IIP-Transformer: Intra-Inter-Part Transformer for Skeleton-Based Action Recognition

Qingtian Wang[1], Jianlin Peng[1,3], Shuze Shi[1,2], Tingxi Liu[1], Jiabin He[1], Renliang Weng[1]
[1]Algorithm Research, Aibee Inc
[2]Beijing Jiaotong University
[3]Nanjing University of Aeronautics and Astronautics
{qtwang,jlpeng,tliu,jhe,rlweng}@aibee.com,19120306@bjtu.edu.cn

Abstract

Recently, Transformer-based networks have shown great promise on skeleton-based action recognition tasks. The ability to capture global and local dependencies is the key to success while it also brings quadratic computation and memory cost. Another problem is that previous studies mainly focus on the relationships among individual joints, which often suffers from the noisy skeleton joints introduced by the noisy inputs of sensors or inaccurate estimations. To address the above issues, we propose a novel Transformer-based network (IIP-Transformer). Instead of exploiting interactions among individual joints, our IIP-Transformer incorporates body joints and parts interactions simultaneously and thus can capture both joint-level (intra-part) and part-level (inter-part) dependencies efficiently and effectively. From the data aspect, we introduce a part-level skeleton data encoding that significantly reduces the computational complexity and is more robust to joint-level skeleton noise. Besides, a new part-level data augmentation is proposed to improve the performance of the model. On two large-scale datasets, NTU-RGB+D 60 and NTU RGB+D 120, the proposed IIP-Transformer achieves the-state-of-art performance with more than 8× less computational complexity than DSTA-Net, which is the SOTA Transformer-based method.

1. Introduction

Human action recognition has been studied during the past decades and achieves promising progress in many applications ranging from human-computer interaction to video retrieval [4, 12, 28, 29]. Recently, skeleton-based representation has received increasing attention due to its compactness of depicting dynamic changes in human body movements [14]. Advanced pose estimation algorithms [2] and advances in the somatosensory cameras such as Kinect [38] and RealSense [15] make it possible to obtain body keypoints accurately and quickly at a low cost. In addition, skeleton-based representation is more robust to variations of illumination and background noises in contrast to RGB representation. These merits attract researchers to develop various methods to exploit skeleton for action recognition.

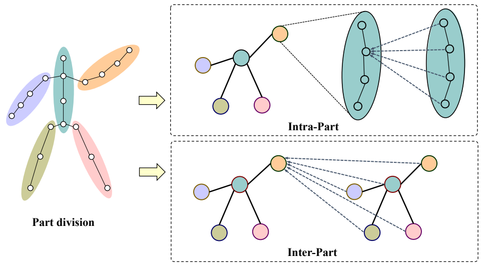

Figure 1. Illustration of our main idea. The body joints are divided into 5 parts. The Inter-Part branch is used to explore relationships between parts and the Intra-Part branch aims to capture dependencies between joints in the same part.

Previous skeleton-based action recognition methods utilize graph topologies or manually designed rules to transform the raw skeleton sequence into a grid-shape structure such as pseudo-image [10, 17, 20] or graph, and then feed it into RNNs, CNNs, or GCNs [36] to extract features. However, there is no guarantee that the hand-crafted structure is the best choice of capturing joints relationships, which limits the generalizability and performance of previous works. Recently, Transformer-based methods [22, 27] have been proposed, relying on the multi-head self-attention mechanism which adaptively explores the potential dependencies between skeleton joints. Specifically, Shi *et al*. [27] treat each individual joint as a token while the calculation of self-attention grows quadratically to the number of tokens, thus introduces a huge amount of calculations. This work also suffers from the noisy skeleton joints collected by sensors or inaccurate estimations.

To solve these problems, we introduce the concept of body parts into transformer network. Most of actions such as standing up or dancing are performed by co-movement of body parts. Actually, body parts can be considered as the minimum units of action execution, which means these actions can be identified only by the movement of body parts. Different from other complex partition strategies, we simply aggregate body joints into several parts according to human body topologies and encourage the model to exploit the complicated interactions. Specifically, we divide v body joints into p parts and encode each part into a token, which reduces the spatial self-attention computation cost by v^2/p^2 times. Another advantage of our proposed partition encoding is that it enables the model to take sparser frames as temporal inputs, which brings in additional computation reduction. To encourage the model to reason globally instead of relying on a particular part, we propose a new data augmentation method named Part-Mask which masks out a part randomly during training. This new strategy makes the model more robust across challenging cases.

Since the body joints are divided into parts, the joint-level information may be lost. For some fine-grained actions, *e.g.* clapping or writing, it is necessary to capture the interactions between body joints additionally. We propose the novel Intra-Inter-Part Transformer network (IIP-Transformer) to tackle this issue and make three main improvements comparing with standard Transformer networks. First, Intra-Inter-Part self-attention mechanism is proposed to simultaneously capture intra-part features and inter-part features without increasing much computational complexity, as depicted in Figure 1. Second, inspired by BERT [9], we introduce a learnable class-token instead of pooling all features extracted by backbone. Last, instead of using two individual transformers to model spatial and temporal dependencies, we propose a new spatial-temporal transformer that reduces model size while increases the generalization of the model. The code and models will be made publicly available at `https://github.com/qtwang0035/IIP-Transformer`.

Overall, our contributions can be summarized as follows:

- We introduce the concept of body parts into transformer-based skeleton action recognition. Our proposed partition encoding significantly reduces self-attention computational complexity and is relatively insensitive to joint noise.
- We propose IIP-Transformer, a novel spatial-temporal transformer network that captures intra-part and inter-part relations simultaneously.
- Extensive experiments on two large-scale skeleton action datasets, *e.g.* NTU RGB+D 60 & 120, show that our proposed IIP-Transformer achieves state-of-the-art performance with 2 $\sim$ 36$\times$ less computational cost.

2. Related Work

Transformer. In recent years, with the development of Natural Language Processing (NLP) tasks, the Transformer structure [34] has been proposed to replace the traditional NLP network structures, *e.g.*, RNNs. Different with RNN architectures, the encoder and decoder of transformers completely rely on the self-attention mechanism, which can effectively solve the problems of long-sequence modeling and parallel processing. Because of these characteristics of the self-attention mechanism, the Transformer networks have been also applied in various computer vision tasks [1, 3, 11, 22] and achieve superior results comparing with the CNN models.

Skeleton-based Action Recognition. Skeleton-based action recognition has received increasing attentions due to its compactness comparing with the RGB-based representations. Previous data-driven methods rely on manual designs of traversal rules to transform the raw skeleton data into a meaningful form such as a point-sequence or a pseudo-image, so that they can be fed into the deep networks such as RNNs or CNNs for feature extraction [8, 16, 17, 35]. Inspired by the booming graph-based methods, Yan *et al.* [36] introduce GCN into the skeleton-based action recognition task, and propose the ST-GCN to model the spatial configurations and temporal dynamics of skeletons simultaneously. The GCN-based methods [7, 26] use the topological structure of the human skeleton to aggregate features of related skeleton nodes and time series. Therefore, the GCN-based methods show better performance than previous methods. Instead of formulating the skeleton data into the images or graphs, Transformer-based methods directly model the dependencies of joints with pure attention blocks. Plizzari *et al.* [22] propose a method that introduces Transformer in skeleton activity recognition and combine it with GCN. Shi *et al.* [27] employ a solely transformer network to exploit relations between joints. Our proposed method utilizes part-level input, thus could efficiently capture both joint-level and part-level relations.

Part-based Methods. Part-based methods are designed to extract features of body parts individually since human body is a natural topology with five major parts. Thakkar *et al.* [33] divide graph into several sub-graphs with shared nodes. They employ GCN operation within sub-graphs of body parts and then propagate information between sub-graphs via the shared nodes. Huang *et al.* [13] propose an

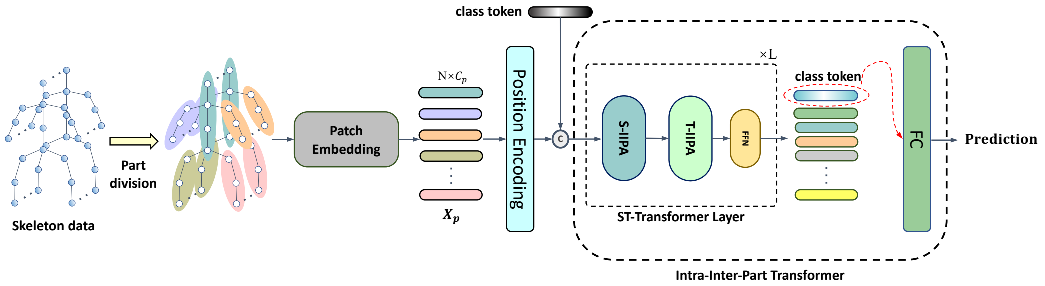

Figure 2. The overall architecture of the proposed pipeline which is composed of Partition Encoding and IIP-Transformer.

automatic partition strategy and utilize a part-based GCN to explore discriminative features from joints and body parts. Song *et al.* [31] propose part-attention mechanism to discover the most informative parts. All these part-based methods employ complex strategies to propagate information individually or fuse information from all parts, while our work focuses on simultaneously capturing discriminative features from intra and inter parts with less computational cost.

3. Methods

3.1. Overall Architecture

Figure 2 shows the overall architecture of our pipeline, which mainly contains two sections, the partition encoding and the IIP-Transformer network. For an input skeleton sequence with V joints, F frames and C channels, we first divide V joints into P parts, and then a patch embedding layer is used to project these part data into N tokens with dimension C_p, where $N = P \times F$. Before being fed into IIP-Transformer, a class-token will be concatenated with tokens above, resulting in $N + 1$ tokens. There are L layers stacked in IIP-Transformer in total and each layer is composed of a spatial IIPA module (S-IIPA) that models the spatial relations between parts in the same frame, a temporal IIPA module (T-IIPA) that captures the temporal relations of parts among different frames and a feed-forward network. Each layer of IIP-Transformer maintains the same number of tokens, and thus we get $N + 1$ tokens as final output features. We feed the feature corresponding to the class-token into a fully-connected layer to obtain the classification scores. The implementation details will be introduced in the following sections.

3.2. Partition Encoding

In order to reduce the computation complexity of self-attention in spatial dimension, we propose a strategy to encode the joints into P parts (5 parts in our case specifically). The partition encoding procedure is illustrated in Figure 3. First, the raw skeleton sequence $X \in R^{C\times F\times V}$ is fed into feature extraction layers $f_J(\cdot)$ to obtain a deeper feature $X_J \in R^{C_o\times F\times V}$, where C_o denotes the channel of features after feature extraction and $f_J(\cdot)$ is implemented by two convolution layers with BatchNorm and ReLU function. Then P individual body parts are obtained by selecting corresponding joints:

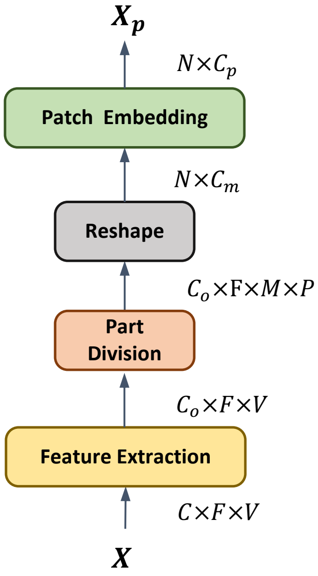

Figure 3. Illustration of the Partition Encoding Procedure.

$$X_J \rightarrow [x_1, x_2, \cdots, x_P], x_i \in R^{C_o\times F\times M} \tag{1}$$

where M denotes the max number of joints in each part. The parts with less than M joints will be padded with zero. Subsequently, we concatenate P parts and permute the dimensions:

$$\widetilde{X_J} = Concat\left(x_i | i = 1, 2, \cdots, P\right) \tag{2}$$

$$\widetilde{X_J} \in R^{C_o\times F\times M\times P} \rightarrow R^{N\times C_m} \tag{3}$$

where $N = P \times F$ and $C_m = C_o \times M$. Finally, $\widetilde{X_J}$ is fed into patch embedding layer $f_P(\cdot)$ to aggregate information of the internal skeleton joints of a part and we get the final partition encoding $X_P \in R^{N \times C_P}$, where C_P is the channels of the partition encoding and $f_P(\cdot)$ is implemented by linear layers with ReLU function.

By aggregating the information of the internal skeleton joints of the parts, an informative encoding is extracted. It drives the model to concentrate on body parts instead of joints, and thus reduce the influence of individual noisy joints. Besides, experiments show that the partition encoding enables model to take sparser temporal inputs.

3.3. Intra-Inter-Part Transformer

The backbone of IIP-Transformer is constructed by stacking L layers of Spatial-Temporal Transformer Layer which is composed of S-IIPA, T-IIPA and feed-forward network. The core of S-IIPA and T-IIPA is Intra-Inter-Part Self-Attention mechanism. In this section, the IIPA mechanism and Spatial-Temporal Transformer will be briefly introduced.

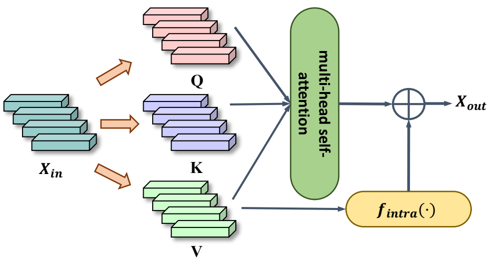

Figure 4. Illustration of the IIPA mechanism. The original multi-head self attention is used to explore relations between tokens while the function f_{intra} is used to capture internal relations of a token.

Intra-Inter-Part Self-Attention. Basic transformer utilizes self-attention mechanism to model the relationships between input tokens. But in part-level cases, each token represents a collection of joints in the same part. Therefore, the relationships of joints in the same part have not been fully exploited and can not be propagated effectively. We present a new self-attention mechanism named Intra-Inter-Part self-attention (IIPA) to simultaneously capture relationships inside and between tokens. As shown in Figure 4, our proposed IIPA mainly consists of a standard multi-head self-attention and an intra-part branch.

Given the input feature X_{in}, we first compute the query Q, key K and value V using three linear projection layers. The information flow across tokens is achieved by the standard multi-head self-attention:

$$X_{inter} = MHSA(Q, K, V) \tag{4}$$

The intra-part branch $f_{intra}(\cdot)$ is implemented by linear layers. It takes value V as input and extracts features inside tokens, termed joint-level feature.

$$X_{intra} = f_{intra}(V) \tag{5}$$

Finally, we fuse the multi-head self-attention output X_{inter} with the intra-part branch output X_{intra}, so that features extracted by IIPA carry both joint-level and part-level information.

$$X_{out} = X_{inter} + X_{intra} \tag{6}$$

Comparing with standard self-attention mechanism, our proposed IIPA introduces no more than 1/4 calculations, but achieves obvious improvement in fine-grained actions, as shown in ablation study.

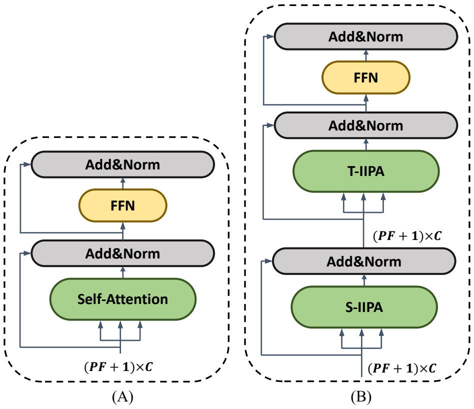

Figure 5. Two Transformer structures. (A) Flatten the spatial-temporal data into a single sequence and use an standard transformer. (B) Explore spatial and temporal relations with S-IIPA and T-IIPA respectively and fuse them by a feed-forward layer.

Spatial-Temporal Transformer Layer. In previous skeleton-based action recognition transformers [22, 25], all features extracted by backbone is average-pooled to obtain the final feature for classification. Inspired by BERT [9], class-token is introduced into our model. In BERT, the input $X_{1d} \in R^{(N+1) \times C}$ is a single dimension sequence with N tokens and a class-token. The single dimension structure naturally fits the self-attention mechanism. But for the skeleton sequence which has two dimensions, there are two problems to solve: one is how to exploit spatial and temporal dimension with self-attention mechanism, another is how to deal with class-token. The first method is flattening the raw skeleton sequence $X_{2d} \in R^{P \times F \times C}$ into a single dimension sequence $X_{flatten} \in R^{PF \times C}$ and concatenating it with a class-token to obtain $X_{in} \in R^{(PF+1) \times C}$, where P denotes the number of parts and F denotes the number of frames. The output $X_{out} \in R^{(PF+1) \times C}$ is calculated by a standard transformer and the corresponding structure is shown in Figure 5-(A):

$$X_{out} = softmax\left(\frac{QK^T}{\sqrt{C}}\right)V \tag{7}$$

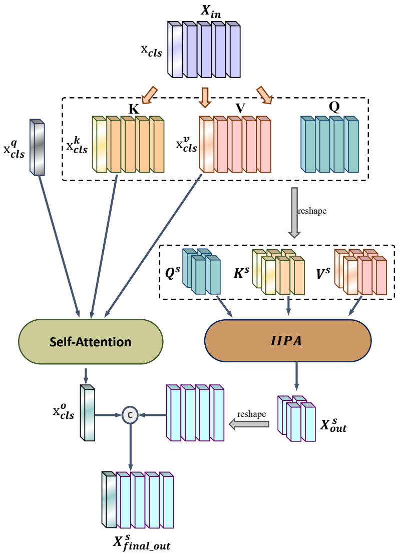

Figure 6. Illustration of S-IIPA. The symbol C stands for concatenate operation. T-IIPA has the same structure as S-IIPA except for reshape size of K, Q, V.

However, it is unreasonable to treat spatial and temporal features equivalently since the semantic information they contain are totally different. Besides, the computational complexity of calculating the attention map is quadratic to sequence length, therefore flattening the skeleton sequence into a long single dimension sequence will introduce a large amount of calculations.

Instead, the spatial and temporal dimension are treated as two different dimensions in Spatial-Temporal Transformer layer and processed by S-IIPA and T-IIPA respectively, as shown in Figure 5-(B). The input of S-IIPA and T-IIPA is the same as above, a flattened sequence with a class-token. S-IIPA is adopted to explore the dependencies of parts in a single frame while T-IIPA is employed to propagate information for each part across different frames. Let's take S-IIPA for an example, we first project the part-level sequence $X_{in} \in R^{(PF+1)\times C}$ with three linear projections layers $f_q(\cdot), f_k(\cdot)$ and $f_v(\cdot)$:

$$Q_{cls}, Q = f_q\left(X_{in}\right) \tag{8}$$

$$K_{cls}, K = f_k\left(X_{in}\right) \tag{9}$$

$$V_{cls}, V = f_v\left(X_{in}\right) \tag{10}$$

where $Q_{cls}, K_{cls}, V_{cls} \in R^{1\times C}$ and $Q, K, V \in R^{PF\times C}$. Due to the existence of class-token, we employ two branches, the self-attention branch and IIPA branch, to model class-token and other tokens separately, as shown in Figure 6. The self-attention branch is similar to the above method, and thus the output $X^o_{cls} \in R^{1\times C}$ could obtain information from all parts at all frames.

$$K_c = Concat\left(K_{cls}, K\right) \tag{11}$$

$$V_c = Concat\left(V_{cls}, V\right) \tag{12}$$

$$X^o_{cls} = softmax\left(\frac{Q_{cls}\left(K_c\right)^T}{\sqrt{C}}\right)V_c \tag{13}$$

In the IIPA branch, we reshape K, Q, V from single dimension into temporal×spatial dimension and repeat $K_{cls}, Q_{cls}, V_{cls}$ in the temporal dimension, then concatenate them in the spatial dimension.

$$Q, K, V \in R^{PF\times C} \rightarrow Q^s, K^s, V^s \in R^{F\times P\times C} \tag{14}$$

$$K_{cls}, V_{cls} \in R^{1\times C} \rightarrow R^{F\times 1\times C} \tag{15}$$

$$K^s = Concat\left(K_{cls},\ K^s\right) \tag{16}$$

$$V^s = Concat\left(V_{cls},\ V^s\right) \tag{17}$$

By following the preceding steps, we get $K^s, V^s \in R^{F\times(P+1)\times C}$, $(P+1)$ means appending a class-token in every frame. Then Q^s, K^s, V^s are fed into IIPA to calculate the output frame by frame, and each frame uses a unique attention map:

$$X_f = softmax\left(\frac{Q^s_f\left(K^s_f\right)^T}{\sqrt{C}}\right)V^s_f,\ f = 1, 2\cdots, F \tag{18}$$

where $X_f \in R^{P\times C}$ is the output for frame f, $Q^s_f \in R^{P\times C}$; K^s_f, $V^s_f \in R^{(P+1)\times C}$. Finally, we concatenate the outputs of all frames and reshape it back to single dimension:

$$X^s_{out} = Concat\left(\{X_f | f = 1, 2, \cdots, F\}\right) \tag{19}$$

$$X^s_{out} \in R^{F\times P\times C} \rightarrow R^{FP\times C} \tag{20}$$

$$X^s_{final\_out} = Concat\left(X^o_{cls}, X^s_{out}\right) \tag{21}$$

where $X^s_{final\_out} \in R^{(FP+1)\times C}$ is the final output of S-IIPA with the same dimension size as the input sequence. The only difference in T-IIPA is that $Q^t \in R^{P\times F\times C}; K^t, V^t \in R^{P\times(F+1)\times C}$, corresponding to Q^s, K^s, V^s, thus the output is calculated part by part.

Comparing with other spatial-temporal transformer methods, *e.g.* DSTA [27], which uses two complete transformers to model space and time, we remove the feed-forward layer in the spatial transformer structure and combine S-IIPA, T-IIPA and feed-forward layer as a new spatial-temporal transformer structure. This structure performs spatial and temporal feature extraction in one stage so that the same-order features can be fully exploited. Meanwhile, it also reduces a large amount of parameters in feed-forward layer and improves the generalizability of the model.

Rotation + Part-based masking

3.4. Data Augmentation

Rao *et al*. [23] propose several joint-level skeleton data augmentation methods, such as *Rotation*, *GaussianNoise*, *GaussianBlur*, *JointMask*, to improve the generalizability of models. However, since the part-level encoding is more robust to joint noises, these methods do not work well on part-level skeleton data, except for *Rotation*. Therefore we propose a new data augmentation method called *PartMask*, to encourage the model to reason globally instead of relying on a particular part. The *Rotation* and *PartMask* methods are defined as follows.

Rotation. Most public skeleton-based datasets are captured in specific view points. We obtain plentiful data through *Rotation* transformation which simulates viewpoint change of the camera.

Based on Euler's rotation theorem, any 3D rotation can be disassembled into a composition of rotations about three axes [35]. The basic rotation matrices are represented as below:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{22}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{23}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{24}$$

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha) \tag{25}$$

where $R_x(\alpha), R_y(\beta), R_Z(\gamma)$ denote the rotation matrices of x, y, z axis with angle α, β, γ respectively, and R is the general rotation matrix that will be applied to original coordinates of the skeleton sequence. In this work, the rotation angles are randomly sampled from $[-\pi/10, \pi/10]$.

PartMask. As an effective augmentation strategy to reduce reliance on specific regions, the mask strategy has been widely used in data augmentation. But simply employing a random zero-mask to a number of body joints in skeleton frames before partition encoding, which is similar to joint-level noises, does not work well due to the anti-noise ability of partition encoding. On the other hand, capturing the global information instead of focusing on a particular part will benefit the action classification tasks. Therefore we employ a part-level mask to encourage the model to reason globally. Specifically, we randomly select a certain body part p from $[1, 2, \cdots, P]$, and apply zero-mask to it in all frames:

$$\begin{aligned} X_J &= X_J \odot mask \\ &= [x_1, x_2, \cdots, x_P] \odot [1, 1, \cdots, \underset{p}{0}, \cdots, 1] \end{aligned} \tag{26}$$

where X_J denotes the intermediate result of partitioning, as discussed in Section 3.2, and mask is an all-ones vector except for the position of the selected part p.

4. Experiments

4.1. DataSets

NTU RGB+D. NTU RGB+D [24] is a widely used large-scale human skeleton-based action recognition dataset, which contains 56,880 skeletal action sequences. These action sequences were performed by 40 volunteers and divided into 60 categories. Each action sequence is completed by one or two subjects and is captured by three Microsoft Kinect-V2 cameras from different views simultaneously. The benchmark evaluations include Cross-Subject (X-Sub) and Cross-View (X-View). In the Cross-Subject, training data comes from 20 subjects, and testing data comes from the other 20 subjects. In the Cross-View, training data comes from camera views 2 and 3, and testing data comes from camera view 1. Note that there are 302 wrong samples that need to be ignored.

NTU RGB+D 120. NTU RGB+D 120 [18] is currently the largest human skeleton-based action recognition dataset. It is an extension of the NTU RGB+D dataset, with 113,945 action sequences and 120 action classes in total. These action sequences were performed by 106 volunteers, captured with three cameras views, and contains 32 setups, each of which represents a different location and background. The benchmark evaluations include Cross-Subject (X-Sub) and Cross-Setup (X-Setup). In the Cross-Subject, training data comes from 53 subjects, and testing data comes from the other 53 subjects. In the Cross-Setup, training data comes from samples with even setup IDs, and testing data comes from samples with odd setup IDs. In this dataset, 532 bad samples should be ignored.

4.2. Implementation Details

All experiments are conducted on 8 GTX 1080Ti GPUs. Our model is trained using SGD optimizer with momentum 0.9 and weight decay 0.0002. The training epoch is set to 300. Learning rate is set to 0.1 and decays with a cosine scheduler. The batch size is 64 and each sample contains 32 frames. The number of Spatial-Temporal Transformer layer is set to 4.

4.3. Ablation Study

In this section, we investigate the effectiveness of the proposed components of the IIP-Transformer. All experiments are conducted on NTU RGB+D 60 with joint stream if no special instruction.

Effect of IIPA. The comparison between IIPA and standard Self-Attention in Table 1 shows that there are 1.8%

Methods	X-Sub	X-View
Standard Self-Attention	85.3	91.2
IIPA (ours)	**87.1**	**93.2**

Table 1. Comparison between IIPA and standard Self-Attention.

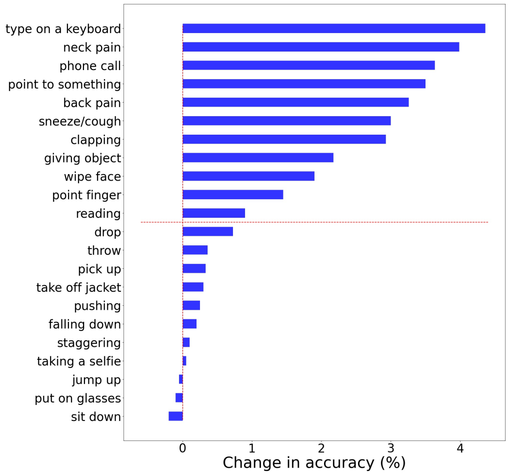

Figure 7. The accuracy comparison on actions with different motion intensity. The categories above the red horizontal line are fine-grained action categories.

and 2% improvements in X-Sub and X-View respectively. We pick out some actions and visualize the changes of accuracy between IIPA and standard Self-Attention in Figure Figure 7. It demonstrates that the IIPA achieves remarkable improvements on fine-grained actions (e.g., type on a keyboard, clapping, reading etc.), while the accuracy of drastic actions (e.g., throw, pushing, pickup etc.) is about the same, which indicates that our proposed IIPA can exploit the joint-level information of body parts more effectively.

Effect of Class-Token. To explore the effect of class-token, we replace it by simply performing global average pooling to the output features from last layer [22, 27]. As shown in Table 2, by introducing the class-token, the performance of X-Sub and X-View improve by 1.3% and 1.7% respectively.

Methods	X-Sub	X-View
Non-CLS	85.8	91.5
CLS (ours)	**87.1**	**93.2**

Table 2. Ablation study on class-token. Non-CLS denotes the previous global average pooling method.

Methods	Frames	X-Sub	FLOPs
Non-PE	32	84.1	44.6G
	64	85.2	97.2G
	128	86.0	198.7G
PE	32	**87.1**	**7.2G**
	64	86.9	19.8G
	128	87.0	45.5G

Table 3. Ablation study of the Partition Encoding.

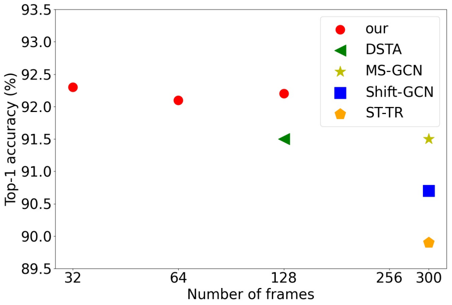

Figure 8. Ablation study over different methods with different input frames on NTU RGB+D 60 (X-Sub). The results are obtained after multi-stream fusion which will be introduced in Section 4.4.

Methods	X-Sub	X-View
Origin	87.1	93.2
Gaussian-Noise	87.1	93.1
Gaussian-Blur	86.5	92.8
Joint-Mask	87.2	92.9
Rotation	87.9	93.9
Part-Mask	88.4	94.0
Rotation+Part-Mask	**88.9**	**94.2**

Table 4. Ablation study of data augmentation on NTU RGB+D 60.

Effect of Partition Encoding. To explore the effect of Partition Encoding, we remove the Partition Encoding module from the proposed pipeline, which increases the number of tokens by 5 times (from 5 parts to 25 joints) and conduct experiments with different frame numbers. Comparing with Non-Partition Encoding (Non-PE), Partition Encoding (PE) improves accuracy by 1.1% with much less computational cost, as shown in Table 3. Besides, PE enables the model to take sparser frames as temporal inputs, while the accuracy of Non-PE drops by 1.9% when reducing the number of frames from 128 to 32, as shown in Table 3. In addition, we compare the accuracy of methods [7,21,22,27] with different number of input frames, as shown in Figure 8. Our IIP-Transformer achieves comparable results with only 32 frames.

Effect of Data Augmentation. To evaluate the impact of the proposed data augmentation strategies, we conduct experiments with different data augmentation (see Table 4). Applying part-level data augmentation strategies(*e.g.*, Rotation and Part-Mask) on X-Sub improves the accuracy

Methods	**X-Sub**	**X-View**	**X-Sub 120**	**X-Set 120**	**Param**	**FLOPs**
ST-LSTM [19]	69.2	77.7	55.7	57.9	-	-
HCN [17]	86.5	91.1	-	-	-	-
ST-GCN [36]	81.5	88.3	-	-	3.1M	16.3G
2s-AGCN [26]	88.5	95.1	82.9	84.9	6.9M	37.3G
AGC-LSTM [30]	89.2	95.0	-	-	22.9M	-
PL-GCN [13]	89.2	95.0	-	-	20.7M	-
DGNN [25]	89.9	96.1	-	-	26.2M	-
Shift-GCN [7]	90.7	96.5	85.9	87.6	2.8M	10.0G
DC-GCN+ADG [6]	90.8	96.6	86.5	88.1	4.9M	25.7G
PA-ResGCN-B19 [31]	90.9	96.0	87.3	88.3	3.6M	18.5G
Dynamic GCN [37]	91.5	96.0	87.3	88.6	14.4M	-
MS-G3D [21]	91.5	96.2	86.9	88.4	2.8M	48.8G
MST-GCN [5]	91.5	**96.6**	87.5	88.8	12.0M	-
EfficientGCN-B4 [32]	91.7	95.7	88.3	89.1	**2.0M**	15.2G
ST-TR [22]	89.9	96.1	82.7	84.7	12.1M	259.4G
DSTA [27]	91.5	96.4	86.6	89.0	4.1M	64.7G
IIP-Transformer	**92.3**	96.4	**88.4**	**89.7**	2.9M	**7.2G**

Table 5. Comparison of top-1 accuracy (%), model size and computational complexity over different methods on the NTU RGB+D 60/120 datasets.

by 0.8% and 1.3% respectively, and the best results are achieved when combining these two strategies. While the effect of joint-level data augmentation (*e.g.*, Gaussian-Noise, Gaussian-Blur and Joint-Mask) is marginal. Intuitively, the reason that Part-Mask strategy can effectively improve the accuracy of the model is that it encourages the model to reason globally and reduce the dependency on any particular part.

4.4. Comparison with State of the Arts

Similar to most SOTA methods, we follow the same multi-stream fusion strategies proposed in [7] for fair comparison. We train four models with different modalities, *e.g.*, joint, bone, joint motion, and bone motion respectively, then average the *softmax* outputs from multiple streams to obtain the final scores during inference. The comparison results are shown in Table 5.

Accuracy Comparison. First we compare our results with GCN-based methods [6,7,21,25,26,31,32,36,37]. Our proposed method outperforms the best GCN method by 0.6% on X-sub of NTU RGB+D 60 and 0.6% on X-Set of NTU RGB+D 120 respectively. In terms of Transformer-based methods [22, 27], our method also surpasses them on most of the metrics by a significant margin.

Complexity Comparison. We evaluate the computational complexity with FLOPs, *e.g*. the number of floating-point multiplication-adds, and measure the model size with the amount of parameters. The computational complexity of our proposed IIP-Transformer is 2.2 times less than ST-GCN [36] and 6.7 times less than MS-G3D [21], while the model sizes are similar. Comparing with the Transformer-based methods, IIP-Transformer achieves superior results with only a fraction of computational complexity and model size.

5. Conclusion

In this work, we propose a novel intra-inter-part transformer network (IIP-Transformer) for skeleton-based action recognition. It effectively captures inter-part and intra-part dependencies. Thanks to the part-level encoding and spatial-temporal separation, our method enjoys high efficiency. Besides, a part-level data augmentation named *Part-Mask* is proposed to encourage the model focus on global parts. On two large scale datasets, NTU RGB+D 60 & 120, the proposed IIP-Transformer notably exceeds the current state-of-the-art methods with $2 \sim 36\times$ less computational cost.

References

[1] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3285–3294, 2019. 2

[2] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(1):172–186, 2018. 1

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi,

Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020. 2

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, July 2017. 1

[5] Zhan Chen, Sicheng Li, Bing Yang, Qinghan Li, and Hong Liu. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1113–1122, 2021. 8

[6] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *European Conference on Computer Vision (ECCV)*, pages 536–553. Springer, 2020. 8

[7] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 180–189, June 2020. 2, 7, 8

[8] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3218–3226, 2015. 2

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert:pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies(NAACL-HLT)*, pages 4171–4186, 2019. 2, 4

[10] Zewei Ding, Pichao Wang, Philip O Ogunbona, and Wanqing Li. Investigation of different skeleton features for cnn-based 3d action recognition. In *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 617–622. IEEE, 2017. 1

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2020. 2

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6201–6210. IEEE, 2019. 1

[13] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. Part-level graph convolutional network for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11045–11052, 2020. 2, 8

[14] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14, 1973. 1

[15] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik. Intel(r) realsense(tm) stereoscopic depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1267–1276, 2017. 1

[16] Guy Lev, Gil Sadeh, Benjamin Klein, and Lior Wolf. Rnn fisher vectors for action recognition and image annotation. In *European Conference on Computer Vision (ECCV)*, pages 833–850. Springer, 2016. 2

[17] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. pages 786–792, 2018. 1, 2, 8

[18] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(10):2684–2701, 2020. 6

[19] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision (ECCV)*, pages 816–833. Springer, 2016. 8

[20] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017. 1

[21] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 143–152, 2020. 7, 8

[22] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Skeleton-based action recognition via spatial and temporal transformer networks. *Computer Vision and Image Understanding*, 208:103219, 2021. 1, 2, 4, 7, 8

[23] Haocong Rao, Shihao Xu, Xiping Hu, Jun Cheng, and Bin Hu. Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. *Information Sciences*, 569:90–109, 2021. 6

[24] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, June 2016. 6

[25] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7912–7921, 2019. 4, 8

[26] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12026–12035, 2019. 2, 8

[27] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In *Asian Conference on Computer Vision (ACCV)*, volume 12626, pages 38–53, 2020. 1, 2, 5, 7, 8

[28] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Action recognition via pose-based graph convolutional networks

with intermediate dense supervision. *Pattern Recognition*, page 108170, 2021. 1

[29] L. Shi, Y. Zhang, J. Hu, J. Cheng, and H. Lu. Gesture recognition using spatiotemporal deformable convolutional representation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1900–1904, 2019. 1

[30] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1227–1236, 2019. 8

[31] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1625–1633, 2020. 3, 8

[32] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Constructing stronger and faster baselines for skeleton-based action recognition. *arXiv preprint arXiv:2106.15125*, 2021. 8

[33] Kalpit C. Thakkar and P. J. Narayanan. Part-based graph convolutional network for action recognition. In *British Machine Vision Conference (BMVC)*, page 270, 2018. 2

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 2

[35] Hongsong Wang and Liang Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 499–508, 2017. 2, 6

[36] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7444–7452. AAAI Press, 2018. 1, 2, 8

[37] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang. Dynamic GCN: context-enriched topology learning for skeleton-based action recognition. *CoRR*, abs/2007.14690, 2020. 8

[38] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, 2012. 1