Skeleton-based action recognition via spatial and temporal transformer networks

Computer Vision and Image Understanding
journal homepage: www.elsevier.com

Skeleton-based Action Recognition via Spatial and Temporal Transformer Networks

Chiara Plizzari[a,b,**], Marco Cannici[a], Matteo Matteucci[a]

[a] *Politecnico di Milano, Via Giuseppe Ponzio 34/5, Milan 20133, Italy*
[b] *Politecnico di Torino, Corso Duca degli Abruzzi, 24, Turin 10129, Italy*

ABSTRACT

Skeleton-based Human Activity Recognition has achieved great interest in recent years as skeleton data has demonstrated being robust to illumination changes, body scales, dynamic camera views, and complex background. In particular, Spatial-Temporal Graph Convolutional Networks (ST-GCN) demonstrated to be effective in learning both spatial and temporal dependencies on non-Euclidean data such as skeleton graphs. Nevertheless, an effective encoding of the latent information underlying the 3D skeleton is still an open problem, especially when it comes to extracting effective information from joint motion patterns and their correlations. In this work, we propose a novel Spatial-Temporal Transformer network (ST-TR) which models dependencies between joints using the Transformer *self-attention* operator. In our ST-TR model, a Spatial Self-Attention module (SSA) is used to understand intra-frame interactions between different body parts, and a Temporal Self-Attention module (TSA) to model inter-frame correlations. The two are combined in a two-stream network, whose performance is evaluated on three large-scale datasets, NTU-RGB+D 60, NTU-RGB+D 120, and Kinetics Skeleton 400, consistently improving backbone results. Compared with methods that use the same input data, the proposed ST-TR achieves state-of-the-art performance on all datasets when using joints' coordinates as input, and results on-par with state-of-the-art when adding bones information.

1. Introduction

Human Action Recognition is achieving increasing interest in recent years for the progress achieved in deep learning and computer vision and for the interest of its applications in human-computer interaction, eldercare and healthcare assistance, as well as video surveillance. Recent advances in 3D depth cameras such as Microsoft Kinect (Zhang (2012)) and Intel RealSense (Keselman et al. (2017)) sensors, and advanced human pose estimation algorithms (Cao et al. (2019)) made it possible to estimate 3D skeleton coordinates quickly and accurately with cheap devices. Nevertheless, several aspects of skeleton-based action recognition still remain open (Zhang et al. (2019); Aggarwal and Ryoo (2011); Ren et al. (2020)). The most widespread method to perform skeleton-based action recognition has nowadays become Graph Neural Networks (GNNs), and in particular, Graph Convolutional Networks (GCNs) since, being an efficient representation of non-Euclidean data, they are able to effectively capture spatial (intra-frame) and temporal (inter-frame) information. Models making use of GCN were first introduced in skeleton-based action recognition by Yan et al. (2018) and they are usually referred to as Spatial-Temporal Graph Convolutional Networks (ST-GCNs). These models process spatial information by operating on skeleton bone-connections along space, and temporal information by considering additional time-connections between each skeleton joint along time. Despite being proven to perform very well on skeleton data, ST-GCN models have some structural limitations, some of them already addressed by Shi et al. (2019a); Simonyan and Zisserman (2014); Cheng et al. (2020); Liu et al. (2020).

First of all, the topology of the graph representing the human body is fixed for all layers and all the actions; this may prevent the extraction of rich representations for the skeleton movements during time, especially if graph links are directed and information can only flow along a predefined path. Secondly, both Spatial and Temporal Convolution are implemented starting from a standard 2D convolution. As such, they are lim-

**Corresponding author:
e-mail: chiara.plizzari@mail.polimi.it (Chiara Plizzari)

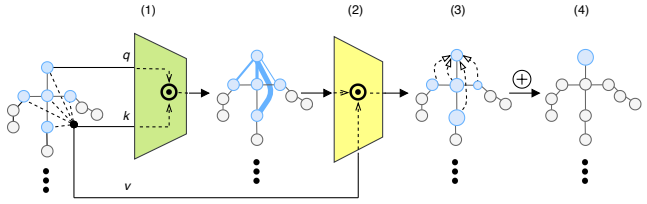


Fig. 1. Self-attention on skeleton joints. (1) For each body joint, a query q, a key k and a value vector v are calculated. (2) Then, the dot product (⊙) between the query of the joint and the key of all the other nodes is performed, representing the connection strength between each pair of nodes. (3) Finally, each node is scaled by its correlation w.r.t. the current node, (4) whose new features are obtained summing the weighted nodes together.

ited to operate in a local neighborhood, somehow restricted by the convolution kernel size. And finally, as a consequence of the previous, correlations between body joints not linked in the human skeleton, e.g., the left and right hands, are underestimated even if relevant in actions such as "clapping".

In this paper, we face all these limitations by employing a modified Transformer self-attention operator, as depicted in Figure 1. Despite being originally designed for Natural Language Processing (NLP) tasks, Transformer self-attention has shown remarkable results on a broad range of computer vision tasks, spanning from classical classification and detection (Dosovitskiy et al. (2020); Bello et al. (2019a); Wang et al. (2018); Carion et al. (2020)), to more complex tasks such as those involving point clouds (Zhao et al. (2020)), generative modeling (Oord et al. (2016); Parmar et al. (2018)) and captioning (He et al. (2020)). In our setting, the sequentiality and hierarchical structure of human skeleton sequences, as well as the flexibility of Transformer self-attention (Vaswani et al. (2017)) in modeling long-range dependencies, make the Transformer a perfect solution to tackle ST-GCN weaknesses. In our work, we aim to apply Transformer to spatial-temporal skeleton-based architectures, and in particular to joints representing the human skeleton, with the goal of modeling long-range interactions within human actions both in space, through a Spatial Self-Attention (SSA) module, and time, through a Temporal Self-Attention (TSA) module. Main contributions of this paper are summarized as follows:

- We propose a novel two-stream Transformer-based model for skeleton activity recognition tasks, employing *self-attention* on both the spatial and the temporal dimensions
- We design a *Spatial Self-Attention* (SSA) module to dynamically build links between skeleton joints, representing the relationships between human body parts, conditionally on the action and independently from the natural human body structure. On the temporal dimension, we introduce a *Temporal Self-Attention* (TSA) module to study the dynamics of a joint along time. We made both layers publicly available for experiments replication and further use [1]
- Our model outperforms ST-GCN (Yan et al. (2018)) and A-GCN (Shi et al. (2019b)) consistently improving backbone results on all datasets and achieving state-of-the-art performance when using joint information, and results on-par with state-of-the-art when bones information is used.

[1] Code at https://github.com/Chiaraplizz/ST-TR

2. Related Works

2.1. Skeleton-based Action Recognition

Most of the early studies in skeleton-based action recognition relied on handcrafted features (Hu et al. (2015); Vemulapalli et al. (2014); Hussein et al. (2013)) exploiting relative 3D rotations and translations between joints. Deep learning revolutionized activity recognition by proposing methods capable of increased robustness (Wang et al. (2019)) and able to achieve unprecedented performance. Methods that fall into this category rely on different aspects of skeleton data: (1) Recurrent neural network (RNN) based methods (Lev et al. (2016); Wang and Wang (2017); Liu et al. (2017b); Du et al. (2015)) leverage on the sequentiality of joint coordinates, treating input skeleton data as time series. (2) Convolutional neural network (CNN) based methods (Chéron et al. (2015); Simonyan and Zisserman (2014); Ding et al. (2017); Liu et al. (2017c); Li et al. (2017)) leverage spatial information, in a complementary way to RNN-based ones. Indeed, 3D skeleton sequences are mapped into a pseudo-image, representing temporal dynamics and skeleton joints respectively in rows and columns. (3) Graph neural network (GNN) based methods (Yan et al. (2018); Li et al. (2019); Shi et al. (2019b,a); Cheng et al. (2020); Liu et al. (2020)), make use of both spatial and temporal data by exploiting information contained in the natural topological graph structure of the human skeleton. These latter methods have demonstrated to be the most expressive among the three, and among these, the first model capturing the balance between spatial and temporal dependencies has been the Spatio-Temporal Graph Convolutional Network (ST-GCN) (Yan et al. (2018)).

In this work, we used ST-GCN as the baseline model; its functioning is presented in details in Section 3.2. Specifically, we propose to substitute regular graph convolutions on both space and time with the transformer self-attention operator. Cho et al. (2020) also proposed a Self-Attention Network (SAN) in which embeddings are extracted by segmenting the action sequence in temporal clips, and self-attention is applied among them in order to model long-term semantic information. However, since it applies self-attention on course-grained embeddings rather than skeleton joints, it hardly captures low-level joints' correlations *within* and *between* frames. Instead, by applying self-attention *directly* on nodes in the graph, both intra- and inter-frame, we efficiently model both spatial and temporal dependencies of the skeleton sequence.

2.2. Graph Neural Networks

Geometric deep learning (Bronstein et al. (2017)) refers to all emerging techniques attempting to generalize deep learning models to non-Euclidean domains such as graphs. The notion of Graph Neural Network (GNN) was initially outlined by Gori et al. (2005) and further elaborated by Scarselli et al. (2008).

The intuitive idea underlying GNNs is that nodes in a graph represent objects or concepts while edges represent their relationships. Due to the success of Convolutional Neural Networks, the concept of convolution has later been generalized from grid to graph data. GNNs iteratively process the graph, each time representing nodes as the result of applying a transformation to nodes' and their neighbors' features. The first formulation of CNNs on graphs is due to Bruna et al. (2014), who generalized convolution to signals using a *spectral* construction. This approach had computational drawbacks that have been subsequently addressed by Henaff et al. (2015) and Defferrard et al. (2016). The latter has been further simplified and extended by Kipf and Welling (2017). A complementary approach is the *spatial* one, where graph convolution is defined as information aggregation (Micheli (2009); Niepert et al. (2016); Such et al. (2017)). In this work we make use of the spectral construction proposed by Kipf and Welling (2017), whose formulation is provided in Section 3.2.

2.3. Transformers in Computer Vision

The Transformer is the leading neural model for Natural Language Processing (NLP), proposed by Vaswani et al. (2017) as an alternative to recurrent networks. It has been designed to face two key problems: (i) the processing of very long sequences, which are often intractable both for LSTMs and RNNs, and (ii) the limitations in parallelizing sentence processing, which is usually performed sequentially, word by word, in standard RNNs architectures. The Transformer follows a usual encoder-decoder structure, but it relies solely on *multi-head self-attention* (Vaswani et al. (2017)). Recently, Transformer self-attention has been applied in many popular computer vision tasks. Wang et al. (2018) proposed a differentiable non-local operator based on self-attention, which allows to capture long-range dependencies both in space and time for a more accurate video classification. After the first attempt of Bello et al. (2019a) to use self-attention as an alternative to convolutional operators, Dosovitskiy et al. (2020) proposed a Vision Transformer (ViT), which shows how Transformers can effectively replace standard convolutions on images. He et al. (2020) proposed a novel image transformer architecture for the image captioning task. Carion et al. (2020) made the first attempt to use a Transformer model to tackle detection problems, namely the Detection Transformer (DeTR). Zhao et al. (2020) proposed Point Transformer, a model which uses transformer self-attention to encode relations between point clouds, exploiting their permutation invariant nature. Other applications of Transformers in segmentation tasks (Huang et al. (2019)), multi-modal tasks (Lee et al. (2020)) and generative modeling (Oord et al. (2016); Parmar et al. (2018)) have been recently developed, showing the potential of Transformer models on a broad range of tasks.

3. Background

In this section, Spatial-Temporal Graph Convolutional Networks (ST-GCN) by Yan et al. (2018) and the original Transformer self-attention by Vaswani et al. (2017) are summarized, being the basic blocks of the model we propose in this paper.

3.1. Skeleton Sequences Representation

Given a sequence of skeletons, we define V as the number of joints representing each skeleton and T as the total number of skeletons composing the sequence, also named frames in the following. In order to represent the sequence, a spatial temporal graph is built, i.e., $G = (N, E)$, where $N = \{v_{ti} | t = 1, ..., T, i = 1, ..., V\}$ represents the set of all the nodes v_{ti} of the graph, i.e., the body joints of the skeleton along all the time sequence, and E represents the set of all the connections between nodes. E consists of two subsets; the first subset $E_S = \{(v_{ti}, v_{tj}) \mid i, j = 1, \ldots, V, t = 1, \ldots, T\}$ is composed by the intra-skeleton connections at each time interval t, for any pair of joints (i, j) connected by a bone in the human skeleton. The subset E_S of intra-skeleton connections is commonly further divided into K disjoint partitions, based on some criterion (Yan et al. (2018)) (e.g., distance from the center of gravity), and encoded using a set of adjacency matrices $\tilde{\mathbf{A}}_k \in \{0, 1\}^{V \times V}$. The second subset $E_T = \{(v_{ti}, v_{(t+1)i}) \mid i = 1, \ldots, V, t = 1, \ldots, T\}$ consists of all the inter-frame connections between joints along consecutive time frames. The result is a graph extending on both the spatial and the temporal dimension.

3.2. Spatial Temporal Graph Convolutional Networks

Spatial Temporal Graph Convolutional Networks (ST-GCN) have been introduced by Yan et al. (2018). A ST-GCN is structured as a hierarchy of stacked spatial-temporal blocks, which are internally composed of a spatial convolution (GCN) followed by a temporal convolution (TCN).

The spatial sub-module uses the Graph Convolution formulation proposed by Kipf and Welling (2017), which can be summarized as it follows:

$$\mathbf{f}_{out} = \sum_{k}^{K_s} (\mathbf{f}_{in} \mathbf{A}_k) \mathbf{W}_k, \tag{1}$$

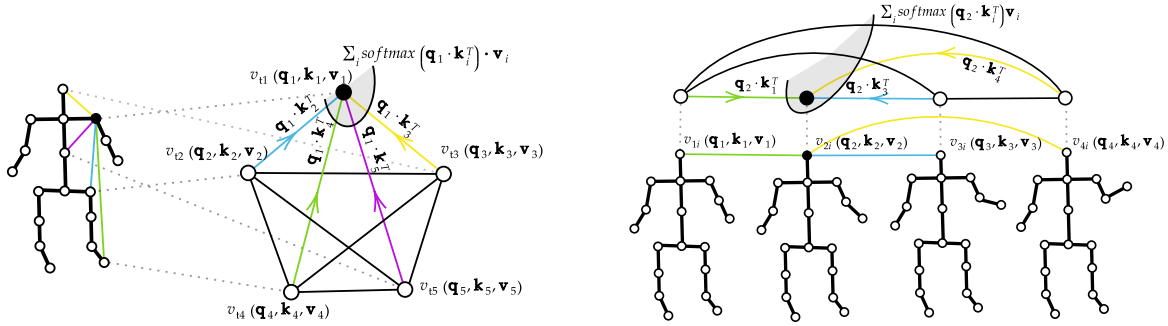
$$\mathbf{A}_k = \mathbf{D}_k^{-\frac{1}{2}} (\tilde{\mathbf{A}}_k + \mathbf{I}) \mathbf{D}_k^{-\frac{1}{2}}, D_{\mathrm{ii}} = \sum_{k}^{K_s} (\tilde{\mathbf{A}}_k^{ij} + \mathbf{I}_{\mathrm{ij}}), \tag{2}$$

where K_s is the kernel size on the spatial dimension, $\tilde{\mathbf{A}}_k$ is the adjacency matrix of the undirected graph representing intra-body connections, $\mathbf{I}$ is the identity matrix and $\mathbf{W}_k$ is a trainable weight matrix. The temporal convolution sub-module (TCN) is implemented as a $1 \times K_t$ 2D convolution operating on (V, T) dimensions of the (C_{in}, V, T) input volume, where K_t is the number of frames considered within the kernel receptive field.

As shown in Equation 1, the graph structure is predefined, being the adjacency matrix fixed. In order to make it *adapative*, Shi et al. (2019b) introduced the *Adaptive Graph Convolutional Network* (A-GCN), where the GCN formulation in Equation 1 is replaced by the following:

$$\mathbf{f}_{out} = \sum_{k}^{K_s} \mathbf{f}_{in} (\mathbf{A}_k + \mathbf{B}_k + \mathbf{C}_k) \mathbf{W}_k, \tag{3}$$

where $\mathbf{A}_k$ is the same as the one in Equation 1, $\mathbf{B}_k$ is learned during training, and $\mathbf{C}_k$ determines whether two vertices are connected or not through a similarity function.



(a) Spatial Self-Attention (b) Temporal Self-Attention

Fig. 2. Spatial Self-Attention (SSA) and Temporal Self-Attention (TSA). Self-attention operates on each pair of nodes, by computing a weight for each of them which represents the strength of their correlation. Those weights are then used to score the contribution of each body joint v_{ti}, proportionally to how relevant the node is w.r.t. to all the others. Please notice that on SSA (a), the procedure is illustrated only of a group of five nodes for simplicity, while in practice it operates on all the nodes.

3.3. Transformer Self-Attention

The original Transformer model of Vaswani et al. (2017) employs *self-attention*, i.e., a *non-local operator* originally designed to operate on words in NLP tasks with the goal of enriching the embedding of each word based on the surrounding context. In the Transformer, new word embeddings are computed by comparing pairs of words an then mixing their embeddings together based on how much a word is relevant w.r.t. the others. By gathering clues from the surrounding context, self-attention enables to extract a better meaning from each word, dynamically building relations within and between phrases.

In particular, for each word embedding $\mathbf{w}_i \in W = \{\mathbf{w}_1, ..., \mathbf{w}_n\}$, a query $\mathbf{q} \in \mathbb{R}^{d_q}$, a key $\mathbf{k} \in \mathbb{R}^{d_k}$ and a value vector $\mathbf{v} \in \mathbb{R}^{d_v}$ are computed through trainable linear transformations, independently. Then, a score for each word embedding is obtained by taking the dot product $\alpha_{ij} = \mathbf{q}_i \cdot \mathbf{k}_j^T \ \forall i, j = 1, ..., n$, where n is the total number of nodes being considered. This score represents how much the word j is relevant for word i. To compute the final embedding for word i, a weighted sum is computed by first multiplying the value vector of each other word $\mathbf{v}_j$ by the corresponding score α_{ij}, scaled through the softmax function, and then summing these vectors together. This process, also called *scaled dot-product attention*, can be written in matrix form as it follows:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_{\mathrm{k}}}}\right)\mathbf{V}, \qquad (4)$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are matrices containing the predicted query, key and value vectors, respectively, packed together and d_{k} is the channel dimension of the key vectors. The division by $\sqrt{d_{\mathrm{k}}}$ is performed in order to increase gradients stability during training. In order to obtain better performance, a mechanism called *multi-headed attention* is usually applied, which consists in applying attention, i.e., a head, multiple times with different learnable parameters and then finally combining the results.

4. Spatial Temporal Transformer Network

We propose the *Spatial Temporal Transformer (ST-TR)* network, an architecture which uses Transformer self-attention to operate on both space and time. We propose to achieve this goal using two modules, the *Spatial Self-Attention (SSA)* and the *Temporal Self-Attention (TSA)* modules, each one focusing on extracting correlations on one of the two dimensions.

4.1. Motivation

The idea behind the original Transformer self-attention is to allow the encoding of both short- and long-range correlations between words in the sentence. Our intuition is that the same approach can be applied to skeleton-based action recognition as well, as correlations between nodes are crucial both on the spatial and on the temporal dimension. We consider the joints comprising the skeleton as a bag-of-words and make use of the Transformer self-attention to extract node embeddings encoding the relation between surrounding joints, just like words in a phrase in NLP. Contrary to a standard graph convolution, where only the adjacent nodes are compared, we discard any predefined skeleton structure and instead let the Transformer self-attention automatically discover joint relations which are relevant for predicting the current action. The resulting operation acts similarly to a graph convolution, but in which the kernel values are *dynamically predicted* based on the discovered joint relations. The same idea is also applied at the sequence level, by analyzing how each joint changes during the action and building *long-range relations* that span different frames, similarly to how relations between phrases are built in NLP. The resulting operator is capable of obtaining a dynamical representation extending both on the spatial and the temporal dimension.

4.2. Spatial Self-Attention (SSA)

The Spatial Self-Attention module applies self-attention *inside each frame* to extract low-level features embedding the relations between body parts. This is achieved by computing correlations between each pair of joints in every single frame independently, as depicted in Figure 2a. Given the frame at time t, for each node v_{ti} of the skeleton, a *query* vector $\mathbf{q}_i^t \in \mathbb{R}^{dq}$, a *key* vector $\mathbf{k}_i^t \in \mathbb{R}^{dk}$ and a *value* vector $\mathbf{v}_i^t \in \mathbb{R}^{dv}$ are first computed by applying trainable linear transformations to the node features $\mathbf{n}_i^t \in \mathbb{R}^{C_{in}}$ with parameters $\mathbf{W}_q \in \mathbb{R}^{C_{in} \times dq}$, $\mathbf{W}_k \in \mathbb{R}^{C_{in} \times dk}$, $\mathbf{W}_v \in \mathbb{R}^{C_{in} \times dv}$, shared across all nodes. Then, for each pair of

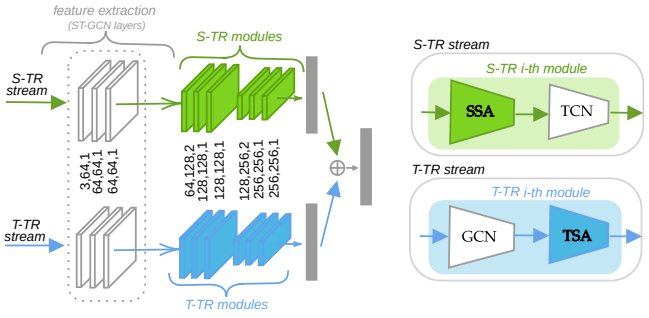


Fig. 3. Illustration of two 2s-ST-TR architecture. On each stream, the first three layers extract low level features through standard ST-GCN (Yan et al. (2018)) layers. At each successive layer, on the S-TR stream (coloured in green), SSA is used to extract spatial information, followed by a 2D convolution on time dimension (TCN), while on the T-TR stream (coloured in blue), TSA is used to extract temporal information, while spatial features are extracted by a standard graph convolution (GCN).

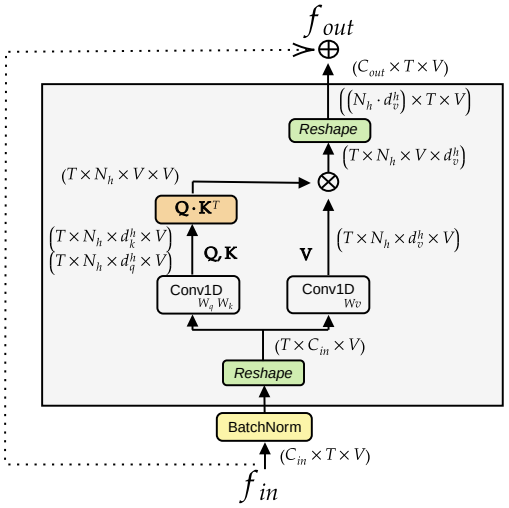


Fig. 4. Illustration of a SSA module (the implementation of TSA is the same, with the only difference that the dimension V corresponds to T and viceversa). The input f_{in} is reshaped by moving T in the batch dimension, such that self-attention operates on each time frame separately. SSA is implemented as a matrix multiplication, where Q, K and V are the query, key and value matrix respectively, and $\otimes$ denotes the matrix multiplication.

body nodes (v_{ti}, v_{tj}), a *query-key dot product* is applied to obtain a weight $\alpha_{ij}^t = \mathbf{q}_i^t \cdot \mathbf{k}_j^{t\,T} \in \mathbb{R}, \forall t \in T$ representing the correlation strength between the two nodes. The resulting score α_{ij}^t is used to weight each joint value $\mathbf{v}_j^t$, and a weighted sum is computed to obtain a new embedding $\mathbf{z}_i^t$ for node v_{ti}, as in the following:

$$\mathbf{z}_i^t = \sum_j softmax_j\left(\frac{\alpha_{ij}^t}{\sqrt{d_k}}\right)\mathbf{v}_j^t, \tag{5}$$

where $\mathbf{z}_i^t \in \mathbb{R}^{C_{out}}$ (with C_{out} the number of output channels) constitutes the new embedding of node v_{ti}.

Multi-head attention is applied by repeating this embedding extraction process N_h times, each time with a different set of learnable parameters. The set $(\mathbf{z}_{i_1}^t, ..., \mathbf{z}_{i_H}^t)$ of node embeddings thus obtained, all referring to the same node v_{ti}, is then combined with a learnable transformation, i.e., $concat(\mathbf{z}_{i_1}^t, ..., \mathbf{z}_{i_H}^t) \cdot \mathbf{W}_o$, and constitutes the output features of SSA.

As shown in Figure 2a, the relations between nodes (i.e., the α_{ij}^t scores) are dynamically *predicted* in SSA; the correlation structure in the skeleton is then not fixed for all the actions, but it changes adaptively for each sample. SSA operates similar to a graph convolution on a fully connected graph where, however, the kernel values (i.e., the α_{ij}^t scores) are predicted dynamically based on the skeleton pose.

4.3. Temporal Self-Attention (TSA)

With the Temporal Self-Attention (TSA) module, the dynamics of each joint is studied separately *along all the frames*, i.e., each single joint is considered as independent and correlations between frames are computed by comparing the change in the embeddings of the same body joint along the temporal dimension (see Figure 2b). The formulation is symmetrical to the one reported in Equation (5) for SSA:

$$\alpha_{tu}^v = \mathbf{q}_t^v \cdot \mathbf{k}_u^v \quad \forall v \in V, \quad \mathbf{z}_t^v = \sum_j softmax_u\left(\frac{\alpha_{tu}^v}{\sqrt{d_k}}\right)\mathbf{v}_u^v, \tag{6}$$

where v_{ti}, v_{ui} indicate the same joint v in two different instants t, u, $\alpha_{tu}^i \in \mathbb{R}$ is the correlation score, $\mathbf{q}_t^i \in \mathbb{R}^{dq}$ is the query associated to v_{ti}, $\mathbf{k}_u^i \in \mathbb{R}^{dk}$ and $\mathbf{v}_u^i \in \mathbb{R}^{dv}$ are the key and value associated to joint v_{ui} (all computed using trainable linear transformations as in SSA), and $\mathbf{z}_t^i \in \mathbb{R}^{C_{out}}$ is the resulting node embedding. Note that the notation used in this section is opposite w.r.t. the one used in Section 4.2; subscripts indicate time while superscripts indicate the joint. Multi-head attention is applied in TSA as in SSA. An example of TSA is depicted in Figure 2b.

The TSA module, by extracting inter-frame relations between nodes in time, can learn how to correlate frames apart from each other (e.g., nodes in the first frame with those in the last one), capturing discriminant features that are not otherwise possible to capture with a standard ST-GCN convolution, being this limited by the kernel size.

4.4. Two-Stream Spatial Temporal Transformer Network

To combine the SSA and TSA modules, a two-stream architecture named ST-TR is used, as similarly proposed by Shi et al. (2019b) and Shi et al. (2019a). In our formulation, the two streams differentiate on the way the proposed self-attention mechanisms are applied: SSA operates on the spatial stream (named S-TR), while TSA on the temporal one (named T-TR). On both streams, node features are first extracted by a three-layers residual network, where each layer processes the input on the spatial dimension through graph convolution (GCN), and on the temporal dimension through a standard 2D convolution (TCN), as done by Yan et al. (2018)[2]. SSA and TSA are then applied on the S-TR and on the T-TR streams in the subsequent layers in substitution to the GCN and TCN feature extraction modules respectively (Figure 3). The S-TR stream and T-TR

[2]In principle other features, e.g., visual features, could be added here but we want in this paper to focus on pure skeleton base action recognition and we leave this option for future investigations.

stream are end-to-end trained separately along with their corresponding feature extraction layers. The sub-networks outputs are eventually fused together by summing up their softmax output scores to obtain the final prediction, as proposed by Shi et al. (2019b) and Shi et al. (2019a).

Spatial Transformer Stream (S-TR). In the spatial stream, self-attention is applied at the skeleton level through a SSA module, which focuses on spatial relations between joints. The output of the SSA module is passed to a 2D convolutional module with kernel K_t on the temporal dimension (TCN), as done by Yan et al. (2018), in order to extract temporally relevant features, as shown in Figure 3 and expressed in the following:

$$\textbf{S-TR}(x) = Conv_{2D(1\times K_t)}(\textbf{SSA}(x)). \tag{7}$$

Following the original Transformer, the input is passes through a Batch Normalization layer (Ioffe and Szegedy (2015); Nguyen and Salazar (2019)), and skip connections are used to sum the input to the output of the SSA module (see Figure 4).

Temporal Transformer Stream (T-TR). The temporal stream, instead, focuses on discovering inter-frame temporal relations. Similarly to the S-TR stream, inside each T-TR layer, a standard graph convolution sub-module (Yan et al. (2018)) is followed by the proposed Temporal Self-Attention module:

$$\textbf{T-TR}(x) = \textbf{TSA}(GCN(x)). \tag{8}$$

TSA operates on graphs linking the same joint along all the time dimension (e.g., all left feet, or all right hands).

4.5. Implementation of SSA and TSA

The matrix implementation of SSA (and of TSA) is based on the implementation of Transformer on pixels by Bello et al. (2019b). As shown in Figure 4, given an input tensor of shape (C_{in}, T, V), where C_{in} is the number of input features, T is the number of frames and V is the number of nodes, a matrix $\mathbf{X}_V \in \mathbb{R}^{T\times C_{in}\times V}$ is obtained by rearranging the input. Here the T dimension is moved inside the batch dimension, effectively implementing parameter sharing along the temporal dimension and applying the transformation separately on each frame:

$$\begin{aligned} head_h(\mathbf{X}_V) &= Softmax\left(\frac{(\mathbf{X}_V\mathbf{W}_\mathrm{q})(\mathbf{X}_V\mathbf{W}_\mathrm{k})^\mathrm{T}}{\sqrt{d_k^h}}\right)(\mathbf{X}_V\mathbf{W}_v) \\ SelfAttention_V &= Concat(head_1, ..., head_{N_h})\mathbf{W}^\mathrm{o}, \end{aligned} \tag{9}$$

where the product with $\mathbf{W}_q \in \mathbb{R}^{C_{in}\times N_h\times d_q^h}$, $\mathbf{W}_k \in \mathbb{R}^{C_{in}\times N_h\times d_k^h}$ and $\mathbf{W}_v \in \mathbb{R}^{C_{in}\times N_h\times d_v^h}$ gives rise respectively to $\mathbf{Q} \in \mathbb{R}^{T\times N_h\times d_q^h\times V}$, $\mathbf{K} \in \mathbb{R}^{T\times N_h\times d_k^h\times V}$ and $\mathbf{V} \in \mathbb{R}^{T\times N_h\times d_v^h\times V}$, being N_h the number of heads, and $\mathbf{W}^o$ a learnable linear transformation combining the heads outputs. The output of the Spatial Transformer is then rearranged back into $\mathbb{R}^{C_{out}\times T\times V}$. The TSA matrix implementation has the same expression as Equation (9), differing only in the way the input $\mathbf{X}$ is processed. Indeed, in order to be processed by each TSA module, the input is reshaped into a matrix $\mathbf{X}_T \in \mathbb{R}^{V\times C_{in}\times T}$, where the V dimension has been moved in the first position and aggregated to the batch dimension, not reported here explicitly, in order to operate separately on each joint along the time dimension. The formulation is analogous to Equation (9), differing only in the shape of matrices, which become $\mathbf{Q} \in \mathbb{R}^{V\times N_h\times d_q^h\times T}$, $\mathbf{K} \in \mathbb{R}^{V\times N_h\times d_k^h\times T}$ and $\mathbf{V} \in \mathbb{R}^{V\times N_h\times d_v^h\times T}$.

5. Model Evaluation

To understand the impact of both the Spatial and Temporal Transformer streams, we analyze their performance separately and in different configurations through extensive experiments on NTU-RGB+D 60 (Shahroudy et al. (2016)) (see Table 1-3). Then, for a comparison with the state-of-the-art, we test the resulting best configurations on the Kinetics dataset (Kay et al. (2017)) and on the NTU-RGB+D 120 dataset (Liu et al. (2019)), which represents to date one of the most complex skeleton-based action recognition benchmarks (see Table 4-5).

5.1. Datasets

NTU RGB+D 60 and NTU RGB+D 120. The NTU RGB+D 60 (NTU-60) dataset is a large-scale benchmark for 3D human action recognition collected using Microsoft Kinect v2 by Shahroudy et al. (2016). Skeleton information consists of 3D coordinates of 25 body joints and a total of 60 different action classes. The NTU-60 dataset follows two different criteria for evaluation. The first one, called *Cross-View Evaluation* (X-View), uses 37, 920 training and 18, 960 test samples, split according to the camera views from which the action is taken. The second one, called *Cross-Subject Evaluation* (X-Sub), is composed instead of 40, 320 training and 26, 560 test samples. Data collection has been performed with 40 different subjects performing actions and divided into two groups, one for training and the other for testing. NTU RGB+D 120 (Liu et al. (2019)) (NTU-120) is an extension of NTU-60, which adds 57, 367 new skeleton sequences representing 60 new actions. To perform the evaluation, the extended dataset follows two criteria: the first one is the *Cross-Subject Evaluation (X-Sub)*, the same used for NTU-60, while the second one is called *Cross-Setup Evaluation (X-Set)*, which substitutes Cross-View by splitting training and testing samples based on the parity of the camera setup IDs.

Kinetics. The Kinetics skeleton dataset (Yan et al. (2018)) is obtained by extracting skeleton annotations from videos composing the Kinetics 400 dataset (Kay et al. (2017)), by using the OpenPose toolbox (Cao et al. (2019)). It consists of 240, 436 training and 19, 796 testing samples, representing a total of 400 action classes. Each skeleton is composed by 18 joints, each one provided with the 2D coordinates and a confidence score. For each frame, a maximum of 2 people are selected based on the highest confidence scores.

5.2. Model Complexity

We perform an analysis on the complexity of the different self-attention modules we designed, and compare them to ST-GCN modules (Yan et al. (2018)), based on standard convolution, and to 1s-AGCN (Shi et al. (2019b)) modules, based on adaptive graph convolution. First, we compare in Figure 5a,

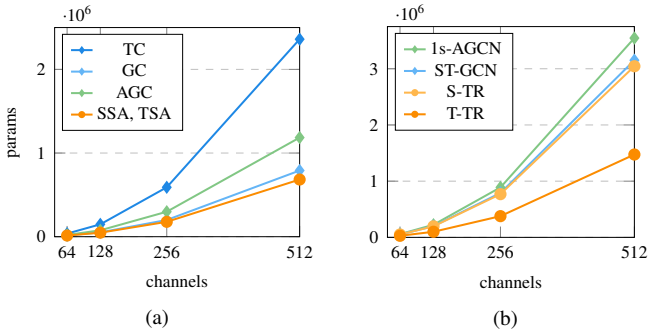


Fig. 5. (a) Difference in terms of parameters between Graph Convolution (GC), Adaptive Convolution (AGC), Spatial Self-Attention (SSA) modules of $C_{in} = C_{out}$ channels, and between Temporal Convolution (TC) and Temporal Self-Attention (TSA) modules; (b) parameters comparison between ST-GCN, 1s-AGCN and our novel S-TR and T-TR. Best viewed in colors.

singularly, a layer of standard convolution with our transformer mechanism, setting $C_{in} = C_{out}$ channels. This results in the same number of parameters for both TSA and SSA, since the convolutions performed internally have the same kernel dimensions and both the query-key dot product and the logit-value product are parameter free. It can be seen that SSA introduces less parameters than GC, especially when dealing with a large number of channels, where the maximum Δ_{GC-SSA}, i.e., the decrease in terms of parameters, is 1.1×10^5. When dealing with adaptive modules (AGC), an additional number of parameters has to be considered, resulting in a difference with respect to SSA of $\Delta_{AGC-SSA} = 5 \times 10^5$. On the temporal dimension Δ_{TC-TSA} reaches a value of 16.8×10^5. Temporal convolution in Yan et al. (2018) is implemented as a $2D$ convolution with filter $1 \times F$, where F is the number of frames considered along the time dimension, and it is usually set to 9, striding along $T = 300$ frames. Thus, substituting it with a self-attention mechanism results in a great complexity reduction, in addition to better performance, as reported in the next sections.

Finally, in Figure 5b we also compare the entire stream architectures, i.e., ST-GCN (Yan et al. (2018)) and 1s-AGCN (Shi et al. (2019b)) with the proposed S-TR and T-TR streams in terms of parameters. As expected from the considerations above, the biggest improvement in parameters reduction is achieved by substituting temporal convolution with TSA, i.e., in T-TR, with a $\Delta_{ST-GCN--T-TR} = 16.7 \times 10^5$. On the spatial dimension the difference in terms of parameters is not as pronounced as in temporal dimension, but it is still significant, with a $\Delta_{ST-GCN--S-TR} = 1.07 \times 10^5$ and $\Delta_{1s-AGCN--S-TR} = 5.0 \times 10^5$.

5.3. Experimental Settings

Using PyTorch (Paszke et al. (2019)) framework, we trained our models for a total of 120 epochs with batch size 32 and SGD as optimizer on NTU-60 and NTU-120, while on Kinetics we trained our models for a total of 65 epochs, with batch size 128. The learning rate is set to 0.1 at the beginning and then reduced by a factor of 10 at the epochs {60, 90} and {45, 55} for NTU and Kinetics respectively. These schedulings have been selected as they have been shown to provide good results on ST-GCN networks used by Shi et al. (2019a). When using adaptive AGCN modules, we performed a linear warmup

Table 1. Comparison between the baseline and our self-attention modules in terms of both performance (accuracy (%)) and efficiency (number of parameters) on NTU-60 (X-View)

Method	GCN	TCN	Params [$\times 10^5$]	Top-1
ST-GCN	✓	✓	31.0	92.7
ST-GCN-fc	$\mathbf{A}_{fc}$	✓	26.5	93.7
1s-AGCN	$\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k$	✓	34.7	93.7
1s-AGCN w/o A	$\mathbf{B}_k, \mathbf{C}_k$	✓	33.1	93.4
S-TR	SSA	✓	30.7	**94.0**
T-TR	✓	TSA	17.6	93.6

of the learning rate during the first epoch. Moreover, we preprocessed the data with the same procedure used by Shi et al. (2019b) and Shi et al. (2019a). In order to avoid overfitting, we also used *DropAttention*, a particular dropout technique introduced by Zehui et al. (2019) for regularizing attention weights in Transformer networks, that consists in randomly dropping columns of the attention logits matrix. In all of these experiments, the *number of heads* for multi-head attention is set to 8, and d_q, d_k, d_v embedding dimensions to $0.25 \times C_{out}$ in each layer, as done in Bello et al. (2019b). We did not perform grid search on these parameters. As far as it concerns the model architecture, each stream is composed by 9 layers, of channel dimension 64, 64, 64, 128, 128, 128, 256, 256 and 256. Batch normalization is applied to input coordinates, a global average pooling layer is applied before the *softmax* classifier and each stream is trained using the standard cross-entropy loss.

5.4. Results

To verify in a fair way the effectiveness of our SSA and TSA modules, we compare the S-TR and T-TR streams individually against the ST-GCN (Yan et al. (2018)) baseline (whose results are reported using our learning rate scheduling) and other models that modify its basic GCN module (see Table 1): (i) *ST-GCN (fc)*: we implemented a version of ST-GCN whose adjacency matrix is composed of all ones (referred as $\mathbf{A}_{fc}$), to simulate the fully-connected skeleton structure underlying our SSA module and verify the superiority of self-attention over graph convolution on the spatial dimension; (ii) *1s-AGCN*: Adaptive Graph Convolutional Network (AGCN) (Shi et al. (2019b)) (see Section 3.2), as it demonstrated in the literature to be more robust than standard ST-GCN, in order to remark the robustness of our SSA module over more recent methods; (iii) *1s-AGCN w/o A*: 1s-AGCN without the static adjacency matrix, to verify the effectiveness of our SSA over graph convolution in a similar setting where all the links between joints are exclusively learnt. All these methods use the same implementation of convolution on the temporal dimension (TCN). We make a comparison both in terms of model accuracy and number of parameters.

Regarding SSA, the performance of S-TR is superior to all methods mentioned above, demonstrating that self-attention can be used in place of graph convolution, increasing the network performance while also decreasing the number of parameters. In fact, as it can be seen from Table 1, S-TR introduces 0.3×10^5 parameters less then ST-GCN and 4×10^5 less than 1s-AGCN, with a performance increment w.r.t. all GCN configu-

Table 2. a) Comparison of S-TR and T-TR streams, and their combination (ST-TR) on NTU-60, w and w/o bones. b) Ablations on different model configurations

Method	Bones	X-Sub	X-View
S-TR		86.4	94.0
T-TR		86.0	93.6
T-TR-agcn		86.9	94.7
ST-TR		88.7	95.6
ST-TR-agcn		**89.2**	**95.8**
S-TR	✓	87.9	94.9
T-TR	✓	87.3	94.1
T-TR-agcn	✓	88.6	94.7
ST-TR	✓	89.9	96.1
ST-TR-agcn	✓	**90.3**	**96.3**

(a)

Ablation	X-View
S-TR-all-layers	93.3
T-TR-all-layers	91.3
ST-TR-all-layers	95.0
S-TR-augmented	94.5
T-TR-augmented	90.2
ST-TR-augmented	94.9
ST-TR-1s	93.3
ST-TR ($k = 0$)	95.4
ST-TR ($k = 1$)	95.6
ST-TR ($k = 2$)	95.7

(b)

rations. Similarly, regarding TSA, what emerges from the comparison between T-TR and the ST-GCN baseline adopting standard convolution, is that by using self-attention on the temporal dimension the model is significantly lighter (13.4×10^5 less parameters), and achieves an increment in accuracy of 0.9%.

In Table 2 we first analyze the performance of the S-TR stream, T-TR stream and their combination by using input data consisting of joint information only. As it can be seen from Table 2a, on NTU-60 the S-TR stream achieves slightly better performance (+0.4%) than the T-TR stream, on both X-View and X-Sub. This can be motivated by the fact that SSA in S-TR operates on 25 joints only, while on temporal dimension the number of correlations is proportional to the huge number of frames. Again, as shown in Table 1a, applying self-attention instead of convolution clearly benefits the model on both spatial and temporal dimensions. The combination of the two streams achieves 88.7% of accuracy on X-Sub and 95.6% of accuracy on X-View, outperforming the baseline ST-GCN and surpassing other two-stream architectures (see Table 3).

As adding the differential of spatial coordinates (bones information) demonstrated to lead to better results in previous works (Shi et al. (2019a); Simonyan and Zisserman (2014)), we also studied our Transformer modules on combined joint and bones information. For each node $\mathbf{v}_1 = (\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_1)$ and $\mathbf{v}_2 = (\mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_2)$, the bone connecting the two is calculated as $\mathbf{b}_{\mathbf{v}_1,\mathbf{v}_2} = (\mathbf{x}_2 - \mathbf{x}_1, \mathbf{y}_2 - \mathbf{y}_1, \mathbf{z}_2 - \mathbf{z}_1)$. Both joint and bone information are concatenated along the channel dimension, and then fed to the network. At each layer, the dimension of the input and output channels are doubled as done by Shi et al. (2019a) and Simonyan and Zisserman (2014). Results are shown again in Table 2a, where all previous configurations improve when bones information is added as input. This highlights the flexibility of our method, which is capable of adapting to different input types and network configurations.

To further test its flexibility, we also perform experiments in which the GCN module is substituted by the AGCN adaptive module on the temporal stream. As it can be seen from Table 2a, these configurations (*T-TR-agcn*) achieve better results than the one using standard GCN on both X-Sub and X-View.

5.5. Effect of Applying Self-Attention since Feature Extraction

We designed our streams to operate starting from high-level features, rather than directly from coordinates, extracted using a sequence of residual GCN and TCN modules as reported in Section 4.4. This set of experiments validates our design choice. In these experiments SSA (TSA) substitutes GCN (TCN) on the S-TR (T-TR) stream, from the very first layer. The configurations reported in Table 2b (named *S-TR-all-layers*), performs worse than the corresponding ones in Table 2a, while still outperforming the baseline ST-GCN (Shi et al. (2019a)) (see Table 3). Indeed, self-attention has demonstrated being more efficient when incorporated in later stages of the network (Carion et al. (2020); Huang et al. (2019); Wang et al. (2018)). Notice that on T-TR, in order to deal with the great number of frames in the very first layers ($T = 300$), we divided them into blocks within which SSA is applied, and then gradually reduce the number of blocks ($d_{block} = 10$ where $C_{out} = 64$, $d_{block} = 10$ where $C_{out} = 128$, and a single block of $d_{block} = T^l$ on layers l with $C_{out} = 256$).

The standard protocol used in recent works (Cheng et al. (2020); Shi et al. (2019b)) that propose alternative modules for ST-GCN based networks is to keep the original ST-GCN backbone architecture fixed in terms of layers composition. Following these works, we kept the three original feature extraction layers for a fair comparison. We further conduct some targeted experiments in which we vary their number k (Table 2b). As it can be seen, performance is not sensitive to variations of k, confirming the effectiveness of the proposed approach.

5.6. Effect of Augmenting Convolution with Self-Attention

Motivated by the results in Bello et al. (2019b), we studied the effect of applying the proposed Transformer mechanism as an augmentation procedure to the original ST-GCN modules. In this configuration, $0.75 \times C_{out}$ features result from GCN (TCN) and they are concatenated to the remaining $0.25 \times C_{out}$ features from SSA (TSA), a setup that has proven to be effective in Bello et al. (2019b). To compensate the reduction of attention channels, *wide attention* is used, i.e., half of the attention channels are assigned to each head, then recombined together while merging heads. The results are reported in Table 2b (referred as *ST-TR-augmented*). Graph convolution is the one that benefits the most from SSA attention (S-TR-augmented, 94.5%), to be compared with S-TR's 94% in Table 2a. Nevertheless, the lower number of output features assigned to self-attention prevent temporal convolution improving on T-TR stream.

5.7. Effect of combining SSA and TSA in a single stream

We tested the efficiency of the model when SSA and TSA are combined in a single stream architecture (see Table 2b, referred as *S-TR-1s*). In this configuration, feature extraction is still performed by the original GCN and TCN modules, while from the 4th layer on, each layer is composed by SSA followed by TSA, i.e., **ST-TR-1s**(x) = **TSA**(**SSA**(x)).

We also tested this configuration on NTU-60, obtaining an accuracy of 93.3%, slightly lower than the 95.6% accuracy of 2s-ST-TR (see Table 1a, ST-TR). However, it should be noted

Table 3. Comparison with state-of-the-art accuracy (%) on NTU-60. Best for both configurations w/ and w/o bones in bold.

NTU-60			
Method	**Bones**	X-Sub	X-View
STA-LSTM (Song et al. (2017))		73.4	81.2
VA-LSTM (Zhang et al. (2017))		79.4	87.6
AGC-LSTM (Si et al. (2019))		89.2	95.0
ST-GCN (Yan et al. (2018))		81.5	88.3
AS-GCN (Li et al. (2019))		86.8	94.2
1s-AGCN (Shi et al. (2019b))		86.0	93.7
SAN (Cho et al. (2020))		87.2	92.7
1s Shift-GCN (Cheng et al. (2020))		87.8	95.1
ST-TR (Ours)		88.7	95.6
ST-TR-agcn (Ours)		**89.2**	**95.8**
2s-AGCN (Shi et al. (2019b))	✓	88.5	95.1
DGNN (Shi et al. (2019a))	✓	89.9	96.1
2s Shift-GCN (Cheng et al. (2020))	✓	89.7	96.0
4s Shift-GCN (Cheng et al. (2020))	✓	90.7	96.5
MS-G3D (Liu et al. (2020))	✓	**91.5**	96.2
ST-TR (Ours)	✓	89.9	96.1
ST-TR-agcn (Ours)	✓	90.3	**96.3**

Table 4. Comparison with state-of-the-art accuracy (%) of S-TR, T-TR, and their combination (ST-TR) on NTU-120. Best for both configurations w/ and w/o bones in bold.

NTU-120			
Method	Bones	X-Sub	X-Set
ST-LSTM (Liu et al. (2016))		55.7	57.9
GCA-LSTM (Liu et al. (2017a))		61.2	63.3
RotClips+MTCNN (Ke et al. (2018))		62.2	61.8
Pose Evol. Map (Liu and Yuan (2018))		64.6	66.9
1s Shift-GCN (Cheng et al. (2020))		80.9	83.2
S-TR (Ours)		78.6	80.7
T-TR (Ours)		78.4	80.5
T-TR-agcn (Ours)		80.1	82.1
ST-TR (Ours)		81.9	84.1
ST-TR-agcn (Ours)		**82.7**	**85.0**
2s-AGCN (Shi et al. (2019b))	✓	82.9	84.9
2s Shift-GCN (Cheng et al. (2020))	✓	85.3	86.6
4s Shift-GCN (Cheng et al. (2020))	✓	85.9	87.6
MS-G3D (Liu et al. (2020))	✓	**86.9**	**88.4**
S-TR (Ours)	✓	81.0	83.6
T-TR (Ours)	✓	80.4	83.0
T-TR-agcn (Ours)	✓	82.7	84.9
ST-TR (Ours)	✓	84.3	86.7
ST-TR-agcn (Ours)	✓	85.1	87.1

that S-TR-1s presents 17.4×10^5 parameters, drastically reducing the complexity of the baseline ST-GCN which consists in 31×10^5 parameters. Moreover, it outperforms the ST-GCN baseline by 0.6% using half of the parameters.

Table 5. Comparison with state-of-the-art accuracy (%) of S-TR, T-TR, and their combination (ST-TR) on Kinetics. Best for both configurations w/ and w/o bones in bold.

Kinetics			
Method	**Bones**	Top-1	Top-5
ST-GCN (Yan et al. (2018))		30.7	52.8
AS-GCN (Li et al. (2019))		34.8	56.5
SAN (Cho et al. (2020))		35.1	55.7
S-TR (Ours)		32.4	55.3
T-TR (Ours)		32.4	55.2
T-TR-agcn (Ours)		34.4	57.1
ST-TR (Ours)		34.5	57.6
ST-TR-agcn (Ours)		**36.1**	**58.7**
2s-AGCN (Shi et al. (2019b))	✓	36.1	58.7
DGNN (Shi et al. (2019a))	✓	36.9	59.6
MS-G3D (Liu et al. (2020))	✓	38.0	**60.9**
S-TR (Ours)	✓	35.4	57.9
T-TR (Ours)	✓	33.1	55.86
T-TR-agcn (Ours)	✓	34.7	56.4
ST-TR (Ours)	✓	37.0	59.7
ST-TR-agcn (Ours)	✓	**38.0**	60.5

6. Comparison with State-Of-The-Art Results

In addition to NTU-60, we compare our methods on NTU-120 and Kinetics. For a fair comparison, we compare the ST-TR configurations on methods trained on the same input data (either with joint information only, or both joint and bones information). On NTU-60 (Table 3), the proposed ST-TR, when using joint information only, outperforms all the state-of-the-art models using the same type of information. In particular, it outperforms SAN (Cho et al. (2020)), another method employing self-attention in skeleton-based action recognition, by up to 3%. When using bones information, the proposed transformer based architecture outperforms 2s-AGCN (Shi et al. (2019b)) on both X-Sub and X-Set and our best configuration making use of the AGCN backbone (ST-TR-agcn) reaches performance on-par with state-of-the-art. We further compare ST-TR against 4s Shift-GCN (Cheng et al. (2020)) which, in addition to the joint and bones streams, also comprises two extra streams making use of additional temporal information, and DGNN (Shi et al. (2019a)), which also makes use of motion information besides joint and bones. We report 4s Shift-GCN and DGNN in Table 3-5 with a different color to highlight the difference in the input. ST-TR-agcn outperforms DGNN on both X-Sub and X-View and crucially, although 4s Shift-GCN uses extra input data and combines two additional streams, ST-TR-agcn still achieves on-par results but with a simpler design.

On NTU-120 (Table 4), the model only based on joints outperforms all state-of-the-art methods that use the same information. When adding bones, both ST-TR and ST-TR-agcn outperform 2s-AGCN by up to 3% on both X-Sub and X-Set. Moreover, ST-TR-agcn's results are on-par with 2s Shift-GCN (Cheng et al. (2020)) on X-Sub, while they improve on X-Set. Our network has only slightly lower performance than MS-G3D (Liu et al. (2020)), which represents to date a very

strong baseline in skeleton-based action recognition. Considering that MS-G3D features a multi-path design with multi-scale graph convolutions, the performance obtained by ST-TR is remarkable given that the latter is based on a simpler backbone. Finally, on Kinetics (Table 5), our model using only joints outperforms the ST-GCN baseline by 5% and all previous methods using only joint information. When bones information is added, it outperforms both 2s-AGCN and DGNN, and achieves results on-par with the very recent state-of-the-art method MS-G3D.

7. Qualitative Results

In Figure 6, we report some actions and the corresponding Spatial Self-Attention maps. On the top we draw the skeleton of the subjects, where the radius of the circles in correspondence to each joint is proportional its relevance predicted by the self-attention. The heatmaps on the bottom represent the attention scores of the last layer; these are 25×25 matrices, where each row and each column represents a body joint. An element in position (i, j) represents the predicted correlation between joint i and joint j in the same frame. As it can be observed, depending on the action, different parts of the body are activated. In Figure 7 are shown the same heatmaps *at each layer*. In the first layers, self-attention captures *low-level* correlations between body joints, as highlighted by the sparsity of the activations. While going deeper through the network, the *global* importance of each node emerges instead, as highlighted by the vertical lines corresponding to the most relevant joints.

8. Conclusions

In this paper we propose a novel approach that introduces Transformer self-attention in skeleton activity recognition as an alternative to graph convolution. Through extensive experiments on NTU-60, NTU-120 and Kinetics, we demonstrated that our Spatial Self-Attention module (SSA) can replace graph convolution, enabling more flexible and dynamic representations. Similarly, Temporal Self-Attention module (TSA) overcomes the strict locality of standard convolution, enabling the extraction of long-range dependencies across the action. Moreover, our final Spatial-Temporal Transformer network (ST-TR) achieves state-of-the-art performance on all dataset w.r.t. methods using same input joint information and stream setup, and results on-par with state-of-the-art methods when bones information is added. As configurations only involving self-attention modules revealed to be sub-optimal, a possible future work is to search for a unified Transformer architecture able to replace graph convolution in a variety of tasks.

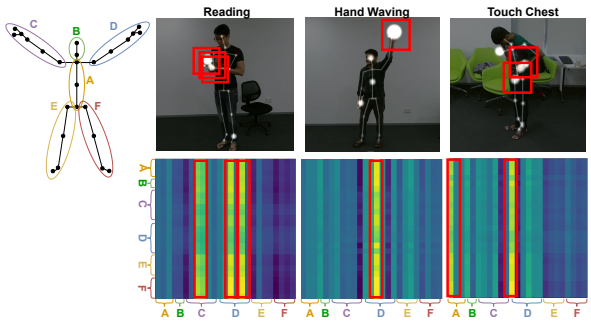


Fig. 6. Skeleton of the subjects performing the action (top) and the corresponding SSA heatmaps (bottom). We display with red boxes the joints which the network identifies as the most relevant, while the corresponding spatial self-attention scores are highlighted in the attention maps.

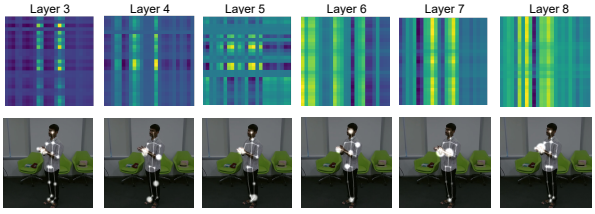


Fig. 7. Layer visualization. Each column represent the k-th layer, along with the corresponding spatial self-attention heatmap on top, and the skeleton of the subjects performing the action on the bottom.

References

Aggarwal, J.K., Ryoo, M.S., 2011. Human activity analysis: A review. ACM Computing Surveys (CSUR) 43, 1–43.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V., 2019a. Attention augmented convolutional networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V., 2019b. Attention augmented convolutional networks. Proceedings of the IEEE International Conference on Computer Vision , 3286–3295.

Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P., 2017. Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine 34, 18–42.

Bruna, J., Zaremba, W., Szlam, A., Lecun, Y., 2014. Spectral networks and locally connected networks on graphs. International Conference on Learning Representations (ICLR2014), CBLS, April 2014 .

Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A., 2019. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer. pp. 213–229.

Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H., 2020. Skeleton-based action recognition with shift graph convolutional network. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition , 183–192.

Chéron, G., Laptev, I., Schmid, C., 2015. P-cnn: Pose-based cnn features for action recognition. Proceedings of the IEEE international conference on computer vision , 3218–3226.

Cho, S., Maqbool, M., Liu, F., Foroosh, H., 2020. Self-attention network for skeleton-based human action recognition, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 635–644.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems , 3844–3852.

Ding, Z., Wang, P., Ogunbona, P.O., Li, W., 2017. Investigation of different skeleton features for cnn-based 3d action recognition. 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) , 617–622.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 .

Du, Y., Wang, W., Wang, L., 2015. Hierarchical recurrent neural network for skeleton based action recognition. Proceedings of the IEEE conference on computer vision and pattern recognition , 1110–1118.

Gori, M., Monfardini, G., Scarselli, F., 2005. A new model for learning in graph domains. Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. 2, 729–734.

He, S., Liao, W., Tavakoli, H.R., Yang, M., Rosenhahn, B., Pugeault, N., 2020.

Image captioning through image transformer, in: Proceedings of the Asian Conference on Computer Vision.

Henaff, M., Bruna, J., LeCun, Y., 2015. Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 .

Hu, J.F., Zheng, W.S., Lai, J., Zhang, J., 2015. Jointly learning heterogeneous features for rgb-d activity recognition. Proceedings of the IEEE conference on computer vision and pattern recognition , 5344–5352.

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W., 2019. Ccnet: Criss-cross attention for semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 603–612.

Hussein, M.E., Torki, M., Gowayyed, M.A., El-Saban, M., 2013. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. Twenty-third international joint conference on artificial intelligence .

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 .

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

Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F., 2018. Learning clip representations for skeleton-based 3d action recognition. IEEE Transactions on Image Processing 27, 2842–2855.

Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A., Bhowmik, A., 2017. Intel realsense stereoscopic depth cameras, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–10.

Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. 5th International Conference on Learning Representations, ICLR .

Lee, S., Yu, Y., Kim, G., Breuel, T., Kautz, J., Song, Y., 2020. Parameter efficient multimodal transformers for video representation learning. arXiv preprint arXiv:2012.04124 .

Lev, G., Sadeh, G., Klein, B., Wolf, L., 2016. Rnn fisher vectors for action recognition and image annotation. European Conference on Computer Vision , 833–850.

Li, B., Dai, Y., Cheng, X., Chen, H., Lin, Y., He, M., 2017. Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn. ICMEW .

Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q., 2019. Actional-structural graph convolutional networks for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3595–3603.

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

Liu, J., Shahroudy, A., Xu, D., Wang, G., 2016. Spatio-temporal lstm with trust gates for 3d human action recognition. European conference on computer vision , 816–833.

Liu, J., Wang, G., Duan, L.Y., Abdiyeva, K., Kot, A.C., 2017a. Skeleton-based human action recognition with global context-aware attention lstm networks. IEEE Transactions on Image Processing 27, 1586–1599.

Liu, J., Wang, G., Hu, P., Duan, L.Y., Kot, A.C., 2017b. Global context-aware attention lstm networks for 3d action recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition , 1647–1656.

Liu, M., Liu, H., Chen, C., 2017c. Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognition 68, 346–362.

Liu, M., Yuan, J., 2018. Recognizing human actions as the evolution of pose estimation maps. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition , 1159–1168.

Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W., 2020. Disentangling and unifying graph convolutions for skeleton-based action recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition , 143–152.

Micheli, A., 2009. Neural network for graphs: A contextual constructive approach. IEEE Transactions on Neural Networks 20, 498–511.

Nguyen, T.Q., Salazar, J., 2019. Transformers without tears: Improving the normalization of self-attention. arXiv preprint arXiv:1910.05895 .

Niepert, M., Ahmed, M., Kutzkov, K., 2016. Learning convolutional neural networks for graphs. International conference on machine learning , 2014–2023.

Oord, A.v.d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., Kavukcuoglu, K., 2016. Conditional image generation with pixelcnn decoders. arXiv preprint arXiv:1606.05328 .

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D., 2018. Image transformer, in: International Conference on Machine Learning, PMLR. pp. 4055–4064.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library, pp. 8026–8037.

Ren, B., Liu, M., Ding, R., Liu, H., 2020. A survey on 3d skeleton-based action recognition using learning method. arXiv preprint arXiv:2002.05907 .

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G., 2008. The graph neural network model. IEEE Transactions on Neural Networks 20, 61–80.

Shahroudy, A., Liu, J., Ng, T.T., Wang, G., 2016. Ntu rgb+d: A large scale dataset for 3d human activity analysis. Proceedings of the IEEE conference on computer vision and pattern recognition , 1010–1019.

Shi, L., Zhang, Y., Cheng, J., Lu, H., 2019a. Skeleton-based action recognition with directed graph neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition , 7912–7921.

Shi, L., Zhang, Y., Cheng, J., Lu, H., 2019b. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition , 12026–12035.

Si, C., Chen, W., Wang, W., Wang, L., Tan, T., 2019. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. Proceedings of the IEEE conference on computer vision and pattern recognition , 1227–1236.

Simonyan, K., Zisserman, A., 2014. Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems , 568–576.

Song, S., Lan, C., Xing, J., Zeng, W., Liu, J., 2017. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence , 4263–4270.

Such, F.P., Sah, S., Domínguez, M., Pillai, S., Zhang, C., Michael, A., Cahill, N.D., Ptucha, R.W., 2017. Robust spatial filtering with graph convolutional neural networks. IEEE Journal of Selected Topics in Signal Processing .

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems , 5998–6008.

Vemulapalli, R., Arrate, F., Chellappa, R., 2014. Human action recognition by representing 3d skeletons as points in a lie group. Proceedings of the IEEE conference on computer vision and pattern recognition , 588–595.

Wang, H., Wang, L., 2017. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition , 499–508.

Wang, L., Huynh, D.Q., Koniusz, P., 2019. A comparative review of recent kinect-based action recognition algorithms. IEEE Transactions on Image Processing 29, 15–28.

Wang, X., Girshick, R., Gupta, A., He, K., 2018. Non-local neural networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7794–7803. doi:10.1109/CVPR.2018.00813.

Yan, S., Xiong, Y., Lin, D., 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. Thirty-second AAAI conference on artificial intelligence .

Zehui, L., Liu, P., Huang, L., Fu, J., Chen, J., Qiu, X., Huang, X., 2019. Dropattention: A regularization method for fully-connected self-attention networks. arXiv preprint arXiv:1907.11065 .

Zhang, H.B., Zhang, Y.X., Zhong, B., Lei, Q., Yang, L., Du, J.X., Chen, D.S., 2019. A comprehensive survey of vision-based human action recognition methods. Sensors 19, 1005.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N., 2017. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. Proceedings of the IEEE International Conference on Computer Vision , 2117–2126.

Zhang, Z., 2012. Microsoft kinect sensor and its effect. IEEE multimedia 19, 4–10.

Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V., 2020. Point transformer. arXiv preprint arXiv:2012.09164 .