Context-Aware Skeleton-based Action Recognition via Spatial and Temporal Transformer Networks

Sharan Ramjee
sramjee@stanford.edu

Sofian Zalouk
szalouk@stanford.edu

Abstract

Human action recognition is an important task in video understanding, with applications ranging from robotics to autonomous driving. Among the variety of approaches, Graph Learning based human action recognition has become extremely popular for its ability to simultaneously learn spatial and temporal patterns from data, in addition to its greater expressive power and stronger generalization capability in comparison to other methods. In particular, Spatial Temporal Graph Convolution Networks (ST-GCN) [1] use a set of spatial and temporal graph convolutions on the human skeleton sequences in order to achieve state-of-the-art performance on on the NTU RGB+D 60 dataset, which is the largest in-house captured benchmark for 3D human action recognition. 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 incorporating contextual information from the video frames. In this paper, we propose a novel Spatial-Temporal Context-aware Transformer Network (ST-CTR), which models dependencies between joints using the Transformer self-attention operator while incorporating contextual information from the human action recognition video frames. Through our extensive experiments, we demonstrate the improved action recognition performance of ST-CTR on the NTU RGB+D 60 dataset in comparison to other state-of-the-art human action recognition methods. The source code is available on GitHub: `https://github.com/sharanramjee/st-ctr`.

1. Introduction

Human action recognition is a crucial task to video understanding, with vast applications in autonomous driving and robotics [2]. Several modalities (such as depth maps, optical flow, skeletons) have been explored to perform human action recognition. Among these, skeleton-based human action recognition remains relatively less explored, only recently gaining popularity due to the recent advances of graph based Deep Learning methods, such as Graph Convolutions Networks (GCNs) [3]. By modelling the dynamics of joint positions over time, skeleton-based methods are a much more natural representation of the video input data; they allow for the training of models with better discriminative capabilities by analyzing the motion of the human skeleton. Furthermore, by significantly reducing the dimensionality of the input video data, skeleton-based methods allow for faster, less computationally expensive inference, which is highly desirable for real-time applications such as autonomous driving.

Given the advantages of skeleton-based action recognition, we aim to explore designs which extend and combine elements from different state-of-the-art models. For one, we aim to explore GCNs to spatially analyze skeletons and product rich feature representations. Moreover, given their recent success, we aim to use transformers [4] in order to model long and short term relationships in the skeletons. By combining GCNs with transformers, we can analyze the human skeleton both spatially and temporally, while using global context information from input video frames in order to enrich our feature representations. Furthermore, since only a few of the joints in the skeleton are important for action recognition, an additional self-attention mechanism [5] will enable our model to focus on those discriminative joints.

Our goal is to reliably perform action recognition on RGB video input using a model that builds on prior work by (1) using GCNs to spatially analyze the skeletons, (2) using a Multimodal Split Attention Fusion (MSAF) module to generate contextual information to incorporate into our model, and (3) using context-aware transformer networks to effectively aggregate skeleton information through time.

In the subsequent sections, we introduce the ST-CTR pipeline and its components, establish a baseline model along with other recently published (in the past year) state-of-the-art human action recognition methods, and proceed to show that our ST-CTR model outperforms the other methods (both qualitatively and quantitatively) on the NTU RGB+D 60 dataset [6] across both the Cross-Subject (X-Sub) and Cross-View (X-View) benchmarks. For the qualitative evaluation, we analyze the performance of our ST-CTR model through the NTU RGB+D 60 test set confusion matrices. For the quantitative evaluation, following prior work [1, 7], we analyze the performance of our ST-CTR model in comparison to other state-of-the-art methods through the NTU RGB+D 60 test set top-1 classification accuracies. Finally, we delve deeper into the components of our ST-CTR pipeline to analyze the performance gain achieved over the other methods.

2. Related Works

Yan *et al*. [1] proposed a Spatial-Temporal Graph Convolutional Network (ST-GCN) model that performs action recognition on skeletons generated using a pre-trained pose estimation model. However, unlike our proposed model, they did not incorporate an RNN model; Instead, they combined the different skeletons by connecting the joints across successive frames. A large limitation of their method is that the ST-GCN cannot perform action prediction for future frames by generating skeletons.

Next, Plizzari *et al*. [7] proposed a Spatial-Temporal Transformer Network (ST-TR) model, which combined GCNs and transformers to perform skeleton-based action recognition. The ST-TR model models dependencies between joints using the transformer self-attention operator. Furthermore, similar to our model, the ST-TR model is able to perform action prediction since it uses an underlying RNN model in the spatial and temporal self-attention modules, which allows ST-TR to generate skeletons for future frames given an initial set of skeletons. However, unlike our proposed model, the ST-TR model does not leverage global video contexts when performing the transformer self-attention operations, which can lead to poor generalizability on unseen data captured in different settings and configurations.

Lastly, Liu *et al*. [8] designed a Global context-aware attention LSTM (GCA-LSTM) network which can effectively perform action recognition on RGB video data. Unlike our work, however, the GCA-LSTM model does not incorporate GCNs to capture spatial patterns in skeletons across frames. We take inspiration from key parts of their design, primarily their video contextual embedding generation, to better leverage global context in our proposed transformer model by using a Multimodal Split Attention Fusion (MSAF) module to generate contextual information to incorporate into our ST-CTR model.

As such, our proposed ST-CTR model is different from each of these prior works, and instead combines several elements from these different designs.

3. Dataset

Given its popularity and significance in the field of Computer Vision, the task of human action recognition has several benchmarks, each with a focus on a particular group of applications. For our work, we evaluate the performance of ST-CTR in comparison to other state-of-the-art human action recognition methods on the NTU RGB+D 60 dataset [6], which is the largest in-house captured benchmark for 3D human action recognition collected using a Microsoft Kinect v2. The dataset comprises of RGB videos, depth sequences, skeleton data (25 joints with 3D pose features), and infrared frames collected for 56,880 RGB+D videos across 60 action classes. We believe that the larger size of the dataset (∼4,000,000 frames), additional skeleton information (25 joints with 3D pose features), and additional action classes (60 classes) make the NTU RGB-D 60 dataset a better benchmark in comparison to others for the evaluation of our approach. The dataset follows two different criteria for evaluation. The Cross-Subject Evaluation (X-Sub) uses 40,320 training and 26,560 test samples split according to the subjects performing the actions. The 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 NTU RGB+D 60 skeleton joint connection configuration is given in Fig. 1.

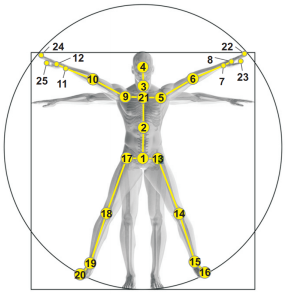


Figure 1. NTU RGB+D 60 skeleton joint configuration [6]

The dataset is pre-processed following the pre-processing performed by Shi *et al*. [9, 10]. The skeleton joints for each frame are used to form a directed acyclic graph (DAG) with the joints as nodes and bones as edges in accordance to the joint configuration specified in Fig. 1. The ”spine joint” (joint 21) is considered to be the the center of gravity of the skeleton and is denoted as the root node. The direction of each edge is determined by the distance between the node and the root node, where the node closer to the root node points to the node further away from the root node. Prior work [1, 7] suggests that this representation is intuitive since the human body is a naturally articulated system where the joints further away from the center of the human body are always physically controlled by an adjacent joint which is closer to the center.

Finally, each graph is normalized to make the distribution of the data for each channel unified. Specifically, the coordinates of each joint are subtracted from the those of the root node. Furthermore, in order to account for different skeletons captured from different viewpoints, the 3D location of the joints are translated from teh camera coordinate system to body coordinates. Specifically, 3D rotations are performed to fix the X axis parallel to the 3D vector from the ”right shoulder” (joint 5) to the ”left shoulder” (joint 9) and the Y axis toward the 3D vector from the ”spine joint” (joint 21) to the ”spine base” (joint 2) [11]. An example of pre-processing is given in Fig. 2.

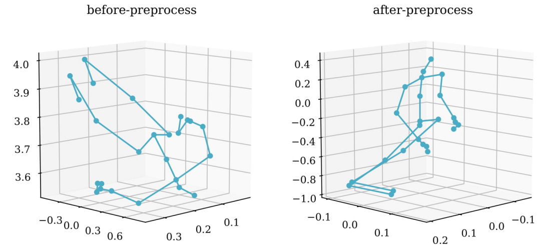


Figure 2. NTU RGB+D 60 skeleton graph pre-processing [11]

4. State-of-the-Art Methods

In order to evaluate the performance of the proposed ST-CTR model in comparison to other human action recognition methods, we explore the performance of recently published (in the past year) GCN-based state-of-the-art action recognition methods, which are detailed in this section.

ST-GCN (Baseline) Yan *et al.* [1] proposed Spatial Temporal Graph Convolution Networks (ST-GCN), which is a skeleton-based human action recognition method that uses GCNs to capture both spatial and temporal patterns in data. Through extensive experiments, they demonstrated that GCNs not only lead to greater expressive power, but also result in stronger generalization capability of the ST-GCN. When considering joint i at time step t (i.e. v_{ti}), the output of the spatial graph convolution are expressed as follows:

$$f_{out}(v_{ti}) = \sum_{v_{qj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{qj})} f_{in}(v_{qj}) \cdot \mathbf{w}(l_{ST}(v_{qj}))$$

where $B(v_{ti})$ is the spatial and temporal neighbourhood of node v_{ti}. l_{ST} is a partitioning function that maps each neighbour of node v_{ti} to one of K subsets. $Z_{ti}(v_{qj})$ is a normalization term that is added to balance the contributions of the different subsets, and $\mathbf{w}$ is the learnable weight function.

As part of their work, [1] explored several different partition strategies. For the sake of our baseline, we selected the spatial configuration partitioning strategy, since it performed best in the original paper. Spatial configuration partitioning assigns labels to nodes based on their distance from a reference node (center of gravity). This partitioning method improves performance by learning more meaningful hierarchical representations of the joints in the skeleton graph [1]. The ST-GCN model is composed of 9 layers of spatial temporal graph convolution operations and a global pooling layer. The output is then fed into a softmax classifier to output the action. In our baseline model, we further implemented dropout layers with $p = 0.5$ in order to combat overfitting.

PeGCN Li *et al.* [12] proposed Predictively Encoded Graph Convolutional Networks (PeGCN), which are highly efficient GCNs that use a parallel structure to gradually fuse motion and spatial information by reducing the temporal resolution as early as possible. PeGCNs achieve state-of-the-art performance on the NTU RGB+D 60 dataset with 86%-93% fewer parameters as a consequence of pre-processing that refines the poses before performing GCN-based action recognition.

RA-GCN Song *et al.* [13] proposed Richly Activated Graph Convolutional Networks (RA-GCN), which are multi-stream GCNs that explore sufficient discriminative features spreading over all skeleton joints in order to reduce the sensitivity of the action models to non-standard skeletons as a consequence of forming distributed redundant representations.

PGCN-TCA Yang *et al.* [14] proposed Pseudo Graph Convolutional Networks with Temporal and Channel-wise Attention (PGCN-TCA), which are GCNs that use a learnable matrix in place of a fixed normalized adjacency matrix in order to capture dependencies between distant joints that are not connected in addition to capturing structural information provided by joints directly connected through bones.

Sem-GCN Ding *et al.* [15] proposed Semantics-guided Graph Convolutional Networks (Sem-GCN), which are GCNs that use a structural graph extraction module, an actional graph inference module, and an attention graph iteration module in order to aggregate L-hop joint neighbourhood information as a means of capturing action-specific latent dependencies.

Mix-Dimension Peng *et al.* [16] proposed Graph Convolutional Networks that, instead of introducing expensive dynamic graph generation paradigms, build GCNs on a Riemann manifold defined via the Poincaré geometry to better model the latent anatomy of the structure data. Different dimensions of the optimal projection in the Riemann space are mixed on the manifold, which provide an efficient way to explore the dimensions for each ST-GCN layer.

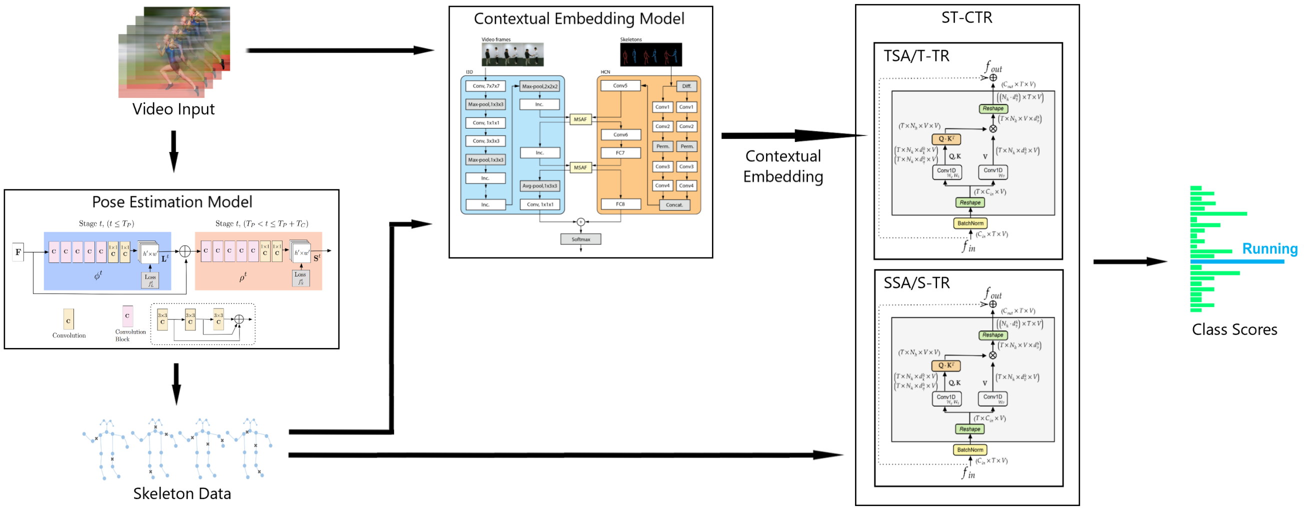


Figure 3. Illustration of the ST-CTR pipeline

PA-ResGCN-B19 Song *et al.* [17] proposed Residual Graph Convolutional Networks (ResGCN), which are GCNs that employ early fused Multiple Input Branches (MIB) along with Part-wise Attention (PartAtt) blocks in order to capture enriched skeleton features that allow the bottleneck-structured ResGCNs to discover essential joints over whole action sequences.

Dynamic GCN Ye *et al.* [18] proposed Dynamic Graph Convolutional Networks (Dynamic GCN), which are GCNs that use stacked Context-encoding Networks (CeN) to learn skeleton topology by incorporating global contextual features from the remaining joints. Dynamic GCNs achieve state-of-the-art performance on the NTU RGB+D 60 dataset with ~ 2 - $4\ \times$ fewer FLOPs than other methods.

ST-TR Plizzari *et al.* [7], as mentioned earlier, propose Spatial-Temporal Transformer Networks (ST-TR), which are GCN-based transformer networks that model dependencies between joints using the Transformer self-attention operator. Spatial and Temporal Self-Attention modules are used to capture intra-frame interactions and correlations among different body parts, which are then combined into a two-stream network to perform action recognition.

5. Spatial-Temporal Context-Aware Transformer Network (ST-CTR)

As illustrated in Fig. 3, the Spatial-Temporal Context-Aware Transformer Network (ST-CTR) pipeline consists of three (four including the pose estimation model) key models that together, perform action recognition. The implementation of ST-CTR is heavily based on that of ST-TR [7], with the same Spatial Transformer Stream (S-TR) module. The Temporal Transformer Stream (T-TR) module is modified to incorporate the contextual embeddings obtained from the MSAF model. The S-TR and T-TR blocks are stacked together and the stream outputs are then fused together by summing up their softmax output scores to obtain the final prediction, as detailed in [9, 10].

Pose estimation model Given that our initial goal was to perform action recognition on raw RGB video data, we decided to use a pose estimation model to generate the skeletons from the video frames, which would then be fed into the ST-CTR model. Following the official implementation of ST-GCN, we used OpenPose [19], an open source library that performs multi-person pose estimation on video input.

However, given that OpenPose is a pre-trained model, it resulted in generating noisy skeletons. As such, we decided to use the ground-truth skeletons provided in the NTU RGB+D 60 dataset for the ST-CTR model as it led to a significant performance boost at the cost of being unable to perform action recognition on RGB video data. Furthermore, the state-of-the-art methods mentioned in the previous section all use the provided ground-truth skeletons, which was another factor in our decision to avoid using a pose estimation model as we wanted to fairly evaluate the considered methods without a pose estimation bottleneck.

Contextual Embedding model One of the key factors that enables ST-CTR to capture global contextual information from video frames through the Transformer attention mechanism is the contextual embedding model. We made use of the Multimodal Split Attention Fusion (MSAF) [20] module in order to generate the context vectors. The MSAF module splits the video and skeleton modalities into channel-wise equal feature blocks and generates a joint representation that is used to then generate soft attention for each channel across the feature blocks. An I3D model [21] is used for the video stream whereas a HCN model [22] is used for the skeleton stream. Two MSAF modules are deployed, one at an intermediate-level in both networks and one at a high-level in both networks. The former module uses an early fusion style using 64 channels per block and is inserted between the 3rd Inception layer of the I3D model and the Conv5 outputs of the HCN model for mixing the two modalities. The latter uses a late fusion style using 256 channels per block and is inserted between the last Inception layer in the I3D model and the FC8 layer in the HCN model for high-level feature recalibration as detailed in [20]. After extensive hyperparameter tuning, we found that a suppression power of $\lambda = 0.5$, which is the default suppression power used in [20], yields the best results for both modules. Finally, the final embedding outputs from both the I3D model and HCN model are averaged together to obtain the contextual embedding.

SSA and S-TR The Spatial Self-Attention (SSA) module applies self-attention within each frame (i.e. skeleton) to extract low-level features that capture relations between body parts [7]. The correlations between each pair of joints are computed independently as illustrated in Fig. 4.

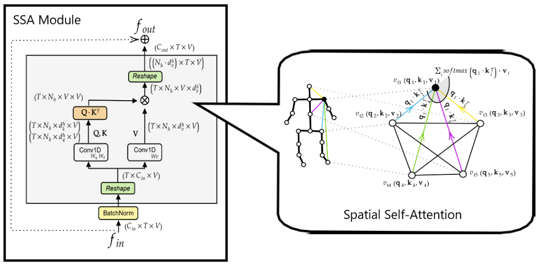

SSA Module
f_{out}
$(C_{out} \times T \times V)$
$((N_h \cdot d_v^h) \times T \times V)$
Reshape
$(T \times N_h \times V \times d_v^h)$
$(T \times N_h \times V \times V)$
$Q \cdot K^T$
$(T \times N_h \times d_k^h \times V)$
$(T \times N_h \times d_q^h \times V)$
Q,K
V
$(T \times N_h \times d_v^h \times V)$
Conv1D W_q W_k
Conv1D Wv
$(T \times C_{in} \times V)$
Reshape
BatchNorm
$(C_{in} \times T \times V)$
f_{in}
$\Sigma_i softmax\left(\mathbf{q}_1 \cdot \mathbf{k}_i^T\right) \cdot \mathbf{v}_i$
v_{t1} $(\mathbf{q}_1, \mathbf{k}_1, \mathbf{v}_1)$
v_{t2} $(\mathbf{q}_2, \mathbf{k}_2, \mathbf{v}_2)$
v_{t3} $(\mathbf{q}_3, \mathbf{k}_3, \mathbf{v}_3)$
v_{t4} $(\mathbf{q}_4, \mathbf{k}_4, \mathbf{v}_4)$
v_{t5} $(\mathbf{q}_5, \mathbf{k}_5, \mathbf{v}_5)$
$\mathbf{q}_1 \cdot \mathbf{k}_2^T$
$\mathbf{q}_1 \cdot \mathbf{k}_3^T$
$\mathbf{q}_1 \cdot \mathbf{k}_4^T$
$\mathbf{q}_1 \cdot \mathbf{k}_5^T$
Spatial Self-Attention

Figure 4. Spatial Self-Attention (SSA) module [7]

Formally, given a frame at time t, for each node v_{ti} of the skeleton, a query vector $\boldsymbol{q}_i^t \in \mathbb{R}^{dq}$, a key vector $\boldsymbol{k}_i^t \in \mathbb{R}^{dk}$, and a value vector $\boldsymbol{v}_i^t \in \mathbb{R}^{dv}$ are computed by applying trainable linear transformation to the node features $\boldsymbol{n}_i^t \in \mathbb{R}^{C_{in}}$ with parameters that are shared across all the nodes. A query-key dot product is then applied to obtain a weight $\alpha_{ij}^t = \boldsymbol{q}_i^t \cdot \boldsymbol{k}_j^{t\top} \in \mathbb{R}, \forall t \in T$ for each pair of nodes (v_{ti}, v_{tj}) that represents the strength of the correlations between the two nodes. The resulting score α_{ij}^t is then used to weight each joint value $\boldsymbol{v}_j^t$. The new embedding $\boldsymbol{z}_i^t \in \mathbb{R}^{C_{out}}$ for node v_{ti} is then computed as a weighted sum of each each of the joint values as follows:

$$\boldsymbol{x}_i^t = \sum_j softmax_j \left(\frac{\alpha_{ij}^t}{\sqrt{d_k}} \right) \boldsymbol{v}_j^t$$

Multi-headed attention is applied by repeating this embedding extraction process N_h times, each time with a different set of learnable parameters. The set $(\boldsymbol{z}_{i1}^t, \ldots, \boldsymbol{z}_{iH}^t)$ of embeddings that are obtained for node v_{ti} is then combined with a learnable transformation $concat(\boldsymbol{z}_{i1}^t, \ldots, \boldsymbol{z}_{iH}^t) \cdot \boldsymbol{W}_o$ to obtain the output features of SSA. Finally, these are then passed to a 2D convolution module with kernel K_t on the temporal dimension (TCN) [1] in order to extract temporally relevant Spatial Transformer Stream (S-TR) features following:

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

TSA and T-TR The Temporal Self-Attention (TSA) module extracts inter-frame relations between the same nodes in time to learn how to correlate frames across time [7]. Each single joint is considered to be independent and correlations between frames are computed by comparing the change in the joint embeddings along the temporal dimension as illustrated in Fig. 5.

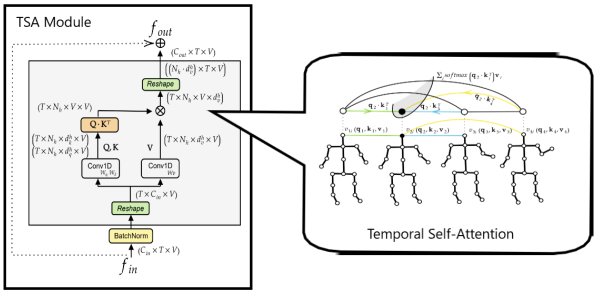

TSA Module
f_{out}
$(C_{out} \times T \times V)$
$((N_h \cdot d_v^h) \times T \times V)$
Reshape
$(T \times N_h \times V \times d_v^h)$
$(T \times N_h \times V \times V)$
$Q \cdot K^T$
$(T \times N_h \times d_k^h \times V)$
$(T \times N_h \times d_q^h \times V)$
Q,K
V
$(T \times N_h \times d_v^h \times V)$
Conv1D W_q W_k
Conv1D Wv
$(T \times C_{in} \times V)$
Reshape
BatchNorm
$(C_{in} \times T \times V)$
f_{in}
$\Sigma_i softmax\left(\mathbf{q}_2 \cdot \mathbf{k}_i^T\right)\mathbf{v}_i$
$\mathbf{q}_2 \cdot \mathbf{k}_1^T$
$\mathbf{q}_2 \cdot \mathbf{k}_3^T$
$\mathbf{q}_2 \cdot \mathbf{k}_4^T$
v_{1i} $(\mathbf{q}_1, \mathbf{k}_1, \mathbf{v}_1)$
v_{2i} $(\mathbf{q}_2, \mathbf{k}_2, \mathbf{v}_2)$
v_{3i} $(\mathbf{q}_3, \mathbf{k}_3, \mathbf{v}_3)$
v_{4i} $(\mathbf{q}_4, \mathbf{k}_4, \mathbf{v}_4)$
Temporal Self-Attention

Figure 5. Temporal Self-Attention (TSA) module [7]. The implementation of the TSA module is symmetrical to that of the SSA module, with the only difference being that the V dimension corresponds to the T dimension and vice versa.

The formulation of the TSA module is symmetrical to that of the SSA module:

$$\alpha_{tu}^v = \boldsymbol{q}_t^v \cdot \boldsymbol{k}_u^t \forall v \in V, \boldsymbol{z}_t^v = \sum_J softmax_u \left(\frac{\alpha_{tu}^v}{\sqrt{d_k}} \right) \boldsymbol{v}_u^v$$

where v_{ti} and v_{ui} correspond to the same joint v in two different instants t and u, $\alpha_{tu}^i \in \mathbb{R}$ is the correlation score, $\boldsymbol{q}_t^i \in \mathbb{R}^{dq}$ is the query associated with v_{ti}, $\boldsymbol{k}_u^i \in \mathbb{R}^{dk}$ and $\boldsymbol{v}_u^i \in \mathbb{R}^{dv}$ are the key and value associated with v_{ui} (all computed using trainable linear transformations as in SSA), and $\boldsymbol{z}_t^i \in \mathbb{R}^{C_{out}}$ is the resulting node embedding. Furthermore, multi-headed attention is then applied in TSA as in SSA. Similar to the S-TR stream, inside each Temporal Transformer Stream (T-TR) layer, a standard Graph Convolution (GCN) [1] sub-module is followed by the TSA module as detailed in [7]. Additionally, in order to incorporate the contextual embeddings generated by our contextual embedding model, we combine the outputs of the GCN module with the outputs of the MSAF module using a linear layer as follows:

$$\mathbf{T\text{-}TR}(x) = \mathbf{TSA}(GCN(x), MSAF(x))$$

Other methods such as summing, averaging, concatenating the outputs of the GCN module and those of the MSAF module were considered. However, we found that using a linear layer to combine the two outputs resulted in the best performance due to the higher expressive power of linear layers [23].

6. Experiments

Experimental Setup The ST-CTR model is trained using a PyTorch framework [24] for a total of 120 epochs with a batch size of 32. After experimenting with several optimizers ranging from first-order stochastic optimizers such as SGD [25] to adaptive learning rate optimizers such as Adam [26], we found that SGD leads to faster convergence early on while maintaining training stability later on during gradient descent and as such, we decided to employ an SGD optimizer to train ST-CTR with an initial learning rate of 0.1 that is decayed by a factor of 10 at epochs 60 and 90. This learning rate schedule has been shown to lead to the best performance empirically by [10].

Additionally, in accordance to the experiments performed in [7], we decided to use DropAttention to avoid overfitting. DropAttention [27] is a dropout technique used to regularize attention weights in Transformers by randomly dropping columns of the attention matrix. We found that using DropAttention allowed ST-CTR to gain a significant performance boost, especially on the NTU RGB+D 60 Cross-View (X-view) benchmark as the model would be able to better generalize the same action across different views. Furthermore, given that [28] found that using 8 multi-head attention heads along with query (d_q), key (d_k), and value (d_v) embedding dimensions of $0.25 \times C_{out}$ in each layer was sufficient in order to achieve state-of-the-art performance, we decided to use the same hyperparameters. The output dimensions obtained from the MSAF model were also set to the same dimensions in order to allow the Temporal T-TR module to easily incorporate the global contextual embeddings from the video frames.

As for the ST-CTR architecture itself, we found that using the hyperparameters tuned by [7] yielded the best performance; Each of the S-TR and T-TR streams are composed of 9 layers with channel dimensions of 64 for the first three layers, 128 for the intermediate three layers, and 256 for the last three layers. Additionally, batch normalization is applied to the input joint and video data along with a global average pooling layer that is applied before the softmax classifier. Finally, each stream is trained using the standard cross-entropy loss.

Qualitative Results The confusion matrices of our ST-CTR model on the NTU RGB+D 60 test set Cross-Subject (X-Sub) and Cross-View (X-View) benchmarks are given in Fig. 6. Given the high dimensionality of output classification space (60 action classes), only the first 30 action classes are included in the confusion matrices, in accordance to the evaluation performed by [7]. Further qualitative evaluation of each of the components of the ST-CTR pipeline can be found in the next section.

Quantitative Results In accordance to the quantitative analysis performed in [1, 7], we report the NTU RGB+D 60 test set Cross-Subject (X-Sub) and Cross-View (X-View) top-1 classification accuracies of our ST-CTR model and the considered state-of-the-art action recognition methods. The quantitative evaluation results are given in Table. 1.

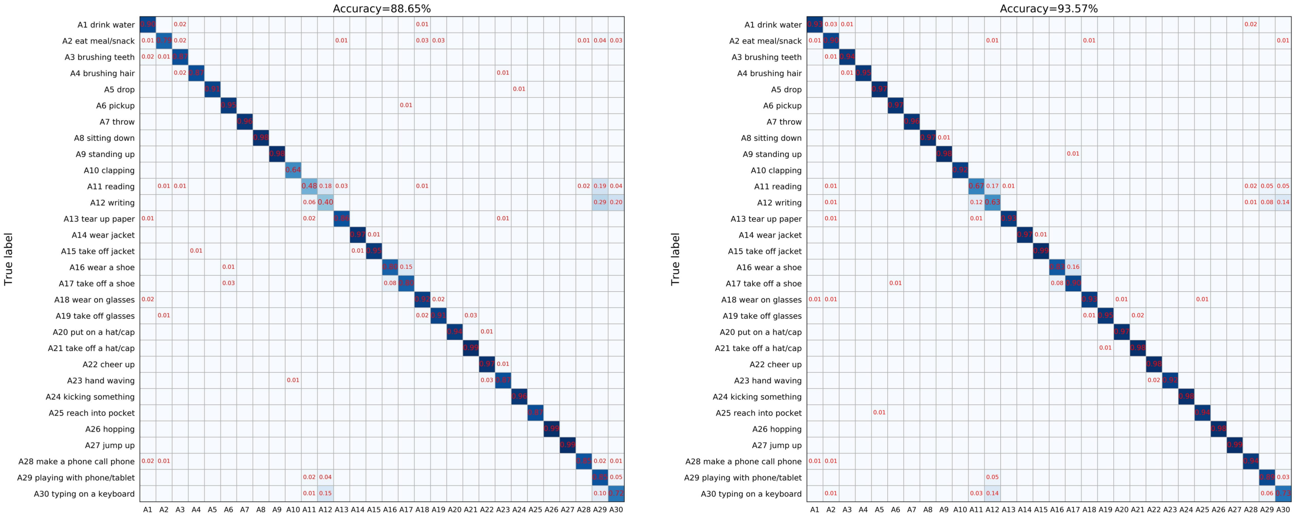
Method	X-Sub	X-View
ST-GCN	77.5%	83.3%
PeGCN	85.6%	93.4%
RA-GCN	87.3%	93.5%
PGCN-TCA	88.0%	93.5%
Sem-GCN	86.2%	92.4%
Mix Dimension	87.2%	93.4%
PA-ResGCN-B19	88.5%	93.5%
Dynamic GCN	87.3%	88.6%
ST-TR	85.9%	91.1%
ST-CTR (ours)	**88.7%**	**93.6%**

Table 1. NTU RGB+D 60 test set top-1 classification accuracies

7. Discussion

Contextual Embedding Model Based on the qualitative evaluation in the previous section and from the confusion matrices, we observe that the main source of classification errors across both the Cross-Subject (X-Sub) and Cross-View (X-View) benchmarks creep in from misclassification between the ”reading” and ”writing” classes. Delving deeper, we found that this misclassification error between the ”reading” and ”writing” classes was worse across all the other state-of-the-art human action recognition methods considered. The main source of this error can be attributed to the fact that the skeleton data, as observed in Fig. 7, on its own, does not convey enough information to the models to accurately distinguish one action from the other.

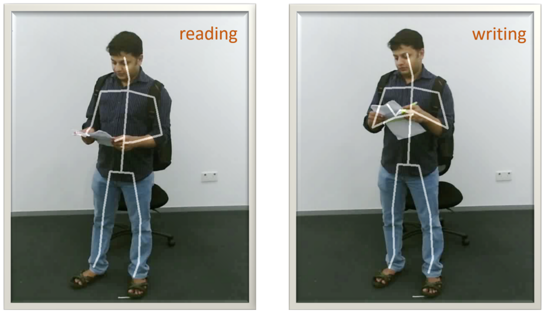
Both actions are performed by a standing human with minimal hand movement and as such, the skeletons are almost impossible to classify, even by a human being. However, viewing the video frames themselves, we observe that subtle movements and nuanced interactions with the pen and paper allow human beings to tell the actions apart. This



(a) Cross-Subject (X-Sub)

(b) Cross-View (X-View)

Figure 6. Confusion matrices of ST-CTR on the first 30 action classes of the NTU RGB+D 60 test set benchmarks



(a) ”reading” class

(b) ”writing” class

Figure 7. Samples from the NTU RGB+D 60 dataset [11]

was one of our primary motivations for using a contextual embedding model as MSAF allows ST-CTR to capture global context features from the video frames themselves, thus, allowing ST-CTR to not only outperform the other state-of-the-art action recognition methods when it comes to variations in the setup (such as lighting, etc.), but also when it comes to classifying actions among which the differences are too subtle to distinguish them based on the skeleton sequences alone.

Ablation Study We analyze the performance gain achieved by each individual component of the ST-CTR pipeline through an ablation study, given in Table. 2.

We observe that S-TR and T-TR, by themselves, do not perform well on either of the benchmarks. Note that no further hyperparameter tuning was performed on either the S-TR or T-TR components during the ablation study and as such, higher performance with either of these components alone is possible with careful hyperparameter tuning.

Components in the pipeline	X-Sub	X-View
S-TR	78.6%	80.7%
T-TR	78.4%	80.5%
MSAF + T-TR	82.1%	85.8%
S-TR + T-TR	85.9%	91.1%
MSAF + S-TR + T-TR (ST-CTR)	**88.7%**	**93.6%**

Table 2. Ablation study of the ST-CTR pipeline

S-TR and T-TR, when combined together, yield a significant performance gain across both the benchmarks. While this can be partially attributed to the fact that the hyperparameters were tuned to use both components together, it can also be attributed to the fact that both spatial and temporal features are crucial to a task such as action recognition. Thus, removing either of these components leads to a large drop in performance.

Including the MSAF contextual embedding model in the T-TR stream also yields a significant performance gain across both benchmarks. As mentioned earlier, this can be attributed to the fact that MSAF generates global contextual features from the video frames themselves, which allows the model to not only generalize better to different setups, but to distinguish between actions whose skeleton data alone does not allow for accurate classification due to subtle differences in motion. Note that since the MSAF generated contextual embeddings are incorporated into ST-CTR through the T-TR stream, an ablation study that removed T-TR but retained MSAF was not possible.

Finally, as mentioned earlier, our initial goal was to perform human action recognition on raw RGB video data. We made use of OpenPose, a pre-trained pose estimation model to generate the skeletons from the raw RGB video data. We found that using the skeletons generated by OpenPose, the ST-CTR model yielded top-1 classification accuracies of 41.6% and 45.9% on the NTU RGB+D 60 test set Cross-Subject (X-Sub) and Cross-View (X-View) benchmarks, respectively. This significant drop in performance can be attributed to the fact that OpenPose is a pre-trained pose estimation model and thus, generates skeletons that are noisy given the raw RGB video data alone. Therefore, we decided to use the provided NTU RGB+D 60 ground-truth skeletons for our model and as such, the ST-CTR pipeline only consists of the MSAF, the S-TR, and the T-TR modules, as given in the last row of Table. 2.

Spatial and Temporal Self-Attention As introduced earlier, the SSA module performs the spatial self-attention mechanism in the S-TR component whereas the TSA module performs the temporal self-attention mechanism in the T-TR component. In order further examine the effect of each of these mechanisms on the overall model, we qualitatively evaluate the outputs of the SSA and TSA modules with respect to the inputs to each of the modules.

The spatial attention maps learned by the SSA module across different layers for two action classes: ”taking a selfie” and ”put on a hat”, are visualized in Fig. 8.

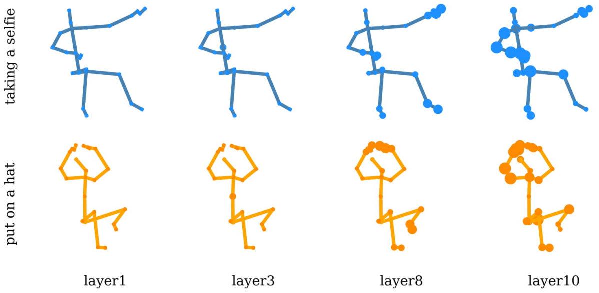


Figure 8. Spatial attention maps learned by the SSA module. Node sizes represent the importance of the corresponding joints [11]

Based on the sizes of the nodes, we observe that the SSA module focuses on the joints in the hands and the head, which suggests that the features from these joints play pivotal roles in enabling accurate classification. This also makes intuitive sense because these joints are prone to more motion when it comes to performing the ”taking a selfie” and ”put on a hat” actions. Furthermore, note that the spatial attention is not evident from the sizes of the nodes in the lower layers. This can be attributed to the fact that the receptive fields of the lower layers are relatively smaller, thus making it harder to learn good attention maps [11].

The temporal attention weights learned by the TSA module across different frames for two action classes: ”taking a selfie” and ”throw”, are visualized in Fig. 9. For the ”taking a selfie” action, we observe that the TSA module focuses more on the movement of the arm as it rises in the fifth layer while it focuses more on the final posture of the body in the seventh layer. As for the ”throw” action, we observe that TSA module focuses more on the arm when the hands are in a lower position. This makes intuitive sense as these are the frames that convey the most information regarding the action being performed, thus, allowing the TSA module to focus on frames that are crucial to action recognition.

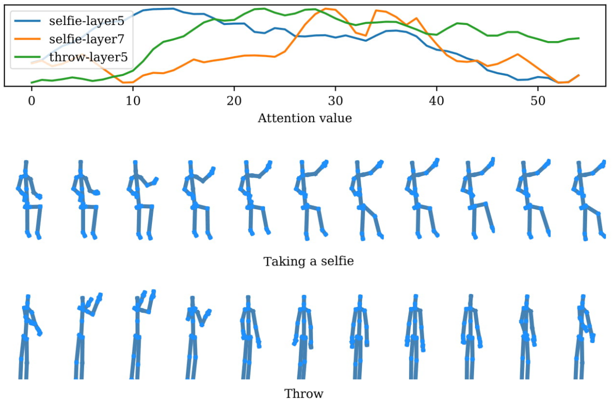


Figure 9. Temporal attention weights learned by the TSA module along with their corresponding skeletons [11]

8. Conclusion

In this paper, we introduced a novel context-aware skeleton-based human action recognition model: the Spatial-Temporal Context-aware Transformer (ST-CTR). Through our extensive experiments, we showed that ST-CTR outperforms the state-of-the-art action recognition methods across both the NTU RGB+D 60 Cross-Subject (X-Sub) and Cross-View (X-View) benchmarks. Furthermore, we showed that the Spatial and Temporal Self-Attention modules enable the model to focus on critical parts of the skeleton for action classification across the spatial and temporal dimensions, respectively. Finally, we showed that the ST-CTR model is able to incorporate global contextual features from the video frames when classifying actions, thus, enabling ST-CTR to distinguish among action classes with subtle differences in skeleton features.

Potential future directions of this work include, but are not limited to: action prediction using the ST-CTR transformer to predict skeletons for future frames given a set of past skeleton sequences, improving pose estimation models to generate better skeletons to enable action recognition on raw RGB video data, and better overfitting mitigation in the spatial processing of the skeleton graphs using techniques such as DropEdge [29] and DropGraph [30].

References

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

[2] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[3] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[4] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.

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

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

[7] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Skeleton-based action recognition via spatial and temporal transformer networks. *Computer Vision and Image Understanding*, page 103219, 2021.

[8] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C Kot. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing*, 27(4):1586–1599, 2017.

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

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

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

[12] Jongmin Yu, Yongsang Yoon, and Moongu Jeon. Predictively encoded graph convolutional network for noise-robust skeleton-based action recognition. *arXiv preprint arXiv:2003.07514*, 2020.

[13] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Richly activated graph convolutional network for robust skeleton-based action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[14] Hongye Yang, Yuzhang Gu, Jianchao Zhu, Keli Hu, and Xiaolin Zhang. Pgcn-tca: Pseudo graph convolutional network with temporal and channel-wise attention for skeleton-based action recognition. *IEEE Access*, 8:10040–10047, 2020.

[15] Xiaolu Ding, Kai Yang, and Wai Chen. A semantics-guided graph convolutional network for skeleton-based action recognition. In *Proceedings of the 2020 the 4th International Conference on Innovation in Artificial Intelligence*, pages 130–136, 2020.

[16] Wei Peng, Jingang Shi, Zhaoqiang Xia, and Guoying Zhao. Mix dimension in poincaré geometry for 3d skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1432–1440, 2020.

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

[18] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang. Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 55–63, 2020.

[19] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[20] Lang Su, Chuqing Hu, Guofa Li, and Dongpu Cao. Msaf: Multimodal split attention fusion. *arXiv preprint arXiv:2012.07175*, 2020.

[21] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

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

[23] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Efficient low-rank multimodal fusion with modality-specific factors. *arXiv preprint arXiv:1806.00064*, 2018.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[25] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] Lin Zehui, Pengfei Liu, Luyao Huang, Junkun Chen, Xipeng Qiu, and Xuanjing Huang. Dropattention: A regularization method for fully-connected self-attention networks. *arXiv preprint arXiv:1907.11065*, 2019.

[28] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3286–3295, 2019.

[29] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

[30] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. Decoupling gcn with dropgraph module for skeleton-based action recognition.