

Article

Graph Convolutional Network with Multi-View Topology for Lightweight Skeleton-Based Action Recognition

Liangliang Wang, Xu Zhang and Chuang Zhang *

School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China; 202312490167@nuist.edu.cn (L.W.); 202312490221@nuist.edu.cn (X.Z.)

* Correspondence: 001981@nuist.edu.cn

Abstract

Skeleton-based action recognition is an important subject in deep learning. Graph Convolutional Networks (GCNs) have demonstrated strong performance by modeling the human skeleton as a natural topological graph, representing the connections between joints. However, most existing methods rely on non-adaptive topologies or insufficiently expressive representations. To address these limitations, we propose a Multi-view Topology Refinement Graph Convolutional Network (MTR-GCN), which is efficient, lightweight, and delivers high performance. Specifically: (1) We propose a new spatial topology modeling approach that incorporates two views. A dynamic view fuses joint information from dual streams in a pairwise manner, while a static view encodes the shortest static paths between joints, preserving the original connectivity relationships. (2) We propose a new MultiScale Temporal Convolutional Network (MSTC), which is efficient and lightweight. (3) Furthermore, we introduce a new temporal topology strategy by modeling temporal frames as a graph, which strengthens the extraction of temporal features. By modeling the human skeleton as both a spatial and a temporal graph, we reveal a topological symmetry between space and time within the unified spatio-temporal framework. The proposed model achieves state-of-the-art performance on several benchmark datasets, including NTU RGB + D (XSub: 92.8%, XView: 96.8%), NTU RGB + D 120 (XSub: 89.6%, XSet: 90.8%), and NW-UCLA (95.7%), demonstrating the effectiveness of our GCN module, TCN module, and overall architecture.



Academic Editor: Zhixun Su

Received: 18 June 2025

Revised: 10 July 2025

Accepted: 15 July 2025

Published: 4 August 2025

Citation: Wang, L.; Zhang, X.; Zhang, C. Graph Convolutional Network with Multi-View Topology for Lightweight Skeleton-Based Action Recognition. *Symmetry* **2025**, *17*, 1235. <https://doi.org/10.3390/sym17081235>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: topological symmetry; skeleton-based action recognition; Multi-view Topology Refinement Graph Convolutional Network; MultiScale Temporal Convolutional Network; temporal topology

1. Introduction

Skeleton-based action recognition plays a crucial role in real-world applications and has long been a prominent topic in the field of deep learning. It is widely used in areas such as intelligent surveillance, human–computer interaction, and medical rehabilitation. Early studies on action recognition primarily relied on video or RGB data as input [1–3]. With the advancement of various sensor technologies, alternative modalities have emerged, including skeleton data [4–6], depth maps [7], and infrared sequences [8]. Among these, skeleton data has attracted increasing attention due to its advantages in preserving user privacy, being robust against background noise, and enabling lightweight network designs.

Initially, skeleton-based action recognition methods were built upon hand-crafted features [9], which often performed well on specific datasets but lacked generalization

ability. Subsequently, deep learning approaches were introduced. RNN-based methods [10,11] treated skeleton sequences as ordered vectors to preserve temporal information but struggled to model spatial dependencies and suffered from slow training. CNN-based methods [12] transformed skeleton data into pseudo-2D images, which also lacked explicit spatial modeling and failed to capture complex joint interactions while offering limited temporal modeling. Since the human skeleton naturally forms a graph structure, GCN-based methods [13] have been proposed to aggregate features based on the topological relationships between joints, effectively capturing intricate inter-joint dependencies and achieving leading performance in both spatial and temporal feature extraction.

Yan et al. [14] made a pioneering contribution by proposing ST-GCN, the first work to apply graph convolutional networks to skeleton-based action recognition. In ST-GCN, joints are treated as graph nodes, and bones (including spatial and temporal connections) are treated as graph edges, thereby establishing a graph-based representation of the human skeleton. However, ST-GCN relies on manually defined skeleton topologies, which are not adaptive and may overlook important connections between joints that are not naturally linked. Subsequent methods [15–17] made improvements in topology learning, but most of them either used simplistic strategies or focused only on a single perspective, leading to incomplete and inaccurate topology representations. BlockGCN [17] pointed out that simple interactions or learning topologies from a single perspective can cause the loss of original skeleton connection information and deviate from the physical topology, which severely impacts recognition performance.

Moreover, as GCN-based methods continue to evolve and recognition accuracy improves, the ever-growing number of parameters and computational demands have emerged as major challenges. Although models such as CTRGCN [16], FR-Head [18], EfficientGCN-B4 [19], and MST-GCN [20] have achieved promising results, they still suffer from heavy model complexity and high computational costs. Therefore, this paper aims to design a high-accuracy, lightweight, and computationally efficient model.

In terms of spatial modeling (GCN), we propose a novel Multi-view Topology Refinement Graph Convolutional Network (MTRGC). In the dynamic view, we dynamically learn pairwise topology relationships among all joints using a dual-stream feature input strategy. Features are projected onto multiple heads to capture richer relational representations. Within each head, we introduce a dimension expansion mechanism that enables pairwise interaction between dual-stream nodes, allowing channel-wise information exchange. We then fuse the multi-head topological information to form a refined representation. For the static view, inspired by BlockGCN [17], we incorporate a topology encoding mechanism that solely depends on the original physical connections and is independent of the input data. By encoding relative positions between statically connected joints, we preserve inherent skeletal structures while complementing the dynamic view.

For temporal modeling, aiming for a lightweight and efficient design, we propose a new MultiScale Temporal Convolutional (MSTC) Network with temporal topology modeling: We employ depthwise separable convolution to extract multi-scale temporal features and integrate a pooling module to capture averaged feature representations. Additionally, another branch is introduced to preserve the original input features. To mitigate the limited receptive field of depthwise separable convolution, we adopt large kernels for broader context modeling. (b) We propose Gated Channel-wise Temporal Topology (GCTT), which models time series as graphs to explicitly capture temporal topology relationships. This approach significantly enhances temporal feature extraction.

We conduct extensive experiments on large-scale skeleton action recognition datasets [21–23]. Experimental results show that our model achieves superior performance across multiple modalities. The main contributions of this paper are summarized as follows:

1. We propose a multi-view topology modeling strategy that captures dynamic joint relationships via a novel pairwise interaction mechanism, while preserving original skeletal connectivity through a complementary static view.
2. We propose a novel MultiScale Temporal Convolutional Network that employs depth-wise separable convolutions with larger kernels for temporal feature extraction. By incorporating a pooling module and a branch preserving input information, it captures richer feature representations. The proposed temporal module achieves a lightweight design while maintaining high accuracy.
3. We propose a novel Gated Channel-wise Temporal Topology (GCTT) that further improves temporal feature extraction on top of the lightweight design. Extensive experiments demonstrate that each component of our model achieves remarkable performance, and the overall model surpasses many state-of-the-art methods. Furthermore, our results highlight the importance of simultaneously leveraging dynamic and static topology information.

(For abbreviations, refer to Table A1.)

2. Related Work

2.1. Skeleton-Based Action Recognition

With the early development of action recognition, most methods relied on videos or RGB data as input. However, with the advancement of various types of sensors, action recognition has expanded to other modalities. Among them, skeleton data, due to its unique advantages, has emerged as a prominent research focus in the field of action recognition.

Early skeleton-based action recognition methods using deep learning mainly adopted RNNs and CNNs for feature extraction. However, these models often lacked effective spatial or temporal modeling capabilities, making it difficult to capture complex and deep inter-joint relationships, which led to a bottleneck in the progress of skeleton-based recognition.

Since human joints naturally form a graph structure in a non-Euclidean space, Graph Convolutional Networks (GCNs) offer an appropriate tool to handle such structures. Yan et al. [14] first introduced Graph Convolutional Networks (GCNs) to extract topological relations among skeletal joints, significantly enhancing feature representation for skeleton-based action recognition.

Further studies built upon this idea. However, these early GCN-based models used non-adaptive topological structures, limiting their ability to flexibly model spatial relationships and preventing dynamic topology updates.

Subsequent works focused on adaptive learning of topology: 2s-AGCN [15] parameterized skeleton nodes into learnable vectors, allowing the network to adaptively learn adjacency weights. CTRGCN [16] proposed a dual-stream subtraction strategy to capture channel-wise topological relationships between joints. This channel-level topological modeling significantly improved model performance and became a baseline choice for many models due to its simplicity. The SA-GC module proposed by InfoGCN [24], based on dot-product attention, can infer a context-dependent intrinsic topology in the spatial modeling of a skeleton. Self-GCN [25] similarly uses attention-based topological modeling, and both demonstrate the widespread application and effectiveness of the attention mechanism in skeleton-based action recognition.

For temporal modeling, early methods like ST-GCN [14] employed simple temporal convolutions. Later, more advanced models were proposed: MST-GCN [20] partitioned feature maps along the channel dimension to better capture features across different temporal ranges. CTRGCN [16] incorporated a multi-scale temporal convolution module combined with dilated convolutions and pooling layers, significantly enhancing the network's ability to model long-term temporal dependencies. Similarly, FR-Head [18], Koopman [26],

and others utilized enhanced TCNs that played crucial roles in improving the extraction of temporal features.

2.2. Relative Position Encoding

Although the aforementioned models are capable of capturing complex relationships between joints, their methods for node interaction remain suboptimal. Moreover, since these models often learn topology from a single perspective, the dynamically learned topology tends to gradually deviate from the original physical structure of the skeleton.

To address this issue, BlockGCN [17] introduced a relative positional encoding mechanism that preserves the inherent skeletal connections during dynamic topology learning.

Building upon this idea, we propose a novel multi-view topology representation, which further enhances the modeling of spatial relationships by integrating multiple perspectives.

2.3. Lightweight Temporal Convolutional Networks

Although the aforementioned models have achieved notable breakthroughs in performance, these improvements often come at the cost of increased parameter counts and higher computational complexity. Moreover, relatively little attention has been paid to enhancing temporal feature extraction.

MobileNet [27] use depthwise separable convolutions to build light-weight deep neural networks. A depthwise separable convolution consists of two steps: first, a depthwise convolution is applied to each input channel separately; second, a pointwise (1×1) convolution is used to fuse features across channels. This structure significantly reduces the complexity of convolutional operations and has been widely adopted in various computation-constrained visual tasks.

In the field of skeleton-based action recognition, methods such as TSGCNext [28] have also applied depthwise separable convolutions. However, the usage was limited to single-branch designs, which restricted the capacity for rich temporal feature extraction.

Meanwhile, models such as CTRGCN [16], FR-Head [18], Koopman [26], and BlockGCN [17] employed multi-scale temporal modules, but they suffered from excessive parameter sizes, low efficiency, and overly complex computations.

Inspired by MobileNet's [27] depthwise separable convolution, our research combines multi-scale temporal modeling and proposes a novel MultiScale Temporal Convolutional Network (MSTC).

On this basis, we further integrate a Gated Channel-wise Temporal Topology (GCTT) module, significantly improving the model's performance while maintaining a lightweight and efficient design.

2.4. Symmetric Topology Modeling

Human skeletons are naturally represented as graphs and processed using Graph Convolutional Networks (GCNs), with the adjacency matrix playing a crucial role. We perform symmetric topology modeling by treating the skeleton as both a spatial and a temporal graph, capturing joint relationships from both dimensions: (a) For spatial modeling, our MTRGC module performs multi-view topology learning, preserving the original static connections while dynamically capturing evolving spatial dependencies. (b) For temporal modeling, we treat skeleton sequences as temporal graphs to learn topological structures along the time dimension. Experimental results demonstrate that our symmetric modeling of spatial and temporal topologies significantly enhances the GCN's ability to capture structural relationships.

3. Methods

In this section, we first present the graph representation of the skeleton data (Section 3.1), followed by a description of the Graph Convolutional Network (GCN) from different views (Section 3.2). Subsequently, we analyze the lightweight design of the Temporal Convolutional Network (TCN) from a mathematical standpoint, along with the modeling process of our Gated Channel-wise Temporal Topology (GCTT) (Section 3.3.2). Finally, we outline the overall structure of the proposed model (Section 3.4).

3.1. Preliminaries

Human skeleton data inherently exhibits a graph structure represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ [14]. Specifically, $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ denotes the set of joints. The set of edges $\mathcal{E}$, representing the connections between each pair of joints, is described by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. Each element a_{ij} in $\mathbf{A}$ indicates the strength of the connection between v_i and v_j .

In this work, $\mathbf{A}$ is the initialized adjacency matrix representing the human skeleton topology, defined following the design in ST-GCN [14]. $\mathbf{A}$ contains three types of edges, corresponding to three channels: self-link, inward, and outward. These three channels represent different types of topologies, enriching the expression of topological relationships. $\mathbf{A}[i]$ refers to the i -th channel of $\mathbf{A}$. In our GCN module, we loop through MTRGC three times, passing in one channel of $\mathbf{A}$ each time.

Our work in Section 3.2 aims to optimize the weights of the adjacency matrix to accurately model the relationships between joints. The original skeleton data is represented as $\mathbf{X} \in \mathbb{R}^{C \times T \times V}$, where C denotes the number of input channels, T denotes the number of input frames, and V denotes the number of joints.

3.2. Multi-View Topology Refinement Graph Convolution

As shown in Figure 1, MTRGC is divided into two parts: a dynamic view and a static view, each capturing different topological relationships. The following sections will introduce these two views separately.

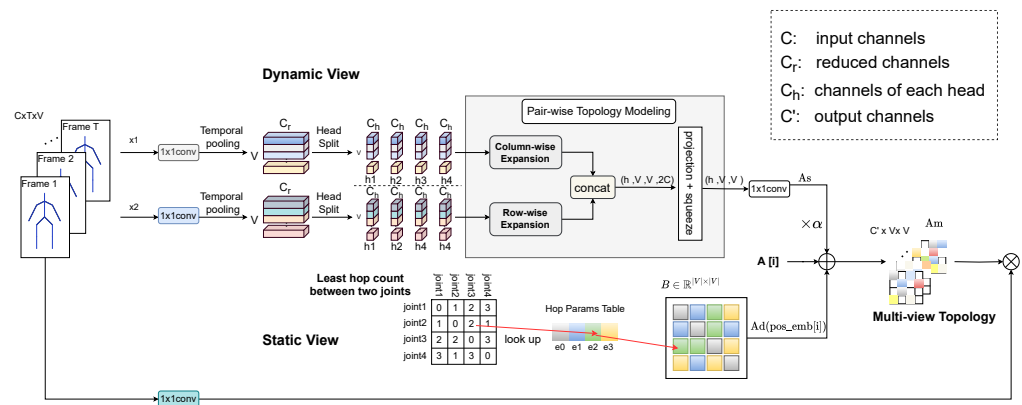


Figure 1. Multi-view Topology Refinement Graph Convolution (MTRGC).

3.2.1. Dynamic View

The dynamic view adopts a two-stream strategy with input features $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{C \times T \times V}$. To reduce computational cost, we first reduce the feature dimension, followed by a temporal pooling layer to obtain the averaged features along the T dimension. This process can be expressed as follows:

$$\mathbf{x}_1 = \frac{1}{T} \sum_{i=1}^T \phi(\mathbf{x}_{:,t,:}), \quad \mathbf{x}_2 = \frac{1}{T} \sum_{i=1}^T \varphi(\mathbf{x}_{:,t,:}), \quad (1)$$

although ‘conv1’ and ‘conv2’ have the same shape and structure, they are two independent layers with different parameters (initialized and updated independently). During model training, they do not share parameters and do not affect each other. Therefore, we use two different function symbols, ϕ and φ , to represent ‘conv1’ and ‘conv2’, respectively. Now $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{V \times C_r}$.

To enrich feature extraction, the features are projected onto multiple heads, and the channels of each head are given by C_h :

$$\mathbf{x}_1 = [\mathbf{x}_{11}, \dots, \mathbf{x}_{1h}], \quad \mathbf{x}_2 = [\mathbf{x}_{21}, \dots, \mathbf{x}_{2h}], \quad (2)$$

Subsequently, we expand the dimensions and concatenate the features to fuse the channel information of each pair of nodes, thereby constructing a topology matrix that fully encodes the interactions among all nodes. As illustrated in Figure 2, we use the first head as an example to demonstrate the process of pairwise topology modeling; the inputs are represented by $\mathbf{x}_{11} \in \mathbb{R}^{1 \times V \times C_h}$ and $\mathbf{x}_{21} \in \mathbb{R}^{1 \times V \times C_h}$. As shown in Figure 2, the V node characteristics of $\mathbf{x}_{11}$ are treated as a column and replicated V times, while the V node characteristics of $\mathbf{x}_{21}$ are treated as a row and replicated V times:

$$\begin{aligned} \mathbf{x}_{11}^{\text{exp}} &= \mathbf{x}_{11} \otimes \mathbf{1}_V^\top \in \mathbb{R}^{V \times V \times C_h}, \\ \mathbf{x}_{21}^{\text{exp}} &= \mathbf{1}_V \otimes \mathbf{x}_{21} \in \mathbb{R}^{V \times V \times C_h}, \end{aligned} \quad (3)$$

where $\mathbf{x}_{11}^{\text{exp}}$ and $\mathbf{x}_{21}^{\text{exp}}$ represent the expanded joint matrices, $\mathbf{1}_V$ represents a matrix of shape $V \times 1$ with all elements equal to one, $\mathbf{1}_V^\top$ represents the transpose of the matrix, and $\otimes$ represents matrix multiplication.

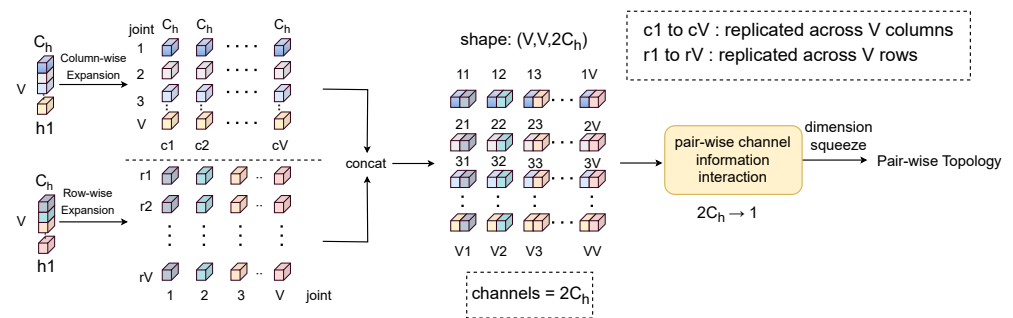


Figure 2. Pair-wise topology modeling in head1.

Then, $\mathbf{x}_{11}^{\text{exp}}$ and $\mathbf{x}_{21}^{\text{exp}}$ are concatenated along the channel dimension. In this way, the feature of each node in $\mathbf{x}_{11}^{\text{exp}}$ is concatenated with the feature of each node in $\mathbf{x}_{21}^{\text{exp}}$, including itself. As a result, all nodes from the two-stream inputs are fused along the channel dimension to form a new feature representation with shape $1 \times V \times V \times 2C_h$. Finally, a linear layer is applied to fuse the channel information of each pair of nodes, resulting in the desired spatial topology matrix:

$$\begin{aligned} \mathbf{Z}_{i,j} &= \left[\left(\mathbf{x}_{11}^{\text{exp}} \right)_{i,j} \parallel \left(\mathbf{x}_{21}^{\text{exp}} \right)_{i,j} \right] \in \mathbb{R}^{1 \times V \times V \times 2C}, \\ \mathbf{S}_{i,j} &= \mathbf{W}_s \mathbf{Z}_{i,j} \in \mathbb{R}^{1 \times V \times V}. \end{aligned} \quad (4)$$

In this process, $\parallel$ denotes the concatenation operation between each joint in $\mathbf{x}_{11}^{\text{exp}}$ and $\mathbf{x}_{21}^{\text{exp}}$. As illustrated in Figure 2, each joint in $\mathbf{x}_{11}^{\text{exp}}$ is concatenated with each joint in $\mathbf{x}_{21}^{\text{exp}}$. $\mathbf{W}_s$ represents the fusion of channel information, ensuring thorough interaction between each joint pair in the dual-stream data. Since we employ h heads, the final derived pairwise topology is denoted as $\mathbf{A}_p \in \mathbb{R}^{h \times V \times V}$.

After performing pairwise topology modeling, we further enhance the extraction of topological relationships by fusing the multi-head topology information, projecting it into a high-level representation:

$$A_d = A_p \times W_c, \quad A_d \in \mathbb{R}^{C' \times V \times V}, \quad (5)$$

where W_c denotes the weights of 2d convolution, A_p represents the topology obtained from the pairwise topology modeling, and A_d represents the final channel-wise topology obtained from the dynamic view.

3.2.2. Static View

BlockGCN [17] identified that the original skeletal connection information may be lost during the dynamic weight updating process, and proposed a relative positional encoding method to mitigate this issue. In this work, we incorporate the encoding strategy from BlockGCN as the static view to complement and enhance the dynamic view.

As illustrated in Figure 1, the numbers in the joint matrix indicate the minimum hop count between two joints on the skeletal graph $\mathcal{G}_s$. The Hop Params Table, which can be dynamically updated, provides weight parameters e_i based on the hop value $d_{i,j}$ between two joints. By selecting the corresponding hop parameter according to the hop value $d_{i,j}$, we construct a topology matrix $B \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ that preserves the original skeletal connectivity. Since the hop parameters are determined by the fixed, unchanging skeletal structure, this representation is referred to as the static view. The encoding process can be expressed as follows:

$$B_{ij} = e_{d_{i,j}} \quad \text{with} \quad d_{i,j} = \min_{D \in \text{Distances}(\mathcal{G}_s)} \{ |D|, D_1 = v_i, D_{|D|} = v_j \}, \quad (6)$$

where D_1 and D_D denote the starting point and the ending point, $|D|$ denotes all the distance values from D_1 to D_D , and hop value $d_{i,j}$ indicates the shortest distance between D_1 and D_D . We denote the topology obtained from the static view as $A_s \in \mathbb{R}^{C' \times V \times V}$,

3.2.3. Multi-View

As shown in Figures 1 and 3, by fusing the topology from the two views and refining them using the CTRGCN method, we obtain a multi-view refinement topology:

$$A_m = (A_d \times \alpha) + A_s + A[i], \quad A_m \in \mathbb{R}^{C' \times V \times V}, \quad (7)$$

where α is a learnable parameter initialized to 0, used to control the strength of A_m .

This part of feature transformation performs dimensional expansion on the original input, producing a feature tensor $\mathbf{x}_3 \in \mathbb{R}^{C' \times T \times V}$ that matches the dimension of A_m . Finally, the multi-view topology A_m is used to aggregate the features of $\mathbf{x}_3$ to produce the new feature representation:

$$\mathbf{Z} = \mathbf{E}(\mathbf{X}_3, \mathbf{A}_m) = [\mathbf{A}_{m1}\mathbf{x}_{3:,1} \parallel \mathbf{A}_{m2}\mathbf{x}_{3:,2} \parallel \cdots \parallel \mathbf{A}_{mC'}\mathbf{x}_{3:,C'}] \quad (8)$$

Here, $\mathbf{E}$ denotes the einsum summation operation. In fact, in our MTRGC, the topology from each channel of A_m is used to aggregate the features of each channel in $\mathbf{x}_3$: A_m and $\mathbf{x}_3$ for each channel are denoted as 1, 2, ..., C' , and matrix multiplication is performed between them. The symbol $\parallel$ denotes the concatenation operation. Finally, we obtain our new representation of the characteristics $\mathbf{Z} \in \mathbb{R}^{C' \times T \times V}$.

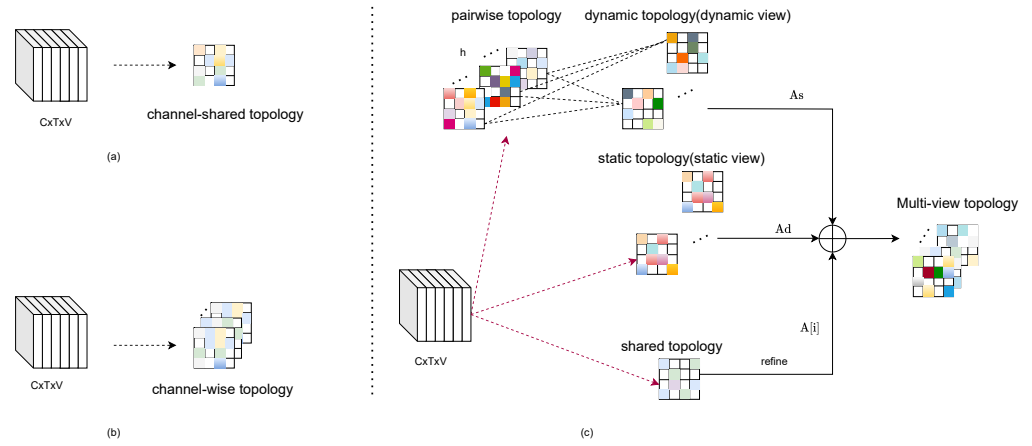


Figure 3. Overall illustration of Multi-view Topology: (a) the channel-shared topology used in ST-GCN, (b) the channel-wise topologies proposed by CTRGCN, and (c) the spatial modeling method adopted in our MTRGCN.

3.3. MultiScale Temporal Convolutional Network (MSTC)

In this section, we introduce the lightweight implementation and the Gated Channel-wise Temporal Topology (GCTT) separately.

3.3.1. Lightweight Implementation

Considering the inconsistent durations of different actions, we propose a lightweight multi-scale temporal modeling strategy to effectively capture features across various temporal spans. In contrast to [29], our method employs fewer branches. As illustrated in the right part of Figure 5, each branch initially applies a convolutional layer for reducing dimensions. The first two branches further pass through a depthwise convolution (dw Conv) and a pointwise convolution (pw Conv), followed by GCTT module to refine the features. The third branch applies a pooling operation, while the fourth branch directly retains the input features without modification. These four branches jointly extract features from multiple perspectives, substantially enhancing the richness of the feature representation.

As pointed out in Section 2.3, although previous research has attempted to utilize multi-branch structures or depthwise separable convolutions for temporal modeling, these approaches still exhibit certain deficiencies. In contrast, our proposed module achieves a lightweight and efficient design without compromising accuracy. Compared with methods such as CTRGCN [16], FR-Head [18], Koopman [26], and BlockGCN [17], which employ conventional convolution operations for feature extraction, our approach significantly reduces both parameters and computational cost.

For conventional 2d convolution:

$$\begin{aligned} \text{Parameters: } \text{Params}_{2d} &= C_{in} \times C_{out} \times k_t, \\ \text{Computational Complexity: } C_{2d} &= O(T \times V \times C_{in} \times C_{out} \times k_t), \\ \text{FLOPs: } \text{FLOPs}_{2d} &= 2 \times T \times V \times k_t \times C_{in} \times C_{out}, \end{aligned} \quad (9)$$

For our depthwise separable convolution:

$$\begin{aligned} \text{Parameters: } \text{Params}_{DSC} &= C_{in} \times k_t + C_{in} \times C_{out}, \\ \text{Computational Complexity: } C_{DTC} &= O(T \times V \times (C_{in} \times k_t + C_{in} \times C_{out})), \\ \text{FLOPs: } \text{FLOP}_{DSC} &= 2 \times T \times V \times (k_t \times C_{in} + C_{in} \times C_{out}), \end{aligned} \quad (10)$$

C_{in} denotes input channels, C_{out} denotes output channels, k_t denotes kernel size, T denotes time dimension, and V denotes number of joints. Here, $O(\cdot)$ denotes the asymptotic

computational complexity, characterizing the growth rate of computation with respect to the input size.

From Equations (9) and (10), it can be observed that our depthwise separable convolution significantly outperforms the conventional 2d convolution in terms of parameter count, computational complexity, and FLOPs, achieving substantial reductions across all three metrics.

Moreover, dilated convolutions are applied in the first two branches to enhance the receptive field. Considering that depthwise separable convolutions operate on individual channels and naturally suffer from a limited receptive field, we utilize larger kernel sizes to mitigate this drawback.

3.3.2. Gated Channel-Wise Temporal Topology Representation

While Section 3.3.1 has achieved lightweight and efficient temporal modeling, there remains potential for further improvement in temporal feature extraction. To address this, we propose Gated Channel-wise Temporal Topology (GCTT). As depicted on the right side of Figure 5, GCTT processes the outputs from the depthwise separable convolutions of the first two branches. The detailed modeling procedure is illustrated in Figure 4. Initially, we use a 1×1 convolution to change the input channel size from C to $2C$, and then apply a chunk operation along the channel dimension to split the input into q and k , where the input feature is denoted as $X \in \mathbb{R}^{C_b \times T \times V}$, and C_b gives the branch channels in MSTC. The dimensional changes are shown in the figure:

$$(Q, K) = \text{split}(\text{Conv}(X)) \quad \text{where} \quad Q, K \in \mathbb{R}^{C_b \times T \times V}, \quad (11)$$

Then, we split Q and K along the channel dimension into h heads, with each head having a feature dimension denoted as $C_{bh} = \frac{C_b}{h}$:

$$Q, K \in \mathbb{R}^{h \times C_{bh} \times T \times V}, \quad (12)$$

and subsequently, spatial pooling is performed to compute the average features over all joints, and Q is transposed:

$$Q = \frac{1}{V} \sum_{v=1}^V Q(:, :, :, v), \quad K = \frac{1}{V} \sum_{v=1}^V K(:, :, :, v), \quad (13)$$

$$Q \in \mathbb{R}^{h \times T \times C_{bh}}, \quad K \in \mathbb{R}^{h \times C_{bh} \times T} \quad (14)$$

After this, we perform matrix multiplication to model dynamic temporal relations and fuse multi-head topological information. To further enhance the feature representation, we introduce a gating mechanism and a residual connection, resulting in the proposed Gated Channel-wise Temporal Topology (GCTT) denoted as A_T :

$$A_T = \Lambda \left[\text{softmax} \left(\frac{QK}{\sqrt{C_{bh}}} \right) \right], A_T \in \mathbb{R}^{C_b \times T \times T}, \quad (15)$$

where Λ denotes a convolution layer used to fuse the information from multiple heads.

Finally, feature extraction is performed on the initial input, and our GCTT module is used to aggregate these features, resulting in a feature representation that captures temporal topological relationships:

$$Y = X + g \times (\text{einsum}('nctt, nctv \rightarrow nctv', A_T, \text{Conv}(X))) \quad (16)$$

Here, X denotes the original input, and g represents the gating coefficient. The einsum operation indicates a tensor contraction over the temporal dimension.

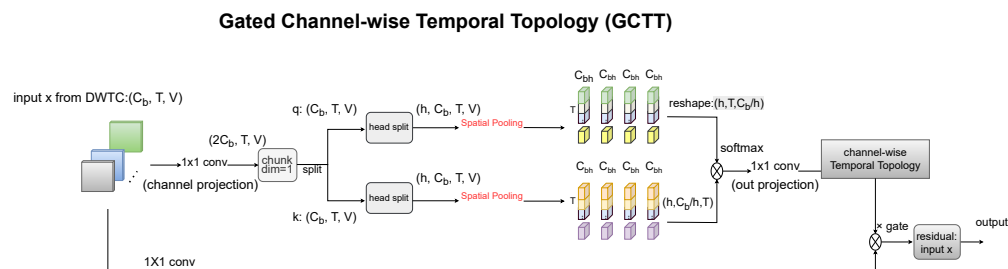


Figure 4. Gated Channel-wise Temporal Topology (GCTT).

We provide a further explanation for the gating coefficient: Our gating coefficient ‘gate’ is defined as a learnable parameter, initialized to one. As shown in Figure 4, after extracting features using the channel-wise temporal topology, the ‘gate’ controls the output. During training, the value of ‘gate’ is continuously updated. On top of the residual connection that preserves the original input information, it controls the influence of the temporal topology on the final result. In other words, the ‘gate’ can adjust the strength of the impact of the proposed GCTT on the outcome, thus finding an optimal level of influence.

3.4. Model Architecture

The basic block of our model is illustrated in the left part of Figure 5. Each Graph Convolutional Network (GCN) consists of three parallel Multi-View Topology Refinement Graph Convolution (MTRGC) modules, whose outputs are summed. Each Temporal Convolution Network (TCN) is implemented using a MultiScale Temporal Convolution (MSTC) module. A basic block is composed of one GCN and one TCN. The complete model is illustrated in Figure 6. Our model is constructed with nine basic blocks, followed by a global average pooling layer and a fully connected layer to generate the final recognition results.

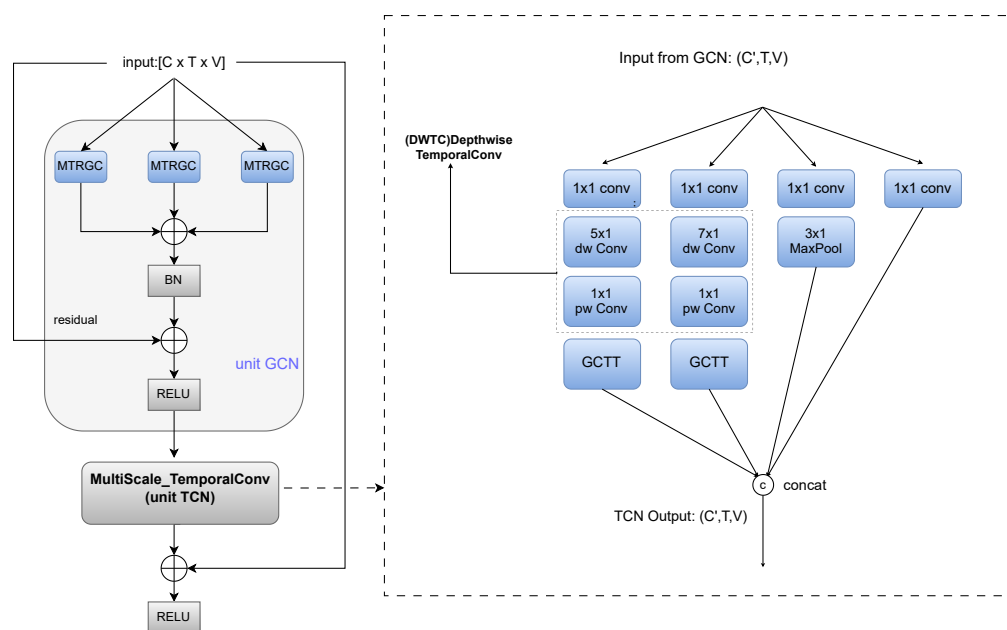


Figure 5. The left side shows our basic block, which consists of a GCN and a TCN, while the right side illustrates the structure of the TCN.

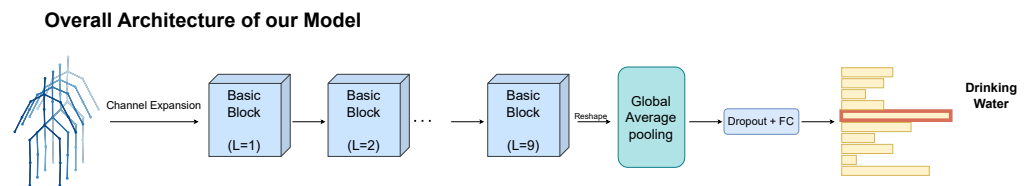


Figure 6. Overall architecture of our model.

4. Experiments

4.1. Datasets

NTU RGB + D. NTU RGB + D [21] is one of the most classical benchmarks for skeleton-based human action recognition. It contains a total of 56,880 skeleton sequences spanning 60 action categories, including daily activities, health-related actions, and person-to-person interactions. The actions were performed by 40 subjects aged between 10 and 35 and were simultaneously recorded from three different horizontal angles using Microsoft Kinect v2 sensors. Each skeleton sequence consists of up to two subjects, and each human body is represented by 25 3D joints tracked over time. To evaluate the generalization ability of models, two standard protocols are adopted: (1) Cross-Subject (X-Sub): A total of 20 subjects are used for training and the remaining 20 for testing. (2) Cross-View (X-View): Samples captured from camera views 2 and 3 are used for training, while those from camera view 1 are reserved for testing.

NTU RGB + D 120. NTU RGB + D 120 [22] is an extension of the original NTU RGB + D dataset and is currently the largest skeleton-based human action recognition benchmark. It consists of 113,945 skeleton sequences spanning 120 action classes, including both the original 60 classes and an additional 60 newly introduced ones. The actions were performed by 106 subjects and were captured using Microsoft Kinect v2 sensors across 32 distinct camera setups, each representing a different location and background environment. To ensure a fair and comprehensive evaluation, the dataset provides two standard evaluation protocols: (1) Cross-Subject (X-Sub): The subjects are evenly split, with 53 used for training and the remaining 53 for testing. (2) Cross-Setup (X-Set): The dataset is divided based on the setup IDs, where even-numbered setups are used for training and odd-numbered setups for testing.

Northwestern-UCLA. The Northwestern-UCLA dataset [23] is a multi-view skeleton-based action recognition dataset comprising 1494 video clips across 10 action classes, each performed by 10 different subjects. The data was captured simultaneously from three distinct viewpoints using Kinect cameras, providing 3D skeleton sequences composed of 20 joints per subject. The dataset follows a cross-view evaluation protocol recommended by the authors: sequences captured from the first two cameras are used for training, while those from the third camera are used for testing.

4.2. Implementation Details

All experiments were conducted on one RTX 3090 GPU with the PyTorch 2.0.0 deep learning framework. Our models are trained with Stochastic Gradient Descent (SGD) with momentum 0.9 and weight decay 0.0004. The training epoch is set to 110 and a warmup strategy is used in the first 5 epochs to make the training procedure more stable. The learning rate is set to 0.05 and decays with a factor 0.1 at epoch 90 and 100. For NTU RGB + D [21] and NTU RGB + D 120 [22], the batch size is 64, each sample is resized to 64 frames, and we adopt the data pre-processing in CTRGCN [16]. For Northwestern-UCLA [23], the batch size is 16, and we also adopt the data pre-processing in [30].

4.3. Ablation Study

The experiments in this subsection were conducted based on the X-sub benchmark of the NTU RGB + D 120 dataset [22] using the joint modality. The results demonstrate the effectiveness of each proposed GCN and TCN module.

We adopt ST-GCN [14] as our baseline, using the same official implementation provided by CTRGCN [16]. ST-GCN employs non-adaptive topology for aggregating spatial features (GCN) and utilizes conventional 2D convolutions for temporal modeling (TCN). However, it suffers from significant limitations in both topology modeling and feature extraction. Building upon ST-GCN, we introduce our model and demonstrate through extensive experiments that our approach achieves advantages of lightweight design, high computational efficiency, and superior recognition accuracy.

As shown in Table 1,

1. Adding only the dynamic topology on top of the baseline improves the accuracy by 0.4%, demonstrating that our pairwise modeling strategy for dynamically capturing spatial joint relationships is highly effective.
2. Adding only the static topology improves the accuracy by 0.3%, indicating that modeling the skeletal topology based on the distance between joints can effectively preserve static connection information.
3. Incorporating the multi-view topology leads to a 1.1% improvement in accuracy, which exceeds the combined gains from 1 and 2. This result proves that the two views are not merely additive but synergistic, validating both the necessity and effectiveness of multi-view topology modeling, as well as the generalizability and universality of the static topology.
4. After introducing the MSTC module, the accuracy slightly drops by 0.1%; however, the model size is reduced by 44% (0.99 M parameters), and FLOPs decrease by 39% (0.99 G), demonstrating the lightweight nature of the proposed design.
5. Finally, with the addition of the GCTT module, the accuracy improves by 1.5% compared to the baseline, proving that our channel-wise temporal topology modeling effectively captures the relationships between different temporal frames, further enriching temporal feature extraction based on MSTC.

Table 1. Ablation study results. ‘✓’ indicates the module is used, ‘-’ indicates it is not used, and ‘↑’ indicates the improvement in percentage points.

Model	Multi-View		MSTC	GCTT	Params	Flops	Acc(%)
	Dynamic	Static					
STGCN (baseline)	-	-	-	-	2.09 M	2.34 G	84.6
+ dynamic only	✓	-	-	-	2.25 M	2.54 G	85.0 (↑0.4)
+ static only	-	✓	-	-	2.09 M	2.34 G	84.9 (↑0.3)
+ Multi-view	✓	✓	-	-	2.26 M	2.54 G	85.7 (↑1.1)
+ MSTC	✓	✓	✓	-	1.27 M	1.55 G	85.6
Whole model	✓	✓	✓	✓	1.37 M	1.65 G	86.1 (↑1.5)

Also, we conduct experiments on the kernel sizes used in the depthwise convolutions within the MSTC module. The results in Table 2 show that using different kernel sizes for the two branches is more beneficial for capturing motion features across different temporal scales. Furthermore, in order to compensate for the limited receptive field of depthwise convolutions, we adopt a larger kernel size of 7 for the second branch.

Table 2. Ablation study on TCN kernel sizes.

Kernel Size 1	Kernel Size 2	Acc (%)
5	5	85.7
5	7	86.1
7	7	85.2
7	9	85.6
9	9	85.5

4.4. Comparison with State-of-the-Art

We follow the same fusion strategy as in CTRGCN [16] and same ensemble strategy in FRhead [18], combining four modalities: joint, bone, joint motion, and bone motion for comparison. To ensure fairness and eliminate randomness, we follow the fixed training protocol of CTRGCN. All experimental results of our model are obtained using a single RTX 3090 GPU. We compare our models with the state-of-the-art methods on NTU RGB + D 120, NTU RGB + D in Table 3 and NW-UCLA in Table 4. On the NTU120 dataset, our model outperforms the state-of-the-art methods under both cross-subject and cross-setup settings. As shown in the Figure 7, our model achieves this with significantly fewer parameters and lower computational cost. Furthermore, Table 3 demonstrates that the fusion of joint and bone modalities in our model achieves performance comparable to the best results. As shown in Table 4, our model also surpasses a wide range of state-of-the-art methods on the NW-UCLA dataset. The comparison with state-of-the-art methods demonstrates that our model is a lightweight, efficient, and high-performance solution.

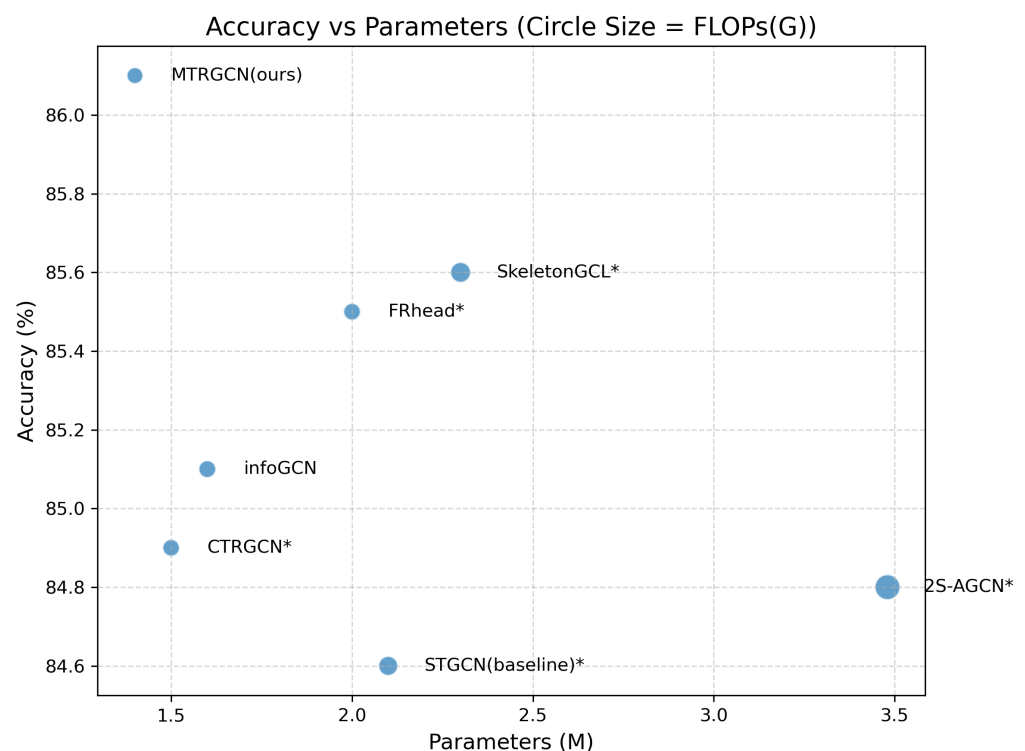


Figure 7. Comparison of Parameters and FLOPs with state-of-the-art methods, where ‘*’ indicates that the results are implemented based on the released code. For fair comparison, we use the results based on the joint modality of NTU RGB + D 120 (X-Sub) without considering the influence of ensemble weights.

Table 3. Comparison with state-of-the-art methods on NTU60 and NTU120 datasets (2S: fusion of joint and bone streams; 4S: fusion of joint, bone, joint-motion, and bone-motion streams).

Methods	NTU60-XSub	NTU60-XView	NTU120-XSub	NTU120-XSet
STGCN [14]	81.5	88.3	—	—
SGN [31]	89.0	94.5	79.2	81.5
AS-GCN [32]	86.8	94.2	—	—
2s-AGCN [15]	88.5	95.1	—	—
DGNN [13]	89.9	96.1	—	—
Shift-GCN [30]	90.7	96.5	85.9	87.6
MS-G3D [29]	91.5	96.2	86.9	88.4
Dynamic-GCN [33]	91.5	96.0	87.3	88.6
MST-GCN [20]	91.5	96.6	87.5	88.8
CTRGCN [16]	92.4	96.8	88.9	90.6
InfoGCN (4S) [24]	—	—	89.4	90.7
Efficient-G4 [19]	92.1	—	88.7	88.9
FRhead [18]	92.8	96.8	89.5	90.9
MTR-GCN (2S)	92.3	96.4	89.2	90.4
MTR-GCN (4S)	92.8	96.8	89.6	90.8

Table 4. Comparison with state-of-the-art methods on NW-UCLA dataset.

Methods	NW-UCLA (%)
Lie Group [34]	74.2
HBRNN-L [35]	78.5
Actionlet Ensemble [36]	76.0
Ensemble TS-LSTM [37]	89.2
AGC-LSTM [38]	93.3
Shift-GCN [30]	94.6
DC-GCN + ADG [39]	95.3
MTR-GCN (Ours)	95.7

5. Conclusions

This work addresses the limitations in spatial and temporal modeling for skeleton-based action recognition. For spatial modeling, we propose the Multi-view Topology Refinement Graph Convolution (MTRGC), which integrates both dynamic and static perspectives to overcome the issues of catastrophic forgetting of skeletal topology and insufficient relational modeling capacity in conventional GCNs. Experimental results demonstrate that MTRGC achieves a synergistic effect—greater than the sum of its individual views—rather than a simple additive gain. For temporal modeling, we introduce the MultiScale Temporal Convolution (MSTC), which enables lightweight design without compromising accuracy; building on this, we propose Gated Channel-wise Temporal Topology (GCTT) to perform topological modeling along the temporal dimension, effectively enhancing temporal feature extraction.

Our model achieves state-of-the-art performance across multiple benchmarks. However, there still exists the issue of incomplete feature extraction. It remains a challenge whether better skeleton features can be extracted using methods other than topology modeling, or if improvements can be made in data preprocessing. These are the challenges we face. Future work may focus on further improving training efficiency and exploring more advanced multi-relational modeling techniques.

6. Visualization

As shown in Figure 8, the Static Topo exhibits a clear diagonal pattern, indicating strong self-connections. The presence of brighter regions near the diagonal further suggests that it effectively captures the relationships between adjacent joints. Overall, its structure is regular and well-aligned with the true human skeleton topology. In contrast, the Dynamic Topo appears more random, reflecting a topology learned entirely from inter-joint relations without relying on any prior skeletal structure. Our Multi-view Topo retains the prior structural knowledge of the original skeleton while dynamically learning the global topology. This results in a more regular and smoother distribution of weights, effectively reducing the impact of noise.

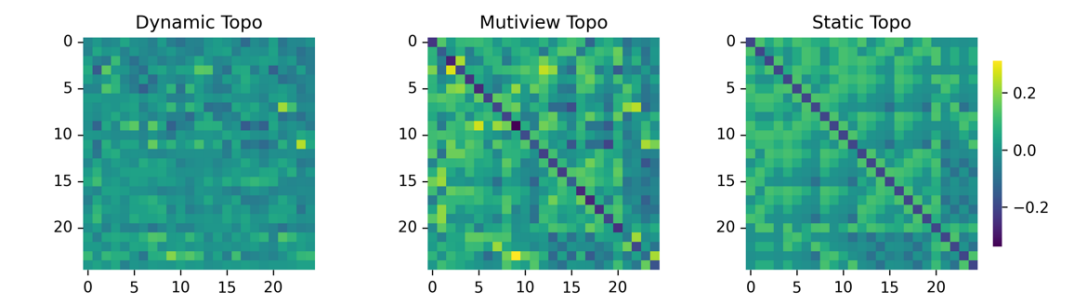


Figure 8. The topological variation of the same MTRGC within a single layer.

As shown in Figure 9, we visualize the static topology across different layers. It can be observed that our static topology is continuously optimized, exhibiting distinct representational characteristics at each layer. Through iterative refinement and updates, the model ultimately learns an optimal static topology representation.

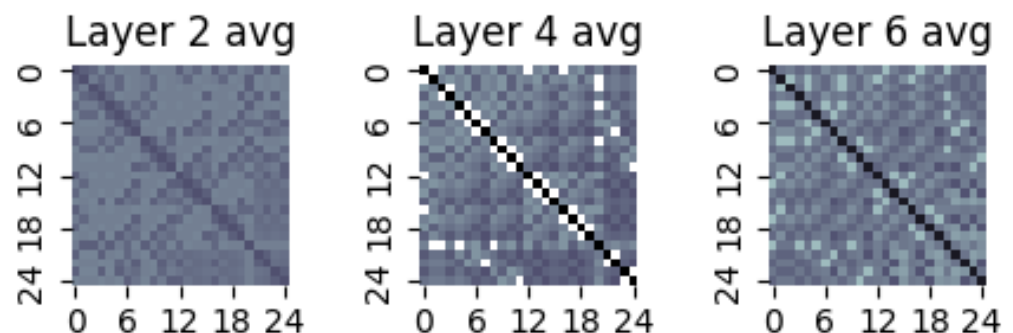


Figure 9. The variation of static topology across different layers.

Author Contributions: L.W.: Investigation, conceptual innovation, experiments, data analysis, manuscript writing. X.Z.: Data analysis, collaborative experiments, assistance with innovation. C.Z.: Project initiation, funding acquisition, critical revision, guidance on key innovations. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No.62272234).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the first author or corresponding author.

Conflicts of Interest: The authors declare no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Appendix A.1

Table A1. Acronym table.

Acronym	Full Form
GCN	Graph Convolutional Network
TCN	Temporal Conbolutional Nwtwork
MTRGC	Multi-view Topology Refinement Graph Convolution
MSTC	MutiScale Temporal Convolution
GCTT	Gated Channel-wise Temporal Topology

References

1. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
2. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 568–576.
3. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. SlowFast networks for video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6202–6211.
4. Do, J.; Kim, M. Skateformer: Skeletal-temporal transformer for human action recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Milan, Italy, 29 September–October 4; pp. 401–420.
5. Ray, A.; Raj, A.; Kolekar, M.H. Autoregressive Adaptive Hypergraph Transformer for Skeleton-based Activity Recognition. In Proceedings of the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 26 February–6 March 2025; pp. 9690–9699.
6. Liu, Q.; Wu, Y.; Li, B.; Ma, Y.; Li, H.; Yu, Y. SHoTGCN: Spatial high-order temporal GCN for skeleton-based action recognition. *Neurocomputing* **2025**, *632*, 129697. [[CrossRef](#)]
7. Ni, B.; Wang, G.; Moulin, P. RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 1147–1153.
8. Gao, C.; Du, Y.; Liu, J.; Lv, J.; Yang, L.; Meng, D.; Hauptmann, A.G. INFAR Dataset: Infrared action recognition at different times. *Neurocomputing* **2016**, *212*, 36–47. [[CrossRef](#)]
9. Hussein, M.E.; Torki, M.; Gowayyed, M.A.; El-Saban, M. Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), Beijing, China, 3–9 August 2013; pp. 2466–2472.
10. Wang, H.; Wang, L. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 499–508.
11. Liu, J.; Shahroudy, A.; Xu, D.; Wang, G. Spatio-temporal LSTM with trust gates for 3D human action recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 September 2016; pp. 816–833.
12. Li, C.; Zhong, Q.; Xie, D.; Pu, S. Skeleton-based action recognition with convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Hong Kong, China, 10–14 July 2017; pp. 597–600.
13. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Skeleton-based action recognition with directed graph neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7912–7921.
14. Yan, S.; Xiong, Y.; Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32, No. 1.
15. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 12026–12035.
16. Chen, Y.; Zhang, Z.; Yuan, C.; Li, B.; Deng, Y.; Hu, W. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 13359–13368.

17. Zhou, Y.; Yan, X.; Cheng, Z.Q.; Yan, Y.; Dai, Q.; Hua, X.S. BlockGCN: Redefine topology awareness for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 2049–2058.
18. Zhou, H.; Liu, Q.; Wang, Y. Learning discriminative representations for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 10608–10617.
19. Song, Y.F.; Zhang, Z.; Shan, C.; Wang, L. Constructing stronger and faster baselines for skeleton-based action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 1474–1488. [[CrossRef](#)]
20. Chen, Z.; Li, S.; Yang, B.; Li, Q.; Liu, H. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, No. 2, pp. 1113–1122.
21. Shahroudy, A.; Liu, J.; Ng, T.T.; Wang, G. NTU RGB+D: A large scale dataset for 3D human activity analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1010–1019.
22. Liu, J.; Shahroudy, A.; Perez, M.; Wang, G.; Duan, L.Y.; Kot, A.C. NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 2684–2701. [[CrossRef](#)] [[PubMed](#)]
23. Wang, J.; Nie, X.; Xia, Y.; Wu, Y.; Zhu, S.C. Cross-view action modeling, learning and recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 2649–2656.
24. Chi, H.G.; Ha, M.H.; Chi, S.; Lee, S.W.; Huang, Q.; Ramani, K. InfoGCN: Representation learning for human skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 20186–20196.
25. Wu, Z.; Sun, P.; Chen, X.; Tang, K.; Xu, T.; Zou, L.; Weise, T. SelfGCN: Graph convolution network with self-attention for skeleton-based action recognition. *IEEE Trans. Image Process.* **2024**, *33*, 4391–4403. [[CrossRef](#)]
26. Wang, X.; Xu, X.; Mu, Y. Neural Koopman Pooling: Control-inspired temporal dynamics encoding for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 10597–10607.
27. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861. [[CrossRef](#)]
28. Liu, D.; Chen, P.; Yao, M.; Lu, Y.; Cai, Z.; Tian, Y. TSGCNeXt: Dynamic-static multi-graph convolution for efficient skeleton-based action recognition with long-term learning potential. *arXiv* **2023**, arXiv:2304.11631.
29. Liu, Z.; Zhang, H.; Chen, Z.; Wang, Z.; Ouyang, W. Disentangling and unifying graph convolutions for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 143–152.
30. Cheng, K.; Zhang, Y.; He, X.; Chen, W.; Cheng, J.; Lu, H. Skeleton-based action recognition with shift graph convolutional network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 183–192.
31. Zhang, P.; Lan, C.; Zeng, W.; Xing, J.; Xue, J.; Zheng, N. Semantics-guided neural networks for efficient skeleton-based human action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 1112–1121.
32. Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; Tian, Q. Actional-structural graph convolutional networks for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3595–3603.
33. Ye, F.; Pu, S.; Zhong, Q.; Li, C.; Xie, D.; Tang, H. Dynamic GCN: Context-enriched topology learning for skeleton-based action recognition. In Proceedings of the 28th ACM International Conference on Multimedia (ACM MM), Seattle, WA, USA, 12–16 October 2020; pp. 55–63.
34. Veeriah, V.; Zhuang, N.; Qi, G.J. Differential recurrent neural networks for action recognition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4041–4049.
35. Du, Y.; Wang, W.; Wang, L. Hierarchical recurrent neural network for skeleton-based action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1110–1118.
36. Wang, J.; Liu, Z.; Wu, Y.; Yuan, J. Learning actionlet ensemble for 3D human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 914–927. [[CrossRef](#)] [[PubMed](#)]
37. Lee, I.; Kim, D.; Kang, S.; Lee, S. Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1012–1020.

38. Si, C.; Chen, W.; Wang, W.; Wang, L.; Tan, T. An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 1227–1236.
39. Cheng, K.; Zhang, Y.; Cao, C.; Shi, L.; Cheng, J.; Lu, H. Decoupling GCN with DropGraph module for skeleton-based action recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 536–553.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.