

## RESEARCH ARTICLE

# Multi-Scale Adaptive Graph Convolution Network for Skeleton-Based Action Recognition

HUANGSHUI HU, (Member, IEEE), YUE FANG<sup>ID</sup>, MEI HAN, AND XINGSHUO QI

College of Computer Science and Engineering, Changchun University of Technology, Changchun, Jilin 130012, China

Corresponding author: Yue Fang (3508901047@qq.com)

This work was supported by the Key Industrial Core Technology Project of the Jilin Provincial Science and Technology Department under Grant 20210201051GX.

**ABSTRACT** The skeleton-based action recognition technology can effectively avoid the background interference and occlusion problems in the image. However, the recognition of similar actions is still a challenge. In this paper, a multi-scale dynamic topological modeling method (MDTM) is proposed to solve this problem. The topological modeling through the convolution kernel generated from the original data, increases the connection of the convolution process to the original data compared to the previous randomly generated ones, effectively distinguishing similar actions. In addition, MDTM uses a multi-scale temporal convolutional network to obtain a wider receptive field, which can effectively extract the temporal information in the action. At the same time, a dynamic topology learning method is utilized to design a spatiotemporal information extractor that can effectively extract the spatiotemporal information in the action to dynamically adjust the topology structure. Extensive experiments have performed on three large-scale datasets, NTU RGB + D 60, NTU RGB + D 120, and NW-UCLA to validate the effectiveness of MDTM. The results show that MART-GCN performs better than the others in terms of accuracy and number of parameters.

**INDEX TERMS** Multi-scale graph convolution, adaptive convolution kernel, action recognition, dynamic topology.

## I. INTRODUCTION

As a central task of video understanding and an important direction of computer vision, human action recognition has always been a hot topic in the field of artificial intelligence. With the rapid development of human action recognition technology, it has been widely used in video surveillance, smart home, sports competition, medical rehabilitation, and other fields [1], [2], [3], [4], [5]. However, due to the complexity of research and the limitations of the dataset, action diversity, occlusion, and light variation have been main challenges and difficulties in human action recognition. In recent years, skeleton-based action recognition technology has received extensive attention. Compared with image-based action recognition technology, it has better robustness

and accuracy. Most importantly, it can avoid background interference and occlusion problems in the image.

Skeleton-based action recognition method is used to identify the actions and behaviors of the human body through the analysis and processing of the skeleton, which usually adopts deep learning models to extract the key points of human skeleton from images or videos, and then analyzes and identifies the movement trajectory of these key points to realize the recognition and classification of human action.

In [6], [7], [8], [9], [10], [11], and [12], graph convolutional neural networks are used for feature extraction and classification of skeleton data. In [6], the ST-GCN model manually defines the adjacency matrix to traverse the neighbor nodes, and multiple adjacency matrices are formed using a partitioning strategy of spatial configurations, with assigned different weights. To better facilitate the discrimination of similar actions, a learnable mask is added to each layer to weigh the different joints to adjust the contribution of

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

the node features to their adjacent nodes. However, the skeleton diagram used in ST-GCN is predefined, which displays only the human physical structure, and the adjacency matrix applied in ST-GCN is fixed in all layers, lacking the flexibility to model the multilayers of semantic information. Accordingly, in [7], the 2s-AGCN model improves the adjacency matrix by dividing into three parts, including the physical structure of the human body, the trainable weights for data learning, the unique graph learned per sample. To improve the flexibility of the adjacency matrix, the second part designs a trainable data-driven weight, and the third part generates unique graphs for each sample using the embedding function. DGNN [8] considers that some parts of the body are strongly related in some action classes, but there is no connection between them in the adjacency matrix based on the physical structure of the human body. To solve this problem, DGNN utilizes adaptive graphs instead of fixed graphs for the aggregation of neighbor nodes. DGNN directly takes the association matrix  $A$  [9] as the model topology, which is fixed before training and then unfixed, which improves the flexibility of topology construction. Unlike DGNN, Dynamic GCN [10] designs a context encoding network and beds it into graph convolution to automatically construct end-to-end skeleton topology. Each sample in the data in Dynamic GCN has a unique topology in each graph convolutional layer, namely, the topology is automatically adjusted during training, resulting in a dynamic topology, which greatly improves the expression and flexibility of the model. Meanwhile, the context encoding network proposed in the paper does not need to construct a prior topology, which is fully data-driven, and able to aggregate the contextual information of the joint. However, the design of the above topology increases the model complexity and requires longer training time. In addition, more context information needs to learn the topology, which has certain requirements for the data set.

In this paper, a multi-scale dynamic topological modeling graph convolutional network is designed to perform skeleton-based action recognition. First, the dynamic learning topology modeling is adopted to dynamically adjust the topological graph to better capture the temporal relationship in the action. At the same time, instead of using the prior topology including the physical structure of the human body, the adjacency matrix directly randomly initialized by the normal distribution as the shared topology of MDTM. By doing so, the topology depends entirely on the experimental data for learning, and enhances the flexibility of the model, and avoids the process of manually defining the topology structure, which reduces the complexity of the model to some extent. Second, topological modeling, node aggregation and other operations through the convolution kernels generated from the original data are used to enable the model to extract richer and more meaningful feature representations. In addition, the spatiotemporal properties of actions are also considered in the topology modeling, which not only models the adjacency matrix with spatial dimensions, but also integrates the action

information of time dimensions into the adjacency matrix, enriching the features extracted by the graph convolution. Third, the three branches of the graph convolution are joined in the spatial module for channel dimensions, with the added edge convolutional module, which greatly reduces the number of model parameters. Finally, three skeleton-based action recognition datasets NTU RGB + D 60 [13], NTU RGB + D 120 [14], and NW-UCLA [15] are used to evaluate the performance of proposed MTRGCN compared with the state-of-the-art methods. The main contributions of this paper are summarized as follows:

- 1) A spatiotemporal topological modeling module using an adaptive convolution kernel is presented to enhance the connection between the adjacency matrix and the raw skeleton data, which is more favorable to the model to distinguish between similar actions.
- 2) A dynamic topological modeling approach as well as multi-scale temporal convolutional modules is used to reduce model training time while increasing flexibility.
- 3) By performing extensive experiments, the effectiveness of the model is demonstrated on three large-scale datasets, NTU RGB + D 60, NTU RGB + D 120, and conducting Northwestern-UCLA.

## II. RELATED WORK

Graph neural network (GNN) [16], [17], [18], [19], [20] is a class of deep learning model used to process graph data. It has become a research hotspot in the fields of computer vision, natural language processing and recommendation system in recent years. There are many kinds of graph neural networks, including graph convolutional network GCN [16], graph attention network GAT [17], graph autoencoder GAE [18], etc. Among them, GCN [16] is like CNN, which has strong feature learning ability. Its essential purpose is to extract the spatial features of topological graphs. It is the mainstream way to achieve the goal in both spatial and spectrum domains. Because the Fourier transform usually assumes that the graph structure is fixed, in the frequency domain, it is difficult for the graph convolutional network to directly process the dynamic graph because the graph changes over time, and involves the eigenvalue decomposition of the matrix, resulting in high complex computation. Therefore, spatial domains-based graph convolutional network is usually used for action recognition [21], [22], [23], [24].

Early skeleton-based behavioral recognition uses RNN [25], [26], [27], [28], [29], [30] or CNN [31], [32], [33], [34], [35] to learn temporal features, however these methods focus on the connection between time series and pay little attention to the effects between joints. Graph convolution network extends CNN to the Euclidean structure, Yan et al. [6] first put forward the temporal graph convolution network (ST-GCN) skeleton model, which extends the graph convolution network to spatiotemporal map model and uses for behavior recognition of skeleton sequence general representation. According to the human body physical structure,

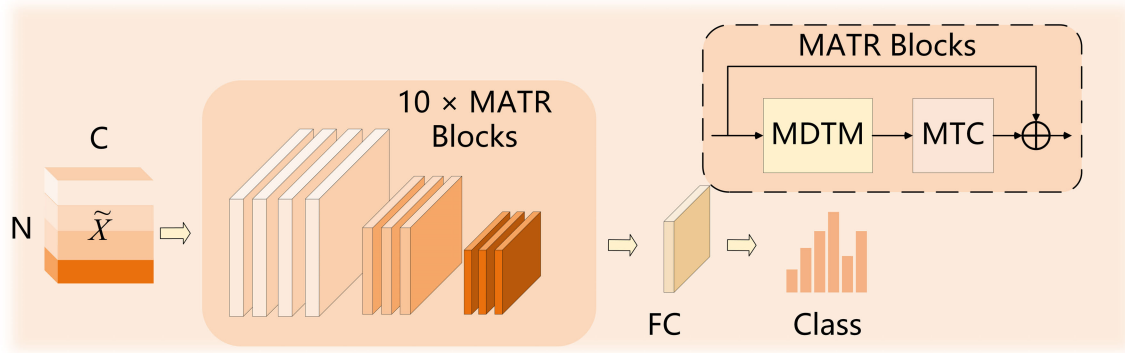


FIGURE 1. Basic network framework.

a manually defined topology is designed in ST-GCN based on the skeleton sequence structure spatiotemporal diagram, which is applied to the spatiotemporal map convolution, and generate higher level of features on the graph. However, the manually defined topology is difficult to achieve the aggregation of nodes without physical connection, which greatly affects the recognition rate and generalization of the network. To obtain more connectivity relationships between the joints, and improve the expressive ability of the model, ST-GAT [21] defines the spatiotemporal adjacent nodes and aggregation functions of the root node through the attention mechanism. The AS-GCN [22] introduces an encoder-decoder structure to capture action-specific potential dependencies directly from the actions. The graph topology of MS-AAGCN [23] can be learned end-to-end based on the input data. The MPA-GC [36] can adaptively learn the topology of each part of the body, and dynamically aggregate their correlations. FLAGCN [37] proposes two graph convolution methods to capture different aspects of human behavioral information. Frame-level graph convolution constructs the human topology for each data frame, while the adjacent graph convolution captures the features of the adjacent joints. These methods enable the dynamic adjustment of the topology by adding the learned topology according to the input data. However, the topology is identical in different channels, which reduces the flexibility of the network to extract features. CTR-GCN [24] can efficiently aggregate features in different channels and improve the expression ability of the model. However, it still relies on fixed topology and requires different refinement networks for different datasets, with poor network generalization.

### III. PROPOSED METHOD

In MTRGCN, a multi-scale graph convolution module using different topology modeling methods in different channels is proposed to increase network generalization, and randomly initialized adjacent matrix as a shared topology is used to enhance the connection between topology structure and input data. At the same time, an adaptive convolution kernel

generated by raw data is designed to improve the recognition performance of the model for similar actions.

#### A. PREPARATION

The graph on the skeleton data can be represented as  $G = (V, E, X)$ , where  $V$  is the set of vertices (or nodes) in the graph. In the context of the skeleton sequence, each vertex represents a body joint, for example, the head, neck, shoulder, elbow, wrist, hip, knee, and ankle.  $E$  is a set of edges in a graph representing the connections between pairs of body joints. For the human body, adjacent joints are usually connected, such as head to neck, neck and shoulder, elbow and shoulder, etc.  $X \in \mathbb{R}^{C \times T \times V}$  ( $C, T$ , and  $V$  represent the number of channels, length of time, and number of skeletal joints, respectively) is a set of joint coordinates or attributes associated with each node. Thus, the graph convolution can be represented by:

$$H^{(l+1)} = \sigma \left( \tilde{D}^{1/2} \tilde{A} \tilde{D}^{1/2} H^{(l)} W^{(l)} \right) \quad (1)$$

where  $H^{(l)} \in \mathbb{R}^{N \times d}$  is the node feature matrix of layer  $l$ ,  $N$  is the number of nodes in the graph, and  $d$  is the feature dimension of each node.  $W^{(l)} \in \mathbb{R}^{N \times d'}$  is the weight matrix of layer  $l$  and  $d'$  is the dimension of the output features.  $\tilde{A} = A + I_N$  is the result after the adjacency matrix  $A$  plus the self-loop and  $I_N$  is the  $N$ -dimensional identity matrix.  $\tilde{D}$  is a matrix of angular angles whose diagonal elements are  $\tilde{D}_{ij} = \sum_j \tilde{A}_{ij}$ .  $\sigma(\cdot)$  is an activation function, usually using the activation function or other non-linear functions.

A common practice in previous works sets  $A$  to a series of manually defined matrices (defined according to the physical structure of the human body) and learns  $\tilde{A}$  further using some methods [7], [23], [24].  $\tilde{A}$  Can be static, as a trainable parameter, or a dynamic parameter generated from the input data. Although the manually defined adjacency matrix can more intuitively show the connection of nodes, it is limited in flexibility and may lead to worse identification performance than pure data segmentation methods. Therefore, we randomly initialize the coefficient matrix  $PA \in \mathbb{R}^{3 \times V \times V}$  to make the PA completely learnable. So, the formula for the graph

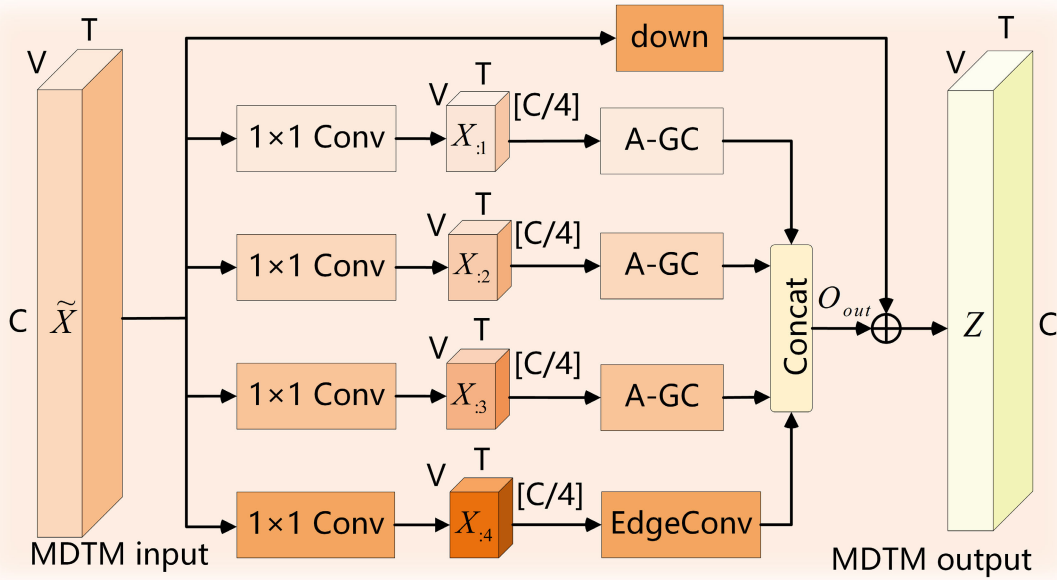


FIGURE 2. The constituent framework of the MDTM.

convolution can be written as follows:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}) \quad (2)$$

where,  $\hat{A}$  is the topology learned by using PA.

### B. OVERALL ARCHITECTURE

Based on multi-scale dynamic topological modeling (MDTM) and multi-scale temporal convolutional (MTC), a light and efficient graph convolutional network MATR-GCN is constructed for skeleton-based action recognition. As shown in Fig. 1, the entire network consists of 10 basic blocks, followed by a global average pool and a classifier to predict action categories. The number of channels for the 10 blocks was 64-64-64-64-128-128-128-256-256-256. The time dimension is halved by time convolution in blocks 5 and 8. The basic blocks of our MATR are shown in Fig. 1 as dashed boxes, each block mainly consists of a spatial modeling module, a temporal modeling module, and a residual connection.

### C. MULTISCALE DYNAMIC TOPOLOGY MODELING MODULE

Manually defined static topologies require more network layers to obtain information about each other, increasing network complexity, and are not the best choice for models. Therefore, MDTM are proposed, which utilizes different modeling functions for different channels of the adjacency matrix, and dynamically adjusts the topology according to the input data.

Fig. 2 shows the proposed MDTM module. The first three branches of the module are the Adaptive Graph

convolution module (A-GC) for channel topology modeling of the adjacency matrix and feature aggregation, and the bottom path is the Edge convolution module (EdgeConv) for fusing local information and its own information. Each branch undergoes a  $1 \times 1$  convolution to reduce the channel dimension to a quarter of the original one. Finally, the four branches are concatenated in the channel dimension to obtain the output.

In terms of adjacency matrix information aggregation, the operation of adding the output results of four branches are changed to concatenate them on channels, which greatly reduces the number of parameters in the model. At the same time, EdgeConv are spliced to capture important features in the input data and reduce redundant information. Its process is expressed as:

$$Z = \sigma(b_n(O_{out} + \text{down})) \quad (3)$$

where,  $Z \in \mathbb{R}^{C_{out} \times T \times V}$  ( $C_{out} = K \times C''$ ) is the output of MDTM,  $\sigma(\cdot)$  is the activation function ReLU,  $b_n(\cdot)$  and  $\text{down}$  are batch normalization and residual connection respectively. The output of each branch is concatenated in the MDTM module in the channel dimension to obtain  $O_{out} \in \mathbb{R}^{K \times C'' \times T \times V}$  ( $K = 4$ ), which is expressed as:

$$O_{out} = [Y(X_{:1}):_1 \parallel Y(X_{:2}):_2 \parallel Y(X_{:3}):_3 \parallel E(X_{:4})] \quad (4)$$

$$X_{:i} = \text{conv}(\tilde{X})_i \quad (5)$$

$\tilde{X} \in \mathbb{R}^{C \times T \times V}$  and  $E \in \mathbb{R}^{C'' \times T \times V}$  are the inputs of the MDTM module and the output of the EdgeConv module;  $\text{conv}(\cdot)$ ,  $O(\cdot)$ , and  $\parallel$  represent  $1 \times 1$  convolution, concatenation of channel dimensions and connection operations, respectively;  $Y_i \in \mathbb{R}^{C'' \times T \times V}$  and  $X_{:i} \in \mathbb{R}^{(C/4) \times T \times V}$  are the input and output of the A-GC in the  $i$ th ( $i \in 1, 2, \dots, K$ ) path.

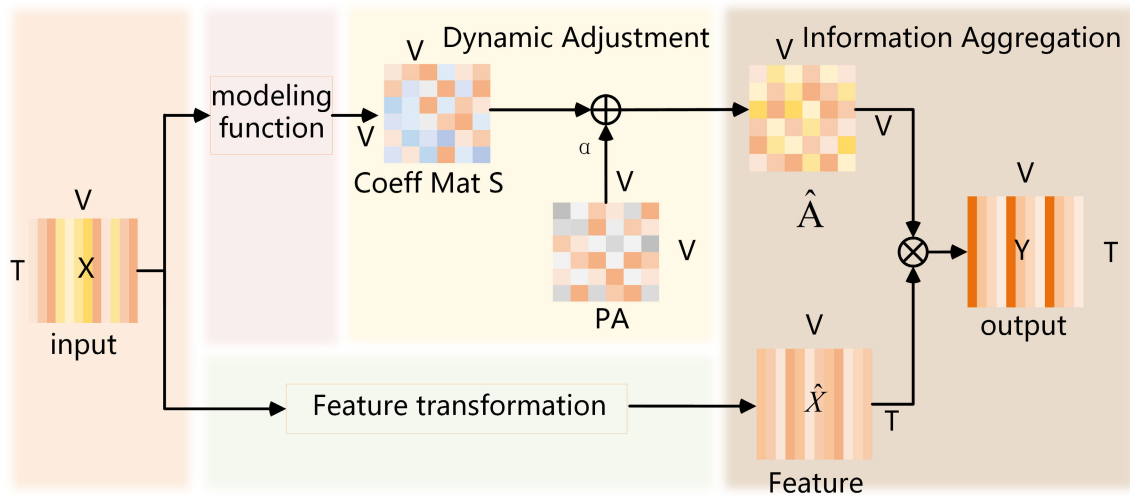


FIGURE 3. The constituent framework of the A-GC.

1) ADAPTIVE GRAPH CONVOLUTION

In the traditional graph convolution [6], [7], [22], the adjacency matrix is predefined according to the physical structure of the human body. However, in some actions, the nodes are not physically connected, for example, the nodes between the two feet play a key role. In this regard, A-GC is proposed in MDTM, in which the adjacency matrix can be automatically adjusted according to different input data. At the same time, three dynamic modeling functions are designed, and the convolution kernel generated by the original data is used to increase the connection between the adjacency matrix and the input data and improve the generalization of the model.

The A-GC uses  $X \in \mathbb{R}^{(C/4) \times T \times V}$  as input, where  $C/4$ ,  $T$ , and  $V$  are expressed as channel number, joint number, and input frame number, respectively. The A-GC mainly consists of four parts, including feature transformation, model building, dynamic adjustment, and information aggregation, as shown in Fig. 3.

As shown in Fig. 3, the lower box is the feature transformation section. The  $T(\cdot)$  function is used to transform the input data into a high-level representation to extract richer features. The process is expressed as follows:

$$\hat{X} = T(X) = XW, \quad T(\cdot) = \text{conv}(X) \quad (6)$$

Among them,  $\hat{X} \in \mathbb{R}^{C' \times T \times V}$  is the transformed feature,  $W \in \mathbb{R}^{C' \times C}$  is the corresponding weight matrix,  $T(\cdot)$  is the function of feature transformation. Moreover, the convolution with  $1 \times 1$  convolution kernel size is chosen.

The upper box part is the model building block, which is used to generate the coefficient matrix  $S \in \mathbb{R}^{C' \times V \times V}$ , ( $C' = (C/4)/r$  ( $r$  is the decay rate)). The middle part is the dynamic adjustment module, which is used to generate the final adjacency matrix  $\hat{A} \in \mathbb{R}^{C' \times T \times V}$ . The right part is the information aggregation module, which uses the adjacency

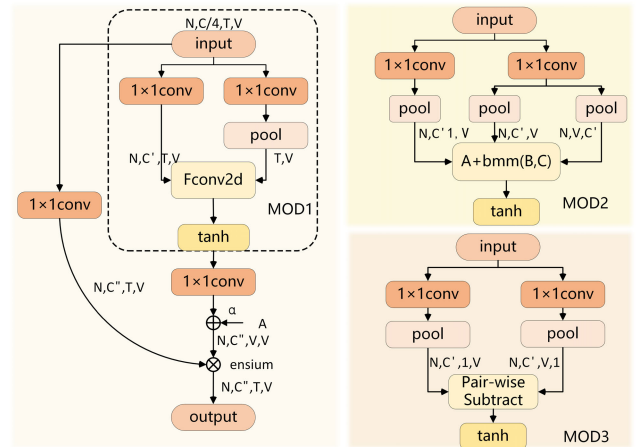


FIGURE 4. The compositional framework of the three modeling functions.

matrix  $\hat{A}$  to aggregate the generated advanced features. The process of the whole A-GC is expressed as:

$$Y = I(D(M(X)), PA, \hat{X}) \quad (7)$$

where  $Y \in \mathbb{R}^{C' \times T \times V}$  is the output of A-GC, and  $I(\cdot)$ ,  $D(\cdot)$ , and  $M(\cdot)$  represent the node aggregation function, dynamic adjustment function and topology modeling function, respectively. The node aggregation function and the dynamic adjustment function are shown in Equations 8 and 9:

$$Y_{C' \times T \times V} = I(\hat{A}, \hat{X}) = \sum_{i=1}^V \hat{A}_{C' \times V, i} \hat{X}_{C' \times T, i} \quad (8)$$

$$\hat{A} = D(S, PA) = \text{conv}(S) + \alpha PA \quad (9)$$

where,  $\alpha$  is the trainable parameter, enabling dynamically adjusting the topology of the adjacency matrix  $\hat{A}$ .

As shown in Fig. 4, as the coefficient matrix PA has three channels, three topological modeling functions (MOD 1,

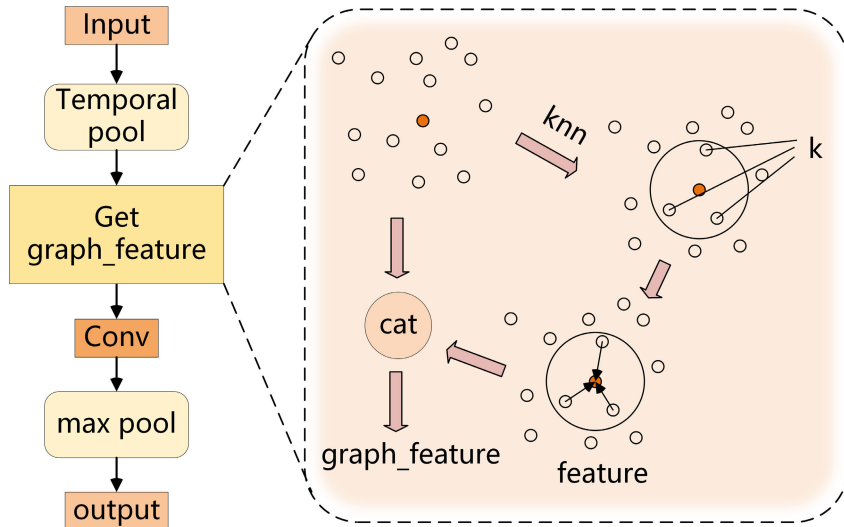


FIGURE 5. The compositional framework of the EdgeConv.

MOD 2, and MOD 3) are constructed and applied to the first three paths of MDTM. The dotted area on the left is the topology modeling function MOD1, with input as  $X \in \mathbb{R}^{(C/4) \times T \times V}$ . To reduce the number of calculation parameters, they are respectively convolved with a convolution kernel size of  $1 \times 1$  to reduce the channel dimension, and an average pooling operation is performed on one of the outputs. Finally, the outputs are  $L \in \mathbb{R}^{C' \times V \times T}$  and  $Ker \in \mathbb{R}^{V \times T}$ , and then Ker is used as the convolution kernel to perform the convolution operation on  $L$ . The process is expressed as follows:

$$S_{C'VV} = M_1(L, Ker) = \sigma\left(\sum_{i=1}^{C'} L_{VTC'} \cdot Ker_{VT1}\right) \quad (10)$$

where  $\sigma(\cdot)$  is the activation function Tanh. In MOD1, the convolution kernel generated from the original data is used, so that the generated topology is more suitable for the training data and can better represent the connections between nodes. As shown in the upper yellow box on the right in Fig. 4, it is the MOD2 topology modeling function. After the input data is convolved with the kernel size of  $1 \times 1$  and the pooling operation, three outputs  $A \in \mathbb{R}^{C'}$ ,  $B \in \mathbb{R}^{C' \times V}$ , and  $C \in \mathbb{R}^{V \times C'}$  are obtained. Batch matrix multiplication of  $B$  and  $C$  is performed and then added to  $A$ . The process is expressed as follows:

$$S_{C'VV} = M_2(A, B, C) = \sigma(A_{C'1V} + \sum_{l=1}^{C'} C_{V1} \cdot B_{lV}) \quad (11)$$

where  $\sigma(\cdot)$  is the activation function Tanh. In MOD 2, we use batch matrix multiplication, which allows simultaneous matrix multiplication operations on multiple matrices, thus enables efficient parallel computing, saves model training time, and allows matrix multiplication between different locations, making it easier for models to capture the relationship between spatial locations. As shown in the pink box on the bottom right side in Fig. 4, it is the MOD 3

topology modeling function. After  $1 \times 1$  convolution and pooling operations, respectively, the output  $\Phi \in \mathbb{R}^{C' \times V}$  and  $\Psi \in \mathbb{R}^{V \times C'}$  are obtained, and the pair subtraction operation is made. The process is expressed as follows:

$$S_{C'VV} = M_3(\Psi, \Phi) = \sigma(\Psi_{C'1V} - \Phi_{C'V1}) \quad (12)$$

where  $\sigma(\cdot)$  is the activation function Tanh.

## 2) EDGE CONVOLUTION MODULE

Since the human body may have small posture adjustments or local movement differences when performing actions, resulting in the overall actions looking similar but not identical, for similar actions, the differences between them are usually reflected in the local changes of the body, for example, in ‘reading’ and ‘writing’, only the hand movement changes are different. For the recognition of similar actions, more local information is needed to distinguish them, and we incorporate the EdgeConv module into the model to learn the relationship between nodes and feature representation in the data. EdgeConv can capture the relationship between nodes, so that the model can better understand the topology of the graph data.

The basic composition of EdgeConv is shown in Fig. 5. The module input is  $X \in \mathbb{R}^{(C/4) \times T \times V}$ , which is input into the Get graph\_feature module after time pooling to obtain the local graph features of each node. The graph features are input into the  $1 \times 1$  convolution and Max pooling layer for update and integration to obtain the output  $E \in \mathbb{R}^{(C'' \times T \times V)}$ .

The Get graph feature module is shown in the dashed box on the right in Fig. 5. The K Nearest Neighbors (KNN) algorithm is used to obtain the indices of the nearest K (K is 3) neighbor nodes of each node, and a feature map containing the relationship between the node and its neighbors is constructed according to the K indices. The generated feature map is spliced with the original data to obtain the final graph feature. In the KNN algorithm,

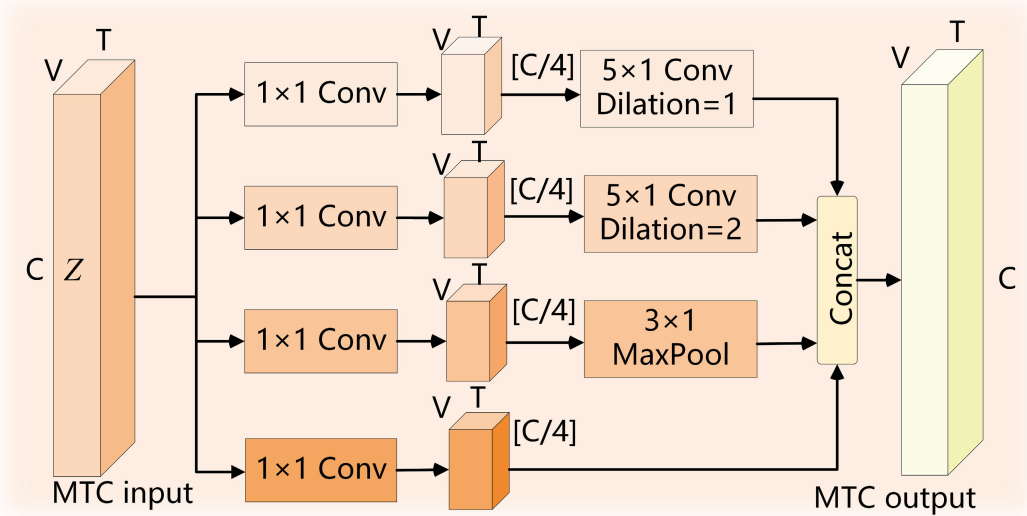


FIGURE 6. The compositional framework of the MTC.

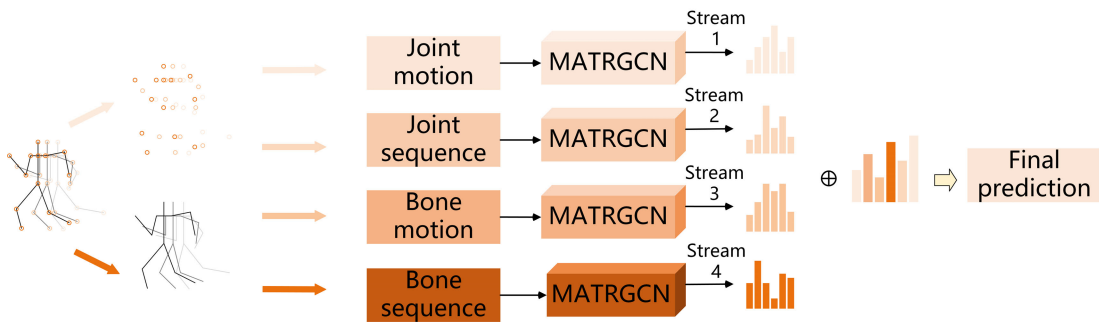


FIGURE 7. Multi-stream fusion framework.

we use the Manhattan distance instead of the commonly used Euclidean distance as a measure of the distance between a node and its neighbor nodes. Compared with Euclidean distance, Manhattan distance is more suitable for dealing with the non-Euclidean structure of graph data, and the calculation is simpler, which reduces the number of parameters of the model.

#### D. MULTI-SCALE TEMPORAL CONVOLUTION MODULE

Time series data usually contains information at different time scales. To model the behavioral information at different time scales, a multi-scale temporal convolution module is used, which can consider information at multiple scales at the same time, to capture the behavioral characteristics of the data more comprehensively. Inspired by the work of Liu et al. [38], temporal convolutions with different dilation coefficients are used to extract temporal features in MARTGCN. As shown in Fig. 6, there are four branches in total. The first two branches use dilated convolution with dilation rate of 1 and 2 respectively to increase the time receptive field; the third branch uses maximum pooling to extract the main

features of the time dimension in the data, and the last branch uses residual connection to enhance the connection with the network of the previous layer. Each branch is subjected to  $1 \times 1$  convolution to reduce the channel dimension, facilitate the later channel splicing, and reduce the number of model parameters.

#### IV. THE EXPERIMENTS

In this section, we conduct experiments on three public datasets: NTU RGB+D 60, NTU RGB+D 120 and Northwest-UCLA. The specific implementation of the experiment is as follows: Firstly, the ablation experiment of the model is conducted on the NTU RGB+D 60 dataset using joint flow with cross-subject evaluation benchmark, in terms of the model accuracy and complexity. Secondly, the accuracy of the four modalities of joint, bone, joint motion and bone motion is tested on the Northwest-UCLA dataset, and the four-stream fusion test is performed. Then, the visualization of a class of samples is performed in the NTU RGB+D 120 dataset and the Northwest-UCLA dataset, and the visualization of confusion matrix and

adjacency matrix on the NTU RGB+D 60 and Northwest-UCLA datasets. Finally, the proposed MART-GCN is compared with the state-of-the-art models on three datasets respectively.

### A. DATASET

**NTU RGB+D 60.** The NTU RGB+D 60 dataset is a widely used skeleton-based action recognition dataset consisting of 56880 samples divided into 60 categories. The samples were performed by 40 different people and captured by three cameras to ensure a variety of actions. Each sample consists of 25 human joints, specifically 3D coordinates. The evaluation of this dataset includes a cross-view (X-view) benchmark and a cross-subject (X-sub) benchmark. They verified the diversity of the model's actions under different viewpoints and different roles. For the X-view benchmark, the training data is from cameras 2 and 3, and the test data is from camera 1. For the X-sub benchmark, the training data are from 20 subjects and the testing data are from the other 20 subjects.

**NTU RGB+D 120.** The NTU RGB+D 120 dataset is a large 3D skeleton dataset for human action recognition, which is an extension of the NTU RGB+D 60 dataset. This version has 113,945 skeleton clips divided into 120 classes. The movements were also performed by three people captured by 106 cameras. There are also two ways to validate benchmarks: across subjects (X-sub) and across Settings (X-set). The training data is from 53 subjects and the testing data is from the other 53 subjects of the X-sub benchmark. For the X-set benchmark, the training data comes from samples with even collector ids and the test data comes from samples with odd ids.

**Northwestern-UCLA.** The Northwest-UCLA dataset is a popular skeleton dataset captured by a Kinect camera. These actions are grouped into 10 classes for 1494 skeleton sequences. Specifically, the actions were performed by 10 different subjects. The evaluation setup is like the cross-view benchmark for the NTU RGB+D 60 dataset. The training data is captured by cameras 1 and 2, and the test data is from camera 3.

### B. IMPLEMENTATION DETAILS

The experiments were performed on a single GPU3090 using the Pytorch platform. The model has an initial learning rate (lr) of 0.1 and a decay factor of 0.1. We use the cross-entropy loss function as our classification loss function, and the optimization method is SGD with momentum 0.9 and weight decay 0.0004. For the stability problem, we adopt the warm-up strategy [35] in the first five epochs. We set the batch size to 64 on the NTU RGB+D 60 and NTU RGB+D 120 datasets and 16 on the Northwestern UCLA dataset. The training Settings on NTU RGB+D 60, NTU RGB+D 120 and Northwestern UCLA datasets are 100, 120 and 65, respectively, and the learning rate decay stages are (35,55,80), (35,55,100) and 65, respectively.

**TABLE 1. Comparisons of the validation accuracy of MATR with different configurations.**

Models	Accuracy(%)	Params(M)
baseline	89.4	2.09
+MDTM	89.6(0.2↑)	1.88(0.29↓)
MDTM+PA	89.7(0.3↑)	1.88(0.29↓)
+MTC	89.5(0.1↑)	1.29(0.8↓)
MTC+PA	89.5(0.1↑)	1.29(0.8↓)
MDTM+MTC	89.8(0.4↑)	1.08(1.01↓)
<b>Ours(MDTM+MTC+PA)</b>	<b>90.0(0.6↑)</b>	<b>1.08(1.01↓)</b>

Replace the modules in the baseline model with modules MDTM, MTC, PA, respectively.

### C. ABLATION EXPERIMENT

To highlight the effectiveness of the proposed MATR-GCN model, extensive experiments are conducted on NTU RGB+D 60 datasets. First, we tested our model with different components to verify the effectiveness of each component of the model. In addition, we construct models with different input data modalities, mainly: joint flow, bone flow, joint motion, and bone motion flow, and then compute the accuracy for each data stream separately.

#### 1) EFFECTS OF MODEL COMPONENTS

We use ST-GCN [6] as the baseline, and due to its static shared topology, the topology structure is untrained and shared in each layer. For fair comparison, we change the model structure, so that the adjacency matrix is trained together with the data, and the adjacency matrix is not passed between each layer. As shown in Table 1, under the X-sub benchmark of NTU-RGB+D dataset, MDTM module, MTC module and randomly initialized adjacency matrix (PA) module are respectively replaced with the modules used by our model MATR. When all of them are replaced with the modules used by our model MATR, the accuracy is the highest (0.6% increase), and the parameter number is the lowest (1.01 M decrease). This shows the effectiveness of MDTM module, MTC module and PA module. Using MDTM instead of the spatial modeling module in the baseline, the accuracy of the model is improved by 0.2%, and the number of parameters is reduced by 0.21 M. On this basis, adding random initialization adjacency matrix, the accuracy of the model is improved by 0.3% compared with the baseline model. We prove the effectiveness of our designed MDTM and PA. Using the MTC module instead of the time modeling module in the baseline, the accuracy of the model is increased by 0.1%, and the number of parameters is reduced by 0.8 M, which proves the effectiveness of the MTC module.

As shown in Table 2, under the X-sub benchmark of NTU-RGB+D dataset, the comparison of model accuracy and parameter number under different combinations of the three topology modeling modules (MOD1, MOD2, MOD3) is carried out. Seeing from the table, when MOD(1,2,3) is used, the model accuracy is the highest, which shows the effectiveness of multi-scale modeling.

As shown in Table 3, it shows the experimental results under the X-sub benchmark of NTU-RGB+D dataset.



**TABLE 2.** Comparisons of accuracy and model parameters after three modeling functions (MOD1,MOD2,M- OD3) are combined.

Models	Accuracy(%)	Params(M)
MOD(1,1,1)	89.9	1.00
MOD(2,2,2)	89.9	1.00
MOD(3,3,3)	89.7	0.99
MOD(1,1,2)	89.8	1.00
MOD(1,1,3)	89.8	1.00
<b>Ours(MOD(1,2,3))</b>	<b>90.0</b>	<b>1.00</b>

**TABLE 3.** Comparison of accuracy and parameter quantity.

Models	Accuracy(%)	Params(M)
baseline	89.4	2.09
+2MATRs	89.5(0.1↑)	2.06(0.03↓)
+5MATRs	89.6(0.2↑)	1.96(0.13↓)
+8MATRs	89.7(0.3↑)	1.55(0.54↓)
<b>Ours</b>	<b>90.0(0.6↑)</b>	<b>1.00(1.09↓)</b>
w/o EdgeConv	89.7(0.3↓)	0.95
w/o CAT	90.0	1.88(0.88↑)

Comparison of accuracy and parameter quantity when the spatial-temporal graph convolutional layer in the baseline was gradually replaced by the MATR layer and the EdgeConv or CAT was removed.

**TABLE 4.** Comparison of validation accuracy of model a for different data modalities.

Data mode	Accuracy(%)
Joint	94.2
Joint motion	90.3
Bone	94.2
Bone motion	93.5
<b>fusion</b>	<b>96.3</b>

We gradually replace the spatial-temporal modeling module in the baseline with MATRs (shown in the dashed box in Fig. 1), and when all are replaced with our MATR basic block, the accuracy is the highest (0.6% improvement), and the parameter number is reduced to half of the original (1.09 M reduction), which verifies the effectiveness and lightweight of our model MATR-GCN. After that, we verify the influence of EdgeConv and channel concatenation (CAT) by either removing EdgeConv from the MATR-GCN model or replacing the CAT method with the element addition method. We can see that the accuracy of MATR-GCN w/o EdgeConv is decreased by 0.3% compared with MATR-GCN, while the number of parameters is only reduced by 0.05 M, which shows that EdgeConv effectively enhances the model's discrimination of similar actions when increasing the number of lower parameters. The number of parameters of MATR-GCN w/o CAT is increased by 0.88 M compared with MATR-GCN, which shows that CAT can effectively reduce the number of parameters of the model.

## 2) THE IMPACT OF MULTI-STREAM EXPERIMENTS

Multi-stream structure [7], [11], [23] is often used in the field of skeleton-based action recognition, which can significantly improve the recognition performance of the model. In this section, we evaluate the importance of the multi-stream structure in the model. We show the recognition performance of the model with four streams, mainly joint, bone, joint motion, and bone motion. The same model is used for all

**TABLE 5.** Comparisons of the recognition accuracy with the state-of-the-art methods on the NTU-RGB+D dataset.

Methods	NTU RGB+D		Params(M)
	X-sub(%)	X-view(%)	
ST-GCN[6]	81.5	88.3	3.10
2s-AGCN[7]	88.5	95.1	6.94
DCA-SGIN[43]	87.2	88.7	—
AS-GCN[22]	86.8	94.2	9.50
MS-AAGCN[23]	90.0	96.2	15.04
DC-GCN[41]	91.1	96.7	—
FLAGCN[37]	89.4	94.8	—
CTR-GCN[24]	92.4	96.8	1.44
DGNN[8]	89.9	96.1	26.24
Dynamic GCN[10]	91.5	96.0	14.4
MSSTNet[39]	89.6	95.3	39.6
2s-ICE-GCN[40]	92.0	96.2	—
<b>Ours</b>	<b>92.5</b>	<b>96.9</b>	<b>1.00</b>

**TABLE 6.** Comparisons of the recognition accuracy with the state-of-the-art methods on the NTU-RGB+D 120 dataset.

Methods	NTU RGB+D 120		Year
	X-sub(%)	X-set(%)	
ST-GCN[6]	70.7	73.2	2018
2s-AGCN[7]	82.5	84.2	2019
AS-GCN[22]	77.9	78.5	2019
Dynamic GCN[10]	87.3	88.6	2020
CTR-GCN[24]	88.9	90.6	2021
IA-ASGCN[42]	85.4	87.4	2022
DCA-SGIN[43]	87.2	88.7	2022
DC-GCN[41]	87.1	88.6	2023
MSSTNet[39]	85.3	86.0	2023
2s-ICE-GCN[40]	89.1	90.2	2023
DC-GCN[41]	87.1	88.6	2023
<b>Ours</b>	<b>88.9</b>	<b>90.6</b>	

**TABLE 7.** Comparisons of the recognition accuracy with the state-of-the-art methods on the NW-UCLA dataset.

Methods	Northwestern-UCLA		Year
	Accuracy(%)		
Ensemble TS-LSTM[29]	89.2		2017
2s-AGC-LSTM[30]	93.3		2019
2s-AGCN[24]	89.6		2019
DCA-SGIN[43]	95.8		2022
IA-ASGCN[42]	93.9		2022
MSSTNet[39]	95.3		2023
ACC-GCN[44]	96.1		2023
<b>Ours</b>	<b>96.3</b>		

flows, and the final prediction result is generated by fusing the weighted sum scores of the four flows, as shown in Fig. 7.

In Table 4, we show the results of multi-stream fusion using the best configuration formulated in the model of MATR-GCN. As shown in the table, the recognition accuracy varies for each flow. Therefore, we utilize the weighted sum to fuse the results of the four streams. According to the table, assigning a high weight to the joint motion flow will result in performance loss because its contribution to the whole multi-flow structure is less than the contribution of the joint and bone flows.

## D. VISUALIZATION

Our dataset, the adjacency matrix, and the visualization of the experimental results are shown in Fig. 8 and Fig. 9. Fig. 8(a) and (b) show the visualization of one sample in the NTU RGB+D 120 dataset and the Northwest-UCLA dataset,

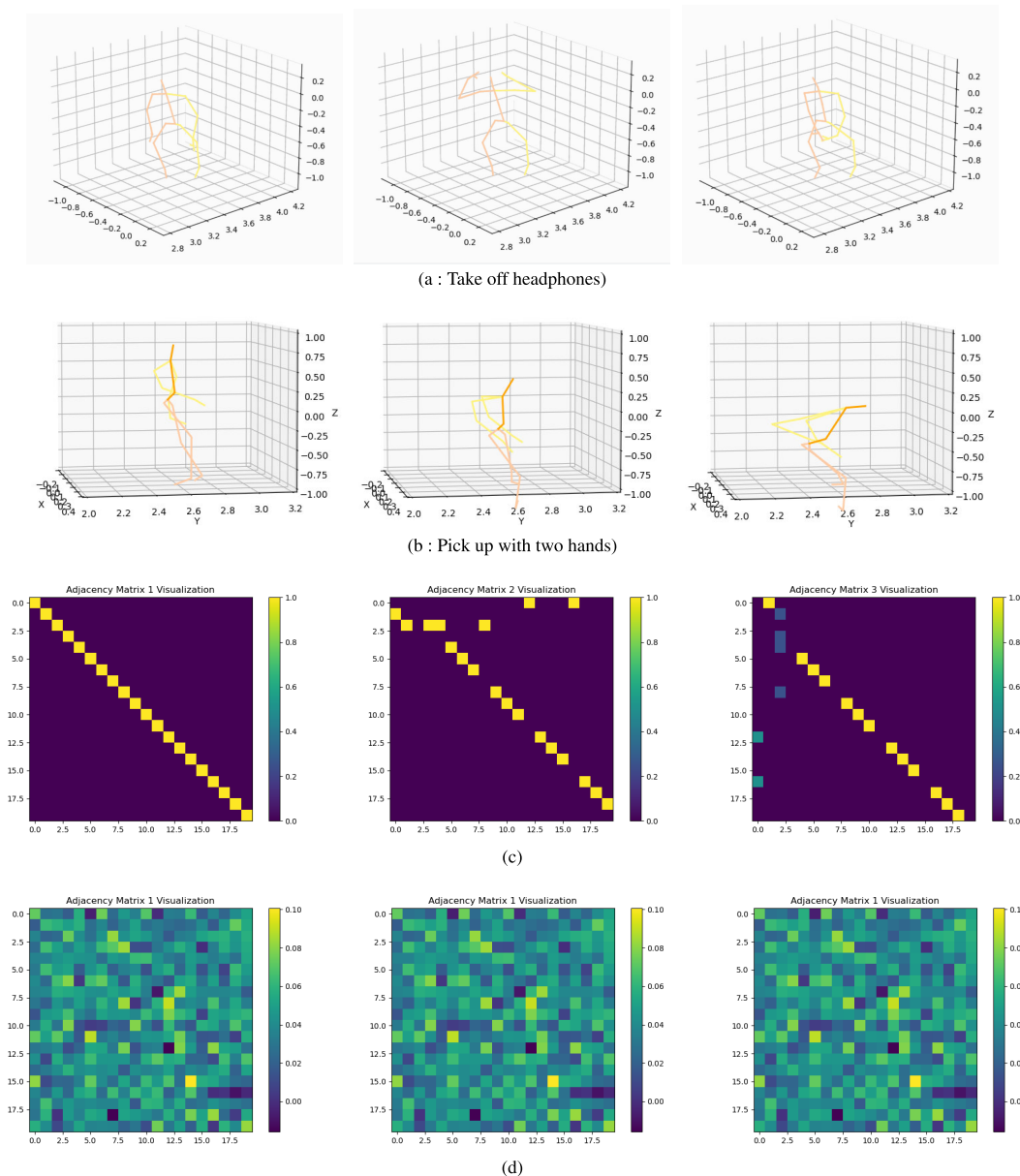


FIGURE 8. Visualization of the dataset and adjacency matrix.

respectively. Fig. 8(c) and (d) show the visualization results using fixed topology and using learnable topology on the NTU RGB+D 60 dataset, respectively, and the comparison shows that our learnable adjacency matrix improves the physically defined topology, thus producing richer features. In addition, the three channel topologies shown in Fig. 8(d) prove that our method can learn unique topologies based on different data from different channels. Fig. 9(a) and (b) show the confusion matrix of our model under NTU RGB+D 60 dataset and Northwest-UCLA dataset, respectively. Moreover, the results prove the superior recognition performance of our model. The red boxes in Fig. 9(a) are two pairs of action categories that are easily confused (reading and writing, putting on shoes and taking off shoes). In Fig. 9(b), ‘moving around’ and ‘carrying’ are easily recognized as ‘throwing’.

### E. COMPARISON WITH OTHER METHODS

To demonstrate the effectiveness of our model, we make a comparison with state-of-art methods on three public datasets: NTU RGB+D 60, NTU RGB+D 120, and Northwest-UCLA. And we do comparisons of model parameters on NTU RGB+D dataset, which proves the lightweight of our model. The compared models are GCN-based methods including [6], [7], [8] [10] [22], [23], [24] [37] [40], [41], [42], [43], and [44], CNN-based method including [39], and RNN-based methods including [30] and [29]. For a comprehensive comparison, the metrics reported by EfficientGCN [45] are used for the evaluation, and the results are shown in Tables 5, 6 and 7.

As shown in Table 5, on the X-sub and X-view benchmarks of the NTU-RGB+D 60 dataset, our model achieves a

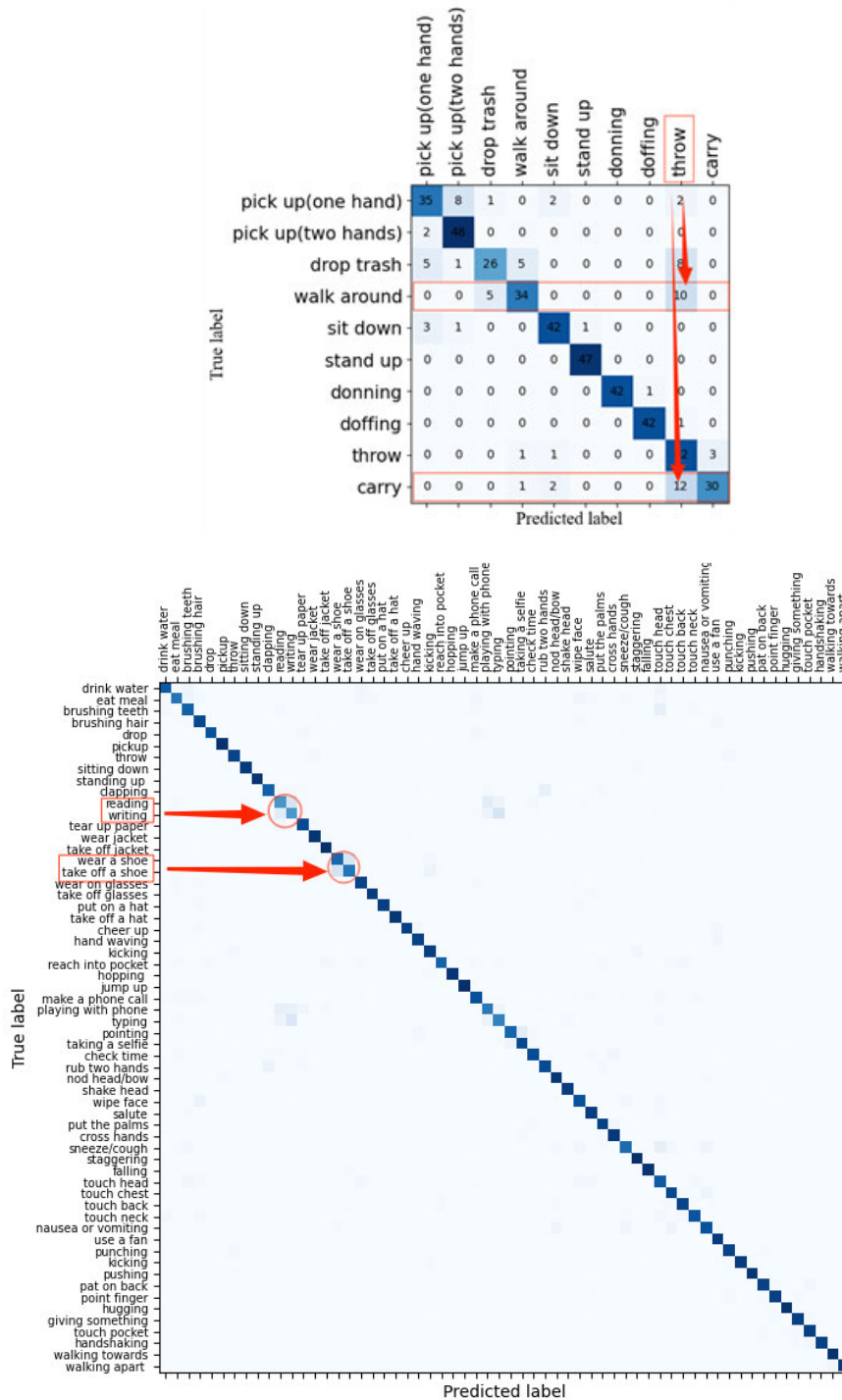


FIGURE 9. Confusion matrix visualization.

significant improvement in recognition accuracy compared with GCN and CNN based methods. For example, compared to MSSTNet [39], our model improves by 2.9% and 1.6% on X-sub and X-view benchmarks, respectively. In GCN-based methods, ST-GCN [6] uses a manually defined topology and is fixed on all layers. On this basis, 2s-AGCN [7] and AS-GCN [22] improve the topology structure by designing embedding functions and encoder-decoder.

Compared with ST-GCN, 2s-AGCN and AS-GCN, our model achieves improvements of 11%, 5% and 5.7% on the X-sub benchmark and 8.6%, 1.8% and 2.7% on the X-view benchmark, respectively. In terms of the number of model parameters, compared with the advanced model CTR-GCN [24], the number of parameters of our model is reduced by 0.44 M, while achieving comparable performance.

Table 6 shows the comparison results of our model on the two benchmarks X-sub and X-set of the NTU-RGB+D 120 dataset, which are 88.9% and 90.6%, respectively. The results show that the performance of our model is better than most methods, which is 0.4% higher than 2s-ICE-GCN [40] under X-set benchmark, and only 0.2% lower than 2s-ICE-GCN [40] under X-sub benchmark. For the CNN-based method, under the X-sub and X-set benchmarks, the recognition accuracy of our model is 3.6% and 4.6% higher than that of the MSSTNet [39] method, respectively. The comparison results demonstrate the superior classification performance of our model on large-scale datasets.

As shown in Table 7, our model exhibits superior recognition performance on the Northwest-UCLA dataset. Compared with RNN-based methods, our recognition accuracy is 7.1% and 3% higher than the Ensemble TS-LSTM [29] and 2s-AGC-LSTM [30], respectively. For the CNN-based method, the recognition accuracy of our model is 1% higher than the MSSTNet [39]. Among the GCN-based methods, our model improves the recognition accuracy by 6.7%, 0.5%, 2.4% and 0.2%, respectively, compared with 2s-AGCN [7], DCA-SGIN [43], IA-ASGCN [42] and ACC-GCN [44]. The comparison results of three datasets demonstrate that our model has superior recognition performance and strong generalization on datasets of various scales.

## V. CONCLUSION

For accurate skeleton-based action recognition, a multi-scale adaptive graph convolutional model (MATR-GCN) is designed in this paper. In MATR-GCN, a multi-scale dynamic topology modeling module is preset to concatenate the channel dimensions of three Adaptive Graph Convolution (MA-GC) branches and one edge convolution branch, which greatly reduces the number of model parameters and running time. The multi-scale temporal convolution module is used to increase the time dimension receptive field. Moreover, random initialization coefficient matrix as the original adjacency matrix input model, and adaptively learn the connection relationship between nodes in the data through three topological modeling functions of MA-GC to enrich the topology structure of the adjacency matrix. Extensive experiment results show that the proposed MATR-GCN performs better than the up-to-date methods in terms of on three datasets: NTU RGB+D, NTU RGB+D 120 and Northwestern UCLA for accuracy and number of parameters.

However, limited by the number of skeleton joints, the recognition of some subtle behavioral actions is still a challenging task. In the future, we will consider using speech text generation method to assist model training, and design a loss function to supervise and regulate the model training process to enhance the model recognition performance.

## REFERENCES

- [1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 1–43, 2011.
- [2] F. Rezazadegan, S. Shirazi, B. Upcroft, and M. Milford, "Action recognition: From static datasets to moving robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3185–3191.
- [3] T. Shu, X. Gao, M. S. Ryoo, and S.-C. Zhu, "Learning social affordance grammar from videos: Transferring human interactions to human-robot interactions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1669–1676.
- [4] T. Saba, A. Rehman, R. Latif, S. M. Fati, M. Raza, and M. Sharif, "Suspicious activity recognition using proposed deep L4-Branching-Actionnet with entropy coded ant colony system optimization," *IEEE Access*, vol. 9, pp. 89181–89197, 2021.
- [5] M. T. Ubaid, T. Saba, H. U. Draz, A. Rehman, M. U. Ghani khan, and H. Kolivand, "Intelligent traffic signal automation based on computer vision techniques using deep learning," *IT Prof.*, vol. 24, no. 1, pp. 27–33, Jan. 2022.
- [6] M. Jiang, J. Dong, D. Ma, J. Sun, J. He, and L. Lang, "Inception spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. Int. Symp. Control Eng. Robot. (ISICER)*, Feb. 2022, pp. 208–213.
- [7] 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. 12026–12035.
- [8] 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.
- [9] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," 2018, *arXiv:1805.07694*.
- [10] 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.
- [11] W. Li, X. Liu, Z. Liu, F. Du, and Q. Zou, "Skeleton-based action recognition using multi-scale and multi-stream improved graph convolutional network," *IEEE Access*, vol. 8, pp. 144529–144542, 2020.
- [12] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1112–1121.
- [13] 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.
- [14] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, Oct. 2020.
- [15] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2649–2656.
- [16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
- [17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *Stat.*, vol. 1050, no. 20, 2017, Art. no. 48550.
- [18] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1025–1035.
- [19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
- [20] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "Graph2vec: Learning distributed representations of graphs," 2017, *arXiv:1707.05005*.
- [21] Q. Huang, F. Zhou, and J. He, "Spatial temporal graph attention networks for skeleton-based action recognition," *J Electron Imag.*, vol. 29, no. 5, 2020, Art. no. 053003.
- [22] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3595–3603.
- [23] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Trans. Image Process.*, vol. 29, pp. 9532–9545, 2020.

- [24] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Jun. 2021, pp. 13359–13368.
- [25] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1110–1118.
- [26] W. Zhu, C. Lan, and J. Xing, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 30, no. 1, 2016, pp. 3697–3703.
- [27] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1963–1978, Aug. 2019.
- [28] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 816–833.
- [29] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1012–1020.
- [30] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1227–1236.
- [31] T. S. Kim and A. Reiter, "Interpretable 3D human action analysis with temporal convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1623–1631.
- [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," 2018, *arXiv:1804.06055*.
- [33] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3D action recognition," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2842–2855, Jun. 2018.
- [34] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3288–3297.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [36] W. Wang, W. Xie, Z. Tu, W. Li, and L. Jin, "Multi-part adaptive graph convolutional network for skeleton-based action recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Padua, Italy, Jul. 2022, pp. 1–7.
- [37] Y. Jiang, X. Yang, and J. Liu, "A lightweight hierarchical model with frame-level joints adaptive graph convolution for skeleton-based action recognition," *Secur. Commun. Netw.*, vol. 2021, pp. 1–13, Sep. 2021.
- [38] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 143–152.
- [39] Q. Cheng, J. Cheng, and Z. Ren., "Multi-scale spatial-temporal convolutional neural network for skeleton-based action recognition," *Pattern Anal. Appl.*, vol. 26, pp. 1303–1315, May 2023.
- [40] S. Wang, J. Pan, B. Huang, P. Liu, Z. Li, and C. Zhou, "ICE-GCN: An interactional channel excitation-enhanced graph convolutional network for skeleton-based action recognition," *Mach. Vis. Appl.*, vol. 34, no. 3, p. 40, May 2023.
- [41] H. Zhou, X. Xiang, Y. Qiu, and X. Liu, "Graph convolutional network with STC attention and adaptive normalization for skeleton-based action recognition," *Imag. Sci. J.*, vol. 71, no. 7, pp. 636–646, Oct. 2023.
- [42] Y. Zhao, J. Wang, H. Wang, M. Liu, and Y. Ma, "Adaptive spatiotemporal graph convolutional network with intermediate aggregation of multi-stream skeleton features for action recognition," *Neurocomputing*, vol. 505, pp. 116–124, Sep. 2022.
- [43] K. Wu and X. Gong, "Dynamic channel-aware subgraph interactive networks for skeleton-based action recognition," *IEEE Signal Process. Lett.*, vol. 29, pp. 2592–2596, 2022.
- [44] T. Alsarhan, O. Harfoushi, A. Y. Shdefat, N. Mostafa, M. Alshinwan, and A. Ali, "Improved graph convolutional network with enriched graph topology representation for skeleton-based action recognition," *Electronics*, vol. 12, no. 4, p. 879, Feb. 2023.
- [45] 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.



**HUANGSHUI HU** (Member, IEEE) received the Ph.D. degree in computer application technology from Jilin University, China, in 2012. He is currently a Professor with the College of Computer Science and Engineering, Changchun University of Technology, China. His research interests include topology control in wireless sensor networks and multifunction vehicle bus networks.



**YUE FANG** received the B.S. degree from the Changchun University of Technology, China, in 2021, where she is currently pursuing the master's degree.

Her research interests include speech emotion recognition and action recognition.



**MEI HAN** received the B.S. degree from the Changchun University of Technology, China, in 2021, where she is currently pursuing the master's degree.

Her research interests include speech emotion recognition and action recognition.



**XINGSHUO QI** received the B.S. degree from the Changchun University of Technology, China, in 2021, where he is currently pursuing the master's degree.

His research interests include speech emotion recognition and facial expression recognition.

• • •