This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3240472

TranSkeleton: Hierarchical Spatial-Temporal Transformer for Skeleton-Based Action Recognition

Haowei Liu, Yongcheng Liu, Yuxin Chen, Chunfeng Yuan, Bing Li, Weiming Hu

***Abstract*—In skeleton-based action recognition, it has been a dominant paradigm to extract motion features with temporal convolution and model spatial correlations with graph convolution. However, it's difficult for temporal convolution to capture long-range dependencies effectively. Meanwhile, commonly used multi-branch graph convolution leads to high complexity. In this paper, we propose TranSkeleton, a powerful Transformer framework which neatly unifies the spatial and temporal modeling of skeleton sequences. For temporal modeling, we propose a novel partition-aggregation temporal Transformer. It works with hierarchical temporal partition and aggregation, and can capture both long-range dependencies and subtle temporal structures effectively. A difference-aware aggregation approach is designed to reduce information loss during temporal aggregation. For spatial modeling, we propose a topology-aware spatial Transformer which utilizes the prior information of human body topology to facilitate spatial correlation modeling. Extensive experiments on two challenging benchmark datasets demonstrate that TranSkeleton notably outperforms the state of the arts.**

***Index Terms*—Skeleton-based action recognition, spatial-temporal Transformer, long-range temporal dependencies.**

I. INTRODUCTION

ACTION recognition is a long-standing research problem of classifying human actions according to the input videos [1]–[3] or skeleton sequences [4]–[9]. It has a wide range of applications, such as human-computer interaction and intelligent monitoring [10]. Compared to video data, skeleton data not only consumes less storage and computation resource, but also has better robustness to viewpoint change and background clutter. Therefore, skeleton-based action recognition has drawn a lot of attention from researchers in recent years.

Generally, the inherent information of skeleton sequences can be decomposed into two orthogonal dimensions, *i.e.* the spatial skeleton pose in each frame and the temporal motion trajectory of each joint. Both of them are important for action recognition. Recently, with the remarkable achievement of deep learning in vision tasks, much effort has been made on applying deep neural networks to skeleton-based action recognition. Early deep learning methods generally arrange the human skeleton as a sequence of 3D joint coordinates or transform it into a pseudo-image, and then use Recurrent Neural Network (RNN) [4], [11]–[13] or Convolutional Neural Network (CNN) [14]–[16] for feature extraction and classification. Though making great progress, these RNN or CNN-based methods neglect to capture the inherent spatial correlations among joints. Inspired by graph learning [17]–[19], researchers find that the human skeleton can be regarded as a graph with joints as nodes and bones as edges. Therefore, recently Graph Convolutional Network (GCN) has been widely applied to this task [5]–[9] and has achieved significant performance boost over conventional methods. However, these methods generally integrate multiple branches of graph convolution to extract richer spatial information, causing huge computation cost. On the other hand, they focus on improving GCNs for better spatial modeling. As for temporal modeling, most of them simply stack multiple temporal convolutional layers to extract motion features, and thus have two non-negligible drawbacks. 1) The increase of the temporal receptive field in this way is rather limited, leading to in fact short-range temporal modeling. 2) Detailed motion information may have largely vanished when reaching deeper layers, hindering the interaction between distant input frames. Therefore, it is pretty intractable for them to fully grasp the temporal motion information, especially long-range temporal dependencies.

Recently, Transformer's strong ability of sequence modeling has been verified in various computer vision tasks, *e.g.* image recognition [20], [21] and video analysis [22], [23]. This motivates us to explore its potential in unifying the spatial and temporal modeling of skeleton sequences. A straightforward way to achieve this is directly substituting GCN and TCN with Transformer for spatial and temporal modeling respectively. Nevertheless, in practice such a simple extension of vanilla Transformer (V-Trans, as illustrated in Fig. 1) cannot achieve

This work is supported by the national key R&D program of China (No. 2018AAA0102802), the Natural Science Foundation of China (Grant No. 61972397, 62036011, 62192782, 61721004, U2033210), Beijing Natural Science Foundation (JQ21017, L223003), the Major Projects of Guangdong Education Department for Foundation Research and Applied Research (Grant: 2017KZDXM081, 2018KZDXM066), Guangdong Provincial University Innovation Team Project (Project No.: 2020KCXTD045). *(Corresponding author: Chunfeng Yuan.)*

Haowei Liu and Yuxin Chen are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: liuhaowei2019@ia.ac.cn; chenyuxin2019@ia.ac.cn).

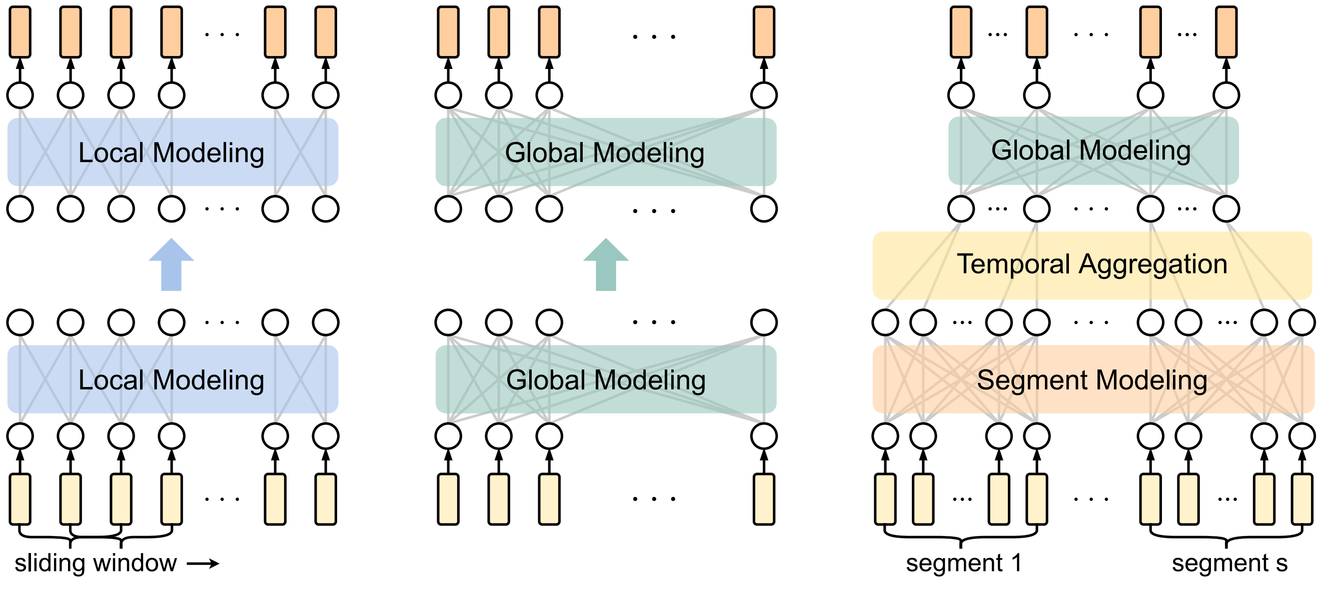
Yongcheng Liu is with Amap, Alibaba, Beijing 100102, China (e-mail: liuyongcheng.lyc@alibaba-inc.com).

Chunfeng Yuan is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: cfyuan@nlpr.ia.ac.cn).

Bing Li is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with People AI Inc., Beijing 100190, China (e-mail: bli@nlpr.ia.ac.cn).

Weiming Hu is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China. He is also with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (e-mail: wmhu@nlpr.ia.ac.cn).

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3240472



(a) TCN (b) V-Trans (c) PAT-Trans

Fig. 1. Comparison of different temporal modeling methods. **Left:** Temporal convolutional network (TCN) performs local modeling in a sliding-window fashion, with relatively limited temporal receptive field. **Middle:** Vanilla Transformer (V-Trans) performs global modeling throughout the model without temporal aggregation. **Right:** Our proposed partition-aggregation temporal Transformer (PAT-Trans) performs temporal modeling to each long segment within the same hierarchy in parallel, and gradually reduces the number of segments to one through difference-aware temporal aggregation. Thus it can capture both long-range dependencies and subtle temporal structures effectively, and also greatly reduces the redundancy of V-Trans.

satisfactory performance. We analyze that this is due to two intrinsic factors: 1) V-Trans maintains the sequence length throughout the model, which leads to huge redundancy as the input skeleton sequences are generally long (*e.g.* 300 frames). 2) Lack of local modeling makes it difficult for V-Trans to capture the subtle temporal structures of the input sequences, especially when trained on limited-scale skeleton datasets.

To address these issues, we propose to unify the spatial and temporal modeling of skeleton sequences within a hierarchical Transformer framework named TranSkeleton (as illustrated in Fig. 2). In this framework, we first propose a novel partition-aggregation temporal Transformer (PAT-Trans, as illustrated in Fig. 1) which works with hierarchical temporal partition and difference-aware temporal aggregation. Then we devise a topology-aware spatial Transformer which incorporates the proposed physical connection constraint to facilitate spatial modeling. Specifically, in each hierarchy, we first partition the input sequence into several long segments along the temporal dimension, and perform attention-based temporal modeling to each segment using stacked Transformer units. Then we concatenate the segments and reduce the sequence length by half through temporal aggregation. We perform such partition-modeling-aggregation process in a hierarchical manner, and gradually reduce the number of segments to one. Through this way, we realize effective local-to-global temporal modeling. In addition, we propose a difference-aware temporal aggregation (DATA) approach. By taking inter-frame differences into consideration, it greatly reduces the information loss brought by multiple temporal aggregations. Compared to V-Trans, the proposed PAT-Trans reduces great redundancy and better grasps the subtle temporal structures of the input sequences. For spatial modeling, we also adopt Transformer to capture the spatial correlations among joints, and devise a physical connection constraint (PCC) to embed the prior information of human body topology into Transformer in a neat way. Note that the spatial and temporal modeling share Transformer's multi-head self-attention (MSA) as their core computation mechanism. Therefore, by combining spatial MSA with temporal MSA in each Transformer unit, we can easily integrate the proposed topology-aware spatial Transformer and partition-aggregation temporal Transformer into a unified spatial-temporal modeling framework. Compared with the prevailing TCN-GCN paradigm, our TranSkeleton captures long-range temporal dependencies more effectively, and avoids cost-expensive multi-branch GCN integration. We conduct extensive experiments and analysis on two challenging skeleton datasets, *i.e.*, NTU RGB+D and NTU RGB+D 120. The experimental results validate the effectiveness and efficiency of our method.

Our contributions can be summarized as follows:

- A powerful Transformer model named TranSkeleton is proposed for skeleton-based action recognition. It effectively and neatly unifies spatial and temporal modeling within a pure Transformer framework.
- A novel temporal modeling method PAT-Trans is proposed. It works with hierarchical partition and aggregation, and can well capture both long-range dependencies and subtle temporal structures simultaneously.
- A difference-aware temporal aggregation approach and a physical connection constraint are devised. The former greatly reduces the information loss in multiple temporal aggregations. The latter provides the prior information of human body topology to facilitate spatial modeling.

II. RELATED WORK

A. Skeleton-Based Action Recognition

Early deep learning methods in skeleton-based action recognition apply RNNs [4], [11]–[13] or CNNs [14]–[16], [24], [25] to model skeleton sequences. As the human skeleton has a natural graph structure, these methods fail to capture the inherent correlations among joints. ST-GCN [5] first applies graph convolution for spatial correlation modeling and temporal convolution for motion feature extraction. Since then, TCN-GCN based methods [26]–[32] have achieved significant performance boost. MS-G3D [7] and STIGCN [33] introduce multi-scale topologies into GCNs to explore the correlations among distant joints. DC-GCN [34] boosts the graph modeling ability by using different learned topologies in different feature channel groups. These methods all employ learned adjacent matrices to model human body topology, and thus lack adaptability to different input samples when inference. 2s-AGCN [6] and SGN [35] introduce self-attention mechanism to model the correlations among joints dynamically according to their features. CTR-GC [32] proposes a channel-wise topology refinement graph convolution for dynamic topology and multi-channel feature modeling. In the semi-supervised scenario, X-CAR [36] proposes a contrastive augmentation and representation learning framework to obtain rotate-shear-scale invariant features. However, most existing methods use multi-branch graph convolution to extract richer spatial information, leading to a drastic increase of parameters and computation cost. In contrast, we avoid such unnecessary increase of complexity by employing multi-head self-attention for spatial modeling.

On the other hand, existing methods mainly focus on improving GCNs to achieve better spatial modeling. For temporal modeling, most of them simply stack multiple temporal convolutional layers to extract motion features. In order to enlarge the temporal receptive field, MS-G3D [7] uses parallel temporal convolutions with different dilation rates. MST-GCN [37] devises a hierarchical residual architecture for multi-scale temporal modeling. DualHead-Net [38] proposes a dual-head graph network consisting of two interleaved branches to extract features at two spatial-temporal resolutions. SEFN [39] proposes a symmetrical enhanced fusion network to fuse multi-level spatial and temporal features. However, as increasing temporal convolution's kernel size will cause a drastic increase of parameters and computation cost, TCN-based methods generally have a small convolution kernel size (*e.g.* 3). This makes it difficult for them to realize a global temporal receptive field and capture long-range dependencies effectively. Recently, [40], [41] introduce vanilla Transformer into the TCN-GCN framework for global modeling. [42] directly adopts global Transformer for temporal modeling. However, as discussed in Sec. I, due to the issues of vanilla Transformer, *i.e.*, huge redundancy and lack of local modeling, they don't achieve satisfactory performance. Different to [40]–[43], in this work, we propose a novel local-to-global temporal modeling method, and unify the spatial and temporal modeling of skeleton sequences within a pure Transformer framework.

B. Vision Transformer

Transformer [44] uses multi-head self-attention for sequence modeling and has been the mainstream approach in natural language processing. Recently, it is introduced into computer vision, and has achieved remarkable performance in various vision tasks, such as image recognition [20], [21], [45], object detection [46] and semantic segmentation [47]. In video-based action recognition, its validity has also been verified [22], [23]. Nevertheless, compared to conventional CNN-based methods, vision Transformer models generally require much more training data to achieve competitive performance. This issue becomes more intractable when applying Transformer to skeleton-based action recognition, as so far skeleton datasets are relatively small compared to image and video datasets. To address this, we propose a hierarchical Transformer architecture which works in a local-to-global manner in the temporal dimension. Introducing local modeling into Transformer not only facilitates the training on limited-scale datasets, but also enhances the model's ability to capture the subtle temporal structures of the input sequences.

It's worth noting that, different from existing vision Transformers in image recognition (*e.g.* Swin Transformer [21]) which adopt local modeling strategy as well, we extend Transformer to unify the much more intractable spatial-temporal modeling task. In particular, our proposed local-to-global temporal modeling method (PAT-Trans) has rarely been studied before, as TCN-based local modeling dominates the area of skeleton-based action recognition and video Transformers [22], [23] generally perform global modeling due to their much shorter input clips (e.g. 8 frames).

III. METHOD

In this section, we first give an overview of the proposed TranSkeleton framework (Sec. III-A). Then, we elaborate its two key components, *i.e.*, Partition-Aggregation Temporal Transformer (Sec. III-B) and Topology-Aware Spatial Transformer (Sec. III-C).

A. Overview of TranSkeleton

As Fig. 2 shows, TranSkeleton processes the embeddings of the input skeleton sequences in a multi-stage manner. Each stage comprises three types of units, *i.e.*, temporal partition, spatial-temporal Transformer unit and temporal aggregation. Specifically, the input sequence is first uniformly partitioned into several segments along the temporal dimension. Then, the spatial-temporal Transformer unit is devised to model each segment. It consists of two multi-head self-attention modules for spatial and temporal modeling respectively, along with an MLP module for feature transformation. Following the original Transformer [44], we also apply LayerNorm [48] before each inside module and use several residual connections to facilitate the training. Finally, temporal aggregation is conducted to reduce the temporal dimension and merge adjacent segments. As the temporal partition and aggregation are performed in a hierarchical way, the number of segments is gradually reduced to one. Therefore, the whole framework works as a local-to-global architecture in the temporal dimension. After the

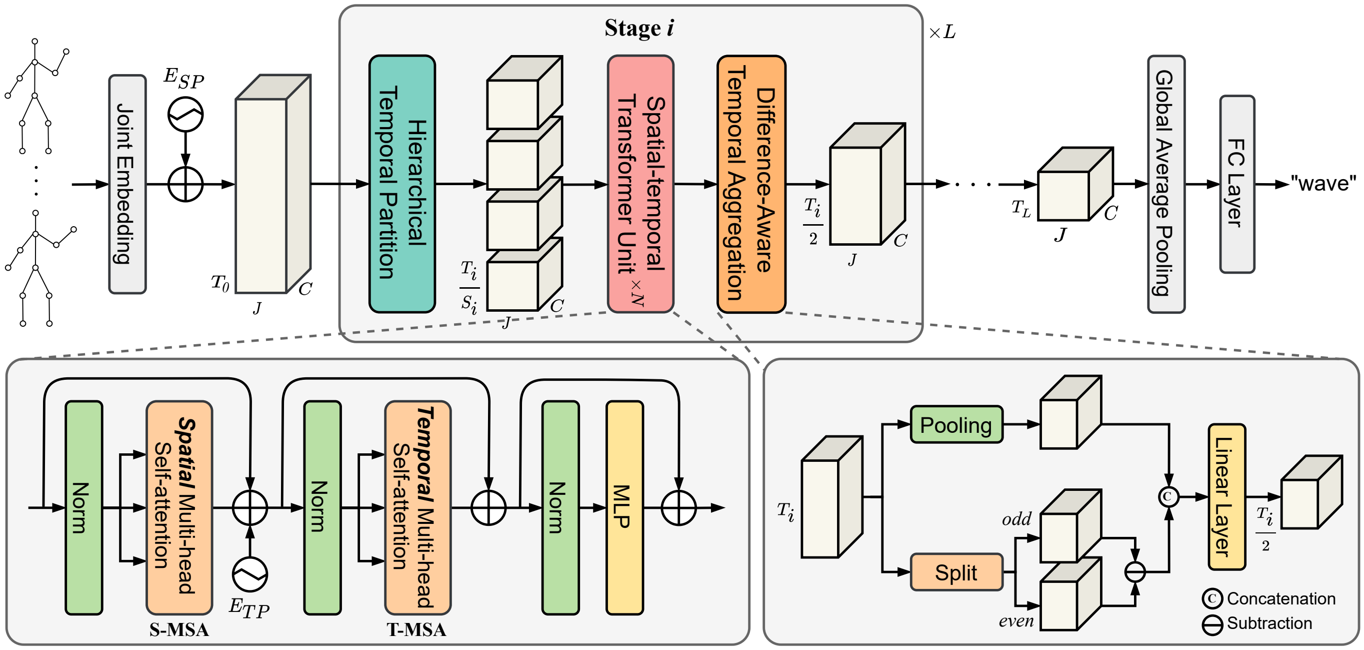


Fig. 2. Illustration of the proposed TranSkeleton with L stages. In each stage, we first partition the input feature into S_i segments along the temporal dimension, then feed them into N stacked spatial-temporal Transformer units for spatial and temporal modeling alternatively. After that, we concatenate the segments into a sequence and apply the proposed difference-aware temporal aggregation approach (Sec. III-B) to reduce the temporal dimension by half. The output is then fed into the next stage. We perform temporal partition and aggregation in a hierarchical manner, such that the number of segments is progressively reduced to one. The final classification score is obtained through global average pooling followed by a fully-connected (FC) layer.

multi-stage partition-modeling-aggregation process, we apply a global average pooling followed by a linear layer to predict the action label. Note that in order to encode the positional information, we add trainable spatial positional embedding E_{SP} after joint embedding and temporal positional embedding E_{TP} in each stage.

Different from the TCN-GCN paradigm, our TranSkeleton unifies spatial and temporal modeling within a pure Transformer framework. It achieves sufficient interaction among joints and deeply-correlated information flow along motion trajectories, and thus can learn discriminative spatial-temporal representations of the input skeleton sequences.

B. Partition-Aggregation Temporal Transformer

Extracting temporal features along the motion trajectories of the joints is important for skeleton-based action recognition. However, this can be intractable since different actions take place on different temporal scales, and thus it requires the model to effectively capture long-range temporal dependencies. The strong ability of Transformer in sequence modeling makes it a desirable option for temporal modeling. Nevertheless, directly employing vanilla Transformer (V-Trans) leads to inferior performance. This is due to: 1) It usually requires long input skeleton sequences for decent performance, while V-Trans maintains such long sequence length throughout the model. This causes huge redundancy when the model deepens. 2) Different from TCN, V-Trans lacks local modeling. This makes it difficult to capture the subtle temporal structures of the input sequences, especially when trained insufficiently on limited-scale skeleton datasets.

Partition-Aggregation Temporal Transformer. To overcome these issues of V-Trans, we propose a partition-aggregation temporal Transformer (PAT-Trans). It works with hierarchical temporal partition and aggregation to capture local-to-global temporal dependencies. Specifically, given an input feature $X \in \mathbb{R}^{T\times J\times C}$, we perform temporal modeling to each joint $X_j \in \mathbb{R}^{T\times C}$ in parallel, where T, J and C are the sequence length, the number of joints and the feature dimension respectively. We first uniformly partition the input feature X_j into S segments (X_j^1, X_j^2, ..., X_j^S) along the temporal dimension. Then feed them into a shared temporal multi-head self-attention (T-MSA) module. T-MSA applies dot-product attention to model the correlations among the elements of the input sequence in a dynamic fashion. Given the feature of the k-th segment $X_j^k \in \mathbb{R}^{\frac{T}{S}\times C}$, T-MSA first employs linear mapping functions $W_Q, W_K, W_V \in \mathbb{R}^{C\times C}$ to generate the corresponding query matrix, key matrix and value matrix, *i.e.*, $Q, K, V \in \mathbb{R}^{\frac{T}{S}\times C}$. Before dot-product attention, each of Q, K and V is uniformly split into h groups (*i.e.* h heads) along the channel dimension. Each head corresponds to a $\hat{C}$-dimensional subspace of the original representation space, where $\hat{C} = \frac{C}{h}$. Then in each subspace, we compute the matrix multiplication of the corresponding query matrix Q_i and key matrix K_i. After normalization, we obtain an attention map and use it to guide the interaction of the elements of the value matrix V_i. The above dot-product attention is applied in these heads in parallel. Then the results are concatenated and fed into a linear layer $W \in \mathbb{R}^{C\times C}$, so as to fuse the features of different heads. The above process can be formulated as

$$Q = X_j^k W_Q, K = X_j^k W_K, V = X_j^k W_V, \tag{1}$$

$$H_i = \mathrm{softmax}\left(\frac{Q_i K_i^T}{\sqrt{\hat{C}}}\right) V_i, i \in \{1, 2, \ldots, h\}, \tag{2}$$

$$\text{T-MSA}(X_j^k) = (H_1 || H_2 || \ldots || H_h)\, W, \tag{3}$$

where $||$ is the concatenate operation and $\frac{1}{\sqrt{\hat{C}}}$ is a scaling factor for normalization.

After that, we concatenate the output of the S segments into a whole sequence $\widetilde{X}_j \in \mathbb{R}^{T \times C}$, and reduce the sequence length by half through temporal aggregation. The whole temporal modeling process in one stage can be formulated as

$$\mathrm{TM}(X_j) = \mathrm{TA}\left(\text{T-MSA}\left(X_j^1\right) || \ldots || \text{T-MSA}\left(X_j^S\right)\right), \tag{4}$$

where $||$ is the concatenate operation and TA denotes temporal aggregation. We perform such partition-modeling-aggregation process in a hierarchical manner, and gradually reduce the number of segments to one. During this process, the temporal receptive field increases rapidly and soon covers the whole sequence. In this way, we realize effective local-to-global temporal modeling. Note that the number of segments S is an important hyper-parameter in this process. The ablation study on its impact can be found in Sec. IV-C.

Difference-Aware Temporal Aggregation. Temporal aggregation plays a significant role in constructing the proposed PAT-Trans due to: 1) It reduces the sequence length and avoids the unnecessary increase of complexity of high-level features. 2) It effectively enlarges the temporal receptive field and facilitates the interaction among distant frames. For instance, if two consecutive stages have the same segment length, the equivalent temporal receptive field would double after the temporal aggregation between them. Generally, average pooling and max pooling are two commonly used operations for dimension reduction. However, average pooling leads to much loss of high-frequency information as it smoothes the motion trajectories. Max pooling preserves the biggest response of each channel and drops the smaller ones, resulting in hidden information loss as well.

To reduce information loss during aggregation, we propose a simple yet effective difference-aware temporal aggregation (DATA) approach. As illustrated in the lower right part of Fig. 2, for feature vectors $\widetilde{X} \in \mathbb{R}^{T_i \times J \times C}$, we first reduce the temporal dimension by half with max pooling or average pooling, and compute the difference between odd frames $\widetilde{X}_{odd}$ and even frames $\widetilde{X}_{even}$. Then we concatenate the two results in the channel dimension, and reduce the number of channels to C with a linear projection $\widetilde{W} \in \mathbb{R}^{2C \times C}$. The whole DATA approach can be formulated as

$$\mathrm{DATA}\left(\widetilde{X}\right) = \left(\mathrm{pooling}\left(\widetilde{X}\right) || \,\mathrm{abs}\left(\widetilde{X}_{odd} - \widetilde{X}_{even}\right)\right) \widetilde{W}, \tag{5}$$

where $||$ is the concatenate operation. By fusing the inter-frame differences and the pooled features, DATA preserves more discriminative information during temporal aggregation, and thus greatly enhances the model's temporal modeling ability.

Comparison with TCN and V-Trans. How far the forward signals from the input elements have to traverse before they meet, is a key factor which affects the model's ability to capture long-range dependencies. The shorter these paths are between any pair of input elements, the easier it is to grasp long-range dependencies [44]. Therefore, the proposed PAT-Trans has two key advantages over TCN: 1) As the segment length of our method (*e.g.* 16) is much larger than TCN's kernel size, its receptive field can soon cover the whole sequence within a few hierarchies. In contrast, TCN needs a great many layers to achieve the same goal, leading to an unacceptable increase of model complexity and computation cost. 2) Even with the same receptive field as TCN, our method has higher efficiency of information interaction. For instance, if we set the segment length to 16, then the aforementioned path length is 1 for any pair of elements within the same segment. This is because for PAT-Trans, all the elements of a segment directly interact with each other by dot-product attention, regardless of their distance. In TCN, however, it takes 6 stacked temporal convolutional layers (kernel size = 5) to realize the interaction between the 1st and the 15th elements. Plenty of subtle motion information has already vanished in such deep layers.

Compared to V-Trans, our PAT-Trans performs temporal modeling in a local-to-global manner. Introducing local modeling into Transformer not only enhances its ability of grasping the subtle temporal structures of the input sequences, but also facilitates the training on limited-scale datasets. Meanwhile, the hierarchical architecture greatly reduces the redundancy of V-Trans. Fig. 1 shows the illustration of the above three temporal modeling methods. A performance comparison of them is presented in Sec. IV-C.

C. Topology-Aware Spatial Transformer

Different from local-to-global temporal modeling, for spatial modeling, we apply multi-head self-attention to capture the spatial correlations among joints in a global manner. This is because unlike a skeleton sequence, the joints within a skeleton have an explicit order, *e.g.*, 1st for "head" and 5th for "left shoulder". Meanwhile, the number of joints is much smaller than the sequence length. Specifically, given an input skeleton sequence, we perform spatial modeling within each frame individually. We treat the 3D coordinates of the joints in the t-th frame $X_t \in \mathbb{R}^{J \times 3}$ as a sequence containing J elements and get the embedding vectors of the joints by mapping the coordinates into a C-dimensional space through linear projection $W_e \in \mathbb{R}^{3 \times C}$. Then we sum the embedding vectors with the trainable spatial positional embedding $E_{SP} \in \mathbb{R}^{J \times C}$ and feed the result into spatial multi-head self-attention (S-MSA) modules. Similar to Eq. (1)-(3), S-MSA generates h attention maps in the representation subspaces and uses the attention maps to guide the information flow among joints. Therefore, it avoids the increase of complexity caused by multi-branch GCN integration. The above stated spatial modeling process can be formulated as

$$\mathrm{SM}(X_t) = \text{S-MSA}\left(X_t W_e + E_{SP}\right). \tag{6}$$

Note that the spatial and temporal modeling share MSA as their core computation mechanism. Thus as illustrated in the lower left part of Fig. 2, by combining S-MSA with T-MSA

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3240472

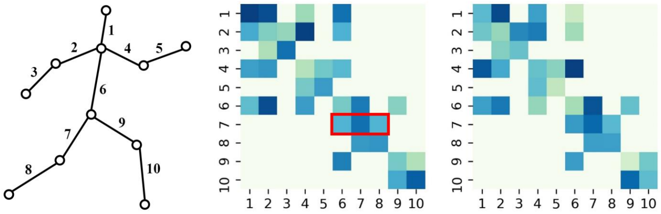


Fig. 3. Illustration of the physical connection constraint. **Left:** A simplified human skeleton. **Right:** The corresponding parameterized adjacent matrices in two attention heads, where only the elements correspond to actual physical connections have non-zero values.

in each Transformer unit, we integrate the proposed spatial Transformer and temporal Transformer into a unified spatial-temporal modeling framework.

Physical Connection Constraint. Apart from joint coordinates, bone vector which represents the length and direction of bones is another commonly used modality in skeleton-based action recognition. However, both absolute position and human body topology information are completely lost in this bone modality, which hinders pure attention-based spatial modeling. To tackle this, we devise a physical connection constraint (PCC) to form a topology-aware spatial Transformer. Specifically, apart from the dynamically generated attention matrices, we define a parameterized adjacent matrix in each attention head and restrict all elements to zero except those correspond to actual physical connections. For instance, as illustrated in Fig. 3, the 7th bone connects to the 6th and the 8th bones in the simplified human skeleton. Thus the 7th row/column in the parameterized adjacent matrices only contains three non-zero values. Combining the original multi-head self-attention with the physical connection constraint, Eq. (2) turns into

$$H_i = \left(\text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{\hat{C}}}\right) + A_i \odot M_{PCC}\right) V_i, \quad (7)$$

where $i \in \{1, 2, \ldots, h\}$ and $\odot$ denotes element-wise multiplication. A_i is the parameterized adjacent matrix in the i-th attention head. M_{PCC} is a zero-one matrix which is used as the physical connection constraint. Besides, adding a parameterized adjacent matrix without any constraint is an option as well. We'll compare the performance of these two schemes in the ablation study.

Note that the devised PCC is quite different from 2s-AGCN [6] in two aspects: 1) We do not perform manual partition as 2s-AGCN, which splits the physically connected neighborhood into three subsets. 2) 2s-AGCN requires multiple GCN branches to perform laborious feature transformations to different subsets. In contrast, our PCC neatly embeds the prior information of human body topology into Transformer by constraining the information flow among joints.

IV. EXPERIMENTS

A. Datasets

NTU RGB+D. NTU RGB+D [49] is the most widely used large-scale dataset for skeleton-based action recognition, which contains 56880 samples of 60 classes ranging from daily actions to medical conditions. The action samples are performed by 40 distinct subjects and captured by Microsoft Kinect v2 cameras from three different views simultaneously. Each sample contains a skeleton sequence with the 3D coordinates of 25 body joints at each frame. The authors of the dataset recommend two evaluation protocols: (1) cross-subject (X-Sub): training on samples from 20 subjects, and testing on those from the other 20 subjects. (2) cross-view (X-View): training on samples captured by camera 2 and 3, and testing on those captured by camera 1.

TABLE I
COMPARISON OF DIFFERENT TEMPORAL MODELING METHODS.

	Methods	Params	FLOPs	Acc (%)	Δ
A	V-Trans	2.21M	7.21G	76.8	-
B	TCN	2.36M	2.36G	82.5	5.7
C	Global	2.21M	2.37G	82.3	5.5
D	Sliding-window	2.21M	2.32G	83.1	6.3
E	PAT-Trans	2.20M	2.31G	**84.1**	7.3

NTU RGB+D 120. NTU RGB+D 120 [50] is currently the largest dataset for skeleton-based action recognition. It extends NTU RGB+D by adding 57600 samples of 60 extra classes. Therefore, it contains 114480 samples of 120 classes in total, which are performed by 106 distinct subjects. There are 32 different setups, each denoting a specific location and background. The authors recommend two evaluation protocols: (1) cross-subject (X-Sub 120): training on samples from 53 subjects, and testing on those from the other 53 subjects. (2) cross-setup (X-Set 120): training on samples with even setup IDs and testing on those with odd setup IDs.

B. Implementation Detail

We implement the proposed TranSkeleton model with Pytorch. Four NVIDIA RTX 2080Ti GPUs are used for training and testing. The whole model is comprised of three stages, each containing two basic Transformer units. We set the feature dimensions of the three stages to 64, 128 and 256 respectively. Each MSA module in the basic unit has four heads. The expansion ratio of the MLP module is set to 2. We adopt the sampling strategy as in [51] and resize each input sequence to 64 frames by interpolation. We adopt Adam [52] optimizer and cross entropy loss to train for 70 epochs with a weight decay of 0.0001. The initial learning rate is set to 0.001, and decays with a factor of 0.1 at epoch 50 and 60.

C. Ablation Study

To assess the contributions of the individual components of the proposed TranSkeleton model, we conduct extensive ablation experiments on the cross-subject benchmark of the NTU RGB+D 120 dataset.

Temporal modeling method. As shown in Table I, all the methods apply multi-head self-attention for spatial correlation modeling, but adopt different temporal modeling methods. Note that V-Trans (vanilla Transformer) is non-hierarchical, while the rest methods conduct temporal aggregation to form a hierarchical architecture. Specifically, the detailed setups are

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3240472

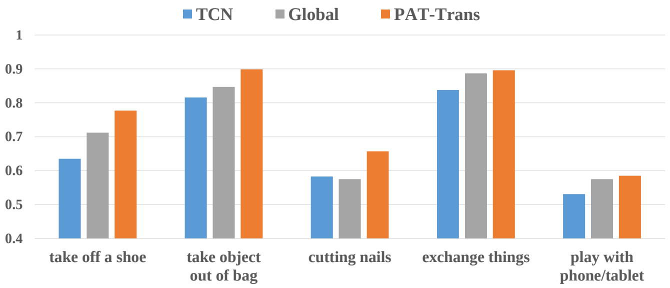


Fig. 4. Performance comparison of different temporal modeling methods on several action classes.

as follows. Global (model C): Applying Transformer for global modeling in all hierarchies without temporal partition. Sliding-window (model D): Applying Transformer in a sliding-window fashion, which is similar to TCN but the core computation is replaced with dot-product attention. PAT-Trans (model E): Our proposed method. For a fair comparison, the window size of model D is set equal to the segment length of ours. As can be seen, V-Trans leads to a drastic increase of computation cost and shows the worst performance. Model C outperforms V-Trans by 5.5%, showing that the hierarchical architecture largely reduces Transformer's redundancy and improves its validity. However, it has a slight performance drop compared to TCN-based local modeling. This indicates that lack of local modeling limits Transformer's temporal modeling ability. Sliding-window outperforms model C by 0.8%, indicating that introducing local modeling into Transformer even in a relatively naive way can improve its performance.

Finally, PAT-Trans achieves the best performance, surpassing model B and C by 1.6% and 1.8% respectively. Note that PAT-Trans in fact has the fewest parameters and FLOPs. Therefore, the performance boost is brought by the local-to-global modeling method itself, rather than the increase of model complexity or computation cost. Moreover, PAT-Trans outperforms model D by 1.0%, showing that our method is a superior way of introducing local modeling into Transformer.

Fig. 4 shows the performance comparison on several action classes where PAT-Trans exceeds TCN-based local modeling the most. "Take off a shoe", "take object out of bag" and "exchange things" are three complex actions that take place on a long temporal scale. "Cutting nails" and "play with phone/tablet" are two actions that are easily confused, as their poses are quite similar and require a relatively long period of observation to distinguish. It can be seen that global modeling outperforms TCN-based local modeling on most of these classes, while PAT-Trans further improves the performance. The results demonstrate our method can better capture long-range temporal dependencies and the subtle temporal structures of the input sequences simultaneously.

The number of segments. The number of segments S plays an important role in the hierarchical temporal partition process. In each hierarchy, a big S leads to local modeling within short segments and thus enhances the ability to capture the subtle temporal structure of the input sequence. In contrast, a small S makes the model perform longer temporal modeling and can better capture long-range dependencies. Therefore, S is a key factor in balancing local and global temporal modeling.

TABLE II
ABLATION STUDY ON THE NUMBER OF SEGMENTS. S_i DENOTES THE NUMBER OF SEGMENTS IN THE i-TH HIERARCHY.

S_1	S_2	S_3	Acc (%)	Δ
1	1	1	82.3	-
2	2	1	82.9	0.6
4	2	1	83.8	1.5
8	4	1	**84.1**	1.8
16	8	1	83.4	1.1

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT INPUT LENGTHS.

Input length	32f	48f	64f	80f	96f
Acc (%)	83.3	83.8	84.1	84.1	84.0

We build a TranSkeleton model of three hierarchies and compare the performance of different S. Note that S_i denotes the number of segments in the i-th hierarchy, and S_3 is set to 1 for global modeling in the last hierarchy. As shown in Table II, setting $S_1 = S_2 = S_3 = 1$ (*i.e.* performing global temporal modeling) yields the worst performance. Increasing S_1 and S_2 introduces local modeling into Transformer and notably improves the model's performance. However, when S_1 and S_2 become too big, the model would concentrate on local modeling within short segments, and thus harms its ability to capture long-range dependencies.

The number of input frames. Existing GCN-based methods [6], [7] generally take 300 frames as input by repeating the skeleton sequences of different lengths, which leads to redundant computation. Different from them, we adopt the sampling strategy as in [51] and resize each input sequence to a certain number of frames through interpolation. Table III shows the performance comparison of different input lengths. As can be seen, the model's performance initially increases with the input length. When the input length is greater than 64 frames, the performance becomes saturated. Therefore, considering the computation efficiency, we set the input length of our TranSkeleton model to 64 frames.

Positional embedding. To evaluate the impact of the spatial and temporal positional embeddings, we compare several different combinations, as shown in Table IV. The baseline model without any positional embedding yields the worst performance. Adding spatial positional embedding facilitates the correlation modeling among joints and brings a performance boost of 0.7%. Adding global temporal positional embedding in each hierarchy further improves the performance by 1.4%. This shows that temporal positional embedding has a greater impact on the performance, as Transformer cannot distinguish the order of the input frames without it. Finally, adding shared temporal positional embedding achieves a slightly better performance compared to the global one. We analyze that this is because a positional embedding shared by all segments may get more sufficient training, as we perform segment temporal modeling in stage 1 and 2.

TABLE IV
ABLATION STUDY ON THE SPATIAL AND TEMPORAL POSITIONAL EMBEDDINGS. PE DENOTES POSITIONAL EMBEDDING.

Spatial PE	Temporal PE	Acc (%)	Δ
✗	✗	81.8	-
✓	✗	82.5	0.7
✓	global	83.9	2.1
✓	shared	**84.1**	2.3

TABLE V
COMPARISON OF DIFFERENT TEMPORAL AGGREGATION APPROACHES. DATA DENOTES DIFFERENCE-AWARE TEMPORAL AGGREGATION.

Methods	Acc (%)	Δ
Avg-pooling	83.4	-
Max-pooling	84.1	0.7
DATA(Avg)	84.4	1.0
DATA(Max)	**84.9**	1.5

TABLE VI
ABLATION STUDY ON THE PHYSICAL CONNECTION CONSTRAINT (PCC). PA DENOTES PARAMETERIZED ADJACENT MATRIX.

Methods	Acc (%)	Δ
w/o PA	84.5	-
PA w/o PCC	84.9	0.4
PA w/ PCC	**85.6**	1.1

TABLE VII
COMPARISON OF MODEL COMPLEXITY WITH GCN-BASED METHODS. THE ACCURACY IS ON THE CROSS-SUBJECT BENCHMARK OF NTU RGB+D.

Methods	Params	FLOPs	Acc (%)
2s-AGCN [6]	3.5M	19.5G	86.5
DC-GCN [34]	3.4M	16.2G	88.2
MS-G3D [7]	3.2M	32.9G	89.4
TranSkeleton	$2.2\text{M}^{\downarrow 31\%}$	$2.3\text{G}^{\downarrow 86\%}$	**90.1**

Temporal aggregation. As shown in Table V, we compare the performance of different temporal aggregation manners. Average pooling yields the worst performance, as it results in a significant loss of high-frequency temporal information. Max pooling preserves the biggest response in each channel and achieves better performance. However, it drops the smaller values and causes much information loss as well. By taking the difference of consecutive frames' features into consideration, the DATA approach using avg-pooling outperforms plain average pooling by 1.0%. Replacing average pooling with max pooling further boosts the performance by 0.5%. These results demonstrate the proposed DATA approach effectively reduces information loss during temporal aggregation and enhances the model's temporal modeling ability.

Physical connection constraint. We evaluate the impact of the devised physical connection constraint. As shown in Table VI, the baseline model without parameterized adjacent (PA) matrix performs the worst. This is due to absolute position and human body topology information are both lost in the bone modality, and thus may hinder pure attention-based spatial modeling. Adding a parameterized adjacent matrix without any constraint boosts the performance, but the improvement is rather limited. Finally, applying the devised physical connection constraint leads to a performance boost of 1.1% compared to the baseline model. The results demonstrate that the devised PCC explicitly embeds the prior information of human body topology into the model, and thus effectively facilitates the spatial modeling.

Model complexity. Here we compare the number of parameters, floating point operations (FLOPs) and the classification accuracy on the cross-subject benchmark of the NTU RGB+D dataset. GCN-based methods [6], [7], [34] generally integrate multiple branches of graph convolution to extract richer spatial information, resulting in huge computation cost. In contrast, we unify the spatial-temporal modeling within a pure Transformer framework, and thus avoid such unnecessary increase of complexity. As shown in Table VII, our model reduces about 1/3 parameters compared to recent GCN-based methods, and has 7× and 14× fewer FLOPs than DC-GCN+ADG [34] and MS-G3D [7] respectively. Noticeably, taking only 64-frame skeleton sequences as input, our model still notably outperforms existing GCN-based methods whose input length is generally 300 frames. Since the computation cost of our model increases linearly with the input length, its FLOPs would still be 33%∼67% fewer than these methods even with the same input length. We also compare the training time of these methods. On NVIDIA RTX 2080Ti, the training of 2s-AGCN [6], DC-GCN [34] and MS-G3D [7] require 25, 33 and 87 GPU hours respectively. In contrast, the training of our model only takes 9 GPU hours. The results validate the high efficiency of the proposed TranSkeleton model.

D. Visualization

Spatial attention. We visualize the attention maps of the spatial MSA module in the 1st Transformer unit of our model. Fig. 5 shows the attention maps of four typical actions, *i.e.*, "drink water" (upper left), "put on a shoe" (upper right), "hand waving" (lower left) and "hopping" (lower right), where blue indicates bigger attention scores. In the upper left attention map, the two red boxes indicate a strong correlation between the head and the right arm, which is vital for classifying the "drink water" action. In the upper right attention map, the red boxes show the strong correlations between the right arm and two feet for the "put on a shoe" action. Likewise, in the lower left attention map, the model manages to capture the correlation between two arms for the "hand waving" action. Finally, in the lower right attention map, strong correlations are shown between "H" (right foot) and "P+Q" (head, shoulder and spine). This is quite interesting, as the "hopping" action means exactly "jumping on one foot", and our model successfully grasps the most discriminative correlations.

In addition, as shown in Fig. 6, we visualize the biggest attention scores of three typical actions with red lines. In the "brush hair" action, there are strong correlations between the head and two hands, which are important for classifying this action. Similarly, the model manages to capture the correlations between the left hand and two feet in the "stand up" action. Note that in the "clapping" action, the strongest correlations are between the base of the spine and two hands. This is kind of interesting as the correlation between two

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3240472

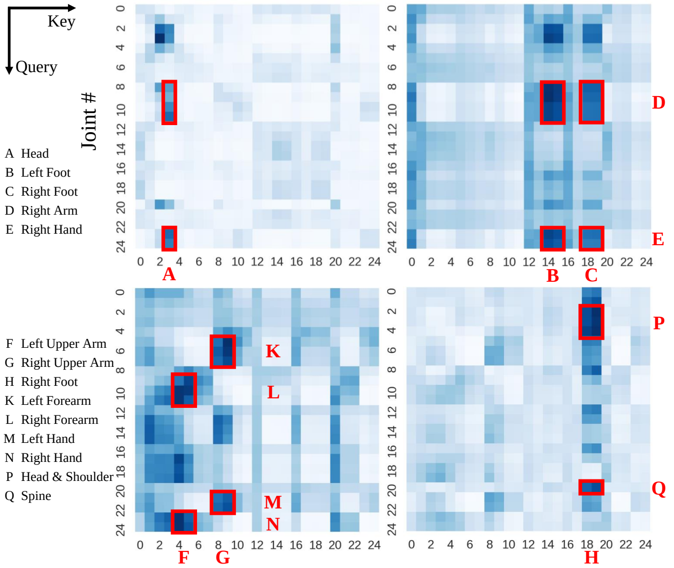
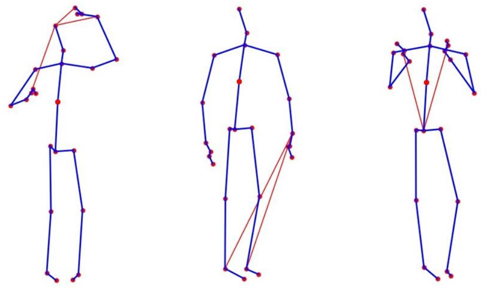


Fig. 5. Spatial attention maps of the "drink water" (upper left), "put on a shoe" (upper right), "hand waving" (lower left) and "hopping" (lower right) actions. Blue indicates bigger values.



(a) brush hair (b) stand up (c) clapping

Fig. 6. Spatial attention visualization of the "brush hair", "stand up" and "clapping" actions. The red lines indicate the biggest attention scores among all the joints.

hands intuitively seems more important. We conjecture this is because it's challenging to directly capture the correlation between two hands, as both of them are moving fast when clapping. The above visualization results demonstrate that the spatial attention in our model can well capture key correlations among joints for correct action classification.

Temporal attention. We visualize the attention maps of the temporal MSA module in the last Transformer unit of our model, but in the form of curves. Specifically, for each joint, we sum its temporal attention scores along the query dimension, and thus turn the 2D attention map into an 1D vector. We term this 1D vector as ***temporal attention intensity***. Each value in this vector indicates how much the corresponding frame contributes in the dot-product attention process, *i.e.*, how much the temporal attention concentrates on the corresponding frame. As shown in Fig 7, we visualize the temporal attention intensity of some typical joints for three actions, *i.e.*, "brush hair", "stand up" and "take object out of bag". Note that for clarity, the marked frames (left part) are visualized using their corresponding images (right part) instead of skeletons.

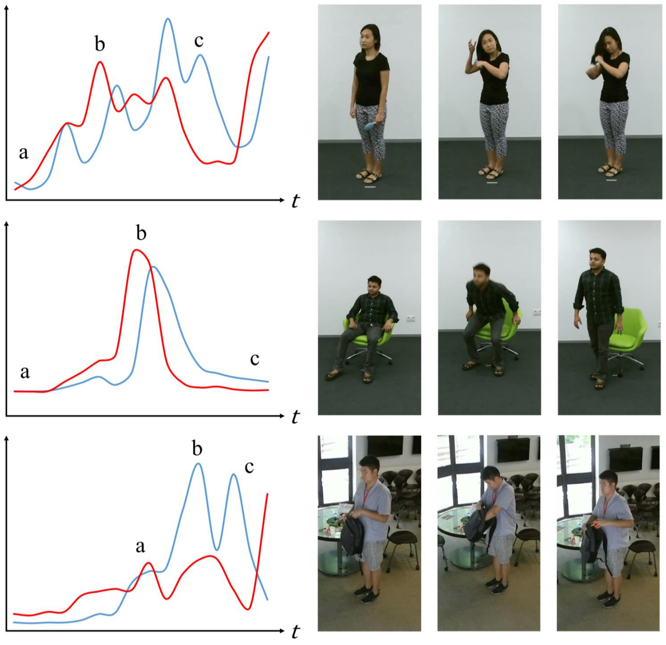


Fig. 7. **Temporal attention intensity** visualization of the "brush hair", "stand up" and "take object out of bag" actions. Note that for clarity, the marked frames (left part) are visualized using their corresponding images (right part) instead of skeletons.

As in Fig. 7, the top row shows the temporal attention intensity curves and the corresponding frames of the "brush hair" action. The red curve represents the left hand and the blue curve represents the right hand. As can be seen, the two curves have similar tendencies. In frame a, the performer is just standing there holding the brush. Therefore, the corresponding temporal attention intensity values are low. In contrast, the attention intensity values become high in frame b and c, as the "brush hair" action is ongoing in these frames. Here we have an interesting finding. In this video, the performer brushes hair with her left and right hands alternatively. As the curves show, our model precisely grasps this subtle temporal structure. This further validates the effectiveness of our proposed partition-aggregation temporal Transformer.

The middle row shows the visualization result of the "stand up" action. The red curve represents the spine and the blue curve represents the right foot. As can be seen, there's a peak in both curves, showing that the temporal attention manages to focus on discriminative frames where the actual "stand up" action takes place. Finally, the bottom row shows the visualization result of the "take object out of bag" action, which is a typical complex action comprising several simple sub-actions. The red curve represents the left hand and the blue curve represents the right hand. Frame a, b and c show the corresponding sub-actions "unzip the bag", "reach into the bag" and "take out the object". As can be seen, both curves cover the second half of the sequence where the action actually happens. We also notice the difference between the two curves especially the opposite trends around frame c. We conjecture this is due to the severe occlusion when the left hand reaches into the bag. The above visualization results demonstrate the temporal attention can well capture

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3240472

TABLE VIII
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE NTU RGB+D AND NTU RGB+D 120 DATASETS.

Methods	X-Sub(%)	X-View(%)	X-Sub 120(%)	X-Set 120(%)	Params(M)	FLOPs(G)
Ind-RNN [55]	81.8	88.0	-	-	-	-
HCN [51]	86.5	91.1	-	-	-	-
ST-LSTM [11]	-	-	55.7	57.9	-	-
RotClips+MTCNN [14]	-	-	62.2	61.8	-	-
ST-GCN [5]	81.5	88.3	70.7⋆	73.2⋆	3.1	16.2
RA-GCN [8]	87.3	93.6	81.1	82.7	2.0	32.8
2s-AGCN [6]	88.5	95.1	82.5⋆	84.2⋆	3.5	39.0
SGN [35]	89.0	94.5	79.2	81.5	0.7	0.8
Shift-GCN [29]	90.7	96.5	85.9	87.6	0.7	10.0
DC-GCN [34]	90.8	96.6	86.5	88.1	3.4	64.8
PA-ResGCN-B19 [30]	90.9	96.0	87.3	88.3	3.6	18.5
MS-G3D [7]	91.5	96.2	86.9	88.4	3.2	132
MST-GCN [37]	91.5	96.6	87.5	88.8	3.0	-
DualHead-Net [38]	92.0	96.6	88.2	89.3	3.0	-
TranSkeleton (Joint)	90.1	95.4	84.9	86.3	-	2.3
TranSkeleton (Bone)	90.3	94.5	85.6	86.8	-	2.3
TranSkeleton	**92.8**	**97.0**	**89.4**	**90.5**	2.2	9.2

⋆: The results are implemented based on the released codes.

both long-range dependencies and subtle temporal structures. Meanwhile, it is adaptive to different input sequences and can focus on discriminative frames.

E. Comparison with the State-of-the-Art

As there is complementarity between different modalities, state-of-the-art methods generally adopt multi-stream fusion to boost their performance. For a fair comparison, we apply a similar score-level fusion strategy to obtain the final results. Specifically, we combine the classification results of four individual modalities, *i.e.*, joint, bone, joint/bone and joint motion. Here joint/bone is a hybrid modality which concatenates the joint and bone data in the channel dimension. And joint motion is the temporal differential between consecutive frames of the joint modality.

To verify the effectiveness of our TranSkeleton model, we compare it with state-of-the-art methods on two large-scale benchmark datasets: NTU RGB+D and NTU RGB+D 120. The corresponding results are shown in Table VIII. On both benchmarks of NTU RGB+D 120 (*i.e.* X-Sub 120 and X-Set 120), TranSkeleton surpasses existing TCN-GCN methods by a large margin. For instance, the state-of-the-art method DualHead-Net [38] in fact combines multi-scale GCN/TCN with MS-G3D [7] to form a complex two-branch network. Thus its computation cost could be larger than MS-G3D, which has 14× FLOPs than ours. TranSkeleton significantly outperforms it by 1.2% on X-Sub 120 and X-Set 120, with much lower model complexity. On the X-Sub benchmark of NTU RGB+D, TranSkeleton also largely outperforms DualHead-Net [38] by 0.8%. Even on the highly saturated X-View benchmark, we also surpass existing state-of-the-art methods by 0.4%. We'd also like to clarify that, as another research direction, extreme efficiency [35], [53], [54] isn't the main focus of our work, for it inevitably harms performance. For instance, we largely outperform the lightweight SGN [35] model by 10.2% and 9.0% on X-Sub 120 and X-Set 120 respectively. The above experimental results verify the effectiveness of our method.

V. CONCLUSION

In this work, we present TranSkeleton, a concise yet powerful Transformer framework which unifies spatial and temporal modeling for skeleton-based action recognition. For temporal modeling, we propose a novel partition-aggregation temporal Transformer which works with hierarchical partition and aggregation. It effectively captures both long-range dependencies and subtle temporal structures, and is demonstrated better than TCN and vanilla Transformer. A difference-aware aggregation approach is also designed to reduce the information loss caused by temporal aggregation. Besides, for effective spatial modeling, we devise a physical connection constraint to form a topology-aware spatial Transformer. Experimental results and comprehensive analysis on two challenging skeleton datasets demonstrate that the proposed TranSkeleton notably surpasses the state-of-the-art counterparts.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014, pp. 568–576.

[2] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*. Springer, 2016, pp. 20–36.

[3] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *ICCV*, 2019, pp. 6202–6211.

[4] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *CVPR*, 2015, pp. 1110–1118.

[5] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI*, vol. 32, no. 1, 2018.

[6] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *CVPR*, 2019, pp. 12 026–12 035.

[7] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *CVPR*, 2020, pp. 143–152.

[8] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Richly activated graph convolutional network for robust skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1915–1925, 2020.

[9] H. Wang, B. Yu, J. Li, L. Zhang, and D. Chen, "Multi-stream interaction networks for human action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 3050–3060, 2021.

[10] X. Shu, J. Yang, R. Yan, and Y. Song, "Expansion-squeeze-excitation fusion network for elderly activity recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[11] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *ECCV*. Springer, 2016, pp. 816–833.

[12] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention lstm networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, 2017.

[13] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," in *CVPR*, 2017, pp. 499–508.

[14] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3d action recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2842–2855, 2018.

[15] C. Cao, C. Lan, Y. Zhang, W. Zeng, H. Lu, and Y. Zhang, "Skeleton-based action recognition with gated convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3247–3257, 2018.

[16] A. Banerjee, P. K. Singh, and R. Sarkar, "Fuzzy integral-based cnn classifier fusion for 3d skeleton action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 6, pp. 2206–2216, 2020.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[18] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[19] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *ICML*. PMLR, 2019, pp. 6861–6871.

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2020.

[21] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021, pp. 10 012–10 022.

[22] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *ICML*. PMLR, 2021, pp. 813–824.

[23] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *ICCV*, 2021, pp. 6836–6846.

[24] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1963–1978, 2019.

[25] R. Xia, Y. Li, and W. Luo, "Laga-net: Local-and-global attention network for skeleton based action recognition," *IEEE Transactions on Multimedia*, vol. 24, pp. 2648–2661, 2021.

[26] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *CVPR*, 2019, pp. 7912–7921.

[27] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *CVPR*, 2019, pp. 3595–3603.

[28] X. Zhang, C. Xu, and D. Tao, "Context aware graph convolution for skeleton-based action recognition," in *CVPR*, 2020, pp. 14 333–14 342.

[29] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *CVPR*, 2020, pp. 183–192.

[30] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition," in *ACM MM*, 2020, pp. 1625–1633.

[31] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," in *AAAI*, vol. 34, no. 03, 2020, pp. 2669–2676.

[32] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *ICCV*, 2021, pp. 13 359–13 368.

[33] Z. Huang, X. Shen, X. Tian, H. Li, J. Huang, and X.-S. Hua, "Spatio-temporal inception graph convolutional networks for skeleton-based action recognition," in *ACM MM*, 2020, pp. 2122–2130.

[34] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, "Decoupling gcn with dropgraph module for skeleton-based action recognition," in *ECCV*. Springer, 2020, pp. 536–553.

[35] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *CVPR*, 2020, pp. 1112–1121.

[36] B. Xu, X. Shu, and Y. Song, "X-invariant contrastive augmentation and representation learning for semi-supervised skeleton-based action recognition," *IEEE Transactions on Image Processing*, 2022.

[37] Z. Chen, S. Li, B. Yang, Q. Li, and H. Liu, "Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition," in *AAAI*, vol. 35, no. 2, 2021, pp. 1113–1122.

[38] T. Chen, D. Zhou, J. Wang, S. Wang, Y. Guan, X. He, and E. Ding, "Learning multi-granular spatio-temporal graph network for skeleton-based action recognition," in *ACM MM*, 2021, pp. 4334–4342.

[39] J. Kong, H. Deng, and M. Jiang, "Symmetrical enhanced fusion network for skeleton-based action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4394–4408, 2021.

[40] C. Plizzari, M. Cannici, and M. Matteucci, "Spatial temporal transformer network for skeleton-based action recognition," in *ICPR*. Springer, 2021, pp. 694–701.

[41] R. Bai, M. Li, B. Meng, F. Li, J. Ren, M. Jiang, and D. Sun, "Gcst: Graph convolutional skeleton transformer for action recognition," *arXiv preprint arXiv:2109.02860*, 2021.

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

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

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[45] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *ICML*. PMLR, 2021, pp. 10 347–10 357.

[46] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*. Springer, 2020, pp. 213–229.

[47] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *CVPR*, 2021, pp. 6881–6890.

[48] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[49] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *CVPR*, 2016, pp. 1010–1019.

[50] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2684–2701, 2019.

[51] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *IJCAI*, 2018, pp. 786–792.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[53] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Adasgn: Adapting joint number and model size for efficient skeleton-based action recognition," in *ICCV*, 2021, pp. 13 413–13 422.

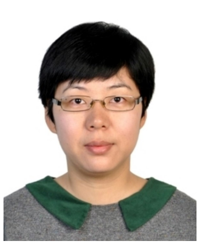
[54] W. Peng, J. Shi, T. Varanka, and G. Zhao, "Rethinking the st-gcns for 3d skeleton-based human action recognition," *Neurocomputing*, vol. 454, pp. 45–53, 2021.

[55] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (indrnn): Building a longer and deeper rnn," in *CVPR*, 2018, pp. 5457–5466.

Haowei Liu received the B.S. degree in automation from Tsinghua University. He is currently pursuing the Ph.D. degree with the Institute of Automation, University of Chinese Academy of Sciences. His research interests mainly include computer vision and machine learning, particularly the theory and applications of human action recognition.

Yongcheng Liu received the Ph.D. degree in School of Artificial Intelligence from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2020. He worked as an assistant professor in the Institute of Automation, Chinese Academy of Sciences, from 2020 to 2022. He is currently an algorithm expert in Amap, Alibaba. His research interests include image semantic segmentation, 3D point cloud processing, and multimodal visual understanding.

Yuxin Chen received the B.S. degree in automation science from Beihang University, Beijing, China. He is currently pursuing the Ph.D. degree with the Institute of Automation, University of Chinese Academy of Sciences (UCAS). His research interests include the theory and applications of action recognition, vision-language pre-training.



Chunfeng Yuan received the Ph.D. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2010. She is currently an associate professor with CASIA. Her research interests and publications range from pattern recognition to computer vision, including sparse representation, deep learning, video understanding, and multimedia analysis.

Bing Li received the Ph.D. degree from the Department of Computer Science and Engineering, Beijing Jiaotong University in 2009. From 2009 to 2011, he worked as a postdoctoral research fellow with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA). He is currently a professor at CASIA. His current research interests include computer vision, color constancy, visual saliency detection, multi-instance learning, and data mining.

Weiming Hu received the Ph.D. degree from the Department of Computer Science and Engineering, Zhejiang University, China, in 1998. From 1998 to 2000, he was a postdoctoral research fellow with the Institute of Computer Science and Technology, Peking University. He is currently a professor with the Institute of Automation, Chinese Academy of Sciences (CASIA). His research interests are visual analysis, recognition of web objectionable information, and network intrusion detection.