STST: Spatial-Temporal Specialized Transformer for Skeleton-based Action Recognition

Yuhan Zhang*
University of Electronic Science and Technology of China
yuhanzhan9@gmail.com

Bo Wu*
MIT-IBM Watson AI Lab
bo.wu@ibm.com

Wen Li†
University of Electronic Science and Technology of China
liwenbnu@gmail.com

Lixin Duan
University of Electronic Science and Technology of China
lxduan@uestc.edu.cn

Chuang Gan
MIT-IBM Watson AI Lab
ganchuang1990@gmail.com

ABSTRACT

Skeleton-based action recognition has been widely investigated considering their strong adaptability to dynamic circumstances and complicated backgrounds. To recognize different actions from skeleton sequences, it is essential and crucial to model the posture of the human represented by the skeleton and its changes in the temporal dimension. However, most of the existing works treat skeleton sequences in the temporal and spatial dimension in the same way, ignoring the difference between the temporal and spatial dimension in skeleton data which is not an optimal way to model skeleton sequences. The posture represented by the skeleton in each frame is proposed to be modeled individually. Meanwhile, capturing the movement of the entire skeleton in the temporal dimension is needed. So, we designed Spatial Transformer Block and Directional Temporal Transformer Block for modeling skeleton sequences in spatial and temporal dimensions respectively. Due to occlusion/sensor/raw video, etc., there are noises on both temporal and spatial dimensions in the extracted skeleton data reducing the recognition capabilities of models. To adapt to this imperfect information condition, we propose a multi-task self-supervised learning method by providing confusing samples in different situations to improve the robustness of our model. Combining the above design, we propose our Spatial-Temporal Specialized Transformer (STST) and conduct experiments with our model on the SHREC, NTU-RGB+D, and Kinetics-Skeleton. Extensive experimental results demonstrate the improved performances and analysis of the proposed method.

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition and understanding**.

KEYWORDS

Skeleton; Spatial-Temporal; Specialized; Self-supervision; Transformer; Action Recognition

ACM Reference Format:
Yuhan Zhang, Bo Wu, Wen Li, Lixin Duan, and Chuang Gan. 2021. STST: Spatial-Temporal Specialized Transformer for Skeleton-based Action Recognition. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475473

1 INTRODUCTION

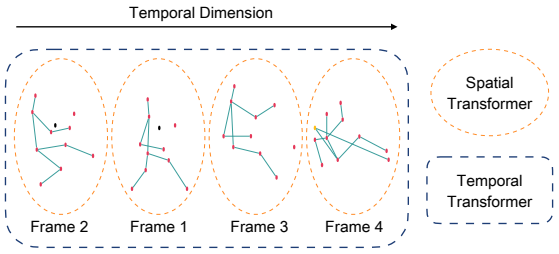

Figure 1: In the real scene, there are some abnormalities in the skeleton data (take the action of falling down as an example) including temporal dislocation (Frame 1 and 2), unknown type of joints (black dots), the missing or shifted (yellow dots) coordinates of joints, etc.

Action recognition plays a vital role in video understanding and is broadly applicable to video surveillance, human-machine interaction, virtual reality, etc. Great progress has been made in this area based on various data representations [13, 16, 28, 33, 45, 48], such as visual appearance, skeleton, depth, optical flows etc. Among those, the skeleton-based action recognition has drawn lots of interests from researchers due to its strong adaptability to dynamic circumstances and complicated backgrounds. Skeleton-based action recognition adopts the human joints coordinates in each frame to represent human actions in videos. Human skeleton, as a tree-structured representation, describes human postures and joints movement dynamically. Therefore, it is more robust in situations with visually noised environments or backgrounds compared to other methods.

*Both authors contributed equally to this research.
†Corresponding author

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475473

Existing skeleton-based methods model human postures by transforming the skeleton into a meaningful form such as a point sequence [11, 30], a pseudo-image [21, 24] or a graph [39, 47] depending on manual designs of traversal rules or graph topology which are not always optimal. Other conventional models utilize the RNNs and CNNs to capture the skeleton movements but always are intractable for long time-span of videos. Recently, with the success of Transformer, it is also applied in skeleton-based models [1, 36, 40]. Transformer is suitable for processing skeleton as sequence data, since it can model the relationships among the intra-frame joints without the limits on the joint numbers as in the manually pre-defined structure. However, most of the Transformer based methods organize and process joints with the same strategy in both temporal and spatial dimension, without analyzing the difference between the temporal and spatial dimension in skeleton sequence data.

To tackle those challenges, we explore two primary aspects to design a better skeleton-based action recognition model. Firstly, it is necessary to model actions from spatial-temporal views separately and design specific module for each view, because two views are not sharing the same mechanism. It requires the model is able to express posture in each frame from spatial view and capture the pattern of movements of posture from temporal view. Secondly, the model should be able to handle the abnormal cases since the skeleton structure is not always stable or robust. Although the skeleton-based methods are capable to learn actions within dynamic circumstances and complicated backgrounds, the skeleton data may be disturbed by occlusion/ sensor/ raw video/ posture estimation algorithms as the Figure 1, and these abnormal cases may lead to unexpected responses of models which are not robust enough.

Under the above purposes, we propose a new model to capture the information of the skeleton sequence respectively in the temporal dimension and the spatial dimension while being robust to various abnormal scenes, which is called Spatial-Temporal Specialized Transformer (STST). For joints in the skeleton sequence data, each of them contains three kinds of information: 1) coordinate information; 2) semantic information; 3) temporal information. A good skeleton-based model should fully capture this information while being robust to the noise on these three types of information. Therefore, firstly, we propose a novel Transformer encoder to model the skeleton sequence where the spatial and temporal operations are designed separately and specially as the Figure 1 shows. To let the encoder to make full use of the skeleton data without losing information, we explicitly extract three kinds of information of joints mentioned above into three kinds of tokens. In the encoder, we design the Spatial Transformer Block to model the posture represented by the skeleton in each frame separately, and the Directional Temporal Transformer Block to model the action based on the movements of the entire skeleton in the temporal dimension with direction-aware strategy. Secondly, all the impact of imperfect information conditions can be regarded as the noises on these three kinds of information mentioned above. Inspired by [9], we proposed a multi-task self-supervised learning method to enhance the robustness of our model to these three kinds of noises for the skeleton data. All these self-supervised learning tasks are designed to be highly parallel and easy to implement, which fit very well with the characteristics of Transformer.

To demonstrate the superiority of our proposed model, we conduct experiments on three widely used datasets: SHREC [8], NTU-RGB+D [37] and Kinetics-Skeleton [20]. Our model achieves state-of-the-art performance on these datasets.

In summary, our contributions are as follows:

- We propose the Spatial-Temporal Specialized Transformer, which captures the movements of skeleton effectively by adopting different joints organization strategies to model the skeleton sequence in spatial and temporal dimensions.
- We further propose a multi-task self-supervised learning method to enhance the robustness of the model to the three imperfect information situations.
- On three widely used datasets for skeleton-based action recognition, our STST outperforms previous methods, even without using the self-supervision strategy.

2 RELATED WORK

2.1 Skeleton-based Action Recognition

Skeleton-based action recognition has been a popular topic in the past years. Existing approaches [19, 44] treat human joints as a set of independent instances, using hand-crafting features to represent the relative 3D rotations and translations between joints in the temporal and spatial dimension. With the rapid progress of deep learning research, several models [11, 30, 42] use RNNs by treating skeleton in an form of frame where each dynamic skeleton sequence indicates a joint changes over time. Some models [6, 10, 21, 24, 31] use CNNs to extract features from 2D pseudo-image which represents temporal dynamics and skeleton joints respectively in rows and columns. Several new approaches [25, 27, 47] achieve a significant increase in performance by constructing graphs based on the artificially defined topological structure of joints and adopt GNNs. However, these methods require the artificial design of the connection characteristics between the joints, which is not always optimal.

As the self-attention mechanism is able to learn relationships among joints, researchers use Transformer to replace the hand-crafted adjacency recently. Some models [36, 40] extended the Transformer to the skeleton-based action recognition task. However, they do not design distinct mechanism according to spatial and temporal characteristics of the skeleton sequence. ST-Transformer [1] proposed an independent transformer for each joint point, but the size of parameters in the model would increase with the number of types of joints in the skeleton data, which makes the model complicated and inflexible. In this paper, we explicitly define three types of encoding strategies to consider main situations of joints and design specific Transformers for the temporal and spatial dimensions.

2.2 Self-supervised Learning

The original intention of self-supervised learning is to learn feature representations from a large amount of unlabeled data. It has been verified that self-supervised pre-training can help supervised learning [12] and it has a variety of applications in video understanding [15, 17].

Recently, for the sequential data like videos, some models [14, 23, 46] learn the temporal patterns by predicting the sequential order of sampled frames or clips. In [34], the video cloze procedure (VCP)

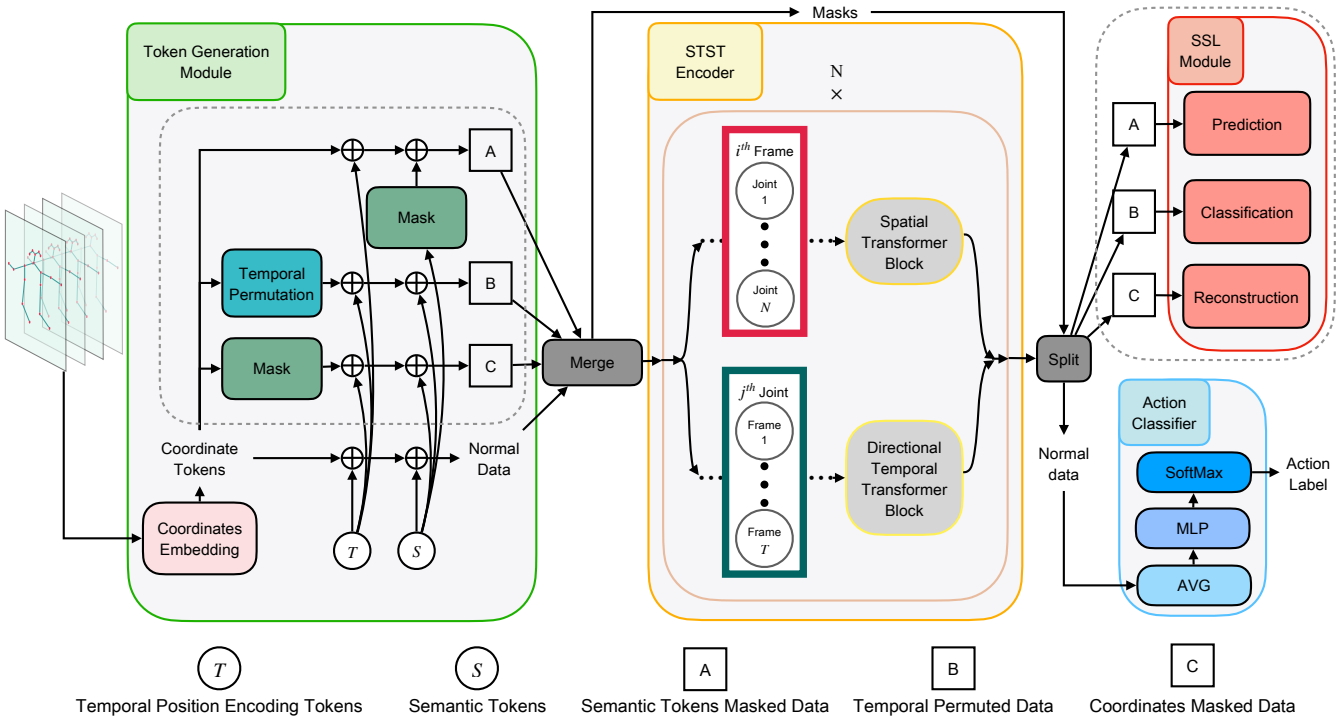

Figure 2: Token Generation Module generates three kinds of tokens for skeleton sequences. The STST-Encoder captures the pattern of skeleton movements. The SSL Module contains three self-supervised learning tasks for boosting the STST-Encoder. The Action Classifier receives the skeleton sequence representation from the STST-Encoder and predicts the action label. The modules and data in the gray dashed line are designed for self-supervision and only involved in model training.

was proposed to learn the spatial-temporal representation of video data based on a method that uses spatial rotations and temporal shuffling method, which enhanced the accuracy in action recognition. In [7], the models learn the video representation via a variable video playback speed prediction task.

For skeleton-based tasks, MS^{2}L [29] proposed self-supervised learning tasks. However the self-supervised learning tasks proposed in MS^{2}L [29] do not cover the three imperfect information situations we mentioned above. Here, according to the characteristics of the skeleton sequence data, we design three self-supervised learning tasks corresponding to the three imperfect information mentioned above, and these tasks are suitable for our powerful encoder.

3 METHOD

3.1 Problem Formulation

Let us denote a skeleton sequence as the X_{seq}. Suppose each X_{seq} consists T frames of skeleton with N joints in each frame, and $X_{seq} \in \mathbb{R}^{T\times S\times C}$ could be expressed as:

$$X_{seq} = \{X_1, X_2, \cdots, X_t, \cdots, X_T\} \tag{1}$$

where $X_t = \{J_t^1, J_t^2, \cdots, J_t^S\} \in \mathbb{R}^{S\times C}$ indicates the skeleton joints of the t^{th} frame in a specific order. Each J is a vector representing the 2D or 3D coordinates of the corresponding joint. From another point of view, the X_{seq} could be expressed as:

$$X_{seq} = \{X^1, X^2, \cdots, X^s, \cdots, X^S\} \tag{2}$$

where $X^s = \{J_1^s, J_2^s, \cdots, J_T^s\} \in \mathbb{R}^{T\times C}$ represents the sequence composed of the s^{th} joint in all frames. Given the skeleton sequence X_{seq}, and our task is to recognize actions of the sequence.

3.2 Framework Overview

We propose a Spatial-Temporal Specialized Transformer Encoder to model the skeleton posture of each frame and capture changes of posture in the temporal dimension. To boost the robustness of our model to noises, we design a multi-task self-supervision module specifically for the skeleton modal. We propose the Spatial-temporal Specialized Transformer (STST) by combining these two into a unified model. As shown in Figure 2, our model includes four functional modules: Token Generation Module, Spatial-Temporal Specialized Transformer Encoder (STST-Encoder), the multi-task self-supervised learning module (SSL Module), and Action Classifier. The Token Generation Module generates tokens to represent joints for the original inputs and transformed samples for self-supervised learning. The STST-Encoder captures the movement information of the skeleton posture. The SSL Module applies self-supervised learning to boost the robustness of our model. And the Action Classifier receives the representation of samples from the STST-Encoder, then predicts the label of action. In this section, we first introduce the

architecture of our Spatial-Temporal Specialized Transformer Encoder in Section 3.4. Then the details of our self-supervised learning tasks are described in Section 3.5.

3.3 Skeleton Token Sequence Generation

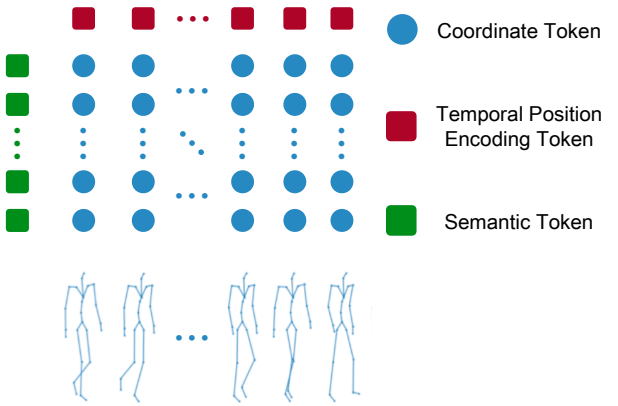

Figure 3: Simulation of the information contained in the skeleton sequence. All joints in the same frame share the same temporal position encoding token. All joints with the same type share the same semantic token.

From our perspectives, each joint in the skeleton sequence contains three kinds of information: 1) coordinate information; 2) semantic information; 3) temporal information. Coordinates information means the coordinates of the joint, which is given by the input data explicitly. Semantic information means the joint (is this joint representing head or other parts of the human body). Moreover, Semantic information is represented by its position in the data implicitly. The temporal information indicates which frame of the skeleton sequence the joint is, which is also is represented by its position in the data implicitly. As the permutation invariance of the self-attention mechanism, location-specific information needs to be used explicitly with the reference to [9]. Therefore, for the modal data of the skeleton sequence, we designed semantic tokens and temporal position encoding tokens to improve the recognition ability of the model without losing any information of the skeleton sequence. We firstly use Multi-Layer Perception (MLP) to map coordinates to high-dimension for the coordinates of joints. Then for the spatial dimension, our model must be able to distinguish the types of joints. So we add the semantic token to each joint:

$$X_t = X_t + P_S \tag{3}$$

where $P_S \in \mathbb{R}^{S\times C}$ is the matrix representing S semantic tokens shared in all frames which means all joints with the same type share the same semantic token. And the P_S is trained jointly with the whole model.

For the temporal dimension, following [43], we use the sine and cosine functions with different frequencies as the encoding functions which :

$$\begin{aligned} P_T(p, 2i) &= \sin(p/10000^{2i/C}) \\ P_T(p, 2i+1) &= \cos(p/10000^{2i/C}) \end{aligned} \tag{4}$$

where p denotes the position of element and i denotes the dimension of the position encoding vector. Then the position encoding is added to the input as:

$$X^s = X^s + P_T \tag{5}$$

where X^s means the sequence composed of the same type of joints in all frames. All kinds of joints share the same P_T, which means all joints in the same frame share the same temporal position encoding.

As shown in Figure 3, we add the semantic tokens and temporal position encoding tokens to coordinate tokens extracted from the input data to form a complete representation of each joint in each frame.

3.4 STST-Encoder

The Spatial-Temporal Specialized Transformer is composed of two Transformer Blocks: the Spatial Transformer Block (STB) and the Directional Temporal Transformer Block (DTTB). The STB is designed to model the posture represented by the skeleton in each frame separately. The DTTB is designed to capture the pattern of movements of human posture. In this section, we first introduce the structure of the basic Transformer Block, and then respectively introduce the modification of STB and DTTB on the basic Transformer.

3.4.1 Basic Transformer Block. Transformer has achieved enormous success in several fields due to utilizing a self-attention mechanism to capture the relationship between all elements. Assuming the matrix $X \in \mathbb{R}^{N\times C}$ represents all elements where N is the number of elements, and C is the channel number of each vector representing each item. An attention operation can be divided into two main steps: (1) getting an attention map, (2) giving new representation to all elements based on the attention map. Attention map represents the correlations between all elements and is obtained by taking the dot product respectively liner transformed input as:

$$\begin{aligned} A &= Att(X) \\ &= \text{SoftMax}(\frac{\phi(X) \times \varphi(X)^\top}{\sqrt{C_{hidden}}}) \end{aligned} \tag{6}$$

where ϕ and φ are two different trainable linear transformations and they share the same output channel C_{hidden} and the X is the matrix representing all elements. The item A_{ij} in A represents the correlations score between the element i and element j. Then on the basis of attention map A, the hidden representation of all elements is produced as in the following:

$$H = \text{LayerNorm}(\psi(AX) + X) \tag{7}$$

where ψ is a linear transformation and LayerNorm is layer normalization [2]. As described in Eq. 7, shortcut connecting is applied to improve the stability of the model. Originally introduced in [43], the multi-head strategy is utilized in the self-attention operation as applying several respective self-attention operations and concatenating all the outputs as the multi-head self-attention operation output. Given the input X, we define the process of updating X in the Transformer Block (TB) as follows:

$$X = \text{LayerNorm}(H + \text{FF}(H)) \tag{8}$$

where FF is any row-wise feedforward layer (i.e., it processes feature of each joint independently and identically). For the convenience of description, we merge Eq. 7 and Eq. 8 into function F one as:

$$X = F(X, A) \tag{9}$$

and merge the Eq. 6 and the Eq. 9 to express the entire Transformer Block as:

$$TB(X) = F(X, Att(X)) \tag{10}$$

3.4.2 Spatial Transformer Block. The Spatial Transformer Block (STB) is a specialized Transformer Block to model the skeleton posture of each frame by computing the relationship between joints intra-frame. Let's take the t^{th} frame as an example, the skeleton in t^{th} frame can be represented by matrix: $X^t \in \mathbb{R}^{S\times C}$, where $1 \le t \le T$. The S denotes the number of joints, and C denotes the number of channels, T is the number of frames. According to the Eq. 6 we can get the attention map of t^{th} frame as $A_t \in \mathbb{R}^{S\times S}$. For the whole skeleton sequence, we get T attention maps. Each frame is processed by remaining part of Transformer Block independently with their own attention maps in parallel. This process can be formulated as:

$$STB(X_{seq}) = \{TB(X_1), \cdots, TB(X_t), \cdots, TB(X_T)\} \tag{11}$$

where all TBs share the same parameters.

3.4.3 Directional Temporal Transformer Block. The Temporal Transformer Block (TTB) is a specialized Transformer Block to capture posture movements over long periods, which can model the entire sequence. As the primary purpose of our Directional Temporal Transformer Block (DTTB) is to capture the relative change of the posture in the temporal dimension, our model must have the ability to perceive the relative positional relationship between frames.

Differing from the Spatial Transformer Block, we need to align the responses of different joints in the temporal dimension instead of treating them separately. So we generate a global attention map for the whole skeleton sequence rather than an exclusive attention map for each joint. In practice, We merge the features of all joints in the same frame as the representation of the posture of the skeleton frame as $\hat{X}_{seq} \in \mathbb{R}^{T\times C}$. Then the self-attention operation is implemented to generate the corresponding attention map representing the posture in the temporal dimension for the whole skeleton sequence as:

$$\hat{A}_{seq} = Att(\hat{X}_{seq}) \tag{12}$$

where $\hat{A}_{seq} \in \mathbb{R}^{S\times S}$. Then the $\hat{A}_{seq}$ is shared in the processing of updating all kinds of joints in the X_{seq} as:

$$TTB(X_{seq}) = \{F(X^1, \hat{A}_{seq}), \cdots, F(X^s, \hat{A}_{seq}), \cdots, F(X^S, \hat{A}_{seq})\} \tag{13}$$

However, due to the periodic symmetry of trigonometric functions, the self-attention operation can hardly distinguish the order of sequence. So we design our Temporal Transformer Block to be direction-aware by applying a directional mask strategy to force the model to recognize the order. We replace one self-attention operation with one forward self-attention operation and one backward self-attention operation. For the mask in forward self-attention operation, the values of items representing the relation of joints between their earlier joints in the temporal dimension equal to zeros, otherwise equal to $-\infty$ (negative infinity). The mask in the backward self-attention operation is the transpose of the mask in the forward self-attention operation. So the Eq. 6 in Directional Temporal Transformer Block transfers to:

$$\begin{aligned} A^{f/b} &= Att^{f/b}(X) \\ &= \text{SoftMax}(M^{f/b} + \frac{(\phi(X) \times \varphi(X)^\top}{\sqrt{C_{hidden}}}) \end{aligned} \tag{14}$$

where f/b means forward or backward, and $M_{f/b}$ means the mask for forward or backward directional self-attention operation. With the direction-aware strategy, the Directional Temporal Transformer Block can be expressed as:

$$\begin{aligned} DTTB(X_{seq}) = &\{F(X^1, \hat{A}^f_{seq}), \cdots, F(X^S, \hat{A}^f_{seq})\} \\ &+ \{F(X^1, \hat{A}^b_{seq}), \cdots, F(X^S, \hat{A}^b_{seq})\} \end{aligned} \tag{15}$$

3.5 Multi-task Self-Supervision

Self-supervised learning aims to learn feature representations from a massive amount of unlabeled data. Inspired by [29], we exploit self-supervised learning to help our model alleviate the harm caused by the noise on the three kinds of information mentioned before of the skeleton sequence data due to various imperfect information situations. We design three self-supervised learning tasks corresponding to the noise on these three kinds of information to deal with different exception situations in skeleton-based action recognition. We now describe our self-supervised learning tasks for Coordinates Reconstruction Task, Temporal Permutation Task, and Semantic Token Prediction Task. To simplify the description, we define our Spatial-Temporal Specialized Transformer Encoder as $f(\cdot)$ in the following parts.

3.5.1 Semantic Token Prediction. As we mentioned in Section 3.3, the semantic tokens of joints are essential for the model to distinguish the types of joints. Meanwhile, a superior skeleton-based model should have the ability to infer the types of joints through its motion trajectory and relative positional relationship with other joints. Here we replace the semantic tokens of some percentage of joints in all frames with $[MASK_S]$ token as X_{MaskS}. The encoder $f(\cdot)$ extract type information of joints to representations by learning relative positional relationship between joints and movement characteristics of the joints. Semantic Token Prediction head $h_S(\cdot)$ receives the representations of joints and predicts the types of joints. The task is trained with the loss L_{PS}, which is formulated as cross-entropy loss for classification as follows:

$$L_{PS} = \sum_{i=1}^{\#MS} -y^i_S \log h_S(f(X^i_{MaskS})) \tag{16}$$

where the $\#MS$ is the number of masked semantic tokens and y^i_S is the correct type of the masked joint. In practice, we do not always replace "masked" token with the actual $[MASK_S]$ token. If the s^th joint is selected, we replace the its semantic token with (1) the $[MASK_S]$ token 80% of the time (2) a random semantic token 10% of the time (3) the unchanged token 10% of the time. We call this Semantic Token Prediction task PS for short.

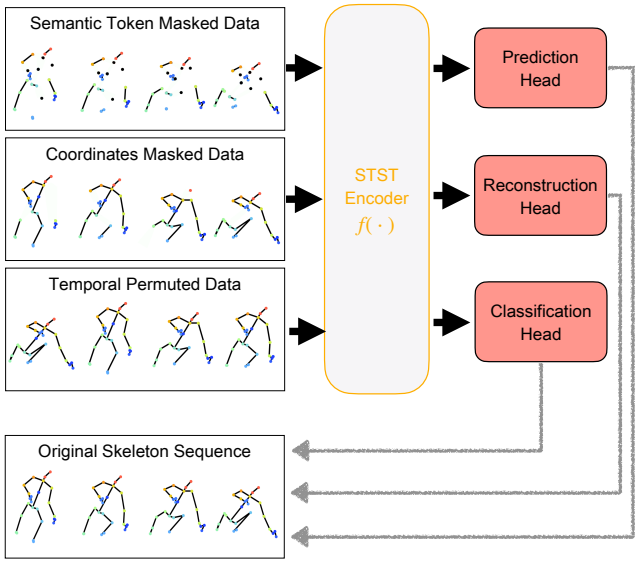

Figure 4: Three kinds of data for self-supervised learning tasks: 1) in the Semantic Token Masked Data, these black dot represent the joints with their semantic token masked; 2) in the Coordinates Masked Data, the missing joints represent the joints with their coordinate token masked; 3) the permuted skeleton sequence.

3.5.2 Masked Coordinates Reconstruction. Intuitively, a skeleton-based model is supposed to perceive the law of posture changes. So it is a basic but necessary ability for a skeleton-based model to predict the coordinates of a joint in a specific frame through the relative relationship with other joints and the state of this joint in other frames. According to a certain proportion, we randomly set the original coordinates of some joints in some frames to zeros and keep the rest part unchanged. Furthermore, we define the data with randomly masked coordinates as zeros. The encoder $f(\cdot)$ reads the input sequences and extracts representations from inputs. Our reconstruction head $h_C(\cdot)$ receives the learned representations from the encoder and reconstructs the coordinates of joints in the input sequences. We use mean square error (MSE) to estimate the parameters of our encoder as follows:

$$L_{PC} = \sum^{\#MC} ||h_C(f(X_{MaskC})) - X||_2^2 \tag{17}$$

where the $\#MC$ is the number of masked coordinates tokens. Actually, we do not set "masked" coordinates token all zeros. If the $s^t h$ joint in $t^t h$ frame is selected, we set its coordinates to (1) the zeros 80% of the time (2) coordinates token Sampled from a normal distribution with its original coordinates as the mean and variance of 1 10% of the time (3) the unchanged coordinates 10% of the time. And we name the Masked Coordinates Reconstruction task as PC.

3.5.3 Temporal Permutation. To enhance the ability of our model on learning temporal patterns, we apply Temporal Order Permutation for skeleton sequences by predicting the correct order from the shuffled sequences X_{PT}. Each sequence is divided into K_T segments equally, and there is $\frac{T}{K_T}$ frames in a segment. So this is $K_T!$ ways to shuffle them, and our classification head $h_T \cdot$ is to predict which way is used to shuffle each sequence. We randomly choose some samples and permute them. We also use cross-entropy loss to formulate our Temporal Order Permutation loss L_PT as:

$$L_{PT} = \sum_{i=1}^{\#TP} -y_T^i \log h_T(f(X_{seq_i})) \tag{18}$$

where the $\#TP$ means the number of the temporal permuted samples and y_T^i means the way of how skeleton sequence permuted. We use PT to refer to the Temporal Permutation task.

3.6 Training and Inference

All these three self-supervised learning tasks are trained jointly with the main action recognition task. Inspired by [41], we also apply the multi-stream strategy to predict the model more accurately and stably. All these self-supervised learning tasks are not required in the processing of inference, as the Figure 2 shows, so it will not affect the speed of the model.

4 EXPERIMENTS

4.1 Dataset

For evaluation, we conduct our experiments on the following three datasets:

SHREC [8] contains 14 hand gestures with 2,800 gesture sequences performed between 1 and 10 times by 28 participants in two ways. It splits the sequences into 1,960 train sequences and 840 test sequences. The length of sample gestures ranges from 20 to 50 frames. Two benchmarks (SHREC-14 and SHREC-28) with 14 and 28 label classes are constructed on this dataset depending on the gesture represented and the number of fingers used.

NTU-RGB+D [37] is a widely used in-door-captured action recognition dataset, which contains 56,000 action clips in 60 action classes. The clips are performed by 40 volunteers captured in a constrained lab environment by 3 KinectV2 cameras with different views. This dataset provides 25 joints with their 3D locations for each subject in the skeleton sequences. It recommends two benchmarks: cross-subject (X-Sub) and cross-view (X-View). X-Sub benchmark provides 40,320 and 16,560 clips for training and evaluation. In this setting, the training clips come from one subset of actors, and the models are evaluated on clips from the remaining actors. X-View provides benchmarks 37,920 and 18,960 clips. Training clips in this set come from the camera views 2 and 3, and the evaluation clips are all from the camera view 1.

Kinetics-Skeleton [20] is a large-scale human action dataset that contains 300,000 videos clips in 400 classes. The video clips are sourced from YouTube videos and transformed to 18 joints for each person by OpenPose [3] toolbox. Two peoples are selected for multi-person clips based on the average joint confidence. The dataset is divided into a training set (240,000 clips) and a validation set (20,000 clips). Following the work [47], we train the models on the training set and report the top-1 and top-5 accuracies on the validation set.

Table 1: Recognition accuracy comparison of our method and state-of-the-art methods on NTU-RGB+D dataset.

Method	NTU-RGB+D (%)	
	X-Sub	X-View
STA-LSTM [42]	73.4	81.2
ST-GCN [47]	81.5	88.3
HCN [26]	86.5	91.1
AS-GCN [27]	86.8	94.2
2S-AGCN [39]	88.5	95.1
Shift-GCN [5]	91.5	96.5
DGNN [38]	89.9	96.1
MS-G3D [32]	91.5	96.2
MS-AAGCN [41]	90.0	96.2
DSTA-Net [40]	91.5	96.4
ST-TR [36]	89.9	96.1
STST (ours)	**91.9**	**96.8**

4.2 Implementation Details

Our Spatial-Temporal Specialized Transformer Encoder is stacked using 8 Spatial-Temporal Specialized Transformer Blocks, which is composed of a Spatial Transformer Block (STB) and a Directional Temporal Transformer Block (DTTB) where the number of attention head is set to be 3. And the output channels of blocks are set to 64, 64, 128, 128, 256, 256, 256 and 256 referring to [40]. All skeleton sequences are sampled to 150 frames and then cropped to 128 frames. During the training process, the loss weights of all self-supervised tasks are set to 0.2. We train our model for a total of 120 epochs with batch size 32 and SGD as optimizer with Nesterov momentum 0.9. Weight decay is set to 0.0005. When training, the initial learning rate is 0.1 and is divided by 10 in 60 and 90 epochs.

4.3 Comparison with previous methods

We evaluate our model with state-of-the-art methods for skeleton-based action recognition on three widely used datasets with a total of six benchmarks. The other models participating in the comparison in this paragraph are all the complete models proposed in their papers. We report the performance of our model on the condition that the multi-stream strategy mentioned in Section 3.6 all proposed self-supervised learning tasks are applied. Table 1 and Table 3 compare our STST with the non-graph methods [26, 42], graph-based methods [5, 32, 38, 39, 41]. Especially compared with the previous most powerful MS-G3D [32] that uses complex manually defined multi-scale Aggregation strategies and a variety of hyper-parameter settings, our STST achieves better performance with a simpler structure. Compared with the ST-TR [36] which uses the global self-attention mechanism in both spatial dimension and temporal dimension as an enhancement of the GNN-based model, our STST achieves higher performance in the case of only using Transformer. Compared with DSTA-Net [40] that organizes and processes joints with the same strategy in both temporal and spatial dimension, without analyzing the difference between the temporal and spatial dimension in skeleton sequence data, our STST achieves a great advantage. On all three datasets, our method outperforms all existing methods under all evaluation settings.

Table 2: Recognition accuracy comparison of our method and state-of-the-art methods on SHREC dataset.

Method	SHREC (%)	
	14 gestures	28 gestures
ST-GCN [47]	92.7	87.7
STA-Res-TCN [18]	93.6	90.7
ST-TS-HGR-NET [35]	94.3	89.4
DG-STA [4]	94.3	90.7
DSTA-Net [40]	97.0	93.9
STST (ours)	**97.6**	**95.3**

Table 3: Recognition accuracy comparison of our method and state-of-the-art methods on Kinetics-Skeleton dataset.

Method	Kinetics-Skeleton (%)	
	Top 1	Top 5
Deep LSTM [37]	16.4	35.3
TCN [22]	20.3	40.0
ST-GCN [47]	32.7	52.8
ST-GR [25]	33.6	56.1
AS-GCN [27]	34.8	58.5
2s-AGCN [39]	36.1	58.7
DGNN [38]	36.9	59.6
MS-G3D [32]	38.0	60.9
ST-TR [36]	37.4	59.8
STST (ours)	**38.3**	**61.2**

4.4 Ablation Study and Further Analysis

We analyze individual designs of our model. Unless stated, all performance is reported as the result of the model with only a single stream for the sake of fairness.

4.4.1 Effectiveness of STST-Encoder. To examine the effectiveness of our STST-Encoder, we verify the validity of the modules that make up our encoder: STB in Section 3.4.2 and TTB/DTTB in Section 3.4.3. We construct several experiments by replacing the STB with STBX which is built in the way as TTB, and replacing the TTB with TTBX which is built in the way as STB. Furthermore, all these results have experimented without any self-supervised learning tasks mentioned before.

By comparing the results of the first four experiments in Table 4, we find that on the basis of the proposed methods (STST-Encoder (STB, TTB)) of modeling spatial and temporal dimensions of skeleton sequences, replacing (STST-Encoder (STBX, TTB) and STST-Encoder (STB, TTBX)) and exchanging (STST-Encoder (STBX, TTBX)) the methods of modeling spatial and temporal dimensions will damage the performance.

Through the above four experiments, we can conclude that the skeleton sequence has different semantic characteristics in the temporal and spatial dimensions, and it is necessary to design correct

Table 4: Comparison of the accuracy of different modeling methods in temporal dimension and spatial dimension on NTU-RGN+D (X-Sub) benchmark.

Method	Spatial	Temporal	Acc (%)
ST-TR-1s[36]			86.16
DSTA-Net-1s [40]			87.13
STST-Encoder	STBX	TTBX	85.79
	STBX	TTB	87.38
	STB	TTBX	86.24
	STB	TTB	**88.03**
	STB	DTTB	**89.99**

modeling methods for different dimensions. Comparing our STST-Encoder with ST-TR-1s and DSTA-Net-1s which are simplified version of their proposed methods with single stream and no other augmentations (both of the ST-TR-1s and the DSTA-Net-1s use self-attention to model the skeleton sequence on temporal and spatial dimensions), we can find out that our STST-Encoder achieves higher performance due to the specially designed strategies on modeling skeleton sequence on temporal and spatial dimensions. The experimental results also prove the effectiveness of our method.

Based on the STB, we examine the effectiveness of our direction-aware strategy by comparing the encoder with DTTB and TTB. In the Table 4, although, compared with the previous outstanding method, the performance of our method without the direction-aware strategy is slightly lower. We can easily find that the performance of our encoder increases about 2% when the direction-aware strategy is applied. As shown in Figure 5, the confusion of our model on the action samples with the opposite temporal characteristic significantly reduces at least about 50%. Therefore, we conclude that the direction-aware strategy can significantly improve the model's ability to recognize opposite actions in temporal dimension.

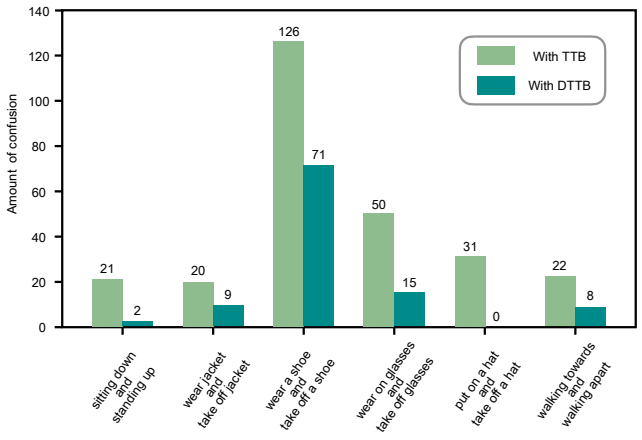

Figure 5: Confusion of opposite actions in temporal dimension on NTU-RGB+D (X-Sub). The lower the better.

Table 5: The effects of self-supervision tasks

Model	PS	PC	PT	SHREC 28(%)
ST-GCN[47]				87.70
DG-STA[4]				90.70
DSTA-Net[40]				93.69
STST-Encoder				93.93
	✓			94.48
		✓		94.19
			✓	94.53
	✓	✓		94.64
	✓		✓	94.75
		✓	✓	94.66
	✓	✓	✓	94.88

4.4.2 Effectiveness of Self-Supervision. To validate the efficacy of self-supervised learning methods, we build up the experiments based on the complete STST-Encoder and show its performance in Table 5. According to the experimental results, all these three self-supervised learning tasks we designed have all played effects on improving the model's performance. Among these three tasks, we can see that the PC task contributes less to the model performance improvement than the other two tasks. We analyze that our model encoder may be robust enough to a small amount of noise on the coordinates of joints, and it is not difficult to infer the coordinates of the original joints through the movements of the joints in the temporal dimension and the relationship with other joints. For the PS task, it boosts the model's performance by forcing the model to infer the types of joints through the motion pattern of the joints and the relative position relationship with other joints. In other words, the model learns to find the relationship between actions and movements of joints. As shown in Table 5, the PT task plays the most significant effect on improving the performance of the model among all self-supervised learning tasks, proving that modeling the changes of posture in the temporal dimension is fundamental to the action recognition task. Enhanced by the above three self-supervised tasks, the model performance has been increased.

5 CONCLUSION

In this work, we present two methods for the skeleton-based action recognition task: 1) a Transformer encoder specially designed for skeleton sequence modal data which can model the posture of each skeleton frame and its movement in the whole time span; 2) a multi-task self-supervised learning method for the three kinds of imperfect information situation. Both of them have had a significant positive impact on skeleton-based action recognition tasks. Sufficient experiments show that our model has achieved the improved performance with only the encoder. With the help of the multi-task self-supervised learning method, our model outperforms existing methods by a notable margin.

ACKNOWLEDGEMENT

This research was supported by the National Key Research and Development Program of China under Grant No. 2018AAA0100400.

REFERENCES

[1] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. 2020. A Spatio-temporal Transformer for 3D Human Motion Prediction. *arXiv e-prints* (2020), arXiv–2004.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7291–7299.

[4] Yuxiao Chen, Long Zhao, Xi Peng, Jianbo Yuan, and Dimitris N Metaxas. 2019. Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. *arXiv preprint arXiv:1907.08871* (2019).

[5] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. 2020. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 183–192.

[6] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. 2015. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE international conference on computer vision*. 3218–3226.

[7] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. 2020. Self-supervised spatio-temporal representation learning using variable playback speed prediction. *arXiv preprint arXiv:2003.02692* (2020).

[8] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, Joris Guerry, Bertrand Le Saux, and David Filliat. 2017. Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] Zewei Ding, Pichao Wang, Philip O Ogunbona, and Wanqing Li. 2017. Investigation of different skeleton features for cnn-based 3d action recognition. In *ICMEW)*. 617–622.

[11] Yong Du, Wei Wang, and Liang Wang. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1110–1118.

[12] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. 2010. Why does unsupervised pre-training help deep learning?. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 201–208.

[13] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. 2018. End-to-end learning of motion representation for video understanding. In *CVPR*. 6016–6025.

[14] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. 2017. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[15] Chuang Gan, Boqing Gong, Kun Liu, Hao Su, and Leonidas J Guibas. 2018. Geometry guided convolutional neural networks for self-supervised video representation learning. In *CVPR*. 5589–5597.

[16] Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alex G Hauptmann. 2015. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*. 2568–2577.

[17] Chuang Gan, Ting Yao, Kuiyuan Yang, Yi Yang, and Tao Mei. 2016. You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images. In *CVPR*. 923–932.

[18] Jingxuan Hou, Guijin Wang, Xinghao Chen, Jing-Hao Xue, Rui Zhu, and Huazhong Yang. 2018. Spatial-temporal attention res-TCN for skeleton-based dynamic hand gesture recognition. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.

[19] Mohamed E Hussein, Marwan Torki, Mohammad A Gowayyed, and Motaz El-Saban. 2013. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Twenty-third international joint conference on artificial intelligence*.

[20] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* (2017).

[21] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. 2017. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[22] Tae Soo Kim and Austin Reiter. 2017. Interpretable 3d human action analysis with temporal convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*. IEEE, 1623–1631.

[23] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2017. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*. 667–676.

[24] Bo Li, Yuchao Dai, Xuelian Cheng, Huahui Chen, Yi Lin, and Mingyi He. 2017. Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 601–604.

[25] Bin Li, Xi Li, Zhongfei Zhang, and Fei Wu. 2019. Spatio-temporal graph routing for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8561–8568.

[26] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. 2018. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. *arXiv preprint arXiv:1804.06055* (2018).

[27] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. 2019. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3595–3603.

[28] Ji Lin, Chuang Gan, and Song Han. 2019. TSM: Temporal Shift Module for Efficient Video Understanding. In *ICCV*. 7082–7092.

[29] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. 2020. MS2L: Multi-Task Self-Supervised Learning for Skeleton Based Action Recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*. 2490–2498.

[30] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. 2016. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European conference on computer vision*. Springer, 816–833.

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

[32] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. 2020. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 143–152.

[33] Xiang Long, Chuang Gan, Gerard De Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. 2018. Attention clusters: Purely attention based local feature integration for video classification. In *CVPR*. 7834–7843.

[34] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. 2020. Video cloze procedure for self-supervised spatio-temporal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11701–11708.

[35] Xuan Son Nguyen, Luc Brun, Olivier Lézoray, and Sébastien Bougleux. 2019. A neural network based on SPD manifold learning for skeleton-based hand gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12036–12045.

[36] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. 2020. Spatial temporal transformer network for skeleton-based action recognition. *arXiv preprint arXiv:2008.07404* (2020).

[37] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. 2016. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1010–1019.

[38] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. 2019. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7912–7921.

[39] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. 2019. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12026–12035.

[40] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. 2020. Decoupled Spatial-Temporal Attention Network for Skeleton-Based Action Recognition. *arXiv preprint arXiv:2007.03263* (2020).

[41] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. 2020. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing* 29 (2020), 9532–9545.

[42] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2017. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, Vol. 31.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[44] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 588–595.

[45] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*. Springer.

[46] Bo Wu, Wen-Huang Cheng, Yongdong Zhang, Qiushi Huang, Jintao Li, and Tao Mei. 2017. Sequential prediction of social media popularity with deep temporal context networks. *International Joint Conferences on Artificial Intelligence* (2017).

[47] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[48] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. 2019. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442* (2019).