sensors

MDPI

Article

MSST-RT: Multi-Stream Spatial-Temporal Relative Transformer for Skeleton-Based Action Recognition

Yan Sun [1,2,*,†], Yixin Shen [1,†] and Liyan Ma [1,2]

[1] School of Computer Engineering and Science, Shanghai University, 99 Shangda Road, Shanghai 200444, China; shenyixin@shu.edu.cn (Y.S.); liyanma@shu.edu.cn (L.M.)
[2] Shanghai Institute for Advanced Communication and Data Science, 333 Nanchen Road, Shanghai 200444, China
* Correspondence: yansun@shu.edu.cn
† These authors contributed equally to this work.

Abstract: Skeleton-based human action recognition has made great progress, especially with the development of a graph convolution network (GCN). The most important work is ST-GCN, which automatically learns both spatial and temporal patterns from skeleton sequences. However, this method still has some imperfections: only short-range correlations are appreciated, due to the limited receptive field of graph convolution. However, long-range dependence is essential for recognizing human action. In this work, we propose the use of a spatial-temporal relative transformer (ST-RT) to overcome these defects. Through introducing relay nodes, ST-RT avoids the transformer architecture, breaking the inherent skeleton topology in spatial and the order of skeleton sequence in temporal dimensions. Furthermore, we mine the dynamic information contained in motion at different scales. Finally, four ST-RTs, which extract spatial-temporal features from four kinds of skeleton sequence, are fused to form the final model, multi-stream spatial-temporal relative transformer (MSST-RT), to enhance performance. Extensive experiments evaluate the proposed methods on three benchmarks for skeleton-based action recognition: NTU RGB+D, NTU RGB+D 120 and UAV-Human. The results demonstrate that MSST-RT is on par with SOTA in terms of performance.

Keywords: action recognition; 3D skeleton; transformer; attention; spatial-temporal

check for updates

Citation: Sun, Y.; Shen, Y.; Ma, L. MSST-RT: Multi-Stream Spatial-Temporal Relative Transformer for Skeleton-Based Action Recognition. *Sensors* **2021**, *21*, 5339. https://doi.org/10.3390/s21165339

Academic Editor: Cosimo Distante

Received: 30 June 2021
Accepted: 3 August 2021
Published: 7 August 2021

1. Introduction

Human action recognition attracts extensive attention due to its important application prospects in many fields, such as healthcare assistance, human–computer interaction and autonomous driving. Many exciting developments have taken place in the study of this research topic in recent years. A consensus has been reached that the key to human action recognition is learning how to capture sufficient spatial-temporal information.

Early research in this field was mostly based on RGB videos. However, RGB videos are often affected by the variations in illuminations, changes in camera viewpoints and background noise. All of these are obstacles in the process of extracting discriminative features. Compared with RGB data, the skeleton is free from these difficulties; it also has fewer restrictions in terms of data size. For example, it provides a compact form to represent dynamic information. These advantages make it possible to design lightweight models based on skeleton data. Meanwhile, with the development of human motion estimation technologies, such as advanced human pose estimation algorithms [1] and multimodal sensors [2], skeleton data are easier and cheaper to obtain than before, which inspires researchers to explore various approaches based on skeleton data for action recognition. In this paper, the action recognition research is based on 3D skeleton data.

To extract information from skeleton data, researchers have mainly explored four aspects: handcrafted-features-based methods, convolution neural-network-based methods, recurrent neural-network-based methods and graph convolution-network-based methods.

In recent years, the graph convolution network (GCN) has become the most widespread method, with the characteristics of low computation and high accuracy. Although these methods have achieved great success in terms of performance, there some defects still exist: (1) Most human actions are performed by a collaborative effort among the joints that not linked in the human skeleton. For example, the left hand and right hand are not linked, but they have a strong correlation in the action of "clapping". Graph convolutions only gather information from the local neighbor joints of two hands, respectively, but neglect the relationship between hands. Applying graph convolution repeatedly would obtain a multi-hop dependence between the two hands; however, this would occur at the expense of increasing computational complexity and would make optimization more difficult. (2) It is hard to achieve long-range dependence utilizing only graph convolution in the model, although it plays an important role in temporal dimension. (3) The rich information derived from different scales of motion can effectively supplement each other, but this is usually neglected in the research.

In this paper, we propose a novel mechanism with a lightweight transformer to overcome the first and second limitations mentioned in the previous paragraph, and call it a relative transformer. The relative transformer mechanism is employed in spatial skeleton-based architecture to build bridges between two distant joints and propagate signals. It is also employed in the temporal dimension to capture long-range interactions between two distant frames. As a consequence, the model is named the spatial-temporal relative transformer (ST-RT). For the third defect, we also propose an effective dynamic representation which fuses three different scales of motion and raw position to obtain richer information from a skeleton sequence in ST-RT.

The overall architecture of the proposed MSST-RT is shown in Figure 1. As well as the joint information, the bone information derived from the joints' positions is also learned by ST-RT. Furthermore, the skeleton sequence, obtained by different sampling strategies, provides supplementary information through model training, e.g., 10 and 20 frames are sampled from the joint sequence and bone sequence. Four ST-RTs are combined to extract features from two joint sequences and two bone sequences. We name this the multi-stream spatial-temporal relative transformer (MSST-RT). Significantly, batch normalization is adopted instead of layer normalization in relative transformer to obtain a faster training time and higher performance.

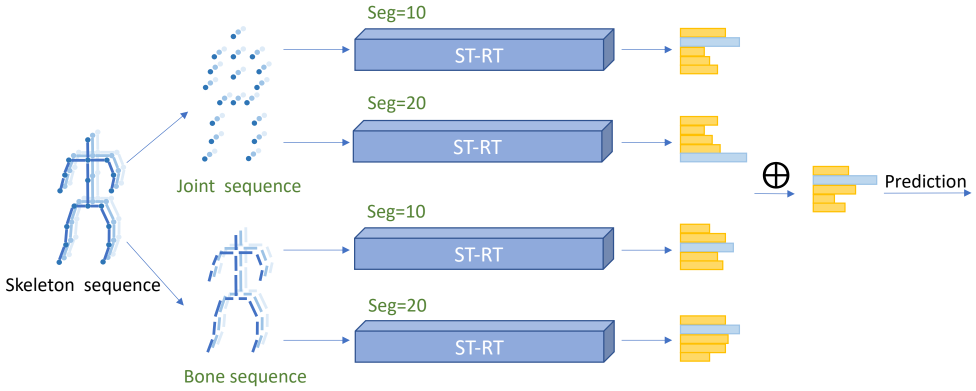


Figure 1. Illustration of the overall architecture of the proposed MSST-RT. The sum of all scores from four ST-RTs is treated as the final prediction.

There are five main contributions of this work, summarized as follows:

- We propose an MSST-RT model comprising four ST-RTs. Each of them extracts features from a corresponding skeleton sequence, which complement each other. It is worth noting that MSST-RT eschews recurrence and graph convolution and instead relies entirely on a relative transformer mechanism to learn long-distance dependencies.

- A multi-scale dynamic representation (DR) combines various scale-of-motion features from one skeleton sequence.
- A lightweight relative transformer module is designed for spatial and temporal modeling. Regarding the spatial dimension, a spatial relative transformer module (SRT) is introduced to establish long-range dependencies while maintaining the origin skeleton topology. In the temporal dimension, the temporal relative transformer module (TRT) studies long-range interaction between nonadjacent frames, with the order of skeleton sequence remaining unchanged.
- Attention responses in SRT and TRT are visualized to show the effects of the attention mechanism. This proves that the proposed ST-RT pays great attention to some nodes and the distribution of attention is consistent with human perception.
- Our MSST-RT achieves state-of-the-art on three benchmark datasets, i.e., NTU RGB+D, NTU RGB+D 120 and UAV-Human, in action recognition based on skeleton.

2. Related Work

2.1. Skeleton-Based Action Recognition

Action recognition based on skeleton can be summarized into two major categories, i.e., traditional methods [3–5] that use hand-crafted features and deep learning methods. In recent years, deep neural networks have gained increasing favor due to their remarkable performance in the following: (1) RNN-based methods [6–10] model the contextual information by splicing the coordinates of the key points at each moment into a vector and concatenating all vectors for each frame. For example, H. Wang and W. Liang introduce a two-stream RNN [7] architecture which incorporates both spatial and temporal RNN networks for skeleton based action recognition. The HBRNN-L model [9] decompose the human skeleton into five parts, and then feed them to five bidirectional RNN subnets. (2) CNN-based methods [11–15] usually reconstructed the skeleton sequenceto a series of pseudo-images to obtain the spatial cues. Wang et al. [16] propose a novel CNN network, which represents the spatial configuration and dynamic information of a joint trajectory as three texture images by color encoding. (3) GCN-based methods [17–19] preserve the inherent topological graph data struct of the skeleton by treating joints and bones as vertices and edges. Taking advantage of the graph data struct, GCN-based methods build operators in the non-Euclidean space and outperform the other two approaches. In particular, the ST-GCN model proposed by Yan et al. [17] is the first construct, a spatial-temporal graph that offers a new partitioning strategy.A lot of other GCN-based methods regard it as a baseline, or improve on it.

2.2. Transformer

Transformer [20] is a novel architecture, which has now become the standard for NLP tasks. It handles long-range dependencies by relying on self-attention rather than sequence-aligned RNNs or convolution. Recently, many transformer-based models have been proposed to improve the original architecture, e.g., Set Transformer [21], Routing Transformer [22] and Star-Transformer [23]. Most of them aim to overcome the computation complexity and large memory overhead. In addition to being developed for NLP, transformers became a research hotspot in the fields of Computer Vision. Carion et al. proposed a Detection Transformer (DETR) [24], which is the first object detection framework that combines the convolutional neural network with a transformer. Vision Transformer (ViT) [25] utilizes transformer architecture without CNN and it outperforms the state-of-the-art convolutional network in various image classification tasks. In our work, the model only consists of relative transformer modules. Inspired by the Star-Transformer, the proposed relative transformer is a variation based on standard transformer architecture designed for skeleton action recognition.

3. Background

The transformer model mainly includes an encoding component and decoding component. Our work is concerned with the encoding part, which is broken down into a self-attention layer and feed-forward neural network layer. This section will briefly review the relevant knowledge.

3.1. Attention in the Encoding

In NLP tasks, each word has a query vector q, a key vector k and a value vector v to calculate a score, which represents the effect of other words on the input sentence on the encoding of the word. The score is calculated by taking the dot product $s_{ij} = q_i \cdot k_j^T$ i, $j = 1, \cdots, n$. The output of the self-attention layer is obtained by multiplying each value vector by the score and summing up the weighted value vectors. While packing all query vectors into a matrix Q, all key vectors into a matrix K and all value vectors into a matrix V, the attention function is defined as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where$\frac{1}{\sqrt{d_k}}$leads to a more stable gradient. Furthermore, the "multi-headed" attention initializes multiple sets of weight matrices randomly and obtains different scores. Concatenating the scores as the result helps the network capture richer features.

3.2. Feed-Forward Neural Network in the Encoding

In addition to the attention layer, the encoder also contains a feed-forward neural network, which consists of two linear transformations and an ReLU activation:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{2}$$

The network projects the refined vector, which was obtained from multi-head attention, into a larger space to improve the ability to capture information. We also apply it in our relative transformer network.

4. Multi-Stream Spatial–Temporal Relative Transformer Network (MSST-RT)

In order to solve the limitations of the small receptive field in traditional convolution, a Transformer is introduced to the skeleton-based action recognition models instead of graph convolution. Since the graph is in an unordered sequence, we propose a new transformer architecture, named the relative transformer to keep the topology of the graph with lower complexity. The proposed multi-stream spatial-temporal relative transformer network (MSST-RT) consists of four spatial-temporal relative transformer networks (ST-RT). Four kinds of skeleton sequence, including a joint sequence of 10 sampled frames, a joint sequence of 20 sampled frames, a bone sequence of 10 sampled frames and a bone sequence of 20 sampled frames (shown in Figure 1). They are fed into four ST-RTs for feature extraction, fusion and prediction.

In this section, we will introduce our model ST-RT, where relative transformers are employed in both space and temporal dimensions; the model architecture is illustrated in Figure 2. It consists of three modules: dynamics representation (DR), spatial relative transformer (SRT) and temporal relative transformer (TRT). Meanwhile, each relative transformer module contains three node update blocks and each block is subdivided into two sub-blocks: joint nodes update block and relay node update block. The feed-forward neural network (FFN) is connected behind them in both a spatial and temporal relative transformer.

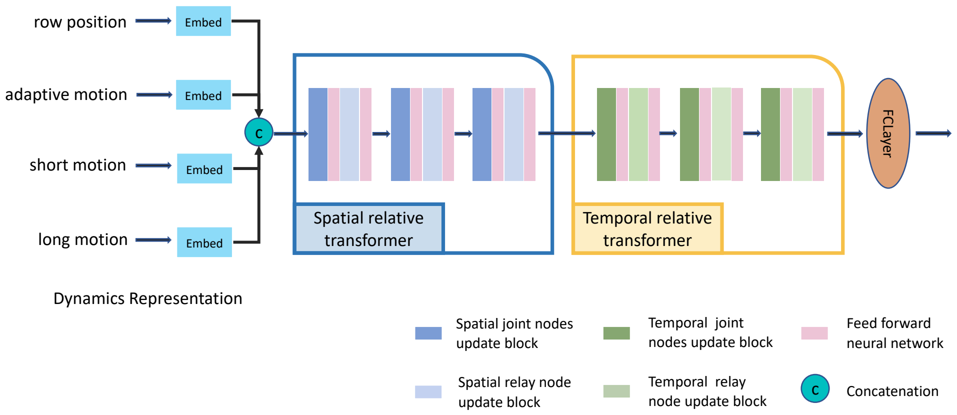

row position
Embed
adaptive motion
Embed
short motion
Embed
long motion
Embed
C
Spatial relative transformer
Temporal relative transformer
FCLayer
Dynamics Representation
Spatial joint nodes update block
Temporal joint nodes update block
Feed forward neural network
Spatial relay node update block
Temporal relay node update block
C Concatenation

Figure 2. Illustration of the spatial-temporal relative transformer (SR-RT). The skeleton data are processed by three modules and then fed into the fully connected layer to predict the score for each action class.

4.1. Dynamics Representation (DR)

Temporal difference operations are always adopted for motion extraction in action recognition, such as TEINet [26] and STM [27]. In action recognition, all the 3D positions of skeleton joints in NTU60 and NTU120 datasets are stored, and UAV-Human stores 2D positions. Figure 3 shows the 3D positions of skeleton joints. A joint in frame a is represented as $J_i^a = (x_i^a, y_i^a, z_i^a)$, the same joint in frame $b(b > a)$ is $J_i^b = (x_i^b, y_i^b, z_i^b)$. The temporal difference is the subtraction between the same joint in the two frames, which can be denoted as $(x_i^b - x_i^a, y_i^b - y_i^a, z_i^b - z_i^a)$.In view of the significant motion variations in actions, we combine different scale motions to model the temporal information of action, as shown in Figure 3. This operation improves the generalization of our network, as the fixed motion focuses on different ranges of motion while the adaptive motion focuses on different durations.

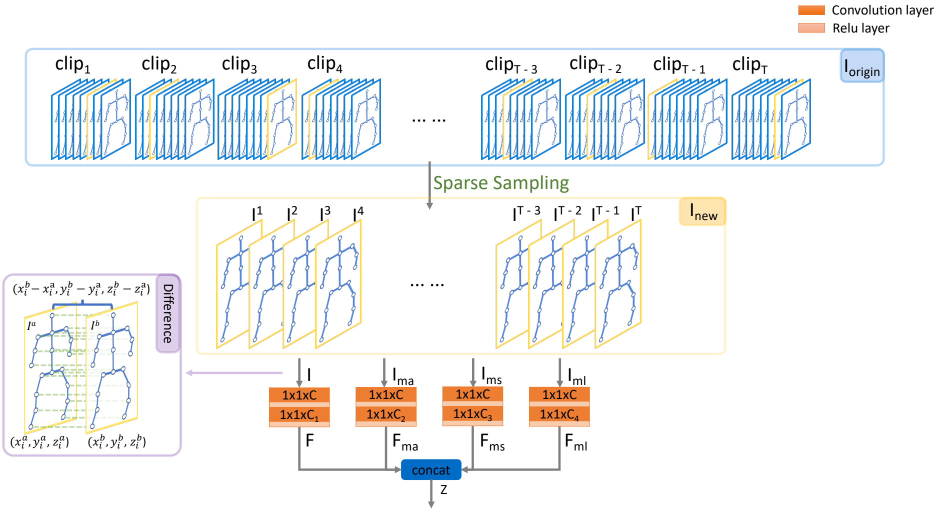


Figure 3. Illustration of dynamics representation (DR). There are four streams of skeleton information embedded into a higher dimension by the embedding block and then concatenated as an input of the spatial relative transformer. Each block consists of two convolution layers and two activation layers.

In more detail, we divide the original sequence $I_{origin} = [I^1, \cdots, I^F]$ into T equal clips and randomly sample one frame from each clip to form a new sequence $I_{new} = [I^1, \cdots, I^T]$ in order. The original sequence I_{origin} is either a joint sequence or bone sequence. The bone extractor proposed by Shi et al. [28] is applied to datasets to obtain bone information, which ensures that the representation of both the bone sequence and joint sequence are exactly the same. I represents the row positions of all joint ponits in single frame. The motion is computed by taking the difference of each joint node $Jo_i^t(Jn_i^t)$ between two frames: Jo_i^t denotes the i-th joint node in frame t of I_{origin} and Jn_i^t denotes the i-th joint node in frame t of I_{new}. The adaptive motion I_{ma} is the difference between consecutive frames in I_{new}, which represents different scales of motion information in the unequal video:

$$I_{ma} = \{I_{ma}^1, I_{ma}^2, \cdots, I_{ma}^T\} \tag{3}$$

$$I_{ma}^t = \{Jn_1^{t+1} - Jn_1^t, Jn_2^{t+1} - Jn_2^t, \cdots, Jn_N^{t+1} - Jn_N^t\} \ t = 1, 2, \cdots, T \tag{4}$$

where I_{ma}^t denotes the adaptive motion of frame t in the new sequence. Note that although the difference is between the adjacent frames in I_{new}, the distance between these two frames depends on their location of I_{origin}, which is interconnected with the length of the skeleton sequence. Hence, each skeleton sequence obtains an adaptive scale motion based on length.

Furthermore, there are two types of fixed motion: short-scale I_{ms} and long-scale I_{ml}. The function is expressed in the following:

$$I_{ms} = \{I_{ms}^1, I_{ms}^2, \cdots, I_{ms}^T\} \tag{5}$$

$$I_{ml} = \{I_{ml}^1, I_{ml}^2, \cdots, I_{ml}^T\} \tag{6}$$

$$I_{ms}^t = \{Jo_1^{f+3} - Jo_1^f, Jo_2^{f+3} - Jo_2^f, \cdots, Jo_N^{f+3} - Jo_N^f\} \ t = 1, 2, \cdots, T \tag{7}$$

$$I_{ml}^t = \{Jo_1^{f+6} - Jo_1^f, Jo_2^{f+6} - Jo_2^f, \cdots, Jo_N^{f+6} - Jo_N^f\} \ t = 1, 2, \cdots, T \tag{8}$$

where I_{ms}^t denotes the short motion of frame t in the original sequence and I_{ml}^t is the long motion of frame t in the original sequence. f represents the frame number in the original video.

Finally, the row position and three different types of motion are embedded into the high-dimension tensor, i.e., F^t, F_{ma}^t, F_{ms}^t and F_{ml}^t, and concatenate them.

$$Z^t = concat([F^t, F_{ma}^t, F_{ms}^t, F_{ml}^t]) = \{Z_1^t, Z_2^t, \cdots, Z_N^t\} \tag{9}$$

$$Z = \{Z^1, Z^2, \cdots, Z^T\} \tag{10}$$

where Z_i^t, Z^t and Z are the dynamic representations of the i-th joint node of frame t, the t-th frame and the new sequence, respectively.

The embedding block consists of two convolution layers and two activation layers, as shown in Figure 3. These blocks extract features from 2D/3D position and motion tensors. The size of the convolution kernel is explained in Section 5.2.

4.2. Spatial Relative Transformer (SRT)

4.2.1. Architecture

Different from the standard transformer, we prepend a virtual node to the graph of skeleton as the input. The virtual node gathers the global information from each joint node and scatters the information to all joint nodes; we named it the spatial-relay node. For the joint node and sptial-relay node, there are two corresponding connections: the inherent connections and the virtual connections.

Spatial Inherent Connections

As shown in Figure 4a, we establish inherent connections for all adjacent joints that have bone connections to preserve the inherent graph topology in skeletons. Such connections with prior knowledge allow each joint node to gather the local information from its adjacent joint nodes. Meanwhile, they enable joints to obtain more direct information from neighbors than non-adjacent joints, consistent with the general perception: neighbor joints are generally more important. A skeleton graph with n joint nodes has $n - 1$ inherent connections.

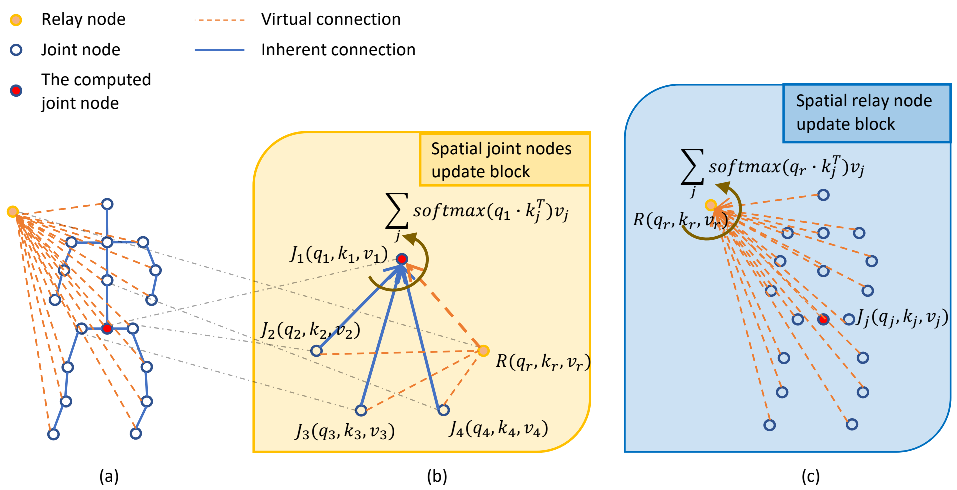


Figure 4. Illustration of the update blocks in a spatial relative transformer (SRT). The graph structure in SRT is described in (**a**). Updating operates on each joint node by obtaining local information from adjacent joint nodes and non-local information from the spatial-relay node in (**b**). Spatial-relay nodes are updated by scoring the contribution of each node, including spatial joint nodes and the spatial-relay node in (**c**).

Spatial Virtual Connections

The connections between every joint node and spatial-relay node are named after virtual connection (see in Figure 4a). Through the virtual connections, the sptial-relay node captures the global composition relationship; therefore, each joint node can obtain the information between the non-adjacent joint nodes. A skeleton graph with n joint nodes has n virtual connections. The combination of inherent and virtual connections makes the relative transformer obtain both local and global information. Compared with the standard transformer, the number of connections includes inherent and virtual connections. As a consequence, the model establishes a long-range dependency with low computational efficienvy and memory overhead.

4.2.2. Implementation of SRT

In the spatial relative transformer model, each frame has its own relative transformer and we look at the model within one single frame. The model input $J_{graph} = \{J_1^t, J_2^t, \cdots, J_N^t\}$ is a sequence of joint nodes at time t, where N is the number of nodes in this frame. $B_{J_i^t}$ is a set which contains the label of the adjacent joint nodes of J_i^t. Each node $J_i^t(R^t)$ has a query vector $q_i^t(q_r^t)$, a key vector $k_i^t(k_r^t)$ and a value vector $v_i^t(v_r^t)$.

Spatial Joint Nodes Update Block (SJU)

For each joint node, we calculate the strength of the correlations between them and their adjacent joint nodes (including the neighbor nodes $J_{B_{J_i^t}}^t$, the relay node R^t and itself J_i^t) by taking the key dot product with the query vector, as shown in equation:

$$\alpha_{ij}^t = q_i^t \cdot {k_j^t}^T, i \in N, j \in [i; B_{J_i^t}; r] \tag{11}$$

where α_{ij}^t represents the importance of node j on node i. Each neighbor node value vector v_j^t is multiplied by the corresponding score α_{ij}^t and added up to update the joint node, as shown below:

$$J_i^t = \sum_j softmax_j(\frac{\alpha_{ij}^t}{\sqrt{d_k}})v_j^t,\ i \in N,\ j \in [i; B_{J_i^t}; r] \tag{12}$$

where J_i^t is the updated result, aggregating local and global information. d_k is the channel dimension of the key value (shown in Figure 4b).

In the model implement, the computations are implemented in matrix form. First, the q_i^t, k_j^t and v_j^t vectors are packed into Q^t, K^t and V^t. Matrix $Q^t \in R^{C\times 1\times N}$ contains all joints' query vectors for a single skeleton. Both matrix $K^t \in R^{C\times A\times N}$ and $V^t \in R^{C\times A\times N}$ contain all key vectors and value vectors, which correspond with an adjacent node matrix M_a (the adjacent node matrix will be introduced in Section 4.4). C denotes the feature dimension; N is the number of joints in one single skeleton; A represents the maximum number of adjacent nodes. The attention in matrix form is defined as follows:

$$Att(Q^t, K^t, V^t) = \sum_{i\in A}(softmax(\frac{mask(Q^t \circ K^t)}{\sqrt{dk}}) \circ V^t) \tag{13}$$

where $\circ$ is a Hadamard product and the mask operation removes the zeros taken by the padding operation.

Spatial Relay Node Update Block (SRU)

To ensure that the spatial-relay node better aggregates the information of all joint nodes, we also apply a transformer (see in Figure 4c). The importance of each joint node α_{rj}^t is computed by the query vector q_r^t and the key vector k_j^t , as shown in the following:

$$\alpha_{rj}^t = q_r^t \cdot {k_j^t}^T, \; j \in [r;N] \tag{14}$$

The relay node R^t is updated by:

$$R^t = \sum_j softmax_j(\frac{\alpha_{rj}^t}{\sqrt{d_k}})v_j^t, j \in [r;N] \tag{15}$$

For the matrix key, all key vectors k_j^t and value vectors v_j^t are packed into matrix $K^t \in R^{C\times L}$ and $V^t \in R^{C\times L}$, respectively. The attention in matrix form is defined as follows:

$$Att(q_r^t, K^t, V^t) = softmax(\frac{q_r^t \cdot K^t}{\sqrt{dk}}) \cdot (V^t)^T \tag{16}$$

where $q_r^t \in R^{1\times C}$ is the spatial relay node, $\cdot$ denotes the matrix product.

By alternately updating the joint nodes and the relay node, the spatial relative transformer will capture all the local and non-local information for an input graph. The overall update algorithm of the SRT is shown in the Algorithm 1.

4.3. Temporal Relative Transformer (TRT)

After designing a spatial relative transformer for each skeleton frame, we then formulate a temporal relative transformer to the skeleton sequence. Similar to the spatial relative transformer, the temporal relative transformer also consists of inherent connections and virtual connections by introducing a temporal-relay node.

Temporal Inherent Connections

Along the temporal dimension, the same joints across consecutive frames are treated as an input sequence into the model. Aside from the same joint nodes used between the adjacent, the joint nodes in the first and last frame are also connected, constituting a ring-shaped structure, as depicted in Figure 5d. A sequence formed by n nodes contains n inherent connections.

Algorithme 1 : The update of spatial relative transformer

```
Input : Embedded skeleton feature $Z_1^t, Z_2^t, \cdots, Z_N^t$
Output : The joint nodes $(J_1^t)^L, (J_2^t)^L, \cdots, (J_N^t)^L$ and relay node $(R^t)^L$ after $L$
         updates

// Initialization
$(J_i^t)^0, (J_i^t)^0, \cdots, (J_N^t)^0 \leftarrow Z_1^t, Z_2^t, \cdots, Z_N^t$
$(R^t)^0 \leftarrow$ average$(Z_1^t, Z_2^t, \cdots, Z_N^t)$
for $l \leftarrow 1$ to $L$ do
    // Update the spatial joint nodes
    for $i \leftarrow 1$ to $N$ do
        $(J_i^t)^l = \mathbf{SJU}((J_1^t)^{l-1}, (J_2^t)^{l-1}, \cdots, (J_N^t)^{l-1}, (R^t)^{l-1})$
    end
    $(J_1^t)^l, (J_2^t)^l, \cdots, (J_N^t)^l = \mathbf{FFN}((J_1^t)^l, (J_2^t)^l, \cdots, (J_N^t)^l)$

    // Update the spatial relay node
    $(R^t)^l = \mathbf{SRU}((R^t)^{l-1}, (J_1^t)^l, (J_2^t)^l, \cdots, (J_N^t)^l)$
    $(R^t)^l = \mathbf{FFN}((R^t)^l)$
end
```

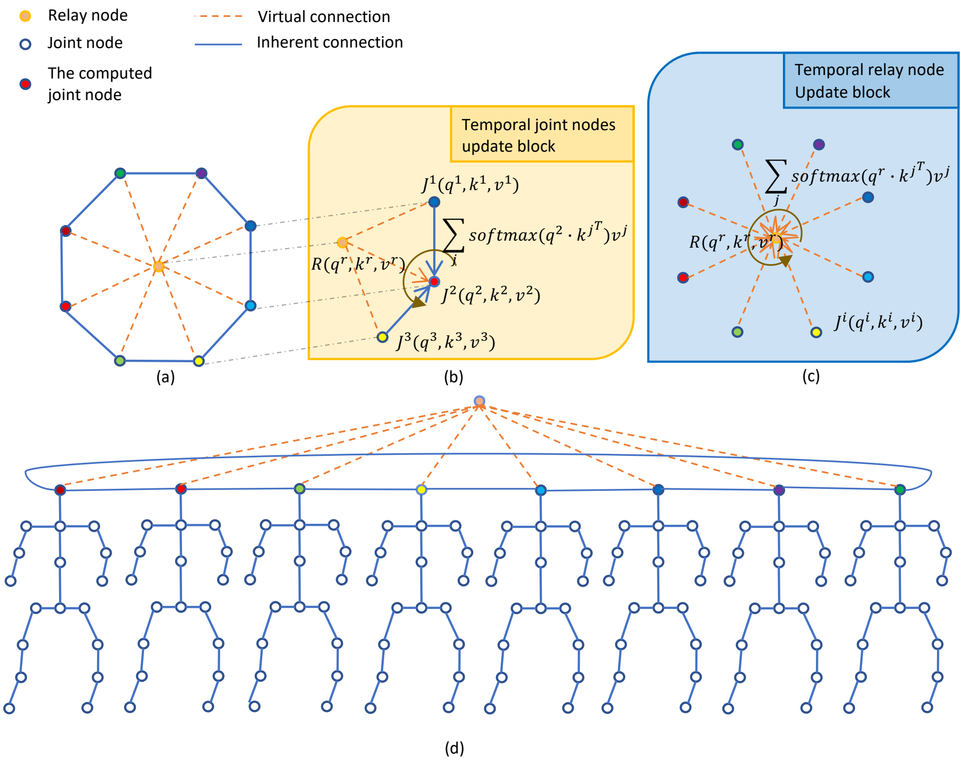


Figure 5. Illustration of the update blocks in temporal relative transformer (TRT). The same joint nodes in all sampled skeleton are connected in order and the joint nodes in the first and last frame are also connected in (**d**). This approach constitutes a ring-shaped structure, as shown in (**a**). Furthermore, each joint node and the temporal-relay node are updated by TJU in (**b**) and TRU in (**c**), respectively, similar to the methods in SRT.

Temporal Virtual Connections

In the temporal relative transformer, each virtual connection links a joint node to the temporal-relay node, similar to the operation in spatial relative transformer. Hence, a sequence which contains n nodes has n virtual connections, as shown in Figure 5a. In a word, temporal relative transformer can capture the relationship between adjacent frames by the inherent connections, and the long-range relationship is captured by the virtual connections. This means that the semantic compositions are divided between the inherent connections and virtual connections, which enables the model to work without heavy pre-training. Furthermore, it reduces the number of connections from n^2 to $2n$, where n is the skeleton sequence length.

4.3.1. Implementation of TRT

In the temporal relative transformer model, each node is considered independently. As a result, the model is applied to the sequence $J = \{J_v^1, J_v^2, \cdots, J_v^T\}$, which represents the same joint node from all frames in the sampled sequence. Each node $J_v^i(R_v)$ consists of a query vector $q_v^i(q_v^r)$, a key vector $k_v^i(k_v^r)$ and a value vector $v_v^i(v_v^r)$.

Temporal Joint Nodes Update Block (TJU)

As shown in Figure 5b, the joint node J_v^i is updated by the relay node r_v, the same joint node in neighbor frames (J_v^{i-1}, J_v^{i+1}) and itself. Their score α_v^{ij} is expressed in the following:

$$\alpha_v^{ij} = q_v^i \cdot {k_v^j}^T,\ i \in T, j \in [i-1; i; i+1; r] \tag{17}$$

where α_v^{ij} represents the importance of the node in j-th frame on the same node in i-th frame.The joint nodes J_v^i are updated by:

$$J_v^i = \sum_j softmax_j(\frac{\alpha_v^{ij}}{\sqrt{d_k}})v_v^j,\ i \in T, j \in [i-1; i; i+1; r] \tag{18}$$

For the matrix key, all query vectors q_v^i are packed into matrix $Q_v \in R^{C \times 1 \times T}$, all key vectors k_v^j are packed into matrix $K_v \in R^{C \times B \times T}$ and all key vectors v_v^j are packed into matrix $V_v \in R^{C \times B \times T}$. B is the total number of j. The attention in matrix form is defined as follows:

$$Att(Q_v, K_v, V_v) = \sum_{i \in B}(softmax(\frac{Q_v \circ K_v}{\sqrt{dk}}) \circ V_v) \tag{19}$$

where $\circ$ denotes Hadamard product.

Temporal Relay Node Update Block (TRU)

The information of all the frames is aggregated to the temporal-relay node in Figure 5c by scaled dot-product attention, as expressed in the following:

$$\alpha_v^{rj} = q_v^r \cdot {k_v^j}^T,\ j \in [r; T] \tag{20}$$

$$R_v = \sum_j softmax_j(\frac{\alpha_v^{rj}}{\sqrt{d_k}})v_v^j,\ j \in [r; T] \tag{21}$$

where α_v^{rj} is the attention score and $\frac{1}{\sqrt{d_k}}$ is a scaling factor. R_v denotes the updated relay node.

For the matrix key, all key vectors k_v^j and value vectors v_v^j are packed into matrix $K_v \in R^{C \times T}$ and $V_v \in R^{C \times T}$, respectively. The attention in matrix form is defined as follows:

$$Att(q_v^r, K_v, V_v) = softmax(\frac{q_v^r \cdot K_v}{\sqrt{dk}}) \cdot (V_v)^T \tag{22}$$

where $q_v^r \in R^{1 \times C}$ is the temporal relay node, $\cdot$ denotes matrix product.

By alternately updating the relay node and the same joint node on all frames, the temporal relative transformer finally captures all the relationships in an input frame sequence. The overall update algorithm of the TRT is shown in Algorithm 2.

Algorithme 2 : The update of temporal relative transformer

```
Input : Embedded skeleton feature $Z_v^1, Z_v^1, \cdots, Z_v^T$
Output : The joint nodes $(J_v^1)^L, (J_v^2)^L, \cdots, (J_v^T)^L$ and relay node $(R_v)^L$ after $L$
         updates

// Initialization
$(J_v^1)^0, (J_v^2)^0, \cdots, (J_v^T)^0 \leftarrow Z_v^1, Z_v^2, \cdots, Z_v^T$
$(R_v)^0 \leftarrow$ average$(Z_v^1, Z_v^2, \cdots, Z_v^T)$
for $l \leftarrow 1$ to $L$ do
    // Update the temporal joint nodes
    for $i \leftarrow 1$ to $T$ do
        $(J_v^i)^l =$ TJU$((J_v^1)^{l-1}, (J_v^2)^{l-1}, \cdots, (J_v^T)^{l-1}, (R_v)^{l-1})$
    end
    $(J_v^1)^l, (J_v^2)^l, \cdots, (J_v^T)^l =$ FFN$((J_v^1)^l, (J_v^2)^l, \cdots, (J_v^T)^l)$
    // Update the temporal relay node
    $(R_v)^l =$ TRU$((R_v)^{l-1}, (J_v^1)^l, (J_v^1)^l, \cdots, (J_v^T)^l)$
    $(R_v)^l =$ FFN$((R_v)^l)$
end
```

4.4. Detail of SRT and TRT

Given an input tensor of shape (B, C, V, T), where B is the number of batch sizes, C is the channel size of dynamic representation, V is the number of joint nodes in a skeleton and T is the number of frames in new sequence. As shown in Figure 6, we move the T dimension into the B dimension to get the new shape $(B \times T, C, V, 1)$ in SRT module and move the V dimension into the B dimension to obtain the new shape $(B \times V, C, T, 1)$ in the TRT module. The former implements the transformer on each frame independently, while the latter applies the transformer separately on each joint along the time dimension.

In a skeleton, each joint node generally has an unequal number of adjacent nodes. To solve this problem, we pad zeros to the nodes whose adjacent nodes are less than A. A denotes the maximum number of adjacent nodes. By doing this, we obtain an adjacent node matrix $M_a \in R^{N \times A}$, where N denotes the number of joint nodes in one skeleton. This operation is contained in step "neighbor" only in the SRT module, as shown in Figure 4b. Since these padding nodes are meaningless, we use a mask before the SoftMax operation, so that the attention mechanism avoids them.

Otherwise, multi-head attention with N_h heads is applied to obtain richer information. The formula for attention translates into the following form:

$$MultiHead(Q', K', V') = Concat(head_1, \cdots, head_N) \tag{23}$$

$$head_i = softmax(\frac{(Q'W_i^Q)(K'W_i^K)^T}{\sqrt{d_k^i}})V'W_i^V,\ i \in [1, N_h] \tag{24}$$

$$X = X'^{W_i^X},\ X \in \{Q, K, V\} \tag{25}$$

where Q', K', V' is the input matrixes, and W_i^Q, W_i^K, W_i^V are learnable parameter matrices. d_k^i is the channel dimension of K for $head_i$.

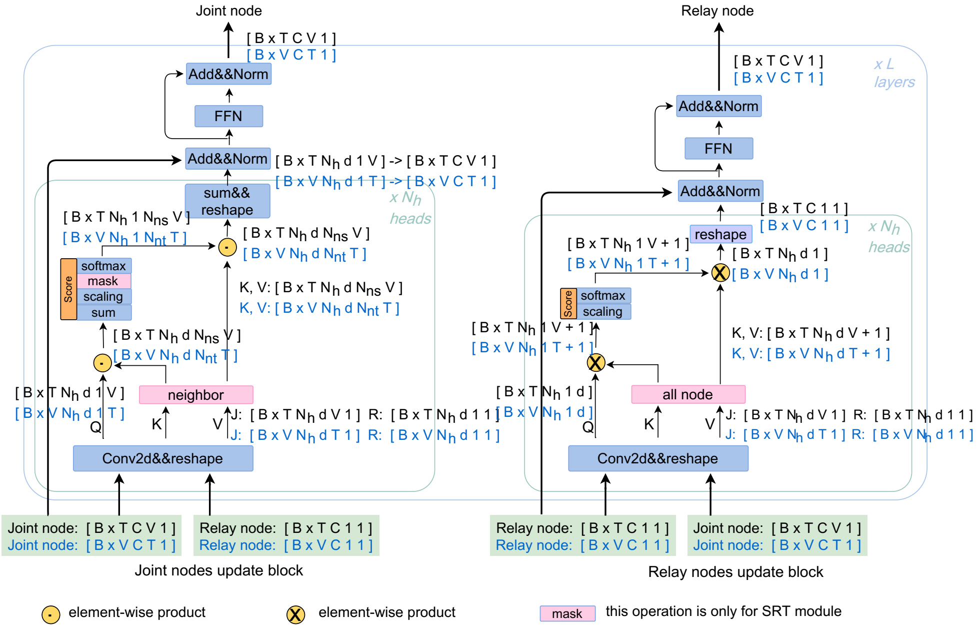

Joint node
[B x T C V 1]
[B x V C T 1]
Add&&Norm
FFN
Add&&Norm
[B x T N_h d 1 V] -> [B x T C V 1]
[B x V N_h d 1 T] -> [B x V C T 1]
sum&& reshape
x N_h heads
[B x T N_h 1 N_{ns} V]
[B x V N_h 1 N_{nt} T]
[B x T N_h d N_{ns} V]
[B x V N_h d N_{nt} T]
Score
softmax
mask
scaling
sum
K, V: [B x T N_h d N_{ns} V]
K, V: [B x V N_h d N_{nt} T]
[B x T N_h d N_{ns} V]
[B x V N_h d N_{nt} T]
[B x T N_h d 1 V]
[B x V N_h d 1 T]
neighbor
Q
K
V
J: [B x T N_h d V 1] R: [B x T N_h d 1 1]
J: [B x V N_h d T 1] R: [B x V N_h d 1 1]
Conv2d&&reshape
Joint node: [B x T C V 1]
Joint node: [B x V C T 1]
Relay node: [B x T C 1 1]
Relay node: [B x V C 1 1]
Joint nodes update block
Relay node
[B x T C V 1]
[B x V C T 1]
x L layers
Add&&Norm
FFN
Add&&Norm
[B x T C 1 1]
[B x V C 1 1]
x N_h heads
reshape
[B x T N_h 1 V + 1]
[B x V N_h 1 T + 1]
[B x T N_h d 1]
[B x V N_h d 1]
Score
softmax
scaling
[B x T N_h 1 V + 1]
[B x V N_h 1 T + 1]
K, V: [B x T N_h d V + 1]
K, V: [B x V N_h d T + 1]
[B x T N_h 1 d]
[B x V N_h 1 d]
all node
Q
K
V
J: [B x T N_h d V 1] R: [B x T N_h d 1 1]
J: [B x V N_h d T 1] R: [B x V N_h d 1 1]
Conv2d&&reshape
Relay node: [B x T C 1 1]
Relay node: [B x V C 1 1]
Joint node: [B x T C V 1]
Joint node: [B x V C T 1]
Relay nodes update block
element-wise product
element-wise product
mask
this operation is only for SRT module

Figure 6. Illustration of the detail of the SRT module and TRT module. The tensor shapes with black font are for the SRT module and the blue are for the TRT module. It can be seen that the "mask" operation is only employed in SRT. In the SRT (TRT) module, the "neighbor" operation finds the adjacent nodes for each node according to Figures 4b and 5b .

5. Experience

In this section, we evaluate the performance of the proposed MSST-RT on three large-scale skeleton datasets, namely, NTU RGB+D [29], NTU RGB+D 120 [30] and UAV-Human [31]. Extensive ablation studies are conducted to validate the contributions of different components in our model. Finally, we visualize the attention probabilities in SRU and TRU block to prove the effectiveness of the proposed ST-RT model.

5.1. Datasets

5.1.1. NTU RGB+D

The data in NTU RGB+D are collected in the lab environment by three Microsoft Kinect v2 cameras, which are widely used for skeleton-based action recognition tasks. They contain 56,680 skeleton sequences performed by 40 subjects, covering a total of 60 action classes. Each sequence contains, at most, two subject skeletons and each skeleton is composed of 25 joints. As the authors of this dataset recommended, we used two evaluation protocols, namely cross-subject (CS) and cross-view (CV). In the CS setting, 20 subjects were used for training and the rest for testing. In the CV setting, the sequences captured by camera 2 and camera 3 were used for training while the rest were used for testing.

5.1.2. NTU RGB+D 120

NTU RGB+D 120 is an extension of NTU RGB+D, whose action classes are increased to 120, subjects are increased to 106 and sequences are increased to 114,480. There are also two benchmarks, namely, cross-subject (C-subject) and cross-setup (C-setup). In the C-subject setting, 53 subjects are used for training and the rest for testing. In the C-setup setting, the dataset is divided by the parity of the setup IDs into two group, with one used for training and another used for testing.

5.1.3. UAV-Human

UAV-Human is a new dataset which is of significance fo practical UAV application scenarios. This dataset covers different types of human behavior and is collected by a flying

UAV in multiple urban and rural districts in the day and night. It contains 155 activity classes in six different modalities including RGB, depth, IR, fisheye, night-vision, and skeleton sequences. There are 22,476 frames with the 2D positions of 17 major key-points of the human body for skeleton-based recognition, 16,169 frames for training and 6307 frames for testing.

5.2. Implementation Details

All experiences are performed on the Pytorch platform. Similar to [20] , we used the Adam optimizer with $\beta = [0.9, 0.98]$ and $\epsilon = 10^{-9}$. Moreover, a two-phase training strategy was utilized to make the model convergence faster and more stable. The gradual warmup strategy linearly increased the learning rate from 4×10^{-7} to 0.0005 for the first training phase (the first 700 steps), while natural exponential decay with the weight decay of 0.9996 proportionally decreased the learning rate for the second phase. During training, the batch sizes for NTU60, UAV-Human and NTU120 were 64, 64 and 100, respectively, and the total training epochs was set to 30. Label smoothing of value $\epsilon_{ls} = 0.1$ was employed for all experiences.

In terms of data processing, the original skeleton coordinates of each frame were replaced with its displacement relative to the first frame. The actions, which contained two skeletons, such as hugging, were divided into two sequences so that each sequence contained only one. Moreover we employed data argumentation by randomly rotating the 3D skeletons to create more variational samples, which could improve the generalization ability of the network.

Raw position, short motion, long motion and adaptive motion were embedded from dimension space 2 (for UAV-Human) or 3 (for NTU and NTU120) to a dimension space of (C) 64 by the first convolution in all the experiments. Through the second convolution, they were embedded from a space of dimension (C) 64 to a space of dimension (C_1, C_2, C_3, C_4) 256, 256, 128 and 128, respectively. Note that the weights of the convolution layers are not shared among them.

5.3. Comparison to State of the Art

The performance of the proposed MSST-RT is compared with other state-of-the-art methods on the NTU60 , NTU120 and UAV-Human datasets in Tables 1–3, respectively. The contents of the "ST-RT()" brackets represent the number of sampled frames (10 or 20) and the type of input skeleton data (joint or bone), respectively.

Table 1. Performance comparison on NTU RGB+D.

Methods	Year	CS(%)	CV(%)
HBRNN-L [9]	2015	59.1	64.0
Part-Aware LSTM [29]	2016	62.9	70.3
ST-LSTM+Trust Gate [32]	2016	69.2	77.7
Two-stream RNN [7]	2017	71.3	79.5
STA-LSTM [33]	2017	73.4	81.2
VA-LSTM [34]	2017	79.4	87.6
ST-GCN [17]	2018	81.5	88.3
DPRL+GCNN [19]	2018	83.5	89.8
HCN [35]	2018	86.5	91.9
AS-GCN [36]	2019	86.8	94.2
TS-SAN [37]	2020	87.2	92.7
ST-RT(seg = 10, joint)	-	86.20	91.03
ST-RT(seg = 10, bone)	-	85.10	90.51
ST-RT(seg = 20, joint)	-	86.44	92.46
ST-RT(seg = 20, bone)	-	85.61	90.61
MSST-RT	-	88.43	93.21

As shown in Table 1, MSST-RT achieves a good performance, with 88.43% and 93.21%, respectively, for Cross Subject (CS) and Cross View (CV) settings of NTU RGB+D 60. It is worth noting that STA-LSTM and TS-SAN both adopt the attention mechanism, which is similar to the model idea that we proposed. The difference is that STA-LSTM uses LSTM in addition to the attention mechanism, while our proposed model only uses the attention mechanism. Moreover, our model outperforms this model by 14.96% with CS and 12.0% with CV. Cmparing our model with TS-SAN, the attention mechanisms in ST-RT are employed not only between video frames, but also between the joint nodes of the skeletons. Our model outperforms it for 1.23% with CS and 0.51% for CV.

Table 2. Performance comparison for NTU RGB+D 120.

Methods	Year	CS(%)	CV(%)
Part-Aware LSTM [29]	2016	25.5	26.3
ST-LSTM + Trust Gate [32]	2016	55.7	57.9
GCA-LSTM [8]	2017	58.3	59.2
Two-Stream GCA-LSTM [38]	2017	61.2	63.3
RotClips+MTCNN [39]	2018	64.6	66.9
SGN [40]	2020	79.2	81.5
ST-RT(seg = 10, joint)	-	76.22	80.26
ST-RT(seg = 10, bone)	-	75.09	78.37
ST-RT(seg = 20, joint)	-	76.67	79.58
ST-RT(seg = 20, bone)	-	76.45	76.85
MSST-RT	-	79.33	82.30

As shown in Table 2, the proposed MSST-RT achieves the best performance, with 79.33% for the C-subject setting and 82.30% for the C-setup setting. Most of the methods we compare are reported in [22].

Table 3. Performance comparison on UAV-Human.

Methods	Year	Accuracy (%)
ST-GCN [17]	2018	30.25
DGNN [18]	2019	29.90
2s-AGCN [28]	2019	34.84
HARD-Net [41]	2020	36.97
Shift-GCN [42]	2020	37.98
ST-RT(seg = 10, joint)	-	36.83
ST-RT(seg = 10, bone)	-	35.86
ST-RT(seg = 20, joint)	-	38.73
ST-RT(seg = 20, bone)	-	37.18
MSST-RT	-	41.22

As shown in Table 3, the proposed MSST-RT achieves the best performance, with 41.12%, and outperforms the second place by 3.24%. UAV-Human is a new dataset, released in 2021, and we compare the performance of our model with the results reported in [31]. Single-stream ST-RT, namely "ST-RT(Seg = 20, joint)", outperformed all methods.

5.4. Ablation Study

Ablation studies of performance were performed by the proposed ST-RT model on the NTU RGB+D dataset with the CS setting. Firstly, two normalization methods were chosen for comparison to provide faster training and better performance. Moreover, we compared dynamic representations with different motion combination methods and different networks with different stream combination methods. Finally, the models were investigated with different hyper-parameters, such as the number of sampled frames, layers and heads. Furthermore, an ablation study regarding computation cost was based on UVA-Human.

5.4.1. The Effect of Different Normalization Methods

Normalization is used in both dynamic representation modules and relative transformer modules. The normalization method in a relative transformer module can either use layer normalization or batch normalization, while the normalization method in the dynamic representation module only can use layer normalization. Table 4 shows that "ST-RT (BatchNorm)" is superior to "ST-RT (LayerNorm)" by 2.03%. The accuracy of the two normalization methods is shown in Figure 7 as a function of the epoch. The accuracy of "ST-RT (BatchNorm)" increases faster than that of "ST-RT (LayerNorm)" and it is stable at a higher value. The experiments demonstrate that batch normalization provides faster training times and a better performance.

Table 4. Accuracy comparison of the ST-RT models between two normalization methods in ST-RT model.

Norm Methods	CS(%)
ST-RT (LayerNorm)	84.42
ST-RT (BatchNorm)	86.46

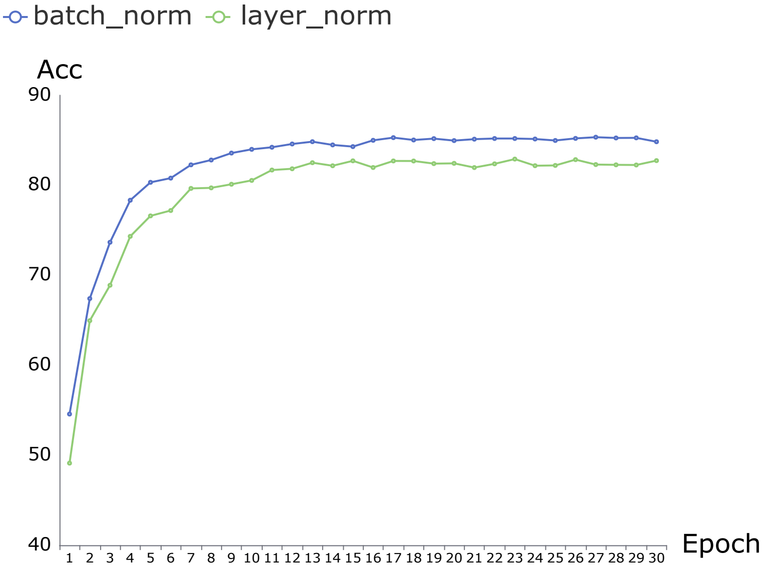


Figure 7. Difference in terms of accuracy and convergence speed between an ST-RT with batch normalization and an ST-RT with layer normalization.

5.4.2. The Effect of Dynamics Representation with Different Combination

To demonstrate that each motion provides different information, we compare our ST-TR model with three different models whose DR module only contains two motions, and show the results in Table 5. "ST-RT(w/o X motion)" denotes that the "X" motion is removed from the DR module in the ST-RT model, "X" is "short", "long" or "adaptive". The accuracy of the ST-TR model has a 0.60% reduction, a 1.17% reduction and a 0.69% reduction when it is without short motion, long motion or adaptive motion, respectively. This means that the information captured from each motion can complement the other two motions. According to the reduction in accuracy, we conclude that long motion contains richer dynamic information compared to short motion.

Table 5. Accuracy comparison of the ST-RT models with different motion and their combination. "w/o" equals without.

Methods	CS (%)
ST-RT(w/o short motion)	85.86
ST-RT(w/o long motion)	85.29
ST-RT(w/o adaptive motion)	85.77
ST-RT	86.46

Table 6 shows the effect of sampling frame numbers on model performance. From the results of "ST-RT(Seg = 15)" to "ST-RT(Seg = 10 && Seg = 15)", we observe that the model underperforms in both superfluous and insufficient frame scenarios. The experiments demonstrate that the model achieves the best performance with 86.46% when sampling 20 frames. In addition, we combined networks with different sampling frames (shown in the last four rows of Table 6, i.e., "ST-RT(Seg = 10 && Seg = 15)" denotes the combination of the network with 10 sample frames and the network with 15 sample frames). "ST-RT(Seg = 10 && Seg = 20)" outperforms other combinations with 87.48%, and this optimal parameter was applied in MSST-RT.

Table 6. Accuracy comparison of the ST-RT models with different sampling strategies and their combination. "X && Y" means the fusion of the model with X sampling strategy and the model with Y sampling strategy.

Methods	CS (%)
ST-RT (Seg = 10)	86.20
ST-RT (Seg = 15)	86.07
ST-RT (Seg = 20)	86.46
ST-RT (Seg=25)	85.21
ST-RT (Seg = 10 && Seg = 15)	86.93
ST-RT (Seg = 10 && Seg = 20)	87.74
ST-RT (Seg = 10 && Seg=25)	87.00
ST-RT (Seg = 15 && Seg = 20)	87.40
ST-RT (Seg = 15 && Seg = 25)	86.92
ST-RT (Seg = 20 && Seg = 25)	87.41

5.4.3. The Effect of Number of Layers and Multi-Heads in ST-RT Model

The results of ST-RT model with a different number of layers and heads are shown in Table 7. "LX" represents the SRT (TRT) module with X layers; each layer contains one SJU (TJU) block, one SRU (TRU) block and two FFNs. "HY" represents each attention mechanism in ST-RT model with Y multi-heads. By comparing the number of heads, we observe that more multi-heads always results in better performance, especially when the model's heads increase from 4 to 8. However, the affect of head number peaks when it increases to 12, and excessive heads can make the accuracy of the lightweight model decrease (shown in "ST-RT (L2H8)" and "ST-RT (L2H12)"). From the results of " ST-RT (L2H8)", " ST-RT (L3H8)" and " ST-RT (L4H8)", we conclude that the model underperforms if the number of layers is too high or too low. In sum, number of layers and heads is set as 3 and 8, respectively, for the proposed model, considering the number of parameters and the accuracy.

Table 7. Accuracy comparison for the ST-RT models with different sampling strategies, and their combination. "X && Y" means the fusion of the model with X sampling strategy and the model with Y sampling strategy.

Methods	CS (%)
ST-RT (L2H4)	85.60
ST-RT (L2H8)	86.26
ST-RT (L2H12)	86.15
ST-RT (L3H4)	84.76
ST-RT (L3H8)	86.46
ST-RT (L3H12)	86.81
ST-RT (L4H4)	81.71
ST-RT (L4H8)	85.21
ST-RT (L4H12)	85.35

5.4.4. The Effect of Dynamics Representation with Different Combination.

The speed and accuracy of different variants of MSST-RT with different numbers of streams are shown in Table 8. When using the same number of frames, the joint stream has a similar computation cost to the bone stream. The computation cost increases with the increase in the number of frame. In the variants with two streams, "ST-RT(s3+s4)" achieves the best performance, which is slightly lower than MSST-RT, and the computation cost is half of MSST-RT's. Hence, the stream numbers can be reduced if there are concerns regarding computation cost.

Table 8. The effect of number of streams in the ST-RT Model on UAV-Human. "s1" is the stream with 10 frames, joint frames, "s2" is the stream with 10 frames, bone frames, "s1" is the stream with 20 frames, joint frames, "s1" is the stream with 20 frames, bone frames. Furthermore, "ST-RT(X + Y)" is the fusion of the model with X stream and the model with Y stream. "Computation cost" means average second per epoch on the train set.

Methods	Accuracy (%)	Computation Cost(s)
ST-RT(s1)	36.83	305.00
ST-RT(s2)	35.86	312.28
ST-RT(s3)	38.73	447.51
ST-RT(s4)	37.18	450.94
ST-RT(s1 + s2)	39.05	617.28
ST-RT(s3 + s4)	41.18	898.45
ST-RT(s1 + s3)	39.46	752.51
ST-RT(s2 + s4)	38.20	763.22
MSST-RT	41.22	1515.73

5.5. Visualization of SRU and TRU

Our model applies the attention mechanism when updating joint nodes and virtual nodes in both the spatial and temporal dimensions. The attention response is visualized from the last SRU block in the Spatial Relative Transformer (SRT) and the last TRU block in the Temporal Relative Transformer (TRT).

The action salute is selected to visualize the attention response for eight multi-heads from the last SRU layer in Figure 8. The red circles represent the spatial-relay nodes and the blue circles represent the joint nodes. The blue lines represent the inherent connections and the yellow lines represent the virtual connections. We zoom in on the five nodes with the highest corresponding attention values, and the other nodes are shown by small circles. The attention response of each head is different and head1, head2, head3 and head4 all focus mainly on the left hand. This suggests that the attention mechanism works in a similar way to human perception. Otherwise, the actions clapping and kicking something are visualized in Appendix A, which shows that different actions focus on different nodes. Both left and right hands are important for "clapping", so most heads focus on hands in Figure A1a. For "kicking something" in Figure A1b, only the right foot is of great value.

Figure 9 shows the attention response for the action "salute" for eight multi-heads from the last TRU layer. The red circles represent the temporal-relay nodes (the left one in the sequence is the node before updating and the right one is the node after updating) and the 20 blue circles represent the eighth joint nodes from 20 sampled frames. The transparency of lines indicates the intensity of the attention response. The brighter color denotes the higher response. This shows that different frames are noted in different heads, and the information from each frame, including the temporal-relay node itself before being updated, is converged to the temporal-relay node. We can see that the temporal-relay node in last layer (the left red node in the sequence) receives a large amount of attention from head1 and head5.

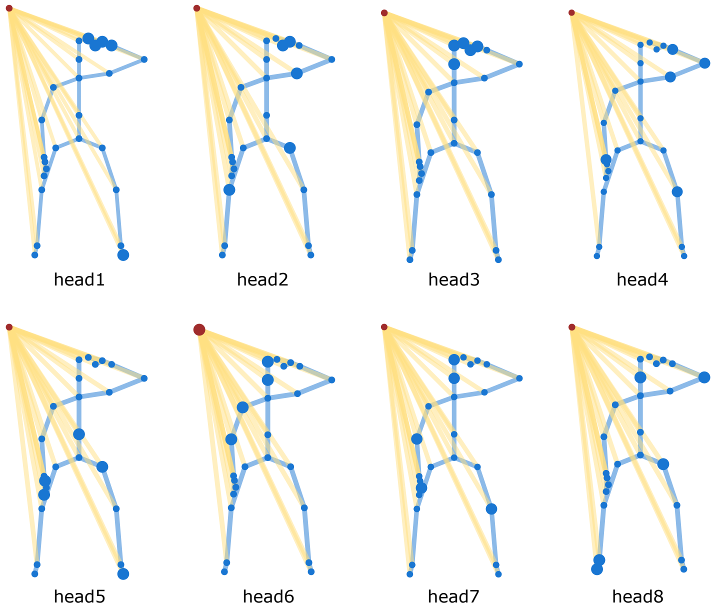


Figure 8. Visualization of the attention responses of the last SRU block in ST-RT model the action is salute.

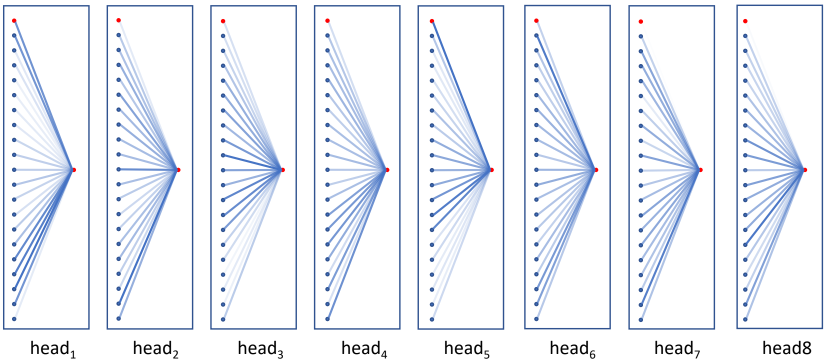


Figure 9. Visualization of the attention responses in the last TRU block in the ST-RT model.

6. Conclusions

In this work, transformer architecture is introduced to establish a long-range dependence instead of graph convolution. Significantly, MSST-RT relies on transformer architecture instead of recurrence, LSTM or graph convolution, which is a full attention model. Specifically, we propose a novel architecture based on standard transformer and named a relative transformer. This compensates for the deficiencies of the standard transformer while retaining the inherent topology of the skeleton, and significantly reduces computational complexity. The architecture, meanwhile, makes it possible for this to work without heavy pre-training. Furthermore, the relative transformer module evolves into a spatial relative transformer and temporal relative transformer, respectively, to extract spatial-temporal features. In addition, the DR module combines multi-scale motion information to adaptively recognize actions with different durations and different ranges of motion. Finally four streams with an ST-RT module, with four dynamic data streams, are fused to complement each other, realizing the further enhancement of performance. The final network, MSST-RT, achieves a state-of-the-art performance in skeleton-based action recognition on NTU RGB+D, NTU RGB+D 120 and UAV-Human. It is worth noting that single-stream ST-RT outperformed other methods on UAV-Human. It outperformed TS-SAN [37], which also adopted attention architecture. The results of attention response visualization verify the effectiveness of the proposed model for skeleton-based action recognition tasks.

Author Contributions: Conceptualization, Y.S. (Yan Sun), Y.S. (Yixin Shen) and L.M.; methodology, Y.S. (Yan Sun), Y.S. (Yixin Shen) and L.M.; software, Y.S. (Yan Sun) and Y.S. (Yixin Shen); validation, Y.S. (Yan Sun) and Y.S. (Yixin Shen); formal analysis, Y.S. (Yan Sun) and Y.S. (Yixin Shen); investigation, Y.S. (Yan Sun) and Y.S. (Yixin Shen); resources, Y.S. (Yan Sun) and Y.S. (Yixin Shen); data curation, Y.S. (Yan Sun) and Y.S. (Yixin Shen); writing—original draft preparation, Y.S. (Yan Sun) and Y.S. (Yixin Shen); writing—review and editing, Y.S. (Yan Sun), Y.S. (Yixin Shen) and L.M.; visualization, Y.S. (Yan Sun) and Y.S. (Yixin Shen); supervision, Y.S. (Yan Sun), Y.S. (Yixin Shen) and L.M.; project administration, Y.S. (Yan Sun) and L.M.; funding acquisition, Y.S. (Yan Sun), Y.S. (Yixin Shen) and L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number: 62002215; This research was funded by Shanghai Pujiang Program (20PJ1404400);This research was funded by Shanghai Municipal Natural Science Foundation (21ZR1423300).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: NTU RGB+D and NTU RGB+D 120 action recognition dataset: https://github.com/shahroudy/NTURGB-D, accessed on 30 June 2021. UAV-Human dataset: https://github.com/SUTDCV/UAV-Human, accessed on 30 June 2021.

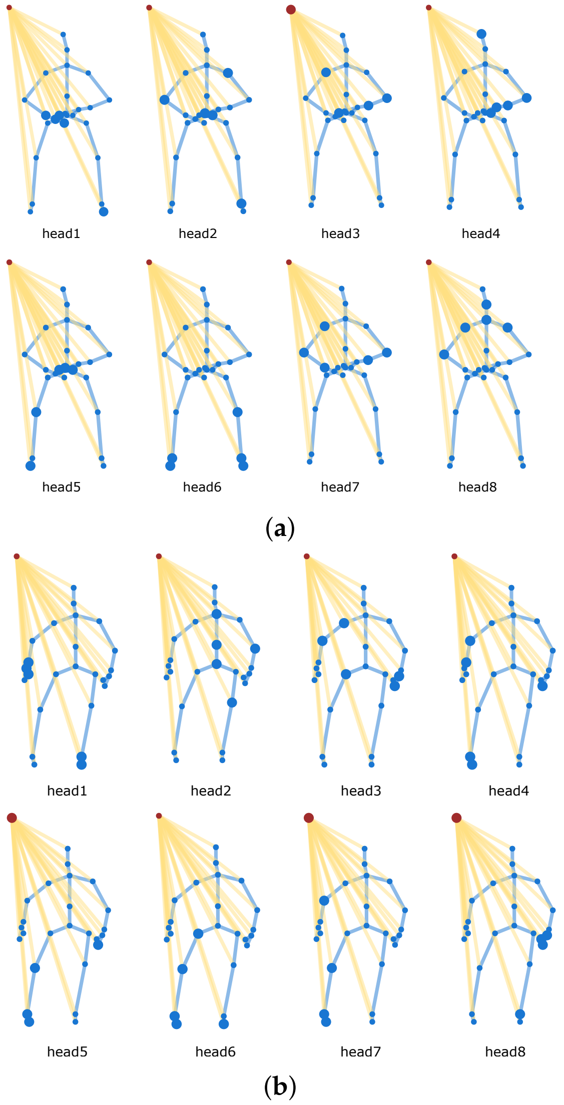
Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MSST-RT	Multi-stream spatial-temporal relative transformer
ST-RT	spatial-temporal relative transformer
FFN	feed forward neural network
DR	dynamic representation
SRT	spatial relative transformer
SJU	spatial joint nodes update block
SRU	spatial relay node update block
TRT	temporal relative transformer
TJU	temporal joint nodes update block
TRU	temporal relay node update block

Appendix A



(**a**)

(**b**)

Figure A1. Visualization of the attention responses of the last SRU block in ST-RT model with respect to two actions, i.e., clapping (**a**) and kicking something (**b**).

References

1. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Pattern Anal.* **2021**, *43*, 172–186. [CrossRef] [PubMed]
2. Zhang, Z. Microsoft Kinect Sensor and Its Effect. *IEEE Multimed.* **2012**, *19*, 4–10. [CrossRef]
3. Hu, J.F.; Zheng, W.S.; Lai, J.H.; Zhang, J. Jointly Learning Heterogeneous Features for RGB-D Activity Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5344–5352.
4. Vemulapalli, R.; Arrate, F.; Chellappa, R. 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 (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 588–595.
5. Hussein, M.; Torki, M.; Gowayyed, M.; El-Saban, M. Human Action Recognition Using a Temporal Hierarchy of Covariance Descriptors on 3D Joint Locations. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013.
6. Lev, G.; Sadeh, G.; Klein, B.; Wolf, L. RNN Fisher Vectors for Action Recognition and Image Annotation. In Proceedings of the Computer Vision–ECCV 2016, European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 833–850.
7. Wang, H.; Wang, L. Modeling Temporal Dynamics and Spatial Configurations of Actions Using Two-Stream Recurrent Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 499–508.

8. Liu, J.; Wang, G.; Hu, P.; Duan, L.Y.; Kot, A.C. Global Context-Aware Attention LSTM Networks for 3D Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1647–1656.
9. Du, Y.; Wang, W.; Wang, L. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1110–1118.
10. Jain, A.; Zamir, A.; Savarese, S.; Saxena, A. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vegas, NV, USA, 26 June–1 July 2016; pp. 5308–5317.
11. Chron, G.; Laptev, I.; Schmid, C. P-CNN: Pose-Based CNN Features for Action Recognition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 3218–3226.
12. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 568–576.
13. Ding, Z.; Wang, P.; Ogunbona, P.; Li, W. Investigation of different skeleton features for CNN-based 3D action recognition. In Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Hong Kong, China, 10–14 July 2017; pp. 617–622.
14. Liu, M.; Liu, H.; Chen, C. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognit.* **2017**, *68*, 346–362. [CrossRef]
15. Li, B.; Dai, Y.; Cheng, X.; Chen, H.; Lin, Y.; He, M. Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN. In Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Hong Kong, China, 10–14 July 2017; pp. 601–604.
16. Wang, P.; Li, W.; Li, C.; Hou, Y. Action recognition based on joint trajectory maps with convolutional neural networks. *Knowl. Based Syst.* **2015**, *158*, 43–53. [CrossRef]
17. Yan, S.; Xiong, Y.; Lin, D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2017; pp. 601–604
18. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Skeleton-based action recognition with directed graph neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7912–7921.
19. Tang, Y.; Tian, Y.; Lu, J.; Li, P.; Zhou, J. Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5323–5332.
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.;Polosukhin, I. Attention is all you need. In Proceedings of the 31th International Conference on Neural Information Processing Systems, Neural Information Processing Systems (NIPS), New York, NY, USA, 4–9 December 2017; pp. 6000–6010.
21. Lee, J.; Lee, Y.; Kim, J.; Kosiorek, A.; Choi, S.; Teh, Y.W. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 3744–3753.
22. Roy, A.; Saffar, M.; Vaswani, A.; Grangier, D. Efficient Content-Based Sparse Attention with Routing Transformers. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 53–68. [CrossRef]
23. Qiu, X.; Liu, P.; Shao, Y.; Xue, X.; Zhang, Z. Star-Transformer. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; pp. 1315–1325.
24. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the Computer Vision–ECCV 2020, European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
25. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 3–7 May 2021.
26. Liu, Z.; Luo, D.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; Lu, T. TEINet: Towards an Efficient Architecture for Video Recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 11669–11676.
27. Jiang, B.; Wang, M.; Gan, W.; Wu, W.; Yan, J. STM: SpatioTemporal and Motion Encoding for Action Recognition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 2000–2009.
28. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 12026–12035.
29. Shahroudy, A.; Liu, J.; Ng, T.; Wang, G. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vegas, NV, USA, 26 June–1 July 2016; pp. 1010–1019.

30. Liu, J.; Shahroudy, A.; Perez, M.; Wang, G.; Duan, L.; Kot, A. NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2684–2701. [CrossRef] [PubMed]
31. Li, T.; Liu, J.; Zhang, W.; Ni, Y.; Wang, W.; Li, Z. UAV-Human: A Large Benchmark for Human Behavior Understanding with Unmanned Aerial Vehicles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 19–25 June 2016; pp. 16266–16275.
32. Liu, J.; Shahroudy, A.; Xu, D.; Wang, G. Spatio-temporal lstm with trust gates for 3d human action recognition. In Proceedings of the Computer Vision–ECCV 2016, European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 816–833.
33. Song, S.; Lan, C.; Xing, J.; Zeng, W.; Liu, J. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. In Proceedings of the AAAI Conference on Artificial Intelligence, Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4263–4270.
34. Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J.; Zheng, N. View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition From Skeleton Data. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2117–2126.
35. Li, C.; Zhong, Q.; Xie, D.; Pu, S. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI, Stockholm, Sweden, 13–19 July 2018; pp. 786–792.
36. Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; Tian, Q. Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3595–3603.
37. Cho, S.; Maqbool, M.; Liu, F.; Foroosh, H. Self-Attention Network for Skeleton-based Human Action Recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, WACV'20, Snowmass Village, CO, USA, 1–5 March 2020; pp. 635–644.
38. Liu, J.; Wang, G.; Duan, L.; Abdiyeva, K.; Kot, A. Skeleton-Based Human Action Recognition with Global Context-Aware Attention LSTM Networks. *IEEE Trans. Image Process.* **2018**, *27*, 1586–1599. [CrossRef] [PubMed]
39. Ke, Q.; Bennamoun, M.; An, S.; Sohel, F.; Boussaind, F. Learning Clip Representations for Skeleton-Based 3D Action Recognition. *IEEE Trans. Image Process.* **2018**, *27*, 2842–2855. [CrossRef] [PubMed]
40. Zhang, P.; Lan, C.; Zeng, W.; Xing, J.; Xue, J.; Zheng, N. Semantics-Guided Neural Networks for Efficient Skeleton-Based Human Action Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 1112–1121.
41. Li, T.; Liu, J.; Zhang, W.; Duan, L. HARD-Net: Hardness-AwaRe Discrimination Network for 3D Early Activity Prediction. In Proceedings of the Computer Vision–ECCV 2020, European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 420–436.
42. Cheng, K.; Zhang, Y.; He, X.; Chen, W.; Cheng, J.; Lu, H. Skeleton-Based Action Recognition With Shift Graph Convolutional Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 183–192.