features. Self-similarity of the query set and the mutual similarity between the query set and the support set samples are calculated to analyze the entire posture of the human body, and similar skeletal features are distinguished according to the similarity of the node spectrum to differentiate between different action categories. Through experimental verification of two public data sets, the proposed method has better recognition accuracy and generalization of small sample behavior. The experiments show that the proposed method outperforms the existing methods on NTU RGB + D and Human36M Dataset.

INDEX TERMS Human action recognition, few-shot, graph filtering, feature matching.

I. INTRODUCTION

With the development and application of computer vision, smart homes [1], intelligent security [2], and other application scenarios have been popularized, and the social security supervision system has been gradually improved. In the data of images and video, humans are the main body of events, and human behavior refers to the execution of certain movements in a continuous period to complete a given task. In image and video data, human beings are the main body of events. The analysis and understanding of human behavior are important. Human behavior recognition (HAR) is a computer vision technology that enables machines to understand, analyze and classify these behaviors into any given effective input [3], which has become a key field of computer vision research at home and abroad [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Huiyu Zhou.

Human behavior is usually carried out coherently. A behavior sequence contains multiple actions, which is difficult to accurately segment and label, and some behaviors themselves contain multiple motion sub-processes. The widely used skeleton model is between 15-30 joint points. The skeleton data can be considered as a sparse representation method of human action, which may result in the feature map extracted from the deep neural network facing challenges when dealing with fine-grained actions. It may not be able to fully complete the recognition task, such as removing headphones and removing glasses, and other similar movements. According to the supervised deep learning method, a large amount of data is needed for model training to improve accuracy. With the development of this behavior recognition field, more and more data sets of different categories have been proposed, and it is difficult to collect enough annotated training samples for each action category. Previous methods [5], [6] have achieved remarkable performance, but there

are some limitations in the robustness of the model. The fixed graph structure can only filter the features in the limited graph, focusing on the similarity of body joints under spatial constraints, without fully considering the differences between different body joints.

This paper proposes a progressive decoding strategy, including a progressive decoding module and a feature matching module. The decoding module decodes the data in the order of time sequence-global skeleton sequence-local joints and filters the local joints step by step according to the transfer law of the human motion chain to obtain multi-level features. Then the feature matching module calculates the similarity matrix by calculating the mutual similarity between the features of the query sample and the support sample and the self-similarity within the query set to distinguish the skeleton features of similar behaviors. The main contributions are as follows:

1) A progressive decoding strategy is proposed to decode human pose features step by step from center to edge, and multi-frequency hierarchical graph features are obtained by using a multi-level graph filter.

2) An adaptive convolution kernel is constructed to match graph features by calculating the self-similarity of the query set and the mutual similarity between the query set and the support set samples.

3) The experiments show that the proposed method outperforms the existing methods on NTU RGB + D and Human36M Dataset.

The content of this paper is organized as follows: Section II describes the action recognition and other methods related to the work. Section III proposes the proposed method in detail. Section IV carries the experiments and Section V gives the conclusion.

II. RELATED WORKS

A. HUMAN ACTION RECOGNITION

In the field of deep learning, skeleton-based methods have attracted a lot of attention, in which information is stored and transmitted in the form of graphs, and the connections between nodes are completely retained in the form of adjacency matrices. Yan et al. [7] first applied graph convolution in the field of action recognition, describing the original human skeleton structure and constructing the spatiotemporal graph convolutional network (ST-GCN) model. Shi et al. extended ST-GCN by proposing the dual-stream adaptive GCN [8] and Directed GCN [9], which respectively modeled first-order and second-order information of the skeleton data using a dual-stream framework, and encoded the skeleton graph in the directed form to further improve the accuracy. Shiraki et al. [10] increased non-physical connections between joints based on the theory of fluid dynamics, considering the dynamic importance of joints in each action. Liu et al. [11] investigated constrained iterative attacks on skeleton-based action recognition using GCN, perturbing the joint positions in action sequences to maintain the temporal and spatial integrity of the resulting adversarial sequence. Liu et al. [12] input spatiotemporal graphs and SkeleMotion images into ST-GCN and ResNeXt, respectively, and then fused the resulting representations, modeling the amplitude and direction of temporal information in the interaction and addressing the issue of isolated temporal information in ST-GCN. Finally, Cha et al. [13] explicitly learned in the context of human action recognition, sampling the maximum information skeleton representation from a reconstructed 3D mesh, while considering the internal and external structure relationships of the 3D mesh and the skeleton captured by the sensor. Zhang et al. [14] introduced a multi-view stereo reconstruction technique that integrates the sparse to dense method with pyramid attention. The sparse to dense approach involves estimating a set of sparse three-dimensional points from the input image and subsequently propagating them intensively to reconstruct a complete three-dimensional scene. This methodology is effective in handling occlusion and blurring in the input data, thereby improving the accuracy of 3D reconstruction results.

B. GRAPH STRUCTURE

Human bones can be naturally represented as a graph structure, whose joints and bones are represented by vertices and edges, respectively. In the processing of human natural mechanisms, the excessive smoothing graph convolution in ST-GCN makes the salient features of fine-grained behavior blurred or lost, which will be further amplified in the case of unbalanced samples. Some studies have proposed to transform the joint space of the human body to enhance different structural features and perform signal filtering or aggregate vertex information based on graph Laplacian feature decomposition [15]. Graph scattering transforms, by changing the bandwidth size [16], using different graph signal filters [17] to obtain the expected spectrum, increasing the richness of the graph. On this basis, the wavelet diffusion method [18], and parametric feature learner [19] have good performance. To enhance the remote interaction of graph structures between different joints, the scattering filter [20] used a GNN-based architecture where all joints are linked together for comprehensive exploration and achieved good results. In this paper, a multi-stage filter is designed to obtain the characteristics of different frequency bands in the graph information.

C. FEW-SHOT

When the number of samples is small or the categories are unbalanced, the Few-shot algorithm aims to enable the model to handle tasks with similar types, rather than only a single classification task. To obtain more representative sample features, Wang et al. [21] used self-training classifiers to predict the pseudo-labels of query set samples. In the measurement method, Zhang et al. [22] introduced a new distance measurement method DeepEMD to calculate the best matching between each block of the query set and the support set of the image to represent the similarity between them.

The two-stream model proposed by Careaga et al. [26] tested the effectiveness of different metric-based few-shot algorithms.

n terms of action recognition tasks, some researchers have proposed a variety of methods, such as Bishay et al. [23] proposed a matching mechanism to measure the depth of alignment representation at the video segment level. Perrett et al. [24] used the CrossTransformers attention mechanism to match frame tuples in time. Wang et al. [25] proposed a hybrid relation module, which uses bidirectional Hausdorff distance to align video frames. The method proposed by Memmesheimer et al. [27] is to represent the signal as an image, extract features from it, and then encode the features into vectors in the embedded space to measure the pairwise similarity. Liao et al. [28] used adaptive convolution to find unique features and perform similarity matching.

III. MATERIALS AND METHODS

As shown in Figure 1, the proposed method consists of two main parts. The progressive decoding module first explores the global and local time relations at multiple time resolutions by temporally decoding the skeletal sequence, and then further uses the hidden motion chain information to perform progressive expansion decoding from the center to the periphery according to the inherent connection properties of the human body structure to obtain better local feature expression. The feature matching module is used to effectively fuse different scale features and create a unified feature representation.

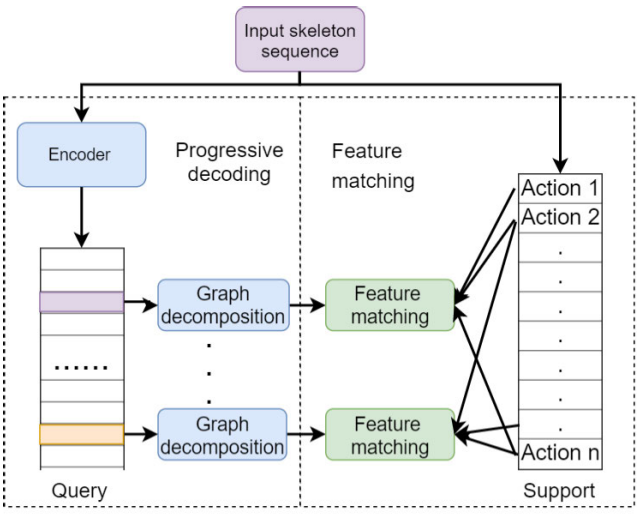

FIGURE 1. Overall framework.

A. BASE MODEL

The joint features of the human body determine the geometric structure of human motion, which can effectively represent the pose information. It aims to generate unobservable future pose sequences based on observable historical pose sequences. For a continuous human motion sequence S in a video, N frames of observable motion pose information is sampled as $S = \{X_i\}_{i=1}^{N}, X_n \in R^{K*3}$, where K is the number of joint points used to represent posture information. There are different values according to the demand K. Such as CMU's perceptual computing laboratory [29] constructed a model containing 18 key points for calculating and tracking the position and posture of the human body. The skeleton points collected by the H36m dataset [30] and the NTU RGB+D dataset [30] were 25.

The positions of all points can be described as a vector as,

$$X_i = [x_1, x_2, x_3, \ldots x_K]^T \in R^{K*3} \tag{1}$$

where X_i is continuous. Let the $x_i = \left[x_{ix}(t), x_{iy}(t), x_{iz}(t)\right]$ reflects the 3D joint positions. Then the displacement vector of the node is as follows,

$$x_i(t) = \left[x_{ix}(t), x_{iy}(t), x_{iz}(t)\right] \tag{2}$$

where $1 \leq t \leq T$ and $t \in Z$, and $x_i(t)$ is continue. Through calculating the derivative of the position vector concerning time, the speed and direction of each key point can be got as follows,

$$\dot{x}_i(t) = \left[\dot{x}_{ix}(t), \dot{x}_{iy}(t), \dot{x}_{iz}(t)\right] \tag{3}$$

where $\dot{x}_i(t)$ is continue. During the process of posture change, the limbs are moving with the connected joints, which can be represented by vectors between adjacent joints. The unit vector from x_j to x_i labeled v_{ij} as follows,

$$v_{ij}(t) = x_j(t) - x_i(t) \tag{4}$$

where i and j represent two adjacent joints, respectively. When the posture of the human body changes, the changes in the limbs labeled V are estimated by calculating the changes in the joint points as follow,

$$\dot{V}_{ij}(t) = \dot{x}_j(t) - \dot{x}_i(t) \tag{5}$$

where $\dot{V}_{ij}(t)$ is continue. The position and change trend of each joint key point and limb can be estimated according to the above equations. In different postures, the joints have different position features.

There is a natural connection between human skeleton joints. Let the $G = (E, V)$ to represent the human posture in a static frame, where V represents the set of limbs and represents the set of joint points. A binary adjacency matrix is set to represent the skeleton relationship as $V \in \{0, 1\}^{K*K}$, When the i-th and j-th individual joints are connected, set the $A_{ij} = 1$, otherwise $A_{ij} = 0$. In addition, when only the edge features are considered, the diagonal of the adjacency matrix is 0. The characteristics of its nodes are considered as $\tilde{A} = A + I$.

B. PROGRESSIVE GRAPH ATTENTION MODULE

Progressive decoding refers to the gradual acquisition of information from images in an incremental manner, starting from low resolution and progressing to high resolution. Similarly, the progressive decoding of skeletal keypoints is

proposed, proceeding from the trunk to the limbs, and iteratively extracting low-level to high-level pose features.

In posture or movement recognition tasks, there is a hidden movement chain between the key points of the skeleton that spreads from the torso to the limbs. For example, the torso dominates the shoulder, and then dominates the movement of the elbow and wrist, and the movement speed and displacement of the torso, shoulder, elbow and wrist show a significant increasing relationship, which indicates that during the movement process, The limbs of the body can usually show richer feature information than that of the trunk. Therefore, the key points of the skeleton are decomposed into multiple levels from the trunk to the limbs, and the layers progressively represent the information of the key points of the skeleton. In the process of decoding the key points of the skeleton, the features of each level are gradually extracted iteratively from the lower level to the higher level, and the features of each level contain the information of the previous level. Further down the hierarchy will contain more granular information.

The progressive decoding of skeletal keypoints offers the advantage of effectively capturing the structural information of human posture from a macro to micro perspective, thereby enabling more precise representation of movements or postures.

As depicted in Figure 2, assuming that all skeletal keypoints are present in the set D, the yellow point in step 1 is selected as the root node, and the yellow nodes at a distance of 1 from the root node in step 2 are assigned to level 1. Similarly, the yellow nodes at a distance of 2 from the root node in step 3 are assigned to level 2, and the yellow nodes at a distance of 3 from the root node in step 4 are assigned to level 3. During the process of progressive decoding, the state sequence is as follows:$\{n = 0, 1, \ldots N\}$, the root node serves as the initial input to the network, and the nodes in each level are subsequently fed into the network. The input sequence is as follows:

$$X_n = \left[X_n^{(1)}, X_n^{(2)}, X_n^{(3)}\right] \tag{6}$$

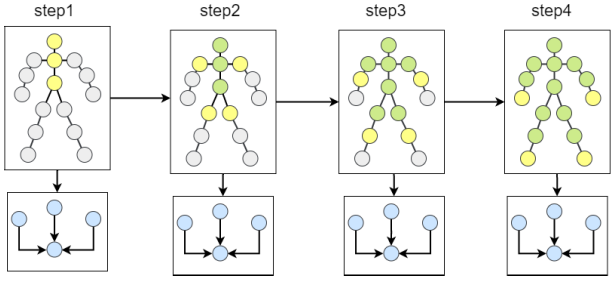

FIGURE 2. Graph progressive decomposition example.

Each decoding level is associated with a corresponding subgraph structure, which can be decomposed into a tree structure network with K channels. These tree nodes undergo graphical filtering to obtain channel characteristics, as illustrated in formula (10).

$$H_{n,k} = \sigma\left(f_{n,k}\left(\tilde{A}\right) X_n W_{n,k}\right) \tag{7}$$

where the $W_{n,k}$ is the training weight corresponding to the k-th filter, then the nonlinear function is used for the frequency representation of the scatter plot. The $f_{n,k}\left(\tilde{A}\right)$ is a parameter matrix based on $\tilde{A}$, which designed to represent the implicit interaction of nodes in the motion subgraph. According to the set number of filters, the given filtering formula is as follows:

$$f_{n,k}\left(\tilde{A}\right) = \begin{cases} \tilde{A} & k = 0 \\ I + \tilde{A} & k = 1 \\ \sum_{j=1}^{k} \tilde{A}^{2^{j-1}} & k = 2, \ldots K \end{cases} \tag{8}$$

Assuming that all edge weights are positive, only the low frequency is retained to enhance the smoothness, while other filters follow different band-pass characteristics in turn to enhance the diverse motion characteristics of other joints.

The structure of the progressive decoding module is illustrated in Figure 3. Initially, temporal encoding is conducted for the unsegmented long-term skeletal sequence, and the behavior recognition problem is formally defined as follows: For a sequence of length T, at any time t, a real behavior label $y_{t,c}$ denotes the action category, where c represents the corresponding action category. Consequently, based on different sampling intervals, each sequence can be divided into T/m non-overlapping fragments during the training period, with each fragment composed of m frames. These distinct sampling intervals encompass time information at various scales, and each segment contains feature information of different scales. As a result, multi-scale samples are provided as input to the decoding module. The decoding module consists of 21 layers, where the first 7 layers have 64 output channels, the middle 7 layers have 128 output channels, and the last 7 layers have 256 output channels, thereby obtaining a 256-dimensional feature vector for each sequence. Additionally, an attention layer is added to each graph filter to perform adaptive fusion of all channel information, and the fused information is then provided to the feature matching module. The weight based on adaptive computation measures the importance of each channel in the subgraph. Given a channel H_n, the fusion feature of the subgraph is expressed as follows,

$$\tilde{H}_n = \omega_k H_{n,k} \tag{9}$$

where the ω_k is the correlation fraction of the k-th channel, which is calculated through,

$$\omega_k = \frac{exp\left(RuLu\left(a\left(\mu H_{n,i} \parallel \mu H_{n,j}\right)\right)\right)}{\sum_{k \in K} exp\left(RuLu\left(a\left(\mu H_{n,i} \parallel \mu H_{n,k}\right)\right)\right)} \tag{10}$$

where the μ is the weigh, which participates in the correlation calculation of the two channels as a shared parameter, and $\parallel$ splices the two features. Then the $a()$ maps the spliced high-dimensional features to a real number and normalizes

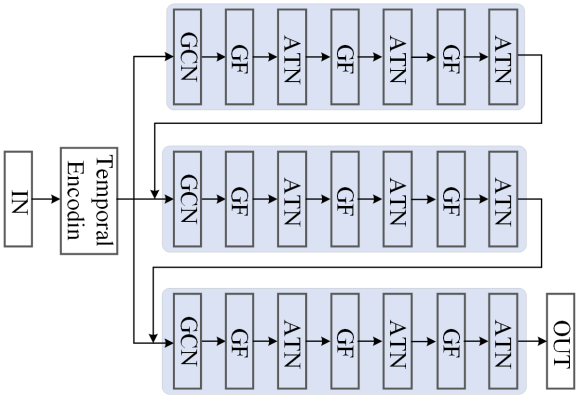

FIGURE 3. Progressive decoding network structure.

them to obtain weights. Note that each node sees itself as its neighbor.

C. FEATURE MATCHING MODULE

The module searches for the most differentiated local responses by matching local features at each position of the feature map. The channel features obtained from the progressive decoding module subgraph are converted into a set of convolution kernels, and another feature map is convoluted to obtain the response value of each position, and then the local part with the highest response is obtained by global pooling.

Firstly, the channel features produced by the image filtering in the progressive decoding module are concatenated, and subsequently, the skeleton sequence of the query set is transformed into a feature graph with dimensions of (M, c, h, w), where M represents the number of query samples, c represents the number of sample frames, h, and w represent the number of subgraphs and subchannels, respectively. For each sample in the query set, the 1*1 region at each position of the feature map is selected as the local convolution kernel. Subsequently, Mhw convolution kernels of size c*1*1 are formed to perform mutual convolution operations on the feature maps of the support set samples. The mathematical expression is as follows.

$$F_{mut} = pooling\left(conv\left(F_n, F_q\right)\right) \tag{11}$$

where F_{mut} represents the local features obtained by traversing the convolution operation, F_n represents the feature map generated by the support set samples, where $n \in [1, 2, \ldots, N]$ is the number of each feature map, F_q refers to the convolution kernel corresponding to each feature map, where $q \in [1, 2, \ldots, Mhw]$ indicates the number of convolutional kernels, and Polling refers to the global pooling operation. For the samples of the support set, the same operation is performed, and each different category in the support sample contains only one sample of data. The feature matching process is shown in Figure 4. The feature map goes through a normalization layer and a fully connected layer. The normalization layer converts the eigenvalues into [1, 0] intervals to alleviate the gradient disappearance problem and accelerate the training speed of the network.

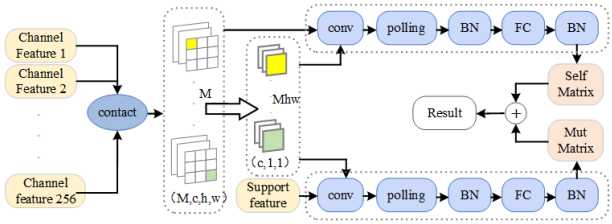

FIGURE 4. Feature matching module.

The fully connected layer is used to aggregate feature information. The SoftMax activation function outputs the category probability of each sample. The mathematical expression is as follows.

$$M_{mut} = softmax\left(BN\left(FC\left(BN\left(F_{mut}\right)\right)\right)\right) \tag{12}$$

where $M_{mut} \in R^{N*M}$ is the mutual similarity matrix, and each item represents the matching score between the samples in the query set and the samples in the support set. M and N refer to the batch size and the number of action categories, respectively.

In addition, the self-matching branch is set to learn the features of the samples in the query set, which is the same as the calculation method of the mutual matching branch. The difference is that the two feature maps of the convolution kernel and the convolution are both from the query set samples. The goal of the branch is to extract the same area of the query sample. The mathematical description of the self-matching branch is as follows.

$$F_{self} = pooling\left(conv\left(F_m, F_q\right)\right) \tag{13}$$

$$M_{self} = softmax\left(BN\left(FC\left(BN\left(F_{self}\right)\right)\right)\right) \tag{14}$$

where F_m represents the feature map generated by the query sample, $m \in [1, 2, \ldots, M]$,$M_{self} \in R^{M*M}$is the self-similar matrix of the query sample, and the elements represent the matching degree of each query sample. In the self-matching matrix, after repeated training and learning, the matching degree of the same category of samples increases, and the similarity between different categories of samples gradually decreases to distinguish visually similar skeleton data.

D. LOSS FUNCTION

According to the similarity matrix and the one-hot code label, the loss function of the model is defined. The position corresponding to the two samples of the same category is set to 1, and the other positions are set to 0. The higher the similarity between the two samples, the more likely the corresponding actions belong to the same category. The loss function is in the form of focus loss. By weighting the cross-entropy loss and increasing the attention to similar data in the training process, the mutual matching loss can be expressed as follows.

$$P_{nm} = \begin{cases} P_{nm} & y_{nm} = 1 \\ 1 - P_{nm} & else \end{cases} \tag{15}$$

$$L_m = -\frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{M}\left(1 - P_{nm}\right)^{\alpha}\log p_{nm} \tag{16}$$

where $P_{nm} \in [0, 1]$ represents the predicted value of the matrix output of N*M. The N and M refer to the number of samples in the support set and the query set, respectively. The self-matching loss is weighted in the same way.

$$L_s = -\frac{1}{M}\sum_{n=1}^{N}\sum_{m=1}^{M}(1 - P_{nm})^{\alpha}\log p_{nm} \tag{17}$$

where M is the number of samples in the query set, the is set to 2. the self-matching loss only participates in the training process, and the number of query samples in the test is not limited. The total loss is obtained by summarizing mutual matching and self-matching. The support set is used as the labeled sample, and its credibility is higher than the query set self-matching. Therefore, the weighted parameter is used to balance the importance of the two losses, so that the mutual matching loss is dominant.

$$loss = L_m + \omega L_s \tag{18}$$

The value of the weight ω is given in the experiment.

IV. EXPERIMENT

A. DATASETS

Two public datasets are collected: NTU RGB + D [31] and Human36M Dataset [30]. Each dataset has different scenarios and falls situations, and the dataset details are as follows, NTU RGB + D contains 3D skeletal information of 60 types of actions, a total of 56880 samples, of which 40 types are daily behavior actions. The Human36M dataset contains 30,000 human poses from 17 scenarios and was collected by 11 testers.

The model is implemented on the NVIDIA TITAN V GPU by PyTorch 1.4. The Adam is used to train the model, the learning rate is set to 0.0005, and the batch size is set to 16. There are 100 epochs used to train.

B. MODEL TEST EXPERIMENT

Figure 5 and Figure 6 display the confusion matrices for partial classification results obtained from the NTU-RGB+D 60 and NTU-RGB+D 120 datasets, respectively. The figures reveal that the proposed model performs better in classifying certain actions with distinct characteristics, such as ''falling down'',''punch'' and ''put on glasses''. Moreover, the model demonstrates effective recognition of visually similar behaviors, such as ''clapping'' and ''rub two hands,'' which indicates the efficacy of filtering enhancement and feature matching in distinguishing similar behaviors.

However, it still does not work well on some actions, such as ''writing'', ''typing'' and ''put on back''.There is still a degree of confusion. This is because the size of the convolution kernel obtained from the query features is fixed, the global dynamic features cannot be obtained, and the discriminant weights are inaccurate for the long term motion in the time series. This is because the current dataset focuses on short-term behavior, and there is not too much design model for long-term conditions. In addition, it is found in the experiment that this method has significantly lower accuracy

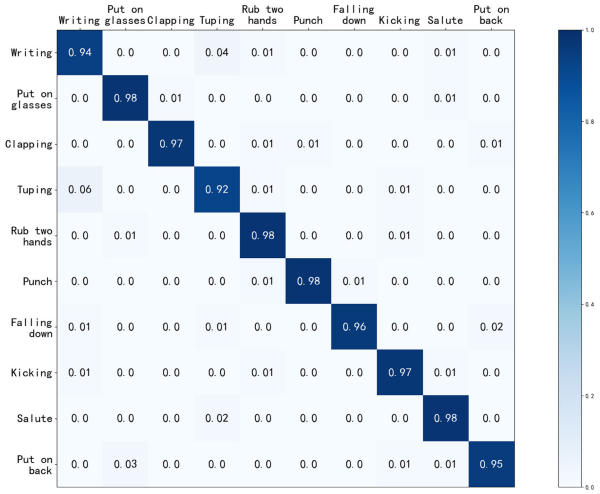

FIGURE 5. NTU-RGB+ D 60 dataset confusion matrix.

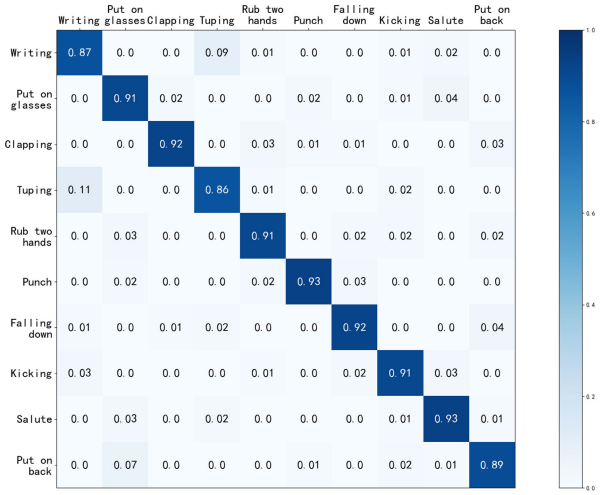

FIGURE 6. NTU-RGB+ D 120 dataset confusion matrix.

in dealing with incomplete action sequences, and it is difficult to infer action categories, which will be the focus of future work.

C. ABLATION EXPERIMENT

The effect of weight on the model has been evaluated on the NTU RGB + D dataset, which determines the proportion of self-matching loss in the model loss function. The average accuracy of 3,000 iterations before the convergence point is calculated as shown in Table 1. When ω is 0 or 1, it means that only L_s and L_m are used for training. The accuracy interval refers to the maximum relative difference between the accuracy and the convergence point accuracy in the iteration before convergence, which represents the absolute value. The accuracy is low when only a single matching branch is used, and the accuracy of the self-matching branch is significantly lower than that of the mutual matching branch. In the absence of labeled samples, the self-matching branch is

TABLE 1. Loss weight ablation experiment.

ω	accuracy (%)	Precision interval
0	63.05	1.29
0.3	65.53	1.14
0.4	69.21	1.54
0.5	68.36	1.63
0.6	65.50	2.16
1	54.88	4.09

similar to unsupervised clustering. The greater the weight of the branch, the stronger the randomness of the model. When set to 0.4 and 0.5, the accuracy is higher, indicating that the information of the two branches complements each other, and the performance in recognition accuracy is significantly improved. In the subsequent experiments, was set to 0.4.

For the progressive decoding module, ablation experiments are conducted with varying numbers of filters to investigate the impact of filtering features at each decoding level on the recognition performance.The results are shown in Table 2. It can be seen that when the filter is 0, the feature matching is directly performed using the skeletal joint point coordinate sequence. As the filter increases, the feature map obtained from the skeletal data is larger and the recognition accuracy is higher. When the number of layers is 2 or 3, a higher accuracy is achieved. This is because when the number of filters is too large, the skeleton graph is decomposed excessively so that the key differences of different behaviors are split and cannot be effectively identified. Considering the influence of the number of frames contained in the sample in the progressive decoding module on the recognition effect, the behavior samples are temporally encoded with multiple different frames, and the number and information of the generated query set samples are different. The ablation experiment results are shown in Table 3. None means that no temporal coding is performed, the sample is directly used as the element of the query set, and the joint point coordinates are convoluted according to the adjacency matrix. When 5 frames are selected as query and support samples, the optimal recognition accuracy is obtained.

TABLE 2. Filter layer ablation experiment.

Filter	accuracy (%)	Precision interval
0	49.28	4.59
1	52.94	3.44
2	65.21	2.87
3	68.08	2.64
4	64.81	2.55

D. COMPARISON WITH OTHER

Query set and support set were established to conduct comparative experiments with mainstream behavior recognition algorithms on the NTU RGB + D60 dataset, NTU RGB + D120 dataset, and Human36m dataset using the proposed method. The average value of repeated experiments was taken as the final result, and the results were shown in Table 4, Table 5 and Table 6 respectively. The selected

TABLE 3. Sampling interval ablation experiment.

sampling interval	1-shot Acc(%)		5-shot Acc(%)	
	Top-1	Top-5	Top-1	Top-5
None	42.28	51.48	43.01	54.10
1-frame	51.65	73.87	62.21	75.17
2-frame	58.50	78.14	66.25	79.94
3-frame	62.44	83.64	70.19	80.87
4-frame	70.84	90.28	74.70	85.08
5-frame	71.01	92.87	75.42	87.21
6-frame	68.05	90.70	70.21	84.99

TABLE 4. comparison on NTU RGB + D 60 dataset.

models	1-shot Acc(%)		5-shot Acc(%)	
	Top-1	Top-5	Top-1	Top-5
HCN[32]	53.6	80.3	59.1	78.4
2stream-3DCNN[33]	55.1	85.6	62.7	83.6
FC[34]	60.9	93.7	64.2	85.8
TCN [37]	64.8	93.8	66.8	92.5
Dynamc GCN [45]	70.2	94.1	74.3	91.2
AdaSGN [36]	71.3	96.1	76.3	91.8
ST-TR[38]	71.6	94.6	75.3	91.2
Eff-GCN[39]	70.3	93.8	76.4	90.8
STST[40]	72.8	96.3	77.1	92.3
Ours	73.7	97.4	78.3	92.8

TABLE 5. comparison on NTU RGB + D 120 dataset.

models	1-shot Acc(%)		5-shot Acc(%)	
	Top-1	Top-5	Top-1	Top-5
HCN[32]	40.6	69.5	55.3	70.2
2stream-3DCNN[33]	45.2	75.6	59.4	81.6
Dynamc GCN [45]	51.2	87.9	61.2	85.4
AdaSGN [36]	53.8	86.0	60.8	85.5
TCN [37]	46.5	84.8	60.3	86.1
MST-GCN[41]	50.4	83.1	63.3	84.0
Ta-GCN[42]	52.8	86.0	60.8	85.3
ShiftGCN++[43]	53.4	85.5	61.1	88.3
2s-STA-GCN[44]	51.6	84.7	60.7	87.4
Ours	58.8	89.4	69.4	88.5

TABLE 6. comparison on human36m dataset.

models	Top-1	Top-5
HCN[32]	45.3	52.4
2stream-3DCNN[33]	50.8	59.1
FC [34]	57.7	63.6
Dynamc GCN [45]	63.2	66.9
AdaSGN [36]	64.4	73.4
STRM[35]	64.8	72.0
TRX[24]	65.1	73.4
ST-TR[38]	63.0	72.5
Eff-GCN[39]	63.8	74.4
STST[40]	65.3	72.5
MST-GCN[41]	65.7	71.0
Ours	70.4	74.9

mainstream behavior recognition algorithms include CNN method [32], [33], RNN method [34] and graph neural network method [36], [37], [38], [39], [40], [41], [42], [43], [44], [45]. It can be seen that in a single recognition task, our method is significantly better than other models, among which the performance of CNN is poor, because CNN's convolution kernel size is fixed, it is not suitable for processing data with a long time series, and the key sparse information of the represented behavior will be lost due to the increase in the number of convolution layers. The RNN based method can effectively capture the time-dependent and dynamic evolution of bone sequences, and the recognition performance is improved, but the problem of gradient disappearance still exists. The method of graph neural network is not limited by bone connection and learns the connection and relationship between bones through the graph structure between joints, and the model performance is better than the previous method.

In the five recognition tasks, our method achieves the highest accuracy, but there is no gap compared with the better algorithm. This is because the self-matching mechanism increases the differentiation degree of such data in the weight in order to distinguish similar but different behavior samples. In the recognition results, if there is a classification situation that the result generated by the matrix is different from the real label, The model will reduce the feature weight of local differentiation, thus increasing the probability of correct samples with similar feature information being misjudged, resulting in a decrease in the accuracy of top5 results. Comparative experiments show that the proposed model is superior to the mainstream methods in top1 and top5 behavior recognition tasks, and has good generalization on different datasets.

V. CONCLUSION

In this paper, a small sample behavior recognition method based on a progressive decoding strategy is proposed. The network consists of two modules: the progressive decoding module realizes the feature extraction of skeleton sequence in time and space dimensions, and the feature matching module realizes local matching by constructing the kernel to perform convolution operations between feature maps. This matching method allows more attention to the information part of the feature map generated from the skeleton sample. The samples in the query and support set are used as input, and a similarity matrix is an output, where each element represents the similarity between the skeleton sequence in the query set and the skeleton sequence in the support set. Through experimental verification of two public data sets, the proposed method has better recognition accuracy and generalization of small sample behavior.

REFERENCES

[1] Y.-L. Hsueh, W.-N. Lie, and G.-Y. Guo, "Human behavior recognition from multiview videos," *Inf. Sci.*, vol. 517, pp. 275–296, May 2020.

[2] J. Carreira and A. Zisserman, "Quo Vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4724–4733.

[3] B. Fernando, E. Gavves, M. J. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5378–5387.

[4] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1–10.

[5] R. Dai, L. Minciullo, L. Garattoni, G. Francesca, and F. Bremond, "Self-attention temporal convolutional network for long-term daily living activity detection," in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2019, pp. 1–7.

[6] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The EPIC-KITCHENS dataset," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 720–736.

[7] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, pp. 7444–7452.

[8] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12018–12027.

[9] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7904–7913.

[10] K. Shiraki, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Spatial temporal attention graph convolutional networks with mechanics-stream for skeleton-based action recognition," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2020, pp. 1–17.

[11] J. Liu, N. Akhtar, and A. Mian, "Adversarial attack on skeleton-based human action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1609–1622, Apr. 2022.

[12] S. Liu, X. Bai, M. Fang, L. Li, and C.-C. Hung, "Mixed graph convolution and residual transformation network for skeleton-based action recognition," *Appl. Intell.*, vol. 52, pp. 1544–1555, May 2021.

[13] J. Cha, M. Saqlain, D. Kim, S. Lee, S. Lee, and S. Baek, "Learning 3D skeletal representation from transformer for action recognition," *IEEE Access*, vol. 10, pp. 67541–67550, 2022, doi: 10.1109/ACCESS.2022.3185058.

[14] K. Zhang, M. Liu, J. Zhang, and Z. Dong, "PA-MVSNet: Sparse-to-dense multi-view stereo with pyramid attention," *IEEE Access*, vol. 9, pp. 27908–27915, 2021, doi: 10.1109/ACCESS.2021.3058522.

[15] C. Zheng, L. Pan, and P. Wu, "Multimodal deep network embedding with integrated structure and attribute information," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1437–1449, May 2020.

[16] Y. Min, F. Wenkel, and G. Wolf, "Scattering GCN: Overcoming oversmoothness in graph convolutional networks," in *Proc. 34th Conf. Neural Inf. Process. Syst.*, 2020, pp. 1–11.

[17] C. Pan, S. Chen, and A. Ortega, "Spatio-temporal graph scattering transform," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–18.

[18] D. Zou and G. Lerman, "Graph convolutional neural networks via scattering," *Appl. Comput. Harmon. Anal.*, vol. 49, no. 3, pp. 1046–1074, Nov. 2020.

[19] Y. Min, F. Wenkel, and G. Wolf, "Geometric scattering attention networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 8518–8522.

[20] F. Wenkel, Y. Min, M. Hirn, M. Perlmutter, and G. Wolf, "Overcoming oversmoothness in graph convolutional networks via hybrid scattering networks," 2022, *arXiv:2201.08932*.

[21] Y. Wang, C. Xu, C. Liu, L. Zhang, and Y. Fu, "Instance credibility inference for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12836–12845.

[22] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: Few-shot image classification with differentiable Earth Mover's Distance and structured classifiers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12200–12210.

[23] M. Bishay, G. Zoumpourlis, and I. Patras, "TARN: Temporal Attentive Relation Network for few-shot and zero-shot action recognition," 2019, *arXiv:1907.09021*.

[24] T. Perrett, A. Masullo, T. Burghardt, M. Mirmehdi, and D. Damen, "Temporal-Relational CrossTransformers for few-shot action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 475–484.
[25] X. Wang, S. Zhang, Z. Qing, M. Tang, Z. Zuo, C. Gao, R. Jin, and N. Sang, "Hybrid relation guided set matching for few-shot action recognition," 2022, *arXiv:2204.13423*.
[26] C. Careaga, B. Hutchinson, N. Hodas, and L. Phillips, "Metric-based few-shot learning for video action recognition," 2019, *arXiv:1909.09602*.
[27] R. Memmesheimer, N. Theisen, and D. Paulus, "SL-DML: Signal level deep metric learning for multimodal one-shot action recognition," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 4573–4580.
[28] S. Liao and L. Shao, "Interpretable and generalizable person re-identification with query-adaptive convolution and temporal lifting," in *Proc. Eur. Conf. Comput. Vis.*, Nov. 2020, pp. 456–474.
[29] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.
[30] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014, doi: 10.1109/TPAMI.2013.248.
[31] A. Shahroudy, J. Liu, T. T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1010–1019.
[32] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 786–792.
[33] H. Liu, J. Tu, and M. Liu, "Two-stream 3D convolutional neural network for skeleton-based action recognition," 2017, *arXiv:1705.08106*.
[34] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 1647–1656.
[35] A. Thatipelli, S. Narayan, S. Khan, R. M. Anwer, F. S. Khan, and B. Ghanem, "Spatio-temporal relation modeling for few-shot action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19926–19935.
[36] A. Sabater, L. Santos, J. Santos-Victor, A. Bernardino, L. Montesano, and A. C. Murillo, "One-shot action recognition in challenging therapy scenarios," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2771–2779.
[37] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "AdaSGN: Adapting joint number and model size for efficient skeleton-based action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13393–13402.
[38] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks," 2020, *arXiv:2008.07404*.
[39] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Constructing stronger and faster baselines for skeleton-based action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1474–1488, Feb. 2023.
[40] Y. Zhang, B. Wu, W. Li, L. Duan, and C. Gan, "STST: Spatial–temporal specialized transformer for skeleton-based action recognition," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3229–3237.
[41] K. Yang, X. Ding, and W. Chen, "Multi-scale spatial temporal graph convolutional LSTM network for skeleton-based human action recognition," in *Proc. Int. Conf. Video, Signal Image Process.*, Oct. 2019, pp. 3–9.
[42] K. Xu, F. Ye, Q. Zhong, and D. Xie, "Topology-aware convolutional neural network for efficient skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 3, pp. 2866–2874.
[43] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 180–189.
[44] R. Hang and M. Li, "Spatial–temporal adaptive graph convolutional network for skeleton-based action recognition," in *Proc. 16th Asian Conf. Comput. Vis. (ACCV)*, Macao, China, Dec. 2023, pp. 172–188.
[45] F. Ye, S. Pu, Q. Zhong, C. Li, D. Xie, and H. Tang, "Dynamic GCN: Context-enriched topology learning for skeleton-based action recognition," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 55–63.

XIANGTAO ZHAO received the bachelor's degree from the Institute of Future, Qingdao University, in 2021, where he is currently pursuing the master's degree. His research interests include image processing and behavior recognition related work.

ZHIHAN LI received the bachelor's degree from the Institute of Future, Qingdao University, in 2020, where he is currently pursuing the master's degree. His research interests include image processing and behavior recognition related work.

YINHUA LIU received the Ph.D. degree in mechanical engineering, Japan, in March 2013. He is currently the Vice President of the Future Research Institute. In August 2017, he joined the School of Automation, Qingdao University, where he start the establishment of the future research, in November 2017. His research interests include advanced energy storage, smart medical, immersive systems, vehicle electronic control systems, mathematical modeling, and control system design.

YUXIAO XIA is currently pursuing the master's degree with Qingdao University. Her research interests include human–computer interaction, computer vision, and recognition of emotion.