Spatio-Temporal Fusion for Human Action Recognition via Joint Trajectory Graph

Yaolin Zheng[1], **Hongbo Huang**[1, 2], **Xiuying Wang**[1, 2], **Xiaoxu Yan**[1], **Longfei Xu**[1]

[1]Computer School, Beijing Information Science & Technology University, Beijing, China
[2]Institute of Computing Intelligence, Beijing Information Science & Technology University, Beijing, China
zhengyaolin574@bistu.edu.cn, hhb@bistu.edu.cn, wxiuying520@126.com, 2021020568@bistu.edu.cn, 2022020601@bistu.edu.cn

Abstract

Graph Convolutional Networks (GCNs) and Transformers have been widely applied to skeleton-based human action recognition, with each offering unique advantages in capturing spatial relationships and long-range dependencies. However, for most GCN methods, the construction of topological structures relies solely on the spatial information of human joints, limiting their ability to directly capture richer spatio-temporal dependencies. Additionally, the self-attention modules of many Transformer methods lack topological structure information, restricting the robustness and generalization of the models. To address these issues, we propose a Joint Trajectory Graph (JTG) that integrates spatio-temporal information into a uniform graph structure. We also present a Joint Trajectory GraphFormer (JT-GraphFormer), which directly captures the spatio-temporal relationships among all joint trajectories for human action recognition. To better integrate topological information into spatio-temporal relationships, we introduce a Spatio-Temporal Dijkstra Attention (STDA) mechanism to calculate relationship scores for all the joints in the JTG. Furthermore, we incorporate the Koopman operator into the classification stage to enhance the model's representation ability and classification performance. Experiments demonstrate that JT-GraphFormer achieves outstanding performance in human action recognition tasks, outperforming state-of-the-art methods on the NTU RGB+D, NTU RGB+D 120, and N-UCLA datasets.

Introduction

Human action recognition aims to accurately identify and classify different human actions from input videos or sequence data. As an important task in computer vision, human action recognition has been extensively researched and widely applied in fields such as human-computer interaction, intelligent surveillance, and motion reconstruction. In particular, skeleton-based methods are less affected by changes in lighting conditions, background clutter, and occlusions. This robustness enhances the model's ability to focus on motion-related information, making skeleton-based methods increasingly popular.

Currently, two popular deep learning models, Graph Convolutional Networks (GCNs) and Transformers, have demonstrated strong performance in skeleton-based human action recognition tasks. GCNs treat human joint data as graph structures and employ convolutional operations based on adjacency matrices to capture spatial dependencies between different body parts, thereby improving the accuracy of action recognition (Yan, Xiong, and Lin 2018; Shi et al. 2019b; Ye et al. 2020). Additionally, Transformers have demonstrated remarkable success due to their strong self-attention mechanisms and capability to capture long-range dependencies (Vaswani et al. 2017; Dosovitskiy et al. 2020).

Despite the promising performance of GCNs and Transformers, accurate and robust skeleton-based action recognition remains a challenging task. This is primarily due to several factors. Firstly, conventional GCN methods do not directly utilize spatio-temporal topology to capture more

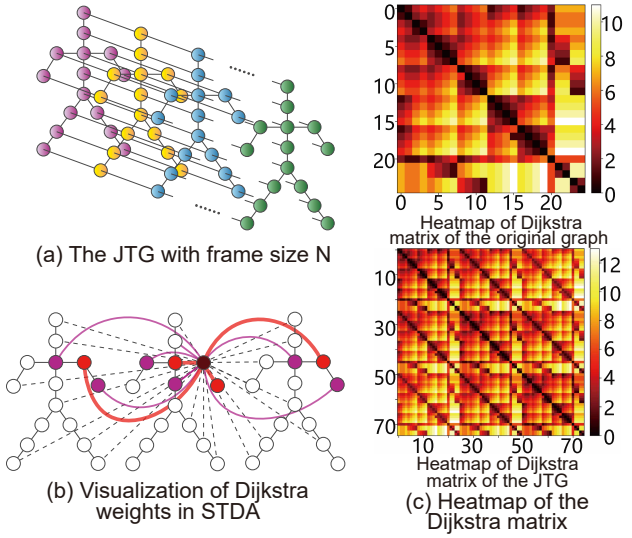

Figure 1: Visualization and Heatmaps. (a) Visualization depicting the Joint Trajectory Graph (JTG). (b) Visualization of Dijkstra weights within the Spatio-Temporal Dijkstra Attention (STDA). The thicker curves denote higher correlation weights among interconnected joints. For illustrative purposes, lower correlation weights are depicted using dotted lines and are selectively sampled. (c) Heatmaps showing Dijkstra matrices for the original human joint graph and the proposed JTG.

comprehensive spatio-temporal dependencies. They aggregate information from neighboring nodes in the graph to update node representations, which is effective for capturing spatial dependencies. However, simply extending the spatial graph is not sufficient for effectively capturing temporal dynamic correlations. Secondly, in joint coordinate sequences, the density of information may vary between the spatial and temporal dimensions, with greater redundancy in the temporal dimension. Finally, although self-attention mechanisms can adaptively compute correlation scores for sequence elements, they may not be able to capture the hidden topological information of each sequence element, leading to a negative impact on the model's robustness and generalization.

Motivated by these observations, we propose a Joint Trajectory GraphFormer (JT-GraphFormer) model with a Joint Trajectory Graph (JTG). The JTG introduces a temporal dimension on top of the original spatial graph structure, enabling it to better encapsulate complex discriminative details associated with joint trajectories. Unlike ST-GCN(Yan, Xiong, and Lin 2018), the proposed JTG focuses on constructing the topological structure between joints within a certain spatio-temple period. This enhancement greatly enriches its potential for action recognition. Specifically, we construct a dynamic trajectory topology for all joints within a certain frame sequence, as shown in Fig. 1 (a).

For action recognition tasks, an action is usually described as a temporal sequence, characterized by temporal order and dynamic evolution. To more effectively capture the intricate spatio-temporal inter-dependencies, JTG extends connections to nodes in neighboring frames. This strategy serves to reduce redundant temporal information and utilize a uniform graph structure to capture the inherent dependencies within the spatio-temporal dimension, facilitating aggregation of features across both spatial and temporal domains.

To extract features more effectively, we utilize an improved Transformer structure. When JTG is used as input data, a node within a single frame will concurrently compute the relationship scores for all nodes within neighboring frames, imposing a strong requirement on the model to handle complex spatio-temporal associations. Inspired by the spatial encoding in Graphormer (Ying et al. 2021), we propose a Spatio-Temporal Dijkstra Attention (STDA) mechanism, which adds the distances between joints in JTG as spatio-temporal topology information to the calculation of attention scores. This enables each node to learn to pay more attention to neighbor nodes that are more relevant to the action. Unlike Graphormer's method of encoding discrete data, our method is more suitable for processing continuous data such as human action recognition. STDA combines the global attention score and shortest path weights and shows stronger expressive power by adding prior information present in the joint trajectory. The correlation weights of a node to its neighbors are shown in Fig. 1 (b) and the heatmaps of Dijkstra matrices are shown in Fig. 1 (c).

Furthermore, we introduce the Koopman operator into the classification stage. The Koopman operator is a linear operator that describes a nonlinear dynamical system by mapping it into an infinite-dimensional Hilbert space. In our approach, the Koopman operator is employed to classify extracted features by learning feature evolution of the different categories. Specifically, the Koopman operator serves to linearize extracted features within either the temporal or spatial dimension and characterizes dynamic shifts inherent to different action categories for effectively capturing dynamic interrelationships of trajectories. This enhances the robustness and generalization ability of the model.

Our contributions can be succinctly summarized as follows:

- Introduction of JTG as an input data representation, leveraging trajectory information to enrich feature aggregation capabilities for nodes and their interactions across frames.
- Proposal of STDA, augmenting feature aggregation among neighboring nodes via the integration of shortest path concepts between joints.
- Incorporation of the Koopman operator for classification, facilitating an encompassing perspective and superior classification performance.
- Rigorous evaluation of our proposed model across three diverse datasets (NTU RGB+D, NTU RGB+D 120, and N-UCLA), revealing its superiority over existing state-of-the-art (SOTA) methods and underscoring its potential as a promising solution for action recognition tasks.

Related Work

GCNs for Action Recognition. The extracted skeleton sequences from videos exhibit non-Euclidean characteristics, and the inherent interconnections among individual joints can be succinctly represented using a graph structure. Representing skeleton data as simple vectors does not fully capture the complex configurations and correlations. In contrast, the topological graph representation may be more suitable for this purpose. As a result, several GCN-based approaches have been emerged to process skeleton data as graphs. For instance, ST-GCN introduces spatio-temporal GCN for skeleton-based human action recognition (Yan, Xiong, and Lin 2018), automatically learning spatial and temporal patterns from skeleton data. Specifically, it estimates pose information from input videos and achieves action representation with strong generalization capability through graphs. However, it focuses solely on the relationships between physically adjacent joints, neglecting implicit joint co-occurrence correlations.

Numerous methods have been proposed in an effort to address this limitation. Actional-Structural GCN (AS-GCN) (Li et al. 2019) combines action links and structure links into a generalized skeleton graph, with a greater emphasis on dependencies between non-physically adjacent joints. A dual-stream adaptive GCN, named 2s-AGCN (Shi et al. 2019b), explores the fusion of diverse input modalities. It utilizes both bone vectors and joint coordinates as inputs, inspiring further research on multiple joint modalities data (Shi et al. 2019a; Ye et al. 2020; Zhang et al. 2020).

In recent years, Channel-wise Topology Refinement GCN (CTR-GCN) has demonstrated promising outcomes in the context of dynamic topology and multi-channel feature

modeling (Chen et al. 2021a). CTR-GCN employs a shared topology matrix as a universal prior for channels and subsequently refines it through the inference of channel-specific correlations. Furthermore, InfoGCN based on information bottleneck learning utilizes attention-based graph convolutions to deduce contextually relevant skeleton topology (Chi et al. 2022). This approach guides the model in acquiring potential representations that are information-rich yet structurally compact.

Transformers for Action Recognition. The capability of Transformers to model long-range dependencies results in their successful application in modeling and classifying human action sequence data. The Spatio-Temporal Transformer network (ST-TR) utilizes spatial and temporal self-attention modules to learn intra-frame joint interactions and motion dynamics, leading to enhanced outcomes through integrating the two input streams (Plizzari, Cannici, and Matteucci 2021). The Spatio-Temporal Tuple Transformer (STTFormer) investigates interdependencies among distinct sequence segments (Qiu et al. 2022). It divides a skeleton sequence into non-overlapping units and subsequently captures multi-joint dependencies between neighboring frames via spatio-temporal self-attention modules, followed by a feature aggregation module for sub-action fusion.

To handle variable-length skeleton inputs without requiring additional preprocessing, a Sparse Transformer-based Action Recognition (STAR) is proposed (Shi et al. 2021). STAR consists of two modules. The first module is a sparse self-attention module that learns spatial relationships using sparse matrix multiplication. The second module is a segment-wise linear self-attention module that models temporal co-dependencies. Additionally, to incorporate more comprehensive skeleton information, the Intra-Inter-Part Transformer (IIP-Transformer) utilizes part-level skeleton data encoding for action recognition (Wang et al. 2021), considering human body joints as five distinct parts to capture dependencies within and between them.

It can be observed that the majority of research efforts are inclined towards handling sequences or modeling spatial topological structures, while the exploration of spatio-temporal topological structures remains relatively scarce. The JT-GraphFormer complements the research in this area, utilizing the trajectory topology and a unified graphical structure known as JTG to unravel the inherent dependencies within trajectories across the spatio-temporal dimension.

Method

Undoubtedly, spatial and temporal information are both crucial for representing human actions. Structures that integrate these two types of features can capture more accurate and richer spatio-temporal dependencies. Therefore, we propose a JTG, which models the spatio-temporal topological structures of the joint trajectories to capture potential spatio-temporal dependencies. Moreover, we propose a STDA mechanism and a simple sequential aggregation module named TCN to augment feature aggregation among neighboring nodes and model the temporal dependencies. These two modules form a JT-GraphFormer block for feature extraction. Furthermore, we introduce the Koopman operator in the classification stage to globally linearize the feature functions and fit the dynamic changes of various action categories. The overall process is illustrated in Figure 2.

Joint Trajectory Graph

We divide the action sequences into several groups. Each group has N frames and describes the joint trajectories with a graph structure, named Joint Trajectory Graph $G_{JT} = (G_t, G_{t+1}, \ldots, G_{t+N-1}, E_T) = (V_{JT}, E_{JT})$, where G_t is a spatial graph of joints in a frame, E_T is the corresponding set of edges, denoting the joint trajectory of the nodes in N frames, and V_{JT}, E_{JT} denote the set of nodes and edges in JTG, respectively. To understand JTG more clearly, we represent it as (1).

$$A_{ST} = \begin{bmatrix} A & A+I & A & \cdots & A \\ A+I & A & A+I & \ddots & \vdots \\ A & A+I & \ddots & \ddots & A \\ \vdots & \ddots & \ddots & A & A+I \\ A & \cdots & A & A+I & A \end{bmatrix}, \quad (1)$$

where A_{JT} is an adjacency matrix of a JTG, A denotes the physical connectivity of all joints in a frame, and I is a unit diagonal matrix indicating the connectivity of the same joints in neighboring frames.

JT-GraphFormer

Positional Encoding. In previous work, a human skeleton graph sequence is usually represented as a joint feature vector $X \in \mathbb{R}^{C \times T \times V}$, where T is the number of frames, C is the number of channels, and V is the number of joints. In this paper, the vector X of JTGs with N frames is denoted as $X \in \mathbb{R}^{C \times T/N \times V*N}$.

In many Transformer structures, embedding map linearly transforms the joint features into a vector with learnable parameters. However, such vectors lack the position information of the joints and have trouble in distinguishing the sequential order of the joints in the subsequent parallel computation (Qiu et al. 2022).

In JTG, the trajectories of the joints involve specific temporal information. Following (Vaswani et al. 2017), we add position encoding (PE) for each frame to express the sequential relationship correctly. PE uses sine and cosine functions with different frequencies to incorporate the inter- and intra-frame position information, as Eq. (2).

$$\begin{aligned} PE(p, 2i) &= \sin(p/10000^{2i/C_{in}}), \\ PE(p, 2i+1) &= \cos(p/10000^{2i/C_{in}}), \end{aligned} \quad (2)$$

where p is the position of the joints in a JTG, i is the dimension of the position encoding vector, and C_{in} is the feature dimension.

STDA Module. To better utilize the spatial information of a graph structure, Graphormer (Ying et al. 2021) adds spatial encoding to compute the self-attention of the nodes. Inspired by this, we propose a STDA mechanism, which adds the

spatio-temporal topological information to the multi-head attention mechanism, increasing the weight of associations between neighboring nodes, thus making the nodes more biased towards aggregating the features of the local neighbors. We compute the Dijkstra matrix $D \in \mathbb{Z}+^{V*N\times V*N}$ of JTG to describe its topology information, where $D_{ij} = D_{ji}$. Then, we compute the topological weights W via Eq. (3):

$$W = \exp(-D) + b, \tag{3}$$

where $-D$ stands for inverting all the entries in the D matrix, $\exp(\cdot)$ computes the exponential values for all the entries in the matrix, and $b \in \mathbb{R}^{V*N\times V*N}$ is a learnable matrix for learning the adaptive inter-joint dynamic weights. W will be multiplied element-wisely with the attention map a_{map} obtained by the self-attention calculation, as:

$$\begin{aligned} a_{map} &= Tanh(QK^T/\sqrt{d_K} \times \alpha), \\ a_{score} &= a_{map} \cdot W, \end{aligned} \tag{4}$$

where Q,K are the query and key vector in the self-attention calculation, which performs a 1×1 convolution operation on the input. d_k represents the dimension of K, α is a learnable parameter that assigns adaptive weights to different heads, a_{score} is the final weighted attention score. In the forward propagation, the STDA output is obtained by a FFN structure and a residual structure via Eq. (5).

$$STDA(H^l) = \sigma(FFN(a_{score}H^l)) + res(H^l)), \tag{5}$$

where H^l is the input feature of the lth block. σ is a activation function, we use the Leaky ReLU function here. The FFN structure contains a $1 \times k_s$ convolution operation and a Batch Normalization (BN) operation. The res represents the residual operation and consists of a 1×1 convolution operation and a BN operation.

TCN Module. For convenience of understanding, we name the sequential aggregation module as TCN, which aims to aggregate features of the joint trajectories and consists of a $k_t \times 1$ convolution operation and a BN operation. The input $H_{in} \in \mathbb{R}^{C_l\times T/N\times V*N}$ is reshaped to the output $H_{out} \in \mathbb{R}^{C_l\times T\times V}$ during this process, where C_l denotes the output channel number of the lth block. The residual operation res is utilized in both the input and output stages, as shown in Fig. 2.

Koopman Operator

The Koopman operator is a linear operator that describes a nonlinear dynamical system by mapping it into an infinite-dimensional Hilbert space. This mapping allows the system's evolution to be depicted in a linear space, which can be easier to analyze than the original nonlinear space (Proctor, Brunton, and Kutz 2018).

In deep learning, the Koopman operator can be utilized to extract evolution features of nonlinear dynamical systems for enhancing classification performance. In this study, we establish the temporal evolution function (for illustrative purposes, we take the temporal connection as an example) $f(\cdot)$ for the JT-GraphFormer's output feature H across distinct frames to relate the feature h_t at the tth frame to the feature h_{t+1} at the next frame step, i.e., $h_{t+1} = f(h_t)$.

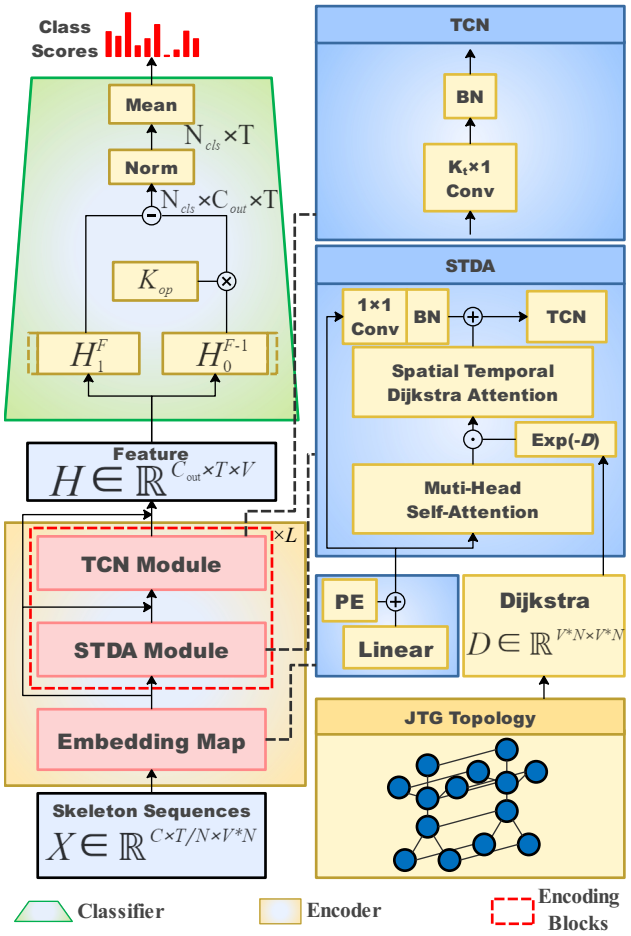

Figure 2: JT-GraphFormer architecture. The model is composed of an encoder and a classifier. The encoder with the STDA module captures context-dependent spatio-temporal joint topology to better represent action. The STDA and TCN modules form a JT-GraphFormer block for feature extraction.

We define the Koopman operator K_{op} as a $N_{cls} \times C_{out} \times C_{out}$ linear operator, where N_{cls} denotes the number of action categories, and C_{out} denotes the output channel amount of the last JT-GraphFormer block. K_{op} applies a linear approach to approximate the interrelations among various categories of action features in the temporal dimension, satisfying Eq. (6).

$$h_{t+1} \approx K_{op}h_t \tag{6}$$

Since we establish the linear correlations at different frame steps, it is possible to approximate the representation of any continuous frame segment feature H_x^y, which denotes the feature segment from the xth frame to the yth frame. Thus the features H_1^{T-1} can be represented as:

$$H_1^{T-1} \approx [h_1, K_{op}h_1, K_{op}^2h_1, \cdots, K_{op}^{T-2}h_1] \tag{7}$$

According to Eq. (6), it can be deduced that:

$$H_{t+1}^{T} \approx K_{op} H_{t}^{T-1} \tag{8}$$

We adopt the DMD algorithm (Kutz et al. 2015) and minimize the Frobenius norm of $\|H_2^T - K_{op}H_1^{T-1}\|_2$ to update the K_{op}. Since K_{op} denotes the feature evolution of various action categories, we can average the K_{op} in the temporal dimension to get the probability distribution of each category and finally complete the classification, as shown in Fig. 2.

Four-stream Ensemble

Previous studies demonstrate that simultaneously using different streams can significantly enhance the performance of human action recognition (Shi et al. 2019b,a). Therefore, we evaluate the performance of the trained models utilizing streams for joint, bone, joint motion, and bone motion. The bone stream utilizes bone modality as input data, proposed by (Shi et al. 2019b). The joint motion and the bone motion streams align with the method presented in (Shi et al. 2019a). The ultimate result is calculated through a weighted average of the models' inference outputs.

Experiments

To demonstrate the advantages of the proposed JT-GraphFormer, we conducted comprehensive experiments on the NTU-60, NTU-120 and N-UCLA datasets. Furthermore, we performed a comparative analysis with current SOTA models and detailed ablation studies to explore the performance of the proposed modules under various conditions.

Datasets

NTU RGB+D (NTU-60) contains 56880 skeleton action sequences in 60 classes (Shahroudy et al. 2016). Each sample contains one action, and each action is performed by up to two subjects and captured by three cameras from different views. The dataset is divided into two test benchmarks based on different subjects and different views, i.e., cross-subject (XSub) and cross-view (XView).

NTU RGB+D 120 (NTU-120) extends NTU RGB+D with 114480 samples in 120 classes (Liu et al. 2019). The dataset was also taken by three cameras and contains 32 settings, each indicating a specific location and background. The dataset is divided into two benchmarks based on parity across subjects and sample IDs, i.e., cross-subject (XSub) and cross-setup (XSet).

Northwestern-UCLA (N-UCLA) contains 1494 video clips in 10 classes. Each action is performed by 10 subjects and captured through three cameras with different camera views. The same evaluation protocol in (Wang et al. 2014) is adopted.

Experimental Setting

All experiments are performed on 2 GTX 3090 GPUs. The skeleton sequences processed as in (Chen et al. 2021a) are resized to 120, 120, 56 frames for NTU-60, NTU-120, and N-UCLA, respectively. No other data processing or augmentation is applied for fair comparisons. Our model is trained utilizing a stochastic gradient descent (SGD) optimizer with a Nesterov momentum of 0.9, and the weight decay is set to 0.0005. The Cross-entropy is taken as the loss function. The training epoch is set to 110 for NTU-60 & 120, and to 30 for N-UCLA. The initial learning rate is 0.1, and a warm up strategy is employed in the first 5 epochs for more stable learning (He et al. 2016). For the NTU-60, NTU-120, and N-UCLA datasets, the learning rate is decayed at epochs on [60, 80, 100], [60, 80, 100], [15], and the batch size is set to 64, 64 and 16, respectively. The JT-GraphFormer block amount is set to 8, and the output channel amounts are [64, 64, 64, 128, 128, 256, 256, 256]. The dimensions d_q, d_k, d_v in each layer are set to $0.25 \times C_{out}$. Convolution kernels k_t, k_s are set to [3, 5]. The shape of K_{op} is [N_{cls}, 256, 256], where N_{cls} denotes the number of categories in the dataset.

frame count N	Top-1 Accuracy (%)
1 (baseline)	87.4
2	89.4 (↑2.0)
4	90.0 (↑2.6)
6	90.4 (↑3.0)
8	89.7 (↑2.3)
fusion (2,4,6,8)	**91.7** (↑4.3)

Table 1: Top-1 accuracy using joint input modality with different frame count N on the NTU-60 dataset under the x-sub setting. The JTG is configured to be dynamic and inter-layer weight-sharing without normalization.

Ablation Studies

To analyze the impact of the different components of the proposed JT-GraphFormer, we examine the performance of our model under different configurations and conditions.

Number of Frames in JTG. Different number of frames in JTG means different sequence and information densities in the temporal dimension, which can affect the effectiveness of STDA and TCN modules. In this regard, we experimentally analyze the effect of the different frame count N to the accuracy. We also explore the performance of fusing models with multiple frame counts. The comparison results are shown in Table 1, with the best performance marked in **bold**.

It can be observed that at the frame count 6, the accuracy of the model on the NTU-60 dataset under the X-Sub setting increases by margins of 3.0% compared to the baseline model, and 0.7% compared to the frame count 8. This phenomenon is consistent with intuitive expectations. In scenarios with excessively simplistic spatio-temporal structures, the model struggles to capture intricate spatio-temporal features. Conversely, within complex spatio-temporal graph structures, the spatio-temporal correlation of actions tends to diminish, thereby affecting the extraction of effective features.

The model with fusion of different frame count can compensate for the need of different actions for the number of adapted frames to some extent, which shows a performance improvement of 1.3% compared to the frame count 6 and 4.3% compared to the baseline.

method	dynamic	normalized	shared	Acc. (%)
baseline				89.8
M_1		✓		89.7
M_2	✓			90.0
M_3	✓	✓		90.0
M_4	✓		✓	**90.4**
M_5	✓	✓	✓	89.8

Table 2: Ablation study of Top-1 accuracy (%) using the JTG with different settings, where baseline represents the method that does not apply these settings.

Dynamic and Weight-sharing. We investigated the performance differences between the static and dynamic configuration of the JTG structure, as well as whether inter-layer weight-sharing is employed. Additionally, an exploration was conducted on whether JTG should be normalized. It is worth noting that the parameters of the static JTG are fixed, hence the issue of inter-layer weight-sharing does not require consideration.

We conducted tests employing solely the joint modality on the NTU-60 dataset under the X-Sub setting, utilizing the JTG with 6 frames as input data, and the results are presented in Table 2.

Experiments demonstrate that the dynamic configuration of the JTG has superior expressive power, particularly when a layer-sharing structure is employed without normalization. We conduct the following analysis.

Firstly, the spatial distances between nodes in a JTG frame are theoretically not uniformly distributed, neither are the distances along the temporal dimension. Thus, the dynamic structure enables the precise modeling of such distances, enhancing the model's expressive capabilities.

Secondly, the improved performance of the layer-sharing dynamic configuration demonstrates the capability of the JTG in global spatio-temporal modeling, which reduces the number of parameters and training costs.

Finally, normalization typically align the variation magnitude of features and improves the optimization conditions. However, in JTG, the difference in the variation magnitude of distance is not significant, thus the advantage of normalization is quite slight. Furthermore, normalization narrows the variation range of distances, which leads to averaging of weights and affects the performance negatively.

Koopman Operator. To capture more trajectory information and dynamic correlation, the Koopman operator is applied in this work. We explored the performance differences between methods using the Koopman operator and those using global average pooling and fully connected (FC). The results are shown in Table 3, where the Temporal (or Spatial) K_{op} denotes global linearization in the temporal (or spatial) dimension of the features.

In this experiment, we utilize four different modalities of skeleton sequences, i.e., joint, bone, joint motion, and bone motion, as input data respectively. The frame count of the JTG is set to 6.

The result indicates that employing the Koopman operator in JT-GraphFormer yields superior results compared to utilizing FC in all the input modalities. It is worth noting that motion modalities encompass information describing variations in joint coordinates or bone vectors. The spatial linearization process of this information involves using the variation trend of the joint to infer its spatial relationship, i.e., to deduce the change of other joints, while its temporal linearization process can be regarded as the evolution of the variation amount. The latter approach is more abstract, which may lose some important dynamic information and may require more complex feature processing methods or more training samples. Therefore, the Spatial K_{op} method is advantageous in capturing the associations of the trajectories and exhibits superior performance for the motion modalities input.

modality	FC	Temporal K_{op}	Spatial K_{op}
Joint	90.4	**90.5**	90.3
Bone	88.9	**89.5**	**89.5**
Joint motion	87.9	87.9	**88.4**
Bone motion	86.9	87.2	**87.3**

Table 3: Top-1 accuracy (%) using the Koopman operator and fully connected methods on the NTU-60 dataset under the X-Sub setting of using various data modalities.

modality	NTU-60		NTU-120	
	X-Sub	X-View	X-Sub	X-Set
Joint	91.8	96.8	87.6	89.6
Bone	91.2	96.4	87.4	89.6
Joint motion	90.2	95.0	84.7	86.7
Bone motion	90.0	94.1	84.6	86.1
Joint+Bone	92.5	97.1	89.0	91.0
4 ensemble	93.4	97.5	89.9	91.7

Table 4: Top-1 accuracy (%) of the methods using various data modalities on the NTU-60 and NTU-120 datasets.

Four-stream Ensemble. We utilize four-stream ensemble method to represent the performance of the trained models, namely joint, bone, joint motion, and bone motion. Each stream and the ensemble methods are tested on the NTU-60 and NTU-120 datasets via the proposed JT-GraphFormer. The results are shown in Table 4.

We observe that the performance of the model gradually improves as the stream number of the ensemble method increases. On the NTU-60 dataset under the X-Sub setting, the accuracy of joint + bone and four-stream ensemble methods increases by margins of 0.7% and 1.6%, respectively, compared with the accuracy of using the joint only modality. It fully demonstrates that the multi-modal representation increases the variety of input features, and improves the representational and the generalization ability of the model.

Comparison with the State-of-the-art Methods

We evaluate the proposed JT-GraphFormer on three popular benchmarks and compare the performance with recent prevailing approaches. Table 5 shows that the JT-GraphFormer

Method	Year	NTU-60		NTU-120		N-UCLA
		X-Sub	X-View	X-Sub	X-Set	
ST-GCN (Yan, Xiong, and Lin 2018)	AAAI 2018	81.5	88.3	-	-	-
2s-AGCN (Shi et al. 2019b)	CVPR 2019	88.5	95.1	82.9	84.9	-
DGNN (Shi et al. 2019a)	CVPR 2019	89.9	96.1	-	-	-
Dynamic-GCN (Ye et al. 2020)	ACM MM 2020	91.5	96.0	87.3	88.6	-
SGN (Zhang et al. 2020)	CVPR 2020	89.0	94.5	79.2	81.5	92.5
DDGCN (Korban and Li 2020)	ECCV 2020	91.1	97.1	-	-	-
DC-GCN+ADG (Cheng et al. 2020)	ECCV 2020	90.8	96.6	86.5	88.1	-
MS-G3D (Liu et al. 2020)	CVPR 2020	91.5	96.2	86.9	88.4	-
MST-GCN (Chen et al. 2021b)	AAAI 2021	91.5	96.6	87.5	88.8	-
CTR-GCN (Yan, Xiong, and Lin 2018)	ICCV 2021	92.4	96.8	88.9	90.6	96.5
InfoGCN (4s) (Chi et al. 2022)	CVPR 2022	92.7	96.9	89.4	90.7	96.6
InfoGCN (6s) (Chi et al. 2022)	CVPR 2022	93.0	97.1	89.8	91.2	97.0
STF (2s) (Ke, Peng, and Lyu 2022)	AAAI 2022	92.5	96.9	88.9	89.9	-
Ta-CNN (4s) (Xu et al. 2022)	AAAI 2022	90.4	94.8	85.4	86.8	96.1
EffificientGCN (3s) (Song et al. 2022)	TPAMI22	91.7	95.7	88.3	89.1	-
CTR-GCN+FR (4s) (Zhou, Liu, and Wang 2023)	CVPR 2023	92.8	96.8	89.5	90.9	96.8
Ours(Joint only)	-	91.8	96.8	87.6	89.6	95.5
Ours(2s)	-	92.5	97.1	89.0	91.0	96.6
Ours(4s)	-	**93.4**	**97.5**	**89.9**	**91.7**	**97.2**

Table 5: Performance comparison between JT-GraphFormer and prevailing SOTA methods in skeleton-based human action recognition tasks on the NTU-60, NTU-120, and N-UCLA datasets. For clarity, we provide multi-stream descriptions for the models published between 2022 and 2023.

surpasses the listed methods under all settings with equivalent streams, even in several cases where fewer streams are applied. By utilizing the joint only modality, the JT-GraphFormer already surpasses the majority of models on the NTU-60 dataset under the X-View setting. When excluding the joint motion and bone motion streams, the JT-GraphFormer outperforms the STF (2s) (Ke, Peng, and Lyu 2022) by a margin of 0.1% on the NTU-120 dataset under both the X-Sub and X-Set settings. On the N-UCLA, the accuracy of the JT-GraphFormer (4s) shows a margin of 0.4% improvement compared to the CTR-GCN+FR (4s) (Chen et al. 2021a; Zhou, Liu, and Wang 2023). Furthermore, despite employing four streams, the JT-GraphFormer (4s) outperforms the InfoGCN (6s) (Chi et al. 2022) by a margin of 0.4% on the NTU-60 dataset in both the X-Sub and X-View configurations. To the best of our knowledge, our model attains superior performances compared with the SOTA methods on the three listed datasets.

Limitations

Despite JT-GraphFormer's advanced performance on the NTU-60, NTU-120, and N-UCLA datasets, testing it on Kinetics-400 (Kay et al. 2017) will be challenging due to its large parameter count when using K_{op} (14.54M parameters for NTU-120). Additionally, JT-GraphFormer only considers integrating four streams and lacks exploration of using more streams fusions to fully exploit its potential. Furthermore, JT-GraphFormer is limited to processing structured data sequences, such as the motion of regular objects. Finally, the use of unlabeled data in JT-GraphFormer is a promising area of future work. In future work, it is suggested and encouraged to explore its methods and potential in unsupervised mode.

Conclusions

In this endeavor, we present a JT-GraphFormer model based on a joint trajectory topology structure. By constructing the JTG, our model effectively captures the semantic information of the input joint trajectory data, enhancing the Transformer's capability. The proposal and use of STDA, which incorporates intra-graph distances of joints within a JTG, empower each node to discerningly allocate attention. Furthermore, our incorporation of the Koopman operator linearizes the extracted features in either the temporal or the spatial dimension, which effectively captures dynamic shifts inherent to distinct action categories. This culminates in augmented representational capacity and classification performance.

Empirical validations unequivocally accentuate the outstanding performance of JT-GraphFormer in human action recognition tasks. Through meticulous comparative analysis and quantitative evaluations across three distinct datasets, the JT-GraphFormer emerges as a standout contender, firmly establishing its superiority over prevailing SOTA methods.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62376286).

References

Chen, Y.; Zhang, Z.; Yuan, C.; Li, B.; Deng, Y.; and Hu, W. 2021a. Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition. In *Proceedings*

of the IEEE/CVF International Conference on Computer Vision.

Chen, Z.; Li, S.; Yang, B.; Li, Q.; and Liu, H. 2021b. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *AAAI*.

Cheng, K.; Zhang, Y.; Cao, C.; Shi, L.; Cheng, J.; and Lu, H. 2020. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *Computer Vision–ECCV 2020: 16th European Conference*.

Chi, H.-g.; Ha, M. H.; Chi, S.; Lee, S. W.; Huang, Q.; and Ramani, K. 2022. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, G., Sylvain; Jakob, U.; and Neil, H. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Ke, L.; Peng, K.-C.; and Lyu, S. 2022. Towards to-at spatio-temporal focus for skeleton-based action recognition. In *AAAI*.

Korban, M.; and Li, X. 2020. Ddgcn: A dynamic directed graph convolutional network for action recognition. In *Computer Vision–ECCV 2020: 16th European Conference*.

Kutz, J. N.; Fu, X.; Brunton, S. L.; and Erichson, N. B. 2015. Multi-resolution dynamic mode decomposition for foreground/background separation and object tracking. In *2015 IEEE international conference on computer vision workshop (ICCVW)*.

Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; and Tian, Q. 2019. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Liu, J.; Shahroudy, A.; Perez, M.; Wang, G.; Duan, L.-Y.; and Kot, A. C. 2019. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2684–2701.

Liu, Z.; Zhang, H.; Chen, Z.; Wang, Z.; and Ouyang, W. 2020. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Plizzari, C.; Cannici, M.; and Matteucci, M. 2021. Spatial temporal transformer network for skeleton-based action recognition. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event*, 694–701.

Proctor, J. L.; Brunton, S. L.; and Kutz, J. N. 2018. Generalizing Koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17: 909–930.

Qiu, H.; Hou, B.; Ren, B.; and Zhang, X. 2022. Spatio-temporal tuples transformer for skeleton-based action recognition. *arXiv preprint arXiv:2201.02849*.

Shahroudy, A.; Liu, J.; Ng, T.-T.; and Wang, G. 2016. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Shi, F.; Lee, C.; Qiu, L.; Zhao, Y.; Shen, T.; Muralidhar, S.; Han, T.; Zhu, S.-C.; and Narayanan, V. 2021. Star: Sparse transformer-based action recognition. *arXiv preprint arXiv:2107.07089*.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019a. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019b. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Song, Y.-F.; Zhang, Z.; Shan, C.; and Wang, L. 2022. Constructing stronger and faster baselines for skeleton-based action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 45: 1474–1488.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, J.; Nie, X.; Xia, Y.; Wu, Y.; and Zhu, S.-C. 2014. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wang, Q.; Peng, J.; Shi, S.; Liu, T.; He, J.; and Weng, R. 2021. Iip-transformer: Intra-inter-part transformer for skeleton-based action recognition. *arXiv preprint arXiv:2110.13385*.

Xu, K.; Ye, F.; Zhong, Q.; and Xie, D. 2022. Topology-aware convolutional neural network for efficient skeleton-based action recognition. In *AAAI*.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

Ye, F.; Pu, S.; Zhong, Q.; Li, C.; Xie, D.; and Tang, H. 2020. Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In *Proceedings of the 28th ACM international conference on multimedia*, 55–63.

Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; and Liu, T.-Y. 2021. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34: 28877–28888.

Zhang, P.; Lan, C.; Zeng, W.; Xing, J.; Xue, J.; and Zheng, N. 2020. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Zhou, H.; Liu, Q.; and Wang, Y. 2023. Learning discriminative representations for skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.