Temporal Decoupling Graph Convolutional Network for Skeleton-Based Gesture Recognition

Jinfu Liu, Xinshun Wang, Can Wang, Yuan Gao, and Mengyuan Liu, *Member, IEEE*

***Abstract*—Skeleton-based gesture recognition methods have achieved high success using Graph Convolutional Network (GCN), which commonly uses an adjacency matrix to model the spatial topology of skeletons. However, previous methods use the same adjacency matrix for skeletons from different frames, which limits the flexibility of GCN to model temporal information. To solve this problem, we propose a Temporal Decoupling Graph Convolutional Network (TD-GCN), which applies different adjacency matrices for skeletons from different frames. The main steps of each convolution layer in our proposed TD-GCN are as follows. To extract deep spatiotemporal information from skeleton joints, we first extract high-level spatiotemporal features from skeleton data. Then, channel-dependent and temporal-dependent adjacency matrices corresponding to different channels and frames are calculated to capture the spatiotemporal dependencies between skeleton joints. Finally, to fuse topology information from neighbor skeleton joints, spatiotemporal features of skeleton joints are fused based on channel-dependent and temporal-dependent adjacency matrices. To the best of our knowledge, we are the first to use temporal-dependent adjacency matrices for temporal-sensitive topology learning from skeleton joints. The proposed TD-GCN effectively improves the modeling ability of GCN and achieves state-of-the-art results on gesture datasets including SHREC'17 Track and DHG-14/28.**

***Index Terms*—Graph convolutional network, gesture recognition, skeleton sequence.**

Manuscript received 25 October 2022; revised 21 February 2023 and 19 April 2023; accepted 19 April 2023. Date of publication 1 May 2023; date of current version 18 January 2024. This work was supported by the National Natural Science Foundation of China under Grant 62203476. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. Guoying Zhao. *(Corresponding author: Mengyuan Liu.)*

Jinfu Liu is with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510275, China, and also with the Hangzhou GOTHEN Technology Co., Ltd., Hangzhou 310000, China (e-mail: liujf69@mail2.sysu.edu.cn).

Xinshun Wang is with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510275, China (e-mail: wangxsh36@mail2.sysu.edu.cn).

Can Wang is with the Multimedia Information Processing Laboratory, Department of Computer Science, Kiel University, 24118 Kiel, Germany, also with the Advanced Institute of Information Technology, Peking University, Beijing 100871, China, and also with the Hangzhou Linxrobot Co., Ltd., Hangzhou 310003, China (e-mail: wangcan@linxrobot.com).

Yuan Gao is with the Faculty of Information Technology and Communication Sciences (ITC), Tampere University, 33100 Tampere, Finland (e-mail: yuan.gao@tuni.fi).

Mengyuan Liu is with the Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Beijing 100871, China (e-mail: nkliuyifang@gmail.com).

Our code is available at: https://github.com/liujf69/TD-GCN-Gesture.

Digital Object Identifier 10.1109/TMM.2023.3271811

I. Introduction

SKELETON-BASED gesture recognition has applications in research fields including sign language communication [1], [2], robot control [3] and virtual reality [4], [5]. Moreover, skeleton-based gesture recognition methods can be generalized to solve the skeleton-based action recognition task, which has potential applications in content-based video retrieval [6], education [7] and human-computer interaction [8]. Actually, skeleton-based gesture recognition and action recognition are different in some respects, such as the degrees of freedom, execution subject, range of motion and temporal dependency. It is worth emphasizing that gesture recognition is more pronounced in terms of temporal dependency than action recognition. Since gesture recognition has high requirements in terms of degrees of freedom and temporal dependency, it makes the recognition task difficult and more distinct from action recognition. The human skeleton whether it represents one hand or the whole body is a natural topological graph, where joints and bones connecting joints can be respectively treated as vertices and edges of the graph. Due to the unique advantages of Graph Convolutional Networks (GCNs) in modeling graph-structured data, many GCN-based methods [9], [10], [11], [12], [13], [14], [15] have been proposed for the skeleton-based gesture and action recognition. The general pipeline of GCN can be divided into three stages. The first stage is adjacency matrix construction, which calculates the dependency matrices between joints. The second stage is high-level feature extraction, which extracts high-level features from raw skeleton data using deep neural networks. The last stage is information fusion, which fuses and updates joint information by multiplying the high-level features and dependency matrices.

Currently, how to construct adjacency matrices remains an open problem. One direct way is to define a fixed adjacency matrix according to the physical connection of joints (Fig. 1(a)). In ST-GCN [14], a learnable mask is additionally used to multiply with a predefined adjacency matrix to construct an edge-attention adjacency matrix (Fig. 1(b)). Moreover, Chen et al. [16] proposed a Channel-wise Topology Refinement Graph Convolution Network (CTR-GCN), focusing on constructing channel-specific adjacency matrices (Fig. 1(c)). Besides, Ye et al. [17] proposed a Dynamic GCN to learn the dependencies between two joints by incorporating the contextual features of the remaining joints.

However, existing methods use shared adjacency matrices for different temporal frames, which ignores the fact that

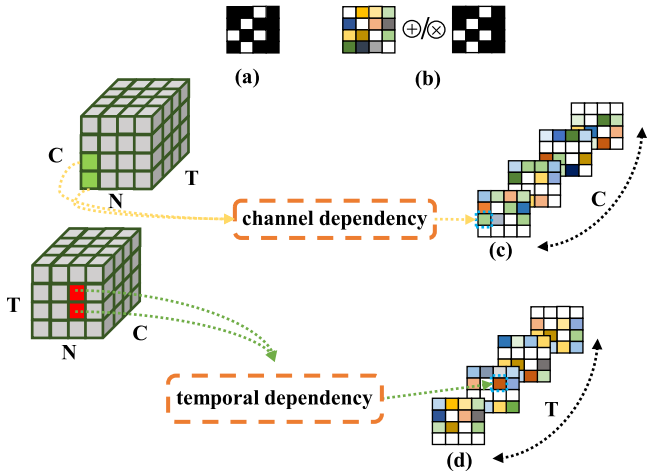

Fig. 1. Comparison of different adjacency matrix construction methods. (a) Previous pre-defined adjacency matrix. (b) Previous learnable mask multiplied or added with a pre-defined adjacency matrix. (c) Previous channel-dependent adjacency matrix. (d) Our proposed temporal-dependent adjacency matrix. Noting that our temporal decoupling graph convolution (TD-GC) involves both channel-wise and temporal-wise adjacency matrices.

dependencies between joints vary over time. Take the gesture "tap on the keyboard with the right hand" as an example. At the beginning and the ending stages, the whole joints on the hand keep nearly still, and the implicit dependencies of joints located on different fingers are quite weak. While, in the mid-term stage, five fingers cooperate to interact with the keyboard, and the implicit dependencies of joints located on different fingers become more obvious. In fact, the dependencies between joints in different frames are usually determined by the joint information of each frame. The joint information in two frames that are far apart is usually different, so it is vital and meaningful to design adjacency matrices for different temporal frames according to the hand joint information of different frames.

Inspired by this observation, we present a Temporal Decoupling Graph Convolution (TD-GC) to compute temporal-dependent adjacency matrices for joints in different frames, which is shown in Fig. 1(d). The main steps of TD-GC are as follows. First, we use a module of Coupling Feature Learning (CFL) to extract high-level spatiotemporal features of skeleton sequences. Then, a pre-defined adjacency matrix A based on the prior knowledge method is introduced, and the channel-dependent and temporal-dependent adjacency matrices are calculated by using the modules of Channel Decoupling Learning (CDL) and Temporal Decoupling Learning (TDL) respectively. Finally, a module of Temporal-Channel Fusion (TCF) is performed on the extracted high-level spatiotemporal features with channel-dependent and temporal-dependent adjacency matrices, and the results are fused to update the features of skeleton vertices. Based on the TD-GC, we propose Temporal Decoupling Graph Convolution Network (TD-GCN) for skeleton-based gesture recognition.

Our contributions can be summarized as follows:

- Compared with GCN-based gesture recognition methods which use the same adjacency matrix across different skeleton frames, we use temporal-dependent adjacency matrices for temporal-sensitive topology learning.
- We present a Temporal Decoupling Graph Convolution Network (TD-GCN) for spatial-temporal topology learning. TD-GCN is implemented by staking multiple Temporal Decoupling Graph Convolution (TD-GC) layers, where both temporal-dependent and channel-dependent adjacency matrices are formulated to model temporal and spatial topology information.
- Extensive experiments on benchmark SHREC'17 Track, DHG-14/28, NTU RGB+D, and NW-UCLA datasets verify the effect of our TD-GCN by outperforming existing GCN-based methods. Compared with the current state-of-the-art method, namely, CTR-GCN, our TD-GCN is more flexible to model the temporal-dependent topology information of joints by using fewer parameters.

II. Related Work

For skeleton-based gesture recognition, deep learning-based methods have been proven more effective than those methods based on hand-crafted features [18], [19]. We will first introduce related works from both DNN-based methods and GCN-based methods. After that, we will introduce topology learning and temporal modeling.

A. DNN-Based Methods

For skeleton-based gesture recognition, DNN-based methods are usually based on Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). RNN usually uses the output of the previous moment as the input of the current moment to form a recursive connection structure, which has unique advantages in processing sequence data. CNN has unique advantages in processing Euclidean data (such as images), which usually convert skeleton sequences into 2D pseudo-images. Some previous works have used RNNs and CNNs to achieve effective results in skeleton-based gesture recognition. For example, Nunez et al. [20] use a combination of a CNN and a Long Short-Term Memory (LSTM) recurrent network for skeleton-based gesture recognition, and propose a data augmentation method to solve the overfitting problem. In fact, skeleton-based gesture recognition and action recognition are closely related. The methods between the two domains can be shared in many cases, while the robustness and generalization of methods can be demonstrated. Similarly, some researchers apply RNNs [21], [22], [23], [24], [25], [26], [27], [28], and CNNs [29], [30], [31], [32], [33], [34] to skeleton-based action recognition. For example, aiming to solve the problem that conventional RNNs tend to ignore the spatial structure of joints, Wang et al. [27] proposed a two-stream recurrent neural network to model the temporal dynamics and spatial structure of skeleton sequences, which breaks through the limitations of RNN in processing raw skeleton data to a certain extent. To address the adverse effects due to view variations and noisy data, Liu et al. [34] visualized skeleton sequences as color images and fed them into a multi-stream CNN for deep feature extraction. For skeleton-based gesture recognition, the

above DNN-based methods achieve better results than methods based on hand-crafted features. But neither the RNN-based methods nor the CNN-based methods could simulate the dependencies between human joints, which restricts the improvement of recognition accuracy.

B. GCN-Based Methods

GCNs are networks that can effectively process structured data, especially graph data. Researchers [10], [11], [12], [13], [14], [15], [35], [36] apply GCNs to skeleton-based gesture recognition and action recognition. For example, Song et al. [11] propose a novel multi-stream improved spatiotemporal graph convolutional network (MS-ISTGCN) for skeleton-based dynamic gesture recognition, which adopts an adaptive spatial graph convolution to learn the relationship between distant hand joints and propose an extended temporal graph convolution to extract informative temporal features from short to long periods. However, the joints of the MS-ISTGCN in different frames share the same adjacency matrix. Yan et al. [14] proposed a Spatial-Temporal Graph Convolutional Network (ST-GCN), which treats joints as vertices and predefines edges representing dependencies based on prior knowledge and learns spatiotemporal features from skeleton data effectively. But the adjacency matrix of ST-GCN is predefined and has the problem of keeping it fixed in both the training and testing phases. To better exploit the first-order and the second-order information of the human skeleton, Shi et al. [13] proposed a Two-Stream Adaptive Graph Convolutional Network (2s-AGCN), but the 2s-AGCN also has the problem that joints in different channels and frames share the same adjacency matrix. Peng et al. [37] proposed a Neural Architecture Search Graph Convolutional Network (NAS-GCN) with the help of neural architecture search, which uses an adaptive way to build a pure learnable correlation matrix. But these methods that do not use prior knowledge may need to spend much time and choose an appropriate strategy to learn the correlation matrix between joints, and the network may converge slowly. For the GCN-based methods mentioned above, their graph convolutions all have the problem that the skeletons of different frames share the same adjacency matrix, which restricts the modeling ability of GCN to a certain extent.

C. Topology Learning

Many methods [16], [17], [35] have shown those dynamic and topology-non-shared methods are more effective than static and topology-shared methods in skeleton-based gesture and action recognition. For example, Cheng et al. [35] proposed a DeCoupling GCN (DC-GCN) which sets different parameterized topologies for different channel groups. However, the DC-GCN faces the problem of optimization caused by excessive parameters when setting channel-wise topologies. Chen et al. [16] proposed a Channel-wise Topology Refinement Graph Convolutional Network (CTR-GCN), which allows skeletons of different channels to have different adjacency matrices. Similarly, the CTR-GCN also suffers from the problem that joints in different frames share the same topology. And it is worth emphasizing that our proposed TD-GCN is a dynamic and topology-non-shared method, which uses different adjacency matrices in different frames and channels.

D. Temporal Modeling

The analysis of the "tap" gesture in the Introduction section above shows that the temporal information of skeleton joints is very important. In the field of skeleton-based gesture recognition, some previous works have used temporal information. For example, Shi et al. [38] proposed a decoupled spatial-temporal attention network (DSTA-Net) for skeleton-based gesture recognition, which involves temporal information in calculating attention maps based on an attention mechanism. Plizzari et al. [39] proposed a novel Spatial–Temporal Transformer network (ST-TR) that uses a Temporal Self-Attention module (TSA) to model inter-frame correlations based on Transformer. Besides, Zhang et al. [40] proposed a Spatial-Temporal Specialized Transformer (STST) for skeleton-based action recognition, which designs a Directional Temporal Transformer Block for modeling skeleton sequences in temporal dimensions. The three methods introduced above all use temporal information for calculating the attention map, which is different from our proposed TD-GCN. Our TD-GCN is the first to introduce temporal dependency modeling for formulating an adjacency matrix, which improves the temporal modeling ability of GCN.

III. Method

In this section, we first define related notations and formulate conventional graph convolution. Then we elaborate on our Temporal Decoupling Graph Convolution (TD-GC) and mathematically analyze the representation capability of TD-GC. Furthermore, we compare the difference between the proposed TD-GC and other graph convolutions. Finally, we introduce the structure of our Temporal Decoupling Graph Convolutional Network (TD-GCN).

A. Graph Convolution

The hand skeleton sequence is a natural topological graph in which joints and bones can be represented as vertices and edges of the graph respectively. The graph is denoted as $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$, where $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N\}$ is the set of N joints and $\boldsymbol{E}$ is the set of bones in the skeleton. For 3D skeleton data, the joint $\boldsymbol{v}_i$ is denoted as $\boldsymbol{V} = \{\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i\}$ where $\boldsymbol{x}_i$, $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$ are features of $\boldsymbol{v}_i$. In the following, we denote the feature set of N vertices by $\boldsymbol{X}$, where $\boldsymbol{X} \in \boldsymbol{R}^{N \times C}$ is a matrix. The edge set $\boldsymbol{E}$ is formulated as an adjacency matrix $\boldsymbol{A} \in \boldsymbol{R}^{N \times N}$ and its element $\boldsymbol{a}_{ij}$ reflects the dependency between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$.

Below we define four modalities of skeleton data. Given two joints data $\boldsymbol{v}_i = \{\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i\}$ and $\boldsymbol{v}_j = \{\boldsymbol{x}_j, \boldsymbol{y}_j, \boldsymbol{z}_j\}$, a bone data of the skeleton is defined as a vector $\boldsymbol{e}_{\boldsymbol{v}_i, \boldsymbol{v}_j} = (\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{y}_i - \boldsymbol{y}_j, \boldsymbol{z}_i - \boldsymbol{z}_j)$. Given two joints data $\boldsymbol{v}_{ti}$, $\boldsymbol{v}_{(t+1)i}$ from two consecutive frames, the data of joint motion is defined as $\boldsymbol{m}_{ti} = \boldsymbol{v}_{(t+1)i} - \boldsymbol{v}_{ti}$. Similarly, given two bones data $\boldsymbol{e}_{\boldsymbol{v}_{(t+1)i}, \boldsymbol{v}_{(t+1)j}}$, $\boldsymbol{e}_{\boldsymbol{v}_{ti}, \boldsymbol{v}_{tj}}$ from two consecutive frames, the data of bone motion is defined as $\boldsymbol{m}_{\boldsymbol{v}_{ti}, \boldsymbol{v}_{tj}} = \boldsymbol{e}_{\boldsymbol{v}_{(t+1)i}, \boldsymbol{v}_{(t+1)j}} - \boldsymbol{e}_{\boldsymbol{v}_{ti}, \boldsymbol{v}_{tj}}$.

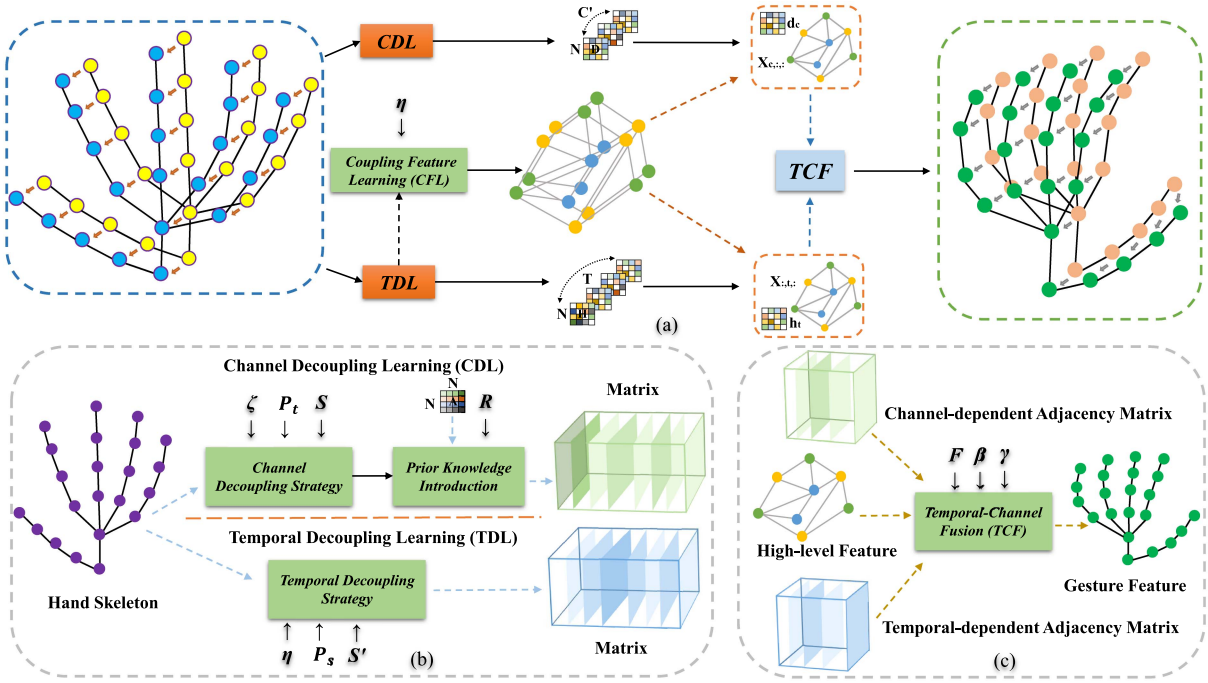

Fig. 2. Framework of our proposed Temporal Decoupling Graph Convolution. (a) The TD-GC uses the module of Coupling Feature Learning (CFL) to learn high-level features from skeleton sequences. (b) The module of Channel Decoupling Learning (CDL) computes the channel-dependent adjacency matrices and the module of Temporal Decoupling Learning (TDL) computes the temporal-dependent adjacency matrices. (c) The module of Temporal-Channel Fusion (TCF) is used to fuse and update high-level spatiotemporal features of joint vertices.

In the context of skeleton-based gesture recognition, a GCN is typically composed of graph convolution and temporal convolution. The normal graph convolution utilizes the weight $\boldsymbol{W}$ for aggregate features of $\boldsymbol{v}_i$'s neighbor vertices through $\boldsymbol{a}_{ij}$ to update its features $\boldsymbol{f}_i$, which is formulated as:

$$\boldsymbol{f}_i = \sum_{\boldsymbol{v}_j \in N(\boldsymbol{v}_i)} \boldsymbol{a}_{ij} \boldsymbol{x}_j \boldsymbol{W}. \tag{1}$$

There are static and dynamic methods to generate $\boldsymbol{a}_{ij}$. For static methods, $\boldsymbol{a}_{ij}$ is defined manually. For dynamic methods, $\boldsymbol{a}_{ij}$ is usually generated from the input data. However, in all previous methods, $\boldsymbol{a}_{ij}$ is shared among different frames.

B. Temporal Decoupling Graph Convolution

The framework of the proposed Temporal Decoupling Graph Convolution is shown in Fig. 2. In detail, TD-GC updates the features of N vertices by:

$$\boldsymbol{X}' = F\left[M(\boldsymbol{D}, \tilde{\boldsymbol{X}}), M(\boldsymbol{H}, \tilde{\boldsymbol{X}})\right], \tag{2}$$

where $\tilde{\boldsymbol{X}} = \eta(\boldsymbol{X}) \in \boldsymbol{R}^{C' \times T \times N}$ are high-level spatiotemporal features, which are obtained from input data $\boldsymbol{X} \in \boldsymbol{R}^{C \times T \times N}$ through a 1×1 convolution operation η of the Coupling Feature Learning (CFL) module. $\boldsymbol{D}$ and $\boldsymbol{H}$ are channel-dependent and frame-dependent adjacency matrices respectively, which represent associations between hand joints. F is a fusion function for aggregating joint features in both temporal and channel dimensions, which is formulated as:

$$F(\boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{y} \cdot \beta + \boldsymbol{z} \cdot \gamma. \tag{3}$$

In (3), we assign adaptive weights to channel features and temporal features via a multiplication operation $\cdot$. The symbols β and γ represent two learnable parameters, which will be optimized during the training process. M is a special matrix multiplication operation with a concatenate operation $\|$, and it is used to update the features of the skeleton by using the extracted high-level features and association matrices between hand joints. Given features $\tilde{\boldsymbol{X}} \in \boldsymbol{R}^{C' \times T \times N}$ and channel-dependent adjacency matrices $\boldsymbol{D} \in \boldsymbol{R}^{C' \times N \times N}$, $M(\boldsymbol{D}, \tilde{\boldsymbol{X}}) \in \boldsymbol{R}^{C' \times T \times N}$ is formulated as:

$$M(\boldsymbol{D}, \tilde{\boldsymbol{X}}) = [\tilde{\boldsymbol{x}}_{1,:,:}\, \boldsymbol{d}_1 \| \tilde{\boldsymbol{x}}_{2,:,:}\, \boldsymbol{d}_2 \| \cdots \| \tilde{\boldsymbol{x}}_{C',:,:}\, \boldsymbol{d}_{C'}], \tag{4}$$

where $\tilde{\boldsymbol{x}}_{c,:,:} \in \boldsymbol{R}^{T \times N}$ and $\boldsymbol{d}_c \in \boldsymbol{R}^{N \times N}$. Similarly, $M(\boldsymbol{H}, \tilde{\boldsymbol{X}}) \in \boldsymbol{R}^{C' \times T \times N}$ is formulated as:

$$M(\boldsymbol{H}, \tilde{\boldsymbol{X}}) = [\tilde{\boldsymbol{x}}_{:,1,:}\, \boldsymbol{h}_1 \| \tilde{\boldsymbol{x}}_{:,2,:}\, \boldsymbol{h}_2 \| \cdots \| \tilde{\boldsymbol{x}}_{:,T,:}\, \boldsymbol{h}_T], \tag{5}$$

where $\tilde{\boldsymbol{x}}_{:,t,:} \in \boldsymbol{R}^{C \times N}$, $\boldsymbol{H} \in \boldsymbol{R}^{T \times N \times N}$ and $\boldsymbol{h}_t \in \boldsymbol{R}^{N \times N}$. $M(\boldsymbol{D}, \tilde{\boldsymbol{X}})$ and $M(\boldsymbol{H}, \tilde{\boldsymbol{X}})$ update the feature information from the channel and temporal dimensions respectively, and different adjacency matrices are used for joints in different channels and different skeleton frames. We use a pre-defined adjacency matrix $\boldsymbol{A}$ to strengthen the local connections between joints, and channel-specific correlation matrix $\boldsymbol{Q}$ are used to refine the connections between joints and expand the receptive field of each

joint to the global. Therefore, the channel-dependent adjacency matrices $\boldsymbol{D}$ are obtained by refining the predefined topology $\boldsymbol{A}$ with channel-specific correlation matrix $\boldsymbol{Q}$ and a learnable parameter α:

$$\boldsymbol{D} = R(\boldsymbol{Q}, \boldsymbol{A}) = \boldsymbol{Q} \cdot \alpha + \boldsymbol{A}. \tag{6}$$

In the specific implementation of TD-GC, the channel-dependent adjacency matrices $\boldsymbol{D}$ are generated dynamically, and the matrices $\boldsymbol{A}$ and $\boldsymbol{Q}$ are continuously optimized and updated during the training process of the network. In (3) and (6), the symbols α, β and γ are all initialized as learnable parameters. And the channel-specific correlation matrix $\boldsymbol{Q}$ are obtained from the input data $\boldsymbol{X}$ by:

$$\boldsymbol{Q}_{i,j} = S\left\{P_t\left[\zeta(\boldsymbol{X}_i)\right], P_t\left[\zeta(\boldsymbol{X}_j)\right]\right\}, \tag{7}$$

where the ζ and P_t are implemented by convolution and global pooling operations. The ζ uses a reduction rate r to reduce the feature dimension and the calculations. We intend to construct the global connection between joints through the difference of features and improve the expressive ability of the connection through a nonlinear activation function. So the function S of (7) uses the input data to perform a self-pair-wise subtraction operation, which is formulated as:

$$S(\boldsymbol{y}, \boldsymbol{z}) = \delta\left\{\sigma\left[\epsilon_i(\boldsymbol{y}) - \epsilon_j(\boldsymbol{z})\right]\right\}, \tag{8}$$

where the symbols δ and σ represent a convolution function and a nonlinear activation function respectively. The ϵ_i and ϵ_j are used to expand the dimension of the input data. Similarly, the temporal-dependent adjacency matrices $\boldsymbol{H}$ are obtained from a function S' by:

$$\boldsymbol{H}_{i,j} = S'\left\{P_s\left[\eta(\boldsymbol{X}_i)\right], P_s\left[\eta(\boldsymbol{X}_j)\right]\right\}, \tag{9}$$

$$S'(\boldsymbol{y}, \boldsymbol{z}) = \sigma\left[\epsilon_i(\boldsymbol{y}) - \epsilon_j(\boldsymbol{z})\right]. \tag{10}$$

In (9) and (10), the symbols σ and P_s are also implemented by a nonlinear activation function and a global pooling operation respectively. The proposed TD-GC uses the difference between skeletons' information to construct the adjacency matrices for different frames and different channels so that the two skeleton graphs with larger differences have more different adjacency matrices, which can improve the modeling and feature representation ability of GCN.

From (7) to (10), ζ and δ are implemented by two 1×1 convolution operations which are used to extract high-level and typical features from complex raw skeletons data or low-level joints features. P_t and P_s represent global spatial pooling along channels and global temporal pooling along frames respectively, and their purposes are to reduce the dimensionality of skeletons data, compress the joints features, and reduce the computation complexity of GCN. σ is implemented to enhance the nonlinear expression capabilities of TD-GC by using a nonlinear activation function tanh. The purpose of ϵ_i and ϵ_j is to expand the dimension of the input data, therefore the difference of joint features can be calculated more conveniently.

Here, we propose TD-GC to make skeletons of different channels and different frames not share the same topology through channel-dependent and temporal-dependent adjacency matrices. Some scholars have proposed a CTR-GC layer, which is related to our work. Fig. 3 compares our proposed TD-GC layer and CTR-GC layer which shows that the CTR-GC layer is a special case of our proposed TD-GC layer. We both use convolution and pooling operations to extract high-level spatiotemporal features, and also emphasize the local connections of joints through a pre-defined adjacency matrix. However, when the CTR-GC layer extracts high-level features of different channels through convolution and pooling, it loses a lot of frame information in the process. Whereas our TD-GC preserves both channel and frame information and reconstructs the spatiotemporal information of skeletons through temporal-dependent and channel-dependent adjacency matrices. The most notable difference between the two graph convolutional layers is that our proposed TD-GC layer makes the skeletons of different frames have different topologies through the method of temporal decoupling, which improves the temporal modeling ability of skeleton-based gesture recognition.

C. Temporal Decoupling GCN

We combine TD-GC and temporal convolution to construct TD-GCN for the skeleton-based gesture recognition task. We construct the backbone as ST-GCN [14] and Fig. 4 shows the basic block of our TD-GCN. In terms of the graph convolution module, we fuse the outputs of three TD-GC layers, each of which is different when introducing a predefined matrix A by using prior knowledge. In terms of the temporal convolution module, we adopt a similar framework to GoogLeNet [41] and design a multi-branch temporal convolution. As shown in Fig. 4, the temporal convolution module contains three types of basic branches (i.e. A, B, and C), and the results of these branches are concatenated to obtain the output. Branch A is mainly composed of two convolutional layers, and the output of the first 1×1 convolution will be used as the input of the second $k\times1$ convolution with dilation d after normalization and nonlinear activation, where k represents the convolution kernel size. Branch B is a simple 1×1 convolutional layer with normalization. Branch C mainly includes a convolutional layer and a max-pooling layer. Similar to branch A, there are normalization and nonlinear activation between the convolutional layer and the max-pooling layer. We conduct ablation experiments on the multi-branch temporal convolution and select the best combination as the final temporal convolution module of our TD-GCN. Our TD-GCN has a total of eleven blocks, the first ten blocks are the basic block shown in Fig. 4, and the last block contains a global average pooling and a softmax classifier for predicting action classes. Moreover, for skeleton-based gesture recognition and action recognition, we follow recent methods [11], [12], [16], and use a multi-stream structure to fuse prediction scores, which has been shown to be more efficient than using a single stream.

IV. Experiments

A. Datasets and Settings

SHREC'17 Track dataset contains 2800 gesture sequences performed in two ways: using one finger and the whole hand.

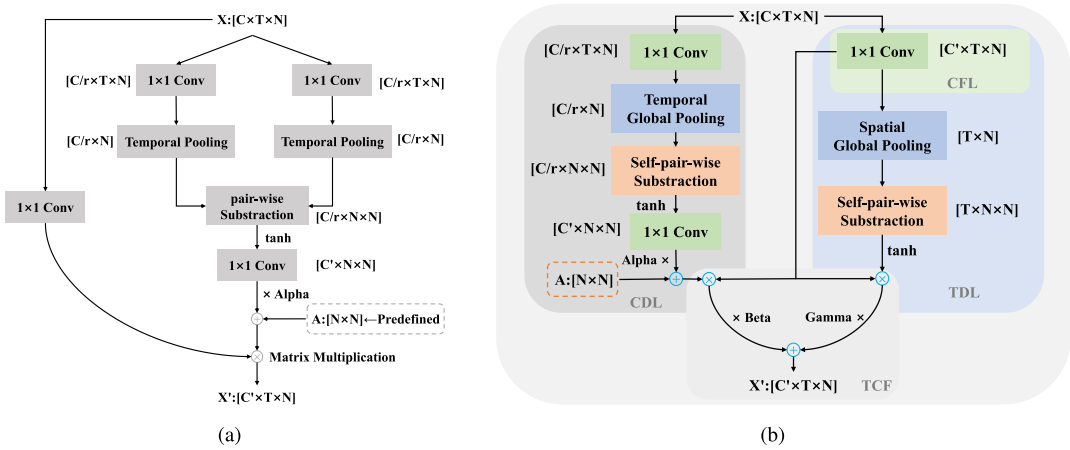

Fig. 3. Comparison between the pipeline of CTR-GC and our proposed TD-GC. (a) The pipeline of the CTR-GC layer. (b) The pipeline of our TD-GC layer.

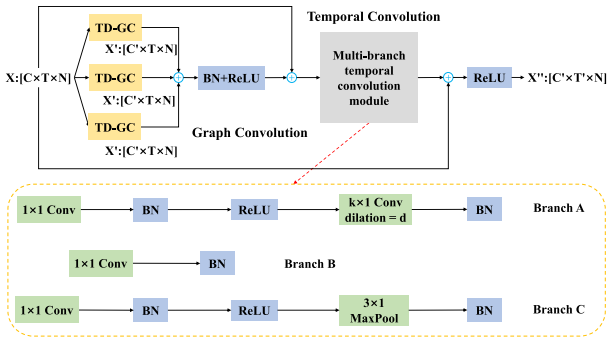

Fig. 4. Basic block of our TD-GCN. The graph convolution of the basic block combines the outputs of three TD-GC blocks, each of which predefines three different adjacency matrices A in the CDL module based on the physical connections of the human skeleton.

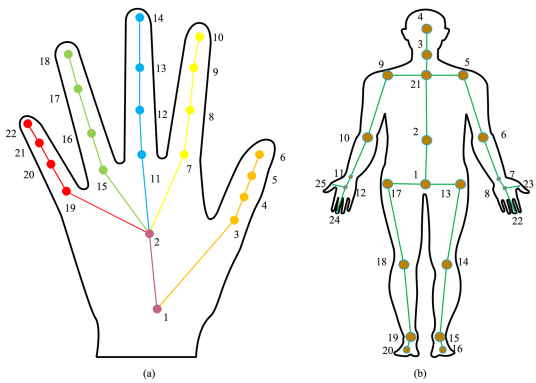

Fig. 5. Skeleton visualization. (a) The hand skeleton from the SHREC'17 Track dataset is captured by the Intel RealSense camera. (b) The body skeleton from the NTU-RGB+D dataset is captured by the Microsoft Kinect v2 sensor.

And each gesture is performed between 1 and 10 times by 28 participants. We follow the same evaluation protocol in [11], [42]: training data comes from 1960 sequences, and the remaining 840 sequences are used for testing. The gesture sequence can be labeled according to 14 or 28 classes, depending on the number of fingers used and the gesture represented. The recognition accuracy of the SHREC dataset will also be computed following 14 or 28 gesture classes. The hand skeleton of the SHREC'17 Track dataset contains 22 joints, which are shown in Fig. 5(a). **DHG-14/28** dataset is also collected by using the Intel RealSense camera, which contains 2800 sequences of 14 gestures performed 5 times by 20 participants. The DHG-14/28 dataset uses a leave-one-subject-out cross-validation strategy [18] for evaluation. In detail about this strategy, the skeleton data of 19 subjects are used for training and the remaining one subject is used for testing. So this evaluation strategy will be conducted 20 times, with the average of 20 times used as the final recognition accuracy. **NTU-RGB+D** is a widely used 3D action recognition dataset containing 56,880 skeleton sequences. The action samples are performed by 40 distinct subjects and categorized into 60 classes. The 3D skeleton data is captured by Kinect V2 through three cameras, which are shown in Fig. 5(b). The original paper [43] recommends two benchmarks: (1) cross-view (X-view): training data comes from camera $0°$ (view 2) and $45°$ (view 3), and testing data comes from camera - $45°$ (view 1). (2) cross-subject (X-sub): training data comes from 20 subjects, and the remaining 20 subjects are used for testing. **Northwestern-UCLA** (NW-UCLA) dataset contains 1494 video clips covering 10 categories and it is captured by three Kinect cameras simultaneously from multiple viewpoints. We follow the same evaluation protocol in [44]: training data from the first two cameras and samples from the other camera are used for testing.

We adopt ST-GCN [14] as the backbone. All experiments are conducted on one Tesla V100S-PCIE-32 GB GPU and one Nvidia GeForce RTX 3080TI GPU. We use SGD to train our TD-GCN model and we use a warm-up strategy [45] to make the training procedure more stable. We also set the learning rate to 0.1 and the momentum to 0.9. For the SHREC'17 Track and the DHG-14/28 dataset, the training ended in 150 epochs. We set the batch size to 32, the weight decay to 0.0001, and let the learning rate decay with a factor of 0.1 at epochs 90 and 130. Besides, we generate random numbers from $(-0.01, 0.01)$ to disturb the hand joint coordinates by applying random translation for data augmentation in the SHREC'17 Track and the DHG-14/28 dataset. For the above two datasets, we also extend

TABLE I
PERFORMANCES OF TD-GCN ON THE SHREC'17 TRACK, NW-UCLA AND NTU-RGB+D DATASETS WITHOUT USING DIFFERENT TOPOLOGIES

Method	Param.	Acc. (%)
TD-GCN w/o A (SHREC'17 Track)	1.34M	91.55
TD-GCN w/o Q (SHREC'17 Track)	1.20M	92.98
TD-GCN w/o D (SHREC'17 Track)	1.20M	91.90
TD-GCN w/o H (SHREC'17 Track)	1.36M	93.21
TD-GCN (SHREC'17 Track)	1.36M	93.57
TD-GCN w/o A (NW-UCLA)	1.33M	91.4
TD-GCN w/o Q (NW-UCLA)	1.19M	93.75
TD-GCN w/o D (NW-UCLA)	1.19M	94.40
TD-GCN w/o H (NW-UCLA)	1.35M	94.18
TD-GCN (NW-UCLA)	1.35M	94.82
TD-GCN w/o A (NTU-RGB+D)	1.35M	94.17
TD-GCN w/o Q (NTU-RGB+D)	1.21M	94.93
TD-GCN w/o D (NTU-RGB+D)	1.21M	93.66
TD-GCN w/o H (NTU-RGB+D)	1.37M	94.66
TD-GCN (NTU-RGB+D)	1.37M	95.00

TABLE II
PERFORMANCES OF TD-GCN ON THE SHREC'17 TRACK, NW-UCLA AND NTU-RGB+D DATASETS WITH DIFFERENT SETTINGS USING PRIOR KNOWLEDGE

Method	Param.	Acc. (%)
CDL w/o A, TDL w/o A (SHREC'17 Track)	1.34M	91.55
CDL w/ A, TDL w/ A (SHREC'17 Track)	1.38M	92.62
CDL w/o A, TDL w/ A (SHREC'17 Track)	1.36M	92.26
CDL w/ A, TDL w/o A (SHREC'17 Track)	1.36M	93.57
CDL w/o A, TDL w/o A (NW-UCLA)	1.33M	91.4
CDL w/ A, TDL w/ A (NW-UCLA)	1.37M	93.97
CDL w/o A, TDL w/ A (NW-UCLA)	1.35M	93.32
CDL w/ A, TDL w/o A (NW-UCLA)	1.35M	94.82
CDL w/o A, TDL w/o A (NTU-RGB+D)	1.35M	94.17
CDL w/ A, TDL w/ A (NTU-RGB+D)	1.39M	94.71
CDL w/o A, TDL w/ A (NTU-RGB+D)	1.37M	94.51
CDL w/ A, TDL w/o A (NTU-RGB+D)	1.37M	95.00

the gesture sequence to 180 frames and 150 frames respectively, and the sequence with less than 180 frames and 150 frames will be padded with 0. For the NTU RGB+D dataset, the training ended in 65 epochs. We set the batch size to 64 and resize each sample to 64 frames. We also set the weight decay to 0.0004, and let the learning rate decay with a factor of 0.1 at epochs 35 and 55. For the NW-UCLA dataset, the training ended in 65 epochs. We set the batch size to 16. We also set the weight decay to 0.0001, and let the learning rate decay with a factor of 0.1 at epochs 50. Similarly, we adopt the methods of [15] and [12] to preprocess the NTU RGB+D dataset and Northwestern-UCLA dataset respectively.

B. Ablation Studies

We conduct ablation experiments on the SHREC'17 Track dataset with the proposed temporal decoupling graph convolution. Besides, to demonstrate the robustness and generalization of our TD-GCN, we also conduct some ablation experiments on two human action datasets namely NTU-RGB+D and NW-UCLA dataset, and a detailed comparison is made with the previous state-of-the-art GCN methods. We also visualize the learned temporal-dependent topologies and analyze the parameters and performances of TD-GC.

First, we validate the effects of the predefined topology A, the channel-specific correlation matrix Q, the channel-dependent topologies D, and the temporal-dependent topologies H respectively by removing any of them from TD-GCN. The results in Table I indicate that removing any of the four topologies reduces the accuracy, where removing the predefined topology A (TD-GCN w/o A) reduces the accuracy by 2.02% relative to the optimal value on the SHREC'17 Track dataset. Compared with the optimal value on the NTU-RGB+D dataset, removing the channel-specific correlation matrix Q (TD-GCN w/o Q) and the channel-dependent topologies D (TD-GCN w/o D) reduces the number of parameters but also reduces the accuracy by 0.07% and 1.34% respectively. When the topologies H is removed (TD-GCN w/o H), the accuracy drops by 0.36%, 0.64% and 0.34% relative to the optimal value on three datasets respectively, which indicates that the temporal-dependent adjacency matrices are calculated by the temporal decoupling method can improve the modeling ability of GCN.

Similar to the CDL module, in Table II we explore the performance of introducing a predefined adjacency matrix A by using prior knowledge in the TDL module. The experimental results (CDL w/ A, TDL w/ A) on three datasets show that introducing an adjacency matrix of prior knowledge into the TDL module will increase parameters and reduce recognition accuracy. Besides, we also conducted ablation studies that introduce prior knowledge in the TDL module and remove prior knowledge in the CDL module (CDL w/o A, TDL w/ A) on three datasets. Compared with the setting that only introduces prior knowledge in the CDL module (CDL w/ A, TDL w/o A), the recognition accuracy of this setting (CDL w/o A, TDL w/ A) will decrease by 1.31%, 1.5% and 0.49% on the SHREC'17 Track, NW-UCLA and NTU-RGB+D datasets respectively.

Then we use the SHREC'17 Track and NW-UCLA datasets to conduct ablation experiments on the multi-branch temporal convolution and explore the performance of the three basic branches in Fig. 4 under different combinations. We design ten different branch combinations in Table III and explore the performance differences in parameters and recognition accuracy while keeping other settings consistent.

The results in Table III show that different combinations of the three basic branches in Fig. 4 will affect the performance of our network. The comparison of types II, III, and IV in Table III shows that reducing the size of the convolution kernel will reduce parameters, but an excessively small convolution kernel will cause the model performance to degrade. Besides, when using a multi-branch temporal convolution module, our model will perform better than using a single-branch module. When the number of branches is set to four and the hyperparameter settings follow Type IX, our TD-GCN performs best among ten different combinations. After considering the performance and parameters of our model, we take the setting of Type IX in Table III as the final multi-branch module.

We also explore different configurations of TD-GC, including the reduction rate r of ζ and the activation function σ. Table IV shows that increasing the reduction rate r can reduce the

TABLE III
COMPARISONS OF THE ACCURACY OF TD-GCN WITH DIFFERENT TEMPORAL CONVOLUTIONAL BRANCH COMBINATIONS

Type	Branches	Comb.	d	k	Param.	Acc. (%)
I	1	A	[1]	5	2.51M	92.74⋆
I	1	A	[1]	5	2.50M	93.10◇
II	2	A	[1,2]	3	1.59M	92.26⋆
II	2	A	[1,2]	3	1.58M	93.53◇
III	2	A	[1,2]	5	1.85M	92.62⋆
III	2	A	[1,2]	5	1.85M	93.75◇
IV	2	A	[1,2]	7	2.11M	91.90⋆
IV	2	A	[1,2]	7	2.11M	91.59◇
V	4	A	[1,2,3,4]	5	1.53M	91.90⋆
V	4	A	[1,2,3,4]	5	1.52M	92.89◇
VI	4	A+B	[1,2,3]	5	1.44M	92.38⋆
VI	4	A+B	[1,2,3]	5	1.43M	94.61◇
VII	4	A+C	[1,2,3]	5	1.44M	92.26⋆
VII	4	A+C	[1,2,3]	5	1.44M	93.10◇
VIII	4	A+B+C	[1,2]	3	1.29M	92.86⋆
VIII	4	A+B+C	[1,2]	3	1.29M	94.61◇
IX	4	A+B+C	[1,2]	5	1.36M	**93.57**⋆
IX	4	A+B+C	[1,2]	5	1.35M	**94.81**◇
X	4	A+B+C	[1,2]	7	1.43M	92.86⋆
X	4	A+B+C	[1,2]	7	1.42M	93.10◇

Symbol ⋆ and symbol ◇ indicate experiments are performed on the SHREC'17 track and NW-UCLA datasets respectively.

TABLE IV
COMPARISONS OF THE ACCURACY OF TD-GC WITH DIFFERENT SETTINGS

Method	r	σ	Param.	Acc. (%)
Baseline	-	-	1.85M	93.31
A	4	Tanh	1.51M	93.53
B	8	Tanh	1.35M	**94.82**
C	12	Tanh	1.30M	94.18
D	12	Sigmoid	1.30M	92.24
E	12	ReLU	1.30M	93.53
F	8	Sigmoid	1.35M	92.03
G	8	ReLU	1.35M	93.53

number of parameters, but an excessive reduction rate will lead to a decline in the model's ability to capture correlations. Comparing models B, F, and G, the activation function Tanh performs better than Sigmoid and ReLU and we argue that non-negative output values of Sigmoid and ReLU constrain the ability to capture correlations. Considering efficiency and performance, we choose model B as our final model configuration.

As mentioned above, skeleton-based gesture recognition can be extended to skeleton-based action recognition. The two fields are closely related, and some of their recognition methods can be shared. To demonstrate the robustness and generalization of our proposed TD-GCN, we conduct ablation experiments on the NTU-RGB+D and NW-UCLA human action datasets. At the same time, we conduct a detailed comparison with the previous state-of-the-art GCN method, namely CTR-GCN [16].

We employ CTR-GCN [16] as the baseline, change the graph convolution method TD-GC in Fig. 4 to CTR-GC, and ensure fair comparison by controlling the rest of the modules to be the same. We compare the performance under the four data modalities and the results are shown in Tables V and VI. We observe that in the NW-UCLA dataset, there are three modalities with higher classification accuracy than CTR-GCN. When using the joint

TABLE V
COMPARISON OF TD-GC AND CTR-GC ON NTU-RGB+D DATASET USING DIFFERENT MODALITIES

Modality	X-view (%)	X-sub (%)
Joint (CTR-GCN)	94.56	89.66
Joint (TD-GCN)	$94.99^{\uparrow 0.43}$	$90.16^{\uparrow 0.50}$
Bone (CTR-GCN)	94.67	89.96
Bone (TD-GCN)	$94.78^{\uparrow 0.11}$	$90.15^{\uparrow 0.19}$
Joint Motion (CTR-GCN)	93.02	88.34
Joint Motion (TD-GCN)	$93.03^{\uparrow 0.01}$	$87.91^{\downarrow 0.43}$
Bone motion (CTR-GCN)	91.66	87.29
Bone motion (TD-GCN)	$91.81^{\uparrow 0.15}$	$87.37^{\uparrow 0.08}$

TABLE VI
COMPARISON OF TD-GC AND CTR-GC ON NW-UCLA DATASET USING DIFFERENT MODALITIES

Modality	NW-UCLA (%)
Joint (CTR-GCN)	93.1
Joint (TD-GCN)	$94.82^{\uparrow 1.72}$
Bone (CTR-GCN)	94.39
Bone (TD-GCN)	$93.53^{\downarrow 0.86}$
Joint Motion (CTR-GCN)	91.38
Joint Motion (TD-GCN)	$92.67^{\uparrow 1.29}$
Bone motion (CTR-GCN)	89.22
Bone motion (TD-GCN)	$89.66^{\uparrow 0.44}$

modality and the joint motion modality, the classification accuracy is 94.82% and 92.67% respectively, which outperforms the CTR-GCN by 1.72% and 1.29%. In the NTU-RGB+D dataset, there are eleven aspects with higher classification accuracy than CTR-GCN. Among them, on the benchmark of cross-view, when using the joint modality and bone motion modality, the classification accuracy is 94.99% and 91.81%, which outperforms the CTR-GCN by 0.43% and 0.15%. And on the benchmark of cross-subject, the classification accuracy is 90.16% and 87.37%, which outperforms the CTR-GCN by 0.5% and 0.08%. In both datasets, TD-GCN performs worse than CTR-GCN in two aspects. In the NW-UCLA dataset, when using the bone modality, the classification accuracy is 93.53%, which is lower than CTR-GCN by 0.86%. On the benchmark of cross-subject, when using the joint motion modality, the classification accuracy is 87.91%, which is lower than CTR-GCN by 0.43%. We infer that this is due to a flaw that self-influence is not considered when using the self-pair-wise subtraction operation to calculate the temporal-dependent adjacency matrices and the channel-dependent adjacency matrices.

Tables V and VI show that our TD-GC adds frame-dependent topologies through temporal decoupling, which significantly improves the performance of GCN. We also compared the parameters and computation cost of CTR-GCN and TD-GCN on the NW-UCLA dataset and NTU-RGB+D dataset respectively. The comparison results are shown in Table VII. We observe that the amount of parameters of our TD-GCN is reduced by 5.5% and 5.6% on the two datasets respectively. Compared with CTR-GCN, our TD-GCN also has a significant improvement in computation cost. On the NW-UCLA dataset, the TD-GCN reduces the computation cost by 5.2%. Similarly, our TD-GCN reduces the computation cost by 5.1% on the NTU-RGB+D

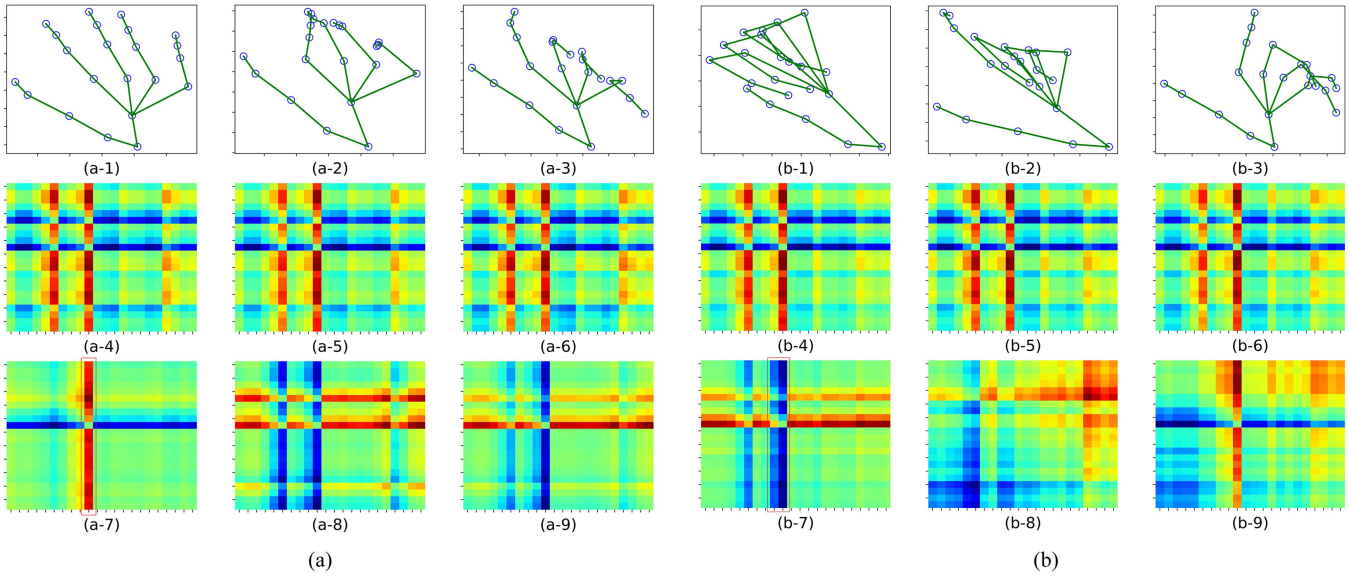

Fig. 6. Visualization of the skeleton and temporal-dependent topologies. (a) The visualization sample of the "tap" gesture. (a-1) Skeleton of frame zero. (a-2) Skeleton of frame thirteen. (a-3) Skeleton of frame forty-nine. (a-4) Topology of frame zero. (a-5) Topology of frame thirteen. (a-6) Topology of frame forty-nine. (a-7) Topological difference between frame zero and frame thirteen. (a-8) Topological difference between frame zero and frame forty-nine. (a-9) Topological difference between frame thirteen and frame forty-nine. (b) The visualization sample of the "expand" gesture. (b-1) Skeleton of frame zero. (b-2) Skeleton of frame seventeen. (b-3) Skeleton of frame Twenty-eight. (b-4) Topology of frame zero. (b-5) Topology of frame seventeen. (b-6) Topology of frame Twenty-eight. (b-7) Topological difference between frame zero and frame seventeen. (b-8) Topological difference between frame zero and frame Twenty-eight. (b-9) Topological difference between frame seventeen and frame Twenty-eight.

TABLE VII
COMPARISON OF PARAMETERS AND COMPUTATION COST BETWEEN CTR-GCN AND TD-GCN ON NW-UCLA AND NTU-RGB+D DATASETS

Method	Param.	FLOPs
CTR-GCN (NW-UCLA)	1.43M	1.16G
TD-GCN (NW-UCLA)	$1.35M^{\downarrow 5.6\%}$	$1.10G^{\downarrow 5.2\%}$
CTR-GCN (NTU-RGB+D)	1.45M	1.78G
TD-GC (NTU-RGB+D)	$1.37M^{\downarrow 5.5\%}$	$1.69G^{\downarrow 5.1\%}$

TABLE VIII
ACCURACY COMPARISON WITH STATE-OF-THE-ART METHODS ON SHREC'17 TRACK DATASET

Method	14 Gestures (%)	28 Gestures (%)
ST-GCN [14]	92.7	87.7
ST-TS-HGR-Net [46]	94.3	89.4
HPEV [47]	94.9	92.3
DSTA-Net [38]	97.0	93.9
MS-ISTGCN [11]	96.7	94.9
TD-GCN (JM Only)	95.24	91.19
TD-GCN (J Only)	96.31	93.57
TD-GCN (J+B+JM)	**97.02**	**95.36**

dataset. Combining the results in Tables V, VI, and VII, it can be concluded that our TD-GCN is superior to CTR-GCN in terms of performance, parameters and computation cost.

C. Visualization of Learned Topologies

To visualize the learned temporal-dependent adjacency matrix of TD-GC more intuitively, we select two gesture and two action samples from the SHREC'17 Track and the NTU-RGB+D datasets for visualization respectively. We first select three frames of skeleton data from the gesture sample "tap" and the gesture sample "expand" for visualization respectively, then we show the learned temporal-dependent topologies of the gesture samples and their differences in Fig. 6. The values close to 0 indicate weak relationships between joints and vice versa. Similarly, we do the same for the action sample "salute" and the action sample "stand up" in Fig. 7. Our observation is four-fold. First, topologies of different frames are different, which are determined by the data of the current frame (Fig. 6(a-4), 6(a-5), and (a-6)). Second, topologies of different frames are easily affected by the core joints of gestures. The core joints of gesture have a greater impact on the rest of hand joints. (The solid red box in Fig. 6(a-7) and (b-7)). Third, adjacent frames have similar topologies (Fig. 7(a-4) and (a-5)), but they still differ (Fig. 7(a-7)), while frames that are farther apart have more different topologies (Fig. 7(a-4) and (a-6)). Finally, for the topologies of different frames, the difference mainly focuses on the main joints where the action occurs. For example, the salute action mainly occurs on the right elbow and right hand, so in the topology of different frames that are far apart, they are more different (Solid yellow box in Fig. 7(a-8)).

D. Comparisons With the State-of-the-Art

We compare our TD-GCN with the state-of-the-art methods on the SHREC'17 Track dataset, DHG-14/28 dataset, NW-UCLA dataset, and NTU-RGB+D dataset in Tables VIII, X, XI and XII. On Four datasets, our model outperforms all existing methods under nearly all evaluation benchmarks. Notably, our TD-GCN is the first model that focuses on the modeling

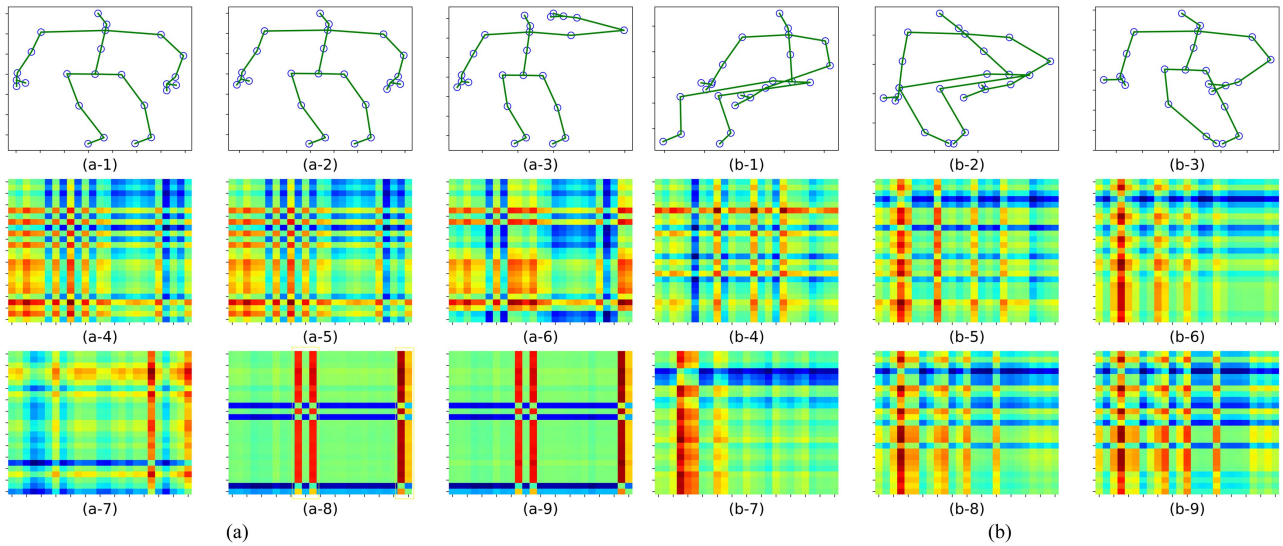

Fig. 7. Visualization of the skeleton and temporal-dependent topologies. (a) The visualization sample of the "salute" action. (a-1) Skeleton of frame zero. (a-2) Skeleton of frame ten. (a-3) Skeleton of frame fifty. (a-4) Topology of frame zero. (a-5) Topology of frame ten. (a-6) Topology of frame fifty. (a-7) Topological difference between frame zero and frame ten. (a-8) Topological difference between frame zero and frame fifty. (a-9) Topological difference between frame ten and frame fifty. (b) The visualization sample of the "stand up" action. (b-1) Skeleton of frame ten. (b-2) Skeleton of frame thirty. (b-3) Skeleton of frame fifty. (b-4) Topology of frame ten. (b-5) Topology of frame thirty. (b-6) Topology of frame fifty. (b-7) Topological difference between frame ten and frame thirty. (b-8) Topological difference between frame ten and frame fifty. (b-9) Topological difference between frame thirty and frame fifty.

TABLE IX
RECOGNITION ACCURACY OF 14 DIFFERENT SUBJECTS AS THE VALIDATION SET OF DHG-14/28 DATASET

Subject	**1**	**3**	**4**	**5**	**8**	**9**	**10**	**11**	**12**	**14**	**15**	**16**	**17**	**19**
acc.($J^{\star}$)(%)	89.29	93.57	88.57	85.00	92.14	83.57	92.86	89.29	92.14	88.57	95.71	87.86	86.43	94.29
acc.($B^{\star}$)(%)	90.00	92.14	86.43	90.00	91.43	84.29	92.14	95.00	90.71	83.57	84.29	85.71	85.00	90.71
acc.($JM^{\star}$)(%)	87.14	95.00	90.71	92.14	88.57	89.29	93.57	92.86	91.43	86.43	96.43	93.57	89.29	95.71
acc.($J^{\star}$+$B^{\star}$+JM)(%)	91.43	96.43	91.43	92.86	92.86	90.00	95.00	95.71	92.14	89.29	97.86	95.00	91.43	95.71
acc.($J^{\diamond}$)(%)	90.00	98.57	93.57	90.00	92.86	91.43	92.86	95.71	90.00	88.57	97.14	96.43	86.43	97.86
acc.($B^{\diamond}$)(%)	90.71	91.43	87.86	91.43	92.86	88.57	92.14	92.86	91.43	85.00	87.86	86.43	87.14	92.14
acc.($JM^{\diamond}$)(%)	88.57	97.86	92.86	92.14	94.29	89.29	95.71	95.71	89.29	87.86	97.86	95.00	90.00	96.43
acc.($J^{\diamond}$+$B^{\diamond}$+$JM^{\diamond}$)(%)	92.86	99.29	94.29	95.00	96.43	96.43	96.43	97.14	92.86	90.00	98.57	96.43	92.14	98.57

Symbol * and symbol $^{\diamond}$ denote experiments with 28 gestures and 14 gestures respectively.

of different frame topologies, and it achieves superior results in skeleton-based gesture recognition.

In the SHREC'17 Track dataset, we use a three-stream architecture to fuse the results of three modalities, namely joint modality, bone modality, and joint motion modality. In Table VIII, when using 14 gesture classes and 28 gesture classes, the classification accuracy after fusion is 97.02% and 95.36% respectively, which outperforms the MS-ISTGCN [11] by 0.32% and 0.46%. In fact, the MS-ISTGCN uses the normalized Gaussian function to calculate the similarity matrix and combined it with the attention mechanism to represent the connection strength of hand joints. However, the similarity matrix computed by MS-ISTGCN suffers from the problem of being shared among different channels and frames, which will inhibit the modeling ability of GCN to a certain extent as mentioned above. Similarly, when we only use the modality of joint, our classification accuracy is 96.31% and 93.57% respectively, which outperforms the ST-GCN by 5.61% and 5.87% respectively. We also show the confusion matrix in Fig. 8(a) and

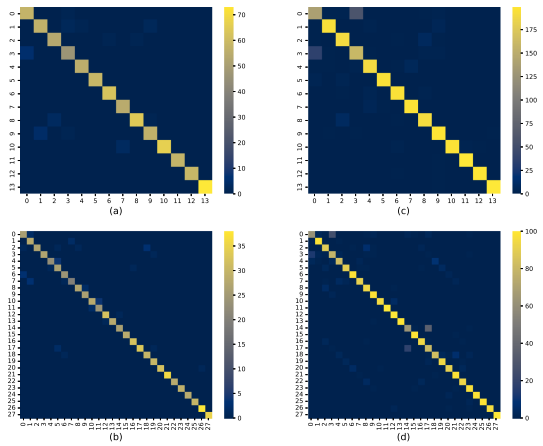

Fig. 8. Visualization of confusion matrices. (a) The confusion matrix of the SHREC'17 Track dataset when using 14 gesture classes. (b) The confusion matrix of the SHREC'17 Track dataset when using 28 gesture classes. (c) The confusion matrix of the DHG-14/28 dataset when using 14 gesture classes. (d) The confusion matrix of the DHG-14/28 dataset when using 28 gesture classes.

TABLE X
ACCURACY COMPARISON WITH STATE-OF-THE-ART METHODS ON DHG-14/28 DATASET

Method	14 Gestures (%)	28 Gestures (%)
CNN+LSTM [20]	85.6	81.1
ST-TS-HGR-Net [46]	87.3	83.4
ST-GCN [14]	91.2	87.1
HPEV [47]	92.5	88.9
DSTA-Net [38]	93.8	90.9
MS-ISTGCN [11]	93.7	91.2
TD-GCN (J+B+JM)	**93.9**	**91.4**

TABLE XI
ACCURACY COMPARISON WITH STATE-OF-THE-ART METHODS ON NORTHWESTERN-UCLA DATASET

Method	NW-UCLA (%)
Lie Group [48]	74.2
Actionlet ensemble [49]	76.0
HBRNN-L [50]	78.5
Skeleton Visualization [34]	86.1
Ensemble TS-LSTM [51]	89.2
AGC-LSTM [52]	93.3
Shift-GCN [12]	94.6
DC-GCN+ADG [35]	95.3
FGCN [10]	95.3
CTR-GCN* [16]	96.5
TD-GCN (Joint Only)	94.8
TD-GCN	**97.4**

Symbol * denotes the rerunning result of authors' original code.

TABLE XII
ACCURACY COMPARISON WITH STATE-OF-THE-ART METHODS ON NTU-RGB+D DATASET

Method	X-Sub (%)	X-View (%)
Ind-RNN [26]	81.8	88.0
Bayesian GC-LSTM [53]	81.8	89.0
ST-GCN [14]	81.5	88.3
HCN [31]	86.5	91.1
SR-TSL [54]	84.8	92.4
2s-AGCN [13]	88.5	95.1
SGN [15]	89.0	94.5
AGC-LSTM [52]	89.2	95.0
NAS-GCN [37]	89.4	95.7
Hyperbolic ST-GCN [55]	89.5	95.1
ST-GDN [56]	89.7	95.9
Poincare-GCN [57]	89.7	96.0
DGNN [58]	89.9	96.1
Tripool [59]	90.0	96.7
Global GNN [60]	90.1	95.9
Shift-GCN [12]	90.7	96.5
DC-GCN+ADG [35]	90.8	96.6
PA-ResGCN-B19 [61]	90.9	96.0
DDGCN [62]	91.1	97.1
Dynamic GCN [17]	91.5	96.0
MS-G3D [63]	91.5	96.2
CTR-GCN* [16]	92.1	96.6
ESE-FN [64]	92.4	96.7
TD-GCN (Joint Only)	90.2	95.0
TD-GCN	**92.8**	**96.8**

Symbol * denotes the rerunning result of authors' original code.

(b) when fusing the three modalities of the SHREC'17 Track dataset.

In the DHG-14/28 dataset, we also use a three-stream architecture to fuse the results of three modalities. In Table X, when using 14 gesture classes and 28 gesture classes, the classification accuracy after fusion is 93.9% and 91.4% respectively, which outperforms the MS-ISTGCN [11] by 0.2% and 0.2%, and outperforms the DSTA-Net [38] by 0.1% and 0.5%. As mentioned above, the DHG-14/28 dataset uses a leave-one-subject-out cross-validation strategy [18] for evaluation. So we selectively show the recognition accuracy of 14 different subjects as validation sets in Table IX. We also show the confusion matrix in Fig. 8(c) and (d) when fusing the three modalities of the DHG-14/28 dataset.

To demonstrate the robustness and generalization of our TD-GCN, we use TD-GCN for skeleton-based action recognition. We conduct experiments on two public benchmark action datasets namely NTU-RGB+D and NW-UCLA and compare them with previous state-of-the-art methods.

In the Northwestern-UCLA dataset, we use a four-stream architecture to fuse the results of the four modalities mentioned in Section III-A. In Table XI, the classification accuracy after fusion is 97.4%, which outperforms current state-of-the-art CTR-GCN by 0.9% and outperforms the FGCN by 2.1%.

When we only use the modality of joint, our classification accuracy is 94.8%, which outperforms the shift-GCN [12] that fuses four modalities by 0.2%. The shift-GCN directly exchanges part of the features between joints through a shift operation without relying on the adjacency matrices, which ignores that the connections between joints are different. Although the shift-GCN compensates for this deficiency with a learnable mask matrix, it still does not effectively exploit the differences in joint information. While our TD-GCN uses a data-driven method to construct connections between joints, the connections are determined by the differences in joint features in different channels and different frames. We also show the confusion matrix in Fig. 9(b) when fusing the four modalities of the NW-UCLA dataset.

In the NTU-RGB+D dataset, we also use a four-stream architecture to fuse the results of four modalities. In Table XII, on the benchmark of cross-view, the classification accuracy after fusion is 96.8%, which outperforms the accuracy of our actual recurrence with CTR-GCN by 0.2% and outperforms the Poincare-GCN by 0.8%. When we only use the modality of joint, our classification accuracy is 95.0%, which outperforms the ST-GCN by 6.7% and outperforms the SGN [15] by 0.5%.

On the benchmark of cross-subject, the classification accuracy is 92.8%, which outperforms current state-of-the-art CTR-GCN by 0.4%. When we only use the modality of joint, the classification accuracy is 90.2%, which outperforms the SGN by 1.2%. The SGN introduces the high-level semantics of joints (joint type and frame index) into the network to enhance the feature representation capability and it uses two FC layers to construct the adjacency matrices. However, the SGN still has the problem that joint features of different channels share the same adjacency matrix. While our TD-GCN constructs different adjacency matrices for joints of different channels and different frames. Besides,

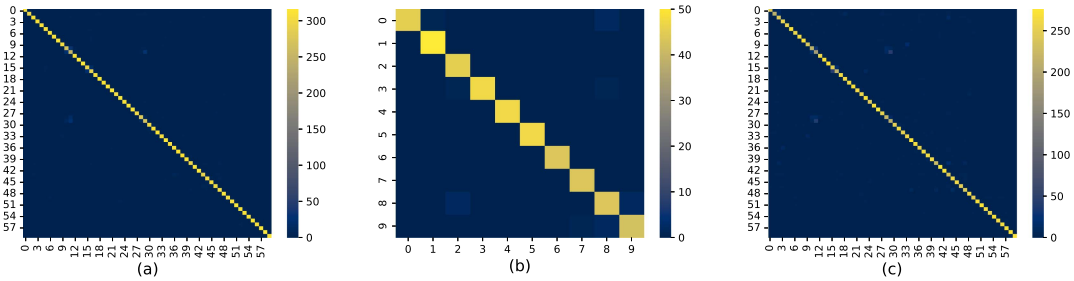

Fig. 9. (a) Confusion matrix of the NTU-RGB+D dataset when fusing four modalities on the benchmark of cross-view. (b) The confusion matrix of the NW-UCLA dataset when fusing four modalities. (c) The confusion matrix of the NTU-RGB+D dataset when fusing four modalities on the benchmark of cross-subject.

TABLE XIII
ACCURACY COMPARISON WITH MOST RECENT TEMPORAL MODELING METHODS ON NTU-RGB+D DATASET

Method	X-Sub (%)	X-View (%)
ST-TR [39]	89.9	96.1
DSTA-Net [38]	91.5	96.4
STST [40]	91.9	96.8
TD-GCN	**92.8**	**96.8**

our TD-GCN performs better without introducing high-level semantics. Similarly, we show the confusion matrix in Fig. 9 when fusing the four modalities of the NTU-RGB+D dataset.

E. Comparisons With Temporal Modeling Methods

We also compare TD-GCN with the temporal-information-based methods mentioned in Section II. We conduct experiments on the NTU-RGB+D dataset, and the comparative experimental results are shown in Table XIII. It should be emphasized that our TD-GCN is the first to introduce temporal dependency modeling for formulating adjacency matrices, and the data in Table XIII shows that our TD-GCN achieves better recognition accuracy than the three methods that utilize temporal information to compute attention maps. For example, on the benchmark of cross-view, the classification accuracy of our TD-GCN is 96.8%, which outperforms the ST-TR by 0.7% and outperforms the DSTA-Net by 0.4%. On the benchmark of cross-subject, the classification accuracy of our TD-GCN is 92.8%, which outperforms the ST-ST by 0.9%.

V. CONCLUSION

We present Temporal Decoupling Graph Convolution Network (TD-GCN) for skeleton-based gesture recognition, which applies temporal-dependant adjacency matrices for skeletons from different frames. Different from previous methods which mainly focus on spatial-dependent adjacency matrices, TD-GCN is the first to use temporal-dependent adjacency matrices for temporal-sensitive topology learning from skeleton joints. Moreover, TD-GCN is flexible to combine both spatial-dependent and temporal-dependent adjacency matrices to learn spatiotemporal topology from skeleton joints. The effectiveness of TD-GCN is verified on the benchmark SHREC'17 Track, DHG-14/28, NTU-RGB+D, and NW-UCLA datasets, where TD-GCN outperforms state-of-the-art methods.

REFERENCES

[1] S. Tang, D. Guo, R. Hong, and M. Wang, "Graph-based multimodal sequential embedding for sign language translation," *IEEE Trans. Multimedia*, vol. 24, pp. 4433–4445, 2022.
[2] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7784–7793.
[3] R. Li, H. Wang, and Z. Liu, "Survey on mapping human hand motion to robotic hands for teleoperation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2647–2665, May 2022.
[4] Y. Zhang, C. Cao, J. Cheng, and H. Lu, "EgoGesture: A new dataset and benchmark for egocentric hand gesture recognition," *IEEE Trans. Multimedia*, vol. 20, pp. 1038–1050, 2018.
[5] G. Wu and W. Kang, "Vision-based fingertip tracking utilizing curvature points clustering and hash model representation," *IEEE Trans. Multimedia*, vol. 19, pp. 1730–1741, 2017.
[6] C. Li et al., "Hierarchical latent concept discovery for video event detection," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2149–2162, May 2017.
[7] M. Li et al., "Rhythm-aware sequence-to-sequence learning for labanotation generation with gesture-sensitive graph convolutional encoding," *IEEE Trans. Multimedia*, vol. 24, pp. 1488–1502, 2022.
[8] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 14–29, Jan. 2016.
[9] K. Liu, L. Gao, N. M. Khan, L. Qi, and L. Guan, "A multi-stream graph convolutional networks-hidden conditional random field model for skeleton-based action recognition," *IEEE Trans. Multimedia*, vol. 23, pp. 64–76, 2021.
[10] H. Yang et al., "Feedback graph convolutional network for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 31, pp. 164–175, 2022.
[11] J.-H. Song, K. Kong, and S.-J. Kang, "Dynamic hand gesture recognition using improved spatio-temporal graph convolutional network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 6227–6239, Sep. 2022.
[12] K. Cheng et al., "Skeleton-based action recognition with shift graph convolutional network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 180–189.
[13] 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.*, 2019, pp. 12018–12027.
[14] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1–6.
[15] P. Zhang et al., "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1109–1118.
[16] Y. Chen et al., "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13359–13368.

[17] F. Ye et al., "Dynamic GCN: Context-enriched topology learning for skeleton-based action recognition," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 55–63.
[18] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 1206–1214.
[19] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Heterogeneous hand gesture recognition using 3D dynamic skeletal data," *Comput. Vis. Image Understanding*, vol. 181, pp. 60–72, 2019.
[20] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vélez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognit.*, vol. 76, pp. 80–94, 2018.
[21] W. Ng, M. Zhang, and T. Wang, "Multi-localized sensitive autoencoder-attention-LSTM for skeleton-based action recognition," *IEEE Trans. Multimedia*, vol. 24, pp. 1678–1690, 2022.
[22] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, "Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures," *IEEE Trans. Multimedia*, vol. 21, pp. 234–245, 2019.
[23] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1586–1599, Apr. 2018.
[24] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-temporal attention-based LSTM networks for 3D action recognition and detection," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3459–3471, Jul. 2018.
[25] C. Li et al., "Memory attention networks for skeleton-based action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4800–4814, Sep. 2022.
[26] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (IndRNN): Building a longer and deeper RNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5457–5466.
[27] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3633–3642.
[28] P. Zhang et al., "Adding attentiveness to the neurons in recurrent neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 136–152.
[29] 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.
[30] R. Xia, Y. Li, and W. Luo, "LAGA-Net: Local-and-global attention network for skeleton based action recognition," *IEEE Trans. Multimedia*, vol. 24, pp. 2648–2661, 2022.
[31] 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. Int. Joint Conf. Artif. Intell.*, 2018, pp. 786–792.
[32] H. Liu, J. Tu, and M. Liu, "Two-stream 3D convolutional neural network for skeleton-based action recognition," 2017, *arXiv:1705.08106*.
[33] 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.*, 2017, pp. 4570–4579.
[34] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognit.*, vol. 68, pp. 346–362, 2017.
[35] K. Cheng et al., "Decoupling GCN with dropgraph module for skeleton-based action recognition," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 536–553.
[36] M. Li et al., "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3590–3598.
[37] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 2669–2676.
[38] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 38–53.
[39] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks," *Comput. Vis. Image Understanding*, vol. 208/209, 2021, Art. no. 103219.
[40] Y. Zhang et al., "STST: Spatial-temporal specialized transformer for skeleton-based action recognition," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 3229–3237.
[41] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[42] Q. de Smedt et al., "SHREC'17 track: 3D hand gesture recognition using a depth and skeletal dataset," in *Proc. 3DOR-10th Eurograph. Workshop 3D Object Retrieval*, 2017, pp. 1–6.
[43] 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.*, 2016, pp. 1010–1019.
[44] 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.*, 2014, pp. 2649–2656.
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[46] X. S. Nguyen, L. Brun, O. Lezoray, and S. Bougleux, "A neural network based on SPD manifold learning for skeleton-based hand gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12028–12037.
[47] J. Liu, et al., "Decoupled representation learning for skeleton-based gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5750–5759.
[48] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4041–4049.
[49] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3D human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 914–927, May 2014.
[50] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1110–1118.
[51] 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.*, 2017, pp. 1012–1020.
[52] 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.*, 2019, pp. 1227–1236.
[53] R. Zhao, K. Wang, H. Su, and Q. Ji, "Bayesian graph convolution LSTM for skeleton based action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6881–6891.
[54] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, "Skeleton-based action recognition with spatial reasoning and temporal stack learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 106–121.
[55] A. Mostafa, W. Peng, and G. Zhao, "Hyperbolic spatial temporal graph convolutional networks," in *Proc. IEEE Int. Conf. Image Process.*, 2022, pp. 3301–3305.
[56] W. Peng, J. Shi, and G. Zhao, "Spatial temporal graph deconvolutional network for skeleton-based human action recognition," *IEEE Signal Process. Lett.*, vol. 28, pp. 244–248, 2021.
[57] W. Peng, J. Shi, Z. Xia, and G. Zhao, "Mix dimension in poincaré geometry for 3D skeleton-based action recognition," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 1432–1440.
[58] 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.*, 2019, pp. 7904–7913.
[59] W. Peng, X. Hong, and G. Zhao, "Tripool: Graph triplet pooling for 3D skeleton-based action recognition," *Pattern Recognit.*, vol. 115, 2021, Art. no. 107921.
[60] W. Peng, J. Shi, T. Varanka, and G. Zhao, "Rethinking the ST-GCNS for 3D skeleton-based human action recognition," *Neurocomputing*, vol. 454, pp. 45–53, 2021.
[61] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 1625–1633.
[62] M. Korban and X. Li, "DDGCN: A dynamic directed graph convolutional network for action recognition," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 761–776.
[63] 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.*, 2020, pp. 140–149.
[64] X. Shu, J. Yang, R. Yan, and Y. Song, "Expansion-squeeze-excitation fusion network for elderly activity recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5281–5292, Aug. 2022.