TSA-Net: Tube Self-Attention Network for Action Quality Assessment

Shunli Wang[1,3], Dingkang Yang[1,2], Peng Zhai[1,4], Chixiao Chen[1], Lihua Zhang[2,1,3,4*]
Academy for Engineering and Technology, Fudan University[1] Ji Hua Laboratory, Foshan, China[2]
Engineering Research Center of AI and Robotics, Shanghai, China[3]
Jilin Provincial Key Laboratory of Intelligence Science and Engineering, Changchun, China[4]
{slwang19,dkyang20,pzhai18,cxchen,lihuazhang}@fudan.edu.cn

ABSTRACT

In recent years, assessing action quality from videos has attracted growing attention in computer vision community and human-computer interaction. Most existing approaches usually tackle this problem by directly migrating the model from action recognition tasks, which ignores the intrinsic differences within the feature map such as foreground and background information. To address this issue, we propose a Tube Self-Attention Network (TSA-Net) for action quality assessment (AQA). Specifically, we introduce a single object tracker into AQA and propose the Tube Self-Attention Module (TSA), which can efficiently generate rich spatio-temporal contextual information by adopting sparse feature interactions. The TSA module is embedded in existing video networks to form TSA-Net. Overall, our TSA-Net is with the following merits: 1) High computational efficiency, 2) High flexibility, and 3) The state-of-the-art performance. Extensive experiments are conducted on popular action quality assessment datasets including AQA-7 and MTL-AQA. Besides, a dataset named Fall Recognition in Figure Skating (FR-FS) is proposed to explore the basic action assessment in the figure skating scene. Our TSA-Net achieves the Spearman's Rank Correlation of 0.8476 and 0.9393 on AQA-7 and MTL-AQA, respectively, which are the new state-of-the-art results. The results on FR-FS also verify the effectiveness of the TSA-Net. The code and FR-FS dataset are publicly available at *https://github.com/Shunli-Wang/TSA-Net*.

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition and understanding**; • **Information systems** → Information extraction.

KEYWORDS

Action Quality Assessment, Self-attention Mechanism, Video Action Analysis

ACM Reference Format:
Shunli Wang[1,3], Dingkang Yang[1,2], Peng Zhai[1,4], Chixiao Chen[1], Lihua Zhang[2,1,3,4]. 2021. TSA-Net: Tube Self-Attention Network for Action Quality Assessment. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475438

* indicates corresponding author.

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

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475438

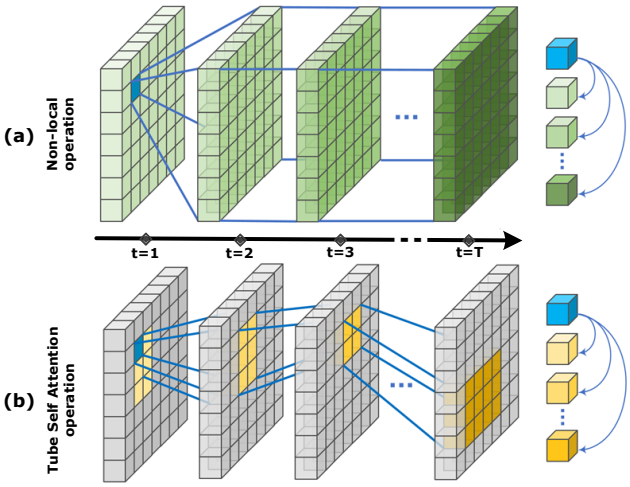

Figure 1: Diagrams of two attention-based feature aggregation methods. (a) For each position (*e.g.* blue), Non-local module [39] performs dense correlation for all features (in green). (b) For each position in the spatio-temporal tube (*e.g.* blue), TSA module only performs sparse correlation for all features located in the tube. After TSA operation, features in the tube can capture rich contextual information from athletes' series of movements in raw videos. Residual connections are ignored for clear display.

1 INTRODUCTION

In addition to identifying human action categories in videos, it is also crucial to evaluate the quality of specific actions, which means that the machine needs to understand not only what has been performed but also how well a particular action is performed. Action quality assessment (AQA) aims to evaluate how well a specific action is performed, which has become an emerging and attractive research topic in computer vision community. Assessing the quality of actions has great potential value for various real-world applications such as analysis of sports skills [22, 25, 27, 33, 40], surgical maneuver training [20, 45, 46] and many others [4, 5].

In recent years, many methods [22, 25, 33, 40] directly applied the network of human action recognition (HAR) such as C3D [34] and I3D [1] to AQA tasks. Although these methods have achieved

considerable performance, they still face many challenges, and their performances and efficiency are indeed limited. Firstly, the huge gap between HAR and AQA should be emphasized. Models in HAR require distinguishing subtle differences between different actions, while models in AQA require evaluating a specific action's advantages and disadvantages. Therefore, the performances of existing methods are inherently limited because of the undifferentiated feature extraction of video content, which leads to the pollution of body features. It is not appropriate to apply the framework in HAR directly to AQA without any modification. Secondly, existing methods cannot perform feature aggregation efficiently. The receptive field of convolution operation is limited, resulting in the loss of long-range dependencies. RNN has the inherent property of storing hidden states, which makes it challenging to be paralleled. An effective and efficient feature aggregation mechanism is desired in AQA tasks.

To solve all challenges above, we propose Tube Self-Attention (TSA) module, an efficient feature aggregation strategy based on tube mechanism and self-attention mechanism, shown in Figure 1. The basic idea of the TSA module is straightforward and intuitive: considering that AQA models require rich temporal contextual information and do not require irrelevant spatial contextual information, we combine the tube mechanism and self-attention mechanism to aggregate action features sparsely to achieve better performance with minimum computational cost. For example, during a diving competition, the athletes' postures are supposed to raise most attentions, instead of distractors such as the audience and advertisements in the background. The merits of the TSA module are three-fold: (1)*High efficiency*, the tube mechanism makes the network only focus on a subset of the feature map, reducing a large amount of computational complexity compared with Non-local module. (2)*Effectiveness*, the self-attention mechanism is adopted in TSA module to aggregate the features in the spatio-temporal tube (ST-Tube), which preserves the contextual information in the time dimension and weakens the influence of redundant spatial information. (3)*Flexibility*, consistent with Non-local module, TSA module can be used in a plug-and-play fashion, which can be embedded in any video network with various input sizes.

Based on TSA module, we proposed Tube Self-Attention Network (TSA-Net) for AQA. Existing visual object tracking (VOT) framework is firstly adopted to generate tracking boxes. Then the ST-Tube is obtained through feature selection. The self-attention mechanism is performed in ST-Tube for efficient feature aggregation. Our method is tested on the existing AQA-7 [24] and MTL-AQA [25] datasets. Sufficient experimental exploration, including performance analysis and computational cost analysis, is also conducted. In addition, a dataset named Fall Recognition in Figure Skating (FR-FS) is proposed to recognize falls in figure skating. Experimental results show that our proposed TSA-Net can achieve state-of-the-art results in three datasets. Extensive comparative results verify the efficiency and effectiveness of TSA-Net.

The main contributions of our work are as follows:

- We exploit a simple but efficient sparse feature aggregation strategy named Tube Self-Attention (TSA) module to generate representations with rich contextual information for action based on tracking results generated by the VOT tracker.
- We propose an effective and efficient action quality assessment framework named TSA-Net based on TSA module, with adding little computational cost compared with Non-local module.
- Our approach outperforms state-of-the-arts on the challenging MTL-AQA and AQA-7 datasets and a new proposed dataset named FR-FS. Extensive experiments show that our method has the ability to capture long-range contextual information, which may not be performed by previous methods.

2 RELATED WORKS

Action Quality Assessment. Most of the existing AQA methods focus on two fields: sports video analysis [22, 25, 27, 33, 40] and surgical maneuver assessment [20, 45, 46]. AQA works focus on sports can be roughly divided into two categories: pose-based methods and non-pose methods. Pose-based methods [22, 27, 37] take pose estimation results as input to extract features and generate the final scores. Because of the atypical body posture in motion scene, the performance of pose-based methods are suboptimal.

Non-pose methods exploit DNNs such as C3D and I3D to extract features directly from the raw video and then predict the final score. For example, Self-Attentive LSTM [40], MUSDL [33], C3D-AVG-MTL [25], and C3D-LSTM [23] share similar network structures, but their difference lies in the feature extraction and feature aggregation method. Although these methods have achieved significant results, the enormous computational cost of feature extraction and aggregation module limits AQA models' development. Different from the aforementioned AQA methods, our proposed TSA module can perform feature extraction and aggregation efficiently.

Self-Attention Mechanism. Self-attention mechanism [36] was firstly applied on the machine translation task in neural language processing (NLP) as the key part of Transformer. After that, researchers put forward a series of transformer-based models including BERT [3], GPT [28], and GPT-2 [29]. These models tremendously impacted various NLP tasks such as machine translation, question answering system, and text generation. Owing to the excellent performance, some researchers introduce self-attention mechanism into many CV tasks including image classification [10, 12, 21], semantic segmentation [2, 42, 44] and object detection [6, 11, 43]. Specifically, inspired by Non-local[39] module, Huang *et al.* [13] proposed criss-cross attention module and CCNet to avoid dense contextual information in semantic segmentation task. Inspired by these methods, our proposed TSA-Net adopts self-attention mechanism for feature aggregation.

Video Action Recognition Video action recognition is a fundamental task in computer vision. With the rise of deep convolutional neural networks (CNNs) in object recognition and detection, some researchers have designed many deep neural networks for video tasks. Two-stream networks [1, 9, 32] take static images and dynamic optical flow as input and fuse the information of appearance and short-term motions. 3D convolutional networks [14, 15, 34] utilize 3D kernels to extract features form raw videos directly. In order to meet the needs in real applications, many works [8, 18, 35]

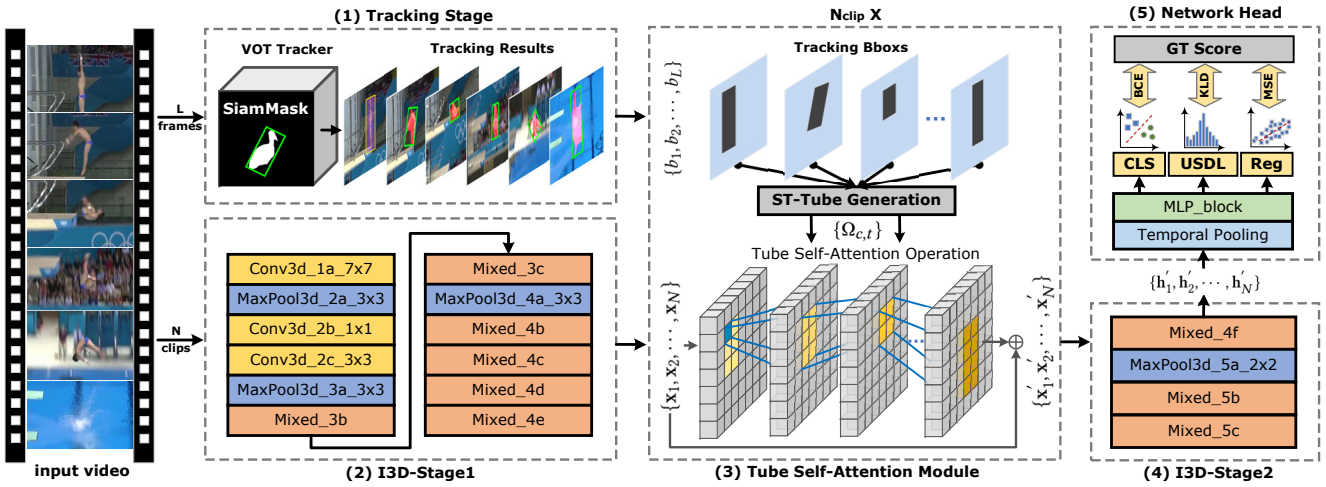

Figure 2: Overview of the proposed TSA-Net for action quality assessment. TSA-Net consists of five steps: (1) Tracking. VOT tracker is adopted to generate tracking results *B*. (2) Feature extraction-s1. The input video is divided into *N* clips and the feature extraction is performed by I3D-Stage1 to generate X. (3) Feature aggregation. ST-Tube is generated given B and X, and then the TSA mechanism is used to complete the feature aggregation, results in X′. (4) Feature extraction-s2. Aggregated feature X′ is passed to I3D-Stage2 to generate H′. (5) Network head. The final scores are generated by *MLP_block*. TSA-Net is trained with different losses according to different tasks.

focus on the efficient designing of networks recently. The proposed TSA-Net take I3D [1] network as the backbone.

3 APPROACH

3.1 Overview

The network architecture is given in Figure 2. Given an input video with L frames $V = \{F_l\}_{l=1}^{L}$, SiamMask[38] is used as the single object tracker to obtain the tracking results $B = \{b_l\}_{l=1}^{L}$, where $b_l = \{(x_p^l, y_p^l)\}_{p=1}^{4}$ represents the tracking box of the l-th frame. In feature extraction stage, V is firstly divided into N clips where each clip contains M consecutive frames. All clips are further sent into the first stage of Inflated 3D ConvNets (I3D) [1], resulting in N features as $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$, $\mathbf{x}_n \in \mathbb{R}^{T\times H\times W\times C}$. Since the temporal length of $\mathbf{x}_n$ is T, we have $\mathbf{x}_n = \{\mathbf{x}_{n,t}\}_{t=1}^{T}$, $\mathbf{x}_{n,t} \in \mathbb{R}^{H\times W\times C}$.

In feature aggregation stage, the TSA module takes tracking boxes B and video feature $\mathbf{X}$ as input to perform feature aggregation, resulting in video feature $\mathbf{X}' = \{\mathbf{x}'_n\}_{n=1}^{N}$ with rich spatio-temporal contextual information. Since the TSA module does not change the size of the input feature map, $\mathbf{x}_n$ and $\mathbf{x}'_n$ have the same size, *i.e.*, $\mathbf{x}'_n \in \mathbb{R}^{T\times H\times W\times C}$. This property enables TSA modules to be stacked in multiple layers to generate features with richer contextual information. The aggregated feature $\mathbf{X}'$ is further sent to the second stage of I3D to complete feature extraction, resulting in $\mathbf{H} = \{\mathbf{h}_n\}_{n=1}^{N}$. $\mathbf{H}$ is the representation of the whole video or athlete's performance.

In prediction stage (*i.e.*, network head), average pooling operation is adopted to fuse $\mathbf{H}$ along clip dimension, *i.e.*, $\overline{\mathbf{h}} = \frac{1}{N}\sum_{n=1}^{N} h_n$, $\overline{\mathbf{h}} \in \mathbb{R}^{T\times H\times W\times C}$. $\overline{\mathbf{h}}$ is further fed into the *MLP_Block* and finally used for the prediction of different tasks according to different datasets.

3.2 Tube Self-Attention Module

The fundamental difference between TSA module and Non-local module is that TSA module can filter the features of participating in self-attention operation in time and space according to the tracking boxes information. The TSA mechanism has the ability to ignore noisy background information which will interfere with the final result of action quality assessment. This operation makes the network pay more attention to the features containing athletes' information and eliminate irrelevant background information interference.

The tube self-attention mechanism can also be called "local Non-local". The first "local" refers to the ST-Tube, while "Non-local" refers to the response between features calculated by self-attention operation. So the TSA module is able to achieve more effective feature aggregation on the premise of saving computing resources. TSA module consists of two steps: (1) spatio-temporal tube generation, and (2) tube self-attention operation.

Step 1: spatio-temporal tube generation. Intuitively, after obtaining tracking information B and feature map $\mathbf{X}$ of the whole video, all features in the ST-Tube can be selected directly. Unfortunately, owing to the existence of two temporal pooling operations in *I3D-stage1*, the corresponding relationship between tracking boxes and feature maps is not $1:1$ but *many* $:1$. Besides, all tracking boxes generated by SiamMask are skew, which complicates the generation of ST-Tube.

To solve these problems, we propose an alignment method which is shown in Figure 3. Since *I3D-Stage1* contains two temporal pooling operations, the corresponding relationship between bounding boxes and feature map is 4:1, *i.e.*, $\{b_l, b_{l+1}, b_{l+2}, b_{l+3}\}$ is correspond to $\mathbf{x}_{c,t}$. All tracking boxes should be converted into mask first, and then used to generate ST-Tube.

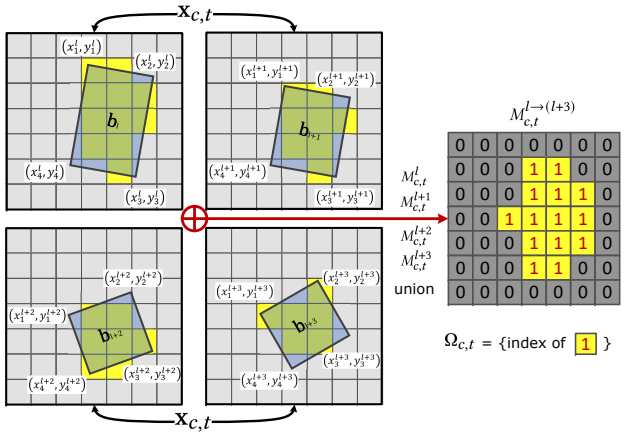

Figure 3: The generation process of spatio-temporal tube. All boxes $\{b_l, b_{l+1}, b_{l+2}, b_{l+3}\}$ are scaled to the same size as the feature map $\mathbf{x}_{c,t}$, and then the separate masks are generated. All masks are aggregated into the final mask $M_{c,t}^{l\rightarrow(l+3)}$ through *Union* operation.

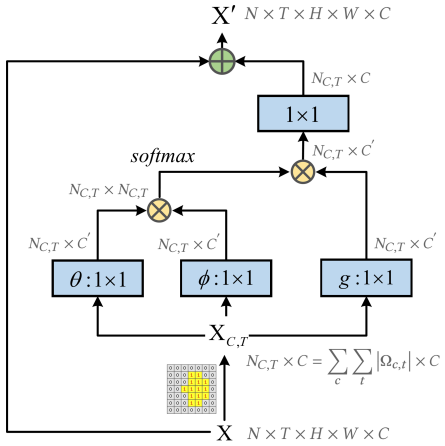

Figure 4: Calculation process of the TSA module. "⊕" denotes matrix multiplication, and "⊗" denotes element-wise sum. Owing to the existence of tube mechanism, only the features inside the ST-Tube can be selected and participate in the calculation of self-attention.

We denote the mask of b_l correspond to $\mathbf{x}_{c,t}$ as $M_{c,t}^{l} \in \{0,1\}^{H\times W}$. The generation process of $M_{c,t}^{l}$ is as follows:

$$M_{c,t}^{l}(i,j) = \begin{cases} 1, S(b_l,(i,j)) \geqslant \tau \\ 0, S(b_l,(i,j)) < \tau \end{cases} \tag{1}$$

Where $S(\cdot,\cdot)$ function calculates the proportion of the feature grid at (i,j) covered by b_l. If the proportion is higher than threshold τ, the feature located at (i,j) will be selected, otherwise it will be discarded. The proportion of each feature grid covered by box ranges from 0 to 1, so we directly took the intermediate value of $\tau = 0.5$ in all experiments of this paper.

Four masks are further assembled into $M_{c,t}^{l\rightarrow(l+3)} \in \{0,1\}^{H\times W}$ through element-wise *OR* operation:

$$M_{c,t}^{l\rightarrow(l+3)} = Union(M_{c,t}^{l}, M_{c,t}^{l+1}, M_{c,t}^{l+2}, M_{c,t}^{l+3}) \tag{2}$$

This mask contains all location information of the features participating in self-attention operation. For the convenience of the following description, $M_{c,t}^{l\rightarrow(l+3)}$ is transformed into the position set of all selected features:

$$\Omega_{c,t} = \left\{(i,j) | M_{c,t}^{l\rightarrow(l+3)}(i,j) = 1\right\} \tag{3}$$

Where $\Omega_{c,t}$ is the basic component of ST-Tube and $|\Omega_{c,t}|$ denotes the number of selected features of $\mathbf{x}_{c,t}$.

Step 2: tube self-attention operation After obtaining $\mathbf{X}$ and $\Omega_{c,t}$, the self-attention mechanism is performed to aggregate all features located in ST-Tube, as shown in Figure 4. The formation of the TSA mechanism adopted in this paper is consistent with [39]:

$$\mathbf{y}_p = \frac{1}{C(\mathbf{x})} \sum_{\forall c} \sum_{\forall t} \sum_{\forall (i,j)\in\Omega_{c,t}} f\left(\mathbf{x}_p, \mathbf{x}_{c,t}(i,j)\right) g(\mathbf{x}_{c,t}(i,j)) \tag{4}$$

Where p denotes the index of an output position whose response is to be computed. (c,t,i,j) is the input index that enumerates all positions in ST-Tube. Output feature map $\mathbf{y}$ and input feature map $\mathbf{x}$ have the same size. $f(\cdot,\cdot)$ denotes the pairwise function, and $g(\cdot)$ denotes the unary function. The response is normalized by $C(\mathbf{x}) = \sum_c \sum_t |\Omega_{c,t}|$.

To reduce the computational complexity, the dot product similarity function is adopted:

$$f\left(\mathbf{x}_p, \mathbf{x}_{c,t}(i,j)\right) = \theta(\mathbf{x}_p)^T \phi(\mathbf{x}_{c,t}(i,j)) \tag{5}$$

Where both $\theta(\cdot)$ and $\phi(\cdot)$ are channel reduction transformations.

Finally, the residual link is added to obtain the final $\mathbf{X}'$:

$$\mathbf{x}'_p = W_z \mathbf{y}_p + \mathbf{x}_p \tag{6}$$

Where $W_z \mathbf{y}_p$ denotes an embedding of $\mathbf{y}_p$. Note that $\mathbf{x}'_p$ has the same size with $\mathbf{x}_p$, so TSA module can be inserted into any position in deep convolutional neural networks. For the trade-off between computational cost and performance, all TSA modules are placed after *Mixed_4e*. Thus, $T = 4$ and $H = W = 14$.

Compared with the Non-local operation, the TSA module greatly reduces the computational complexity in time and space from

$$O\left((N\times T\times H\times W)\times(N\times T\times H\times W)\right) \tag{7}$$

to

$$O\left(\left(\sum_c \sum_t |\Omega_{c,t}|\right) \times \left(\sum_c \sum_t |\Omega_{c,t}|\right)\right) \tag{8}$$

Note that the computational cost of TSA can only be measured after forwarding propagation because $\Omega_{c,t}$ is generated from B.

3.3 Network Head and Training

To verify the effectiveness of the TSA module, we extend the network head to support multiple tasks, including classification, regression, and score distribution prediction. All tasks can be achieved by changing the output size of *MLP_block* and the definition of the loss function. The implementation details of these three tasks are as follows:

Figure 5: The tracking results and predicted scores of four cases from four datasets. Four manually annotated initial frames are coloured in yellow, and the subsequent boxes generated by SiamMask are coloured in green. The predicted scores of TSA-Net and GT scores are shown on the right. More visualization cases can be found in supplementary materials.

Classification. When dealing with classification tasks, the output dimension of *MLP_block* is determined by the number of categories. Binary Cross-Entropy loss (BCELoss) is adopted.

Regression. When dealing with regression tasks, the output dimension of *MLP_block* is set to 1. Mean Square Error loss (MSELoss) is adopted.

Score distribution prediction. Tang *et al.* [33] proposed an uncertainty-aware score distribution learning (USDL) approach and its multi-path version MUSDL for AQA tasks. Although experiment results in [33] proved the superiority of MUSDL compared with USDL, a multi-path strategy will lead to a significant increase in computational cost. However, the TSA module can generate features with rich contextual information by adopting a self-attention mechanism in ST-Tube with less computational complexity.

To verify the effectiveness of the TSA module, we embed the TSA module into the USDL model. The loss function is defined as Kullback-Leibler (KL) divergence of predicted score distribution and ground-truth (GT) score distribution:

$$KL\left\{p_c \parallel s_{pre}\right\} = \sum_{i=1}^{m} p(c_i) log \frac{p(c_i)}{s_{pre}(c_i)} \tag{9}$$

Where s_{pre} is generated by *MLP_block*, and p_c is generated by GT score. Note that for dataset with difficulty degree (DD), $s = DD \times s_{pre}$ is used as the final predicted score.

4 EXPERIMENTS

We carry out comprehensive experiments on AQA-7 [24], MTL-AQA [25], and FR-FS datasets to evaluate the proposed method. Experimental results demonstrate that TSA-Net achieves state-of-the-art performance on these datasets. In the following subsections, we first introduce two public datasets and a new dataset named Fall Detection in Figure Skating (FD-FS) proposed by us. After that, a series of experiments and computational complexity analysis are performed on AQA-7 and MTL-AQA datasets. Finally, the detection results on FD-FS are reported, and the network prediction results are analyzed visually and qualitatively.

4.1 Datasets and Evaluation Metrics

AQA-7 [24]. The AQA-7 dataset comprising samples from seven actions. It contains 1189 videos, in which 803 videos are used for training and 303 videos used for testing. To ensure the comparability with other models, we delete *trampoline* category because of its long time.

MTL-AQA [25]. The MTL-AQA dataset is currently the largest dataset for AQA tasks. There are 1412 diving samples collected from 16 different events in MTL-AQA. Furthermore, MTL-AQA provides detailed scoring of each referee, diving difficulty degree, and live commentary. We followed the evaluation protocol suggested in [25], so that there are 1059 samples used for training and 353 used for testing.

FR-FS (Fall Recognition in Figure Skating). Although some methods have been proposed [27, 40] to evaluate figure skating skills, they are only based on long-term videos which last nearly 3 minutes. These coarse-grained methods will lead to the inundation of detailed information in a long time scale. However, these details are crucial and indispensable for AQA tasks. To address this issue, we propose a dataset named FR-FS to recognize falls in figure skating sports. We plan to start from the most basic fault recognition and gradually build a more delicate granularity figure skating AQA system.

The FR-FS dataset contains 417 videos collected from FIV [40] and *Pingchang 2018 Winter Olympic Games*. FR-FS contains the critical movements of the athlete's take-off, rotation, and landing. Among them, 276 are smooth landing videos, and 141 are fall videos. To test the generalization performance of our proposed model, we randomly select 50% of the videos from the fall and landing videos as the training set and the testing set.

Table 1: Comparison with state-of-the-arts on AQA-7 Dataset.

Method	Diving	Gym Vault	Skiing	Snowboard	Sync. 3m	Sync. 10m	Avg. Corr.
Pose+DCT [27]	0.5300	-	-	-	-	-	-
ST-GCN [41]	0.3286	0.577	0.1681	0.1234	0.6600	0.6483	0.4433
C3D-LSTM [23]	0.6047	0.5636	0.4593	0.5029	0.7912	0.6927	0.6165
C3D-SVR [23]	0.7902	0.6824	0.5209	0.4006	0.5937	0.9120	0.6937
JRG [22]	0.7630	0.7358	0.6006	0.5405	0.9013	0.9254	0.7849
USDL [33]	0.8099	0.757	0.6538	**0.7109**	0.9166	0.8878	0.8102
NL-Net	0.8296	0.7938	**0.6698**	0.6856	0.9459	0.9294	0.8418
TSA-Net (Ours)	**0.8379**	**0.8004**	0.6657	0.6962	**0.9493**	**0.9334**	**0.8476**

Table 2: Study on different settings of the number of TSA module.

Method	Diving	Gym Vault	Skiing	Snowboard	Sync. 3m	Sync. 10m	Avg. Corr.
TSA-Net	0.8379	0.8004	0.6657	0.6962	**0.9493**	0.9334	0.8476
TSAx2-Net	0.8380	0.7815	**0.6849**	**0.7254**	0.9483	**0.9423**	**0.8526**
TSAx3-Net	**0.8520**	**0.8014**	0.6437	0.6619	0.9331	0.9249	0.8352

Table 3: Comparisons of computational complexity and performance on AQA-7. GFLOPs is adopted to measure the computational cost.

Method	NL-Net	TSA-Net	Comp. Dec.	Corr. Imp.
Diving	2.2G	0.864G	-60.72%	↑0.0083
Gym Vault	2.2G	0.849G	-61.43%	↑0.0066
Skiing	2.2G	0.283G	-87.13%	↓0.0041
Snowboard	2.2G	0.265G	-87.97%	↑0.0106
Sync. 3m	2.2G	0.952G	-56.74%	↑0.0034
Sync. 10m	2.2G	0.919G	-58.24%	↑0.0040
Average	2.2G	0.689G	-68.70%	↑0.0058

Table 4: Comparison with state-of-the-arts on MTL-AQA.

Method	Avg. Corr.
Pose+DCT [27]	0.2682
C3D-SVR [23]	0.7716
C3D-LSTM [23]	0.8489
C3D-AVG-STL [25]	0.8960
C3D-AVG-MTL [25]	0.9044
MUSDL [33]	0.9273
NL-Net	**0.9422**
TSA-Net	0.9393

Evaluation Protocols. Spearman's rank correlation is adopted as the performance metric to measure the divergence between the GT score and the predicted score. The Spearman's rank correlation is defined as follows:

$$\rho = \frac{\sum(p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum(p_i - \bar{p})^2 \sum(q_i - \bar{q})^2}} \tag{10}$$

Where p and q represent the ranking of GT and predicted score series, respectively. Fisher's z-value [23] is used to measure the average performance across multiple actions.

4.2 Implementation Details

Our proposed methods were built on the Pytorch toolbox [26] and implemented on a system with the Intel (R) Xeon (R) CPU E5-2698 V4 @ 2.20GHz. All models are trained on a single NVIDIA Tesla V100 GPU. Faster-RCNN[30] pretrained on MS-COCO[19] is adopted to detect the athletes in all initial frames. All videos are normalized to $L = 103$ frames. For all experiments, the I3D[1] pretrained on Kinetics [16] is utilized as the feature extractor. All videos are select from high-quality sports broadcast videos, and the athletes' movements are apparent. Therefore, we argue that the performance of TSA-Net is not sensitive to the choice of the tracker. SiamMask[38] was chosen only for high-speed and tight boxes. Each training mini-batch contains 4 samples. Adam [17] optimizer was adopted for network optimization with initial learning rate 1e-4, momentum 0.9, and weight decay 1e-5.

Considering the complexity differences between datasets, we adopt different experimental settings. In AQA-7 and MTL-AQA datasets, all videos are divided into 10 clips consistent with [33]. Random horizontal flipping and timing offset are performed on videos in training phase. Training epoch is set to 100. All video score normalization are consistent with USDL [33]. In FR-FS dataset, all videos are divided into 7 segments to prevent overfitting. Training epoch is set to 20.

4.3 Results on AQA-7 Dataset

The TSA module and the Non-local module are embedded after *Mixed_4e* of I3D to create TSA-Net and NL-Net. Experimental results in Table 1 show that TSA-Net achieves 0.8476 on Avg. Corr. , which is higher than 0.8102 of USDL. TSA-Net outperforms USDL in all categories except the *snowboard*. This is mainly caused by

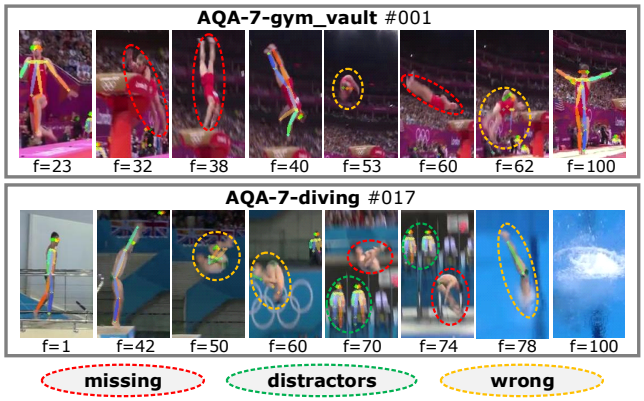

Figure 6: Alphapose [7] is selected as the pose estimator. The estimation results of two sports videos are visualized.

the size issue: the small size of the target leads to the small size of the ST-Tube, resulting in invalid feature enhancement (AQA-7 *snow. #056* in Figure 5). Note that the TSA module is used in a plug-and-play fashion, comparative experiments in Table 1 can also be regarded as ablation studies. Therefore, we didn't set up a separate part of ablation in this paper.

The effect of different number of TSA module. Inspired by the multi-layer attention mechanism in Transformer [36], we stack multiple TSA modules and test these variants on AQA-7. Experimental results in Table 2 show that TSA-Net achieves the best performance when $N_{stack} = 2$. Benefit from the feature aggregation operations conducted by two subsequent TSA modules, the network can capture richer contextual features compared with USDL. When $N_{stack} = 3$, the performance of the model becomes worse, which may be caused by overfitting.

Computational cost analysis. Computational cost comparison results are shown in Table 3. Note that only the calculation of TSA module or Non-local module is counted, not the whole network. Compared with NL-Net, TSA-Net can reduce the computation by 68.7% on average and bring 0.0058 AVG. Corr. improvement. This is attributed to the tube mechanism adopted in TSA module, which can avoid dense attention calculation and improve performance simultaneously. Among all categories in AQA-7, the TSA module saves up to 87% of the computational complexity on *skiing* and *snowboard*. Such a large reduction is caused by the small size of the ST-Tube. However, the small ST-Tube will hinder the network from completing effective feature aggregation and ultimately affect the final performance. This conclusion is consistent with the analysis of the results in Table 1.

4.4 Results on MTL-AQA Dataset

As shown in Table 4, the TSA-Net and NL-Net is compared with existing methods. Regression network head and MSELoss are adopted in two networks. Experimental results show that both TSA-Net and NL-Net can achieve state-of-the-art performance and the NL-Net is better. The performance fluctuation of TSA-Net is mainly caused by different data distribution between two datasets. Videos in MTL-AQA have higher resolution (640x360 to 320x240) and broader field of view, which leads to smaller ST-Tubes in TSA-Net and affects the performance. It should be emphasized that this impact is feeble. TSA-Net saves half of the computational cost and achieves almost the same performance as NL-Net. This proves the effectiveness and efficiency of the TSA-Net, which is not contradictory to the final conclusion.

Table 5: Comparisons of computational complexity and performance between NL-Net and the variants of TSA-Net on MTL-AQA.

Method	Sp. Corr.↑	MSE↓	FLOPs↓
NL-Net	**0.9422**	47.83	2.2G
TSA-Net	0.9393	**37.90**	**1.012**G
TSAx2-Net	0.9412	46.51	2.025G
TSAx3-Net	0.9403	47.77	3.037G

Table 6: Recognition accuracy on FR-FS.

Method	Acc.
Plain-Net	94.23
TSA-Net	**98.56**

Studys on the the stack number of TSA modules and computational cost. As shown in Tabel 5, three parallel experiments are conducted with only the number of TSA module changed just as the experiments on AQA-7. If NL-Net is excluded, the best *Sp. Corr.* is achieved when $N_{clip} = 2$ (*i.e.*, TSAx2-Net), while TSA-Net with only one TSA module achieves minimum MSE and computational cost simultaneously. This phenomenon is mainly caused by low computational complexity of TSA-Net. The sparse feature interaction characteristics of TSA module achieve more efficient feature enhancement and have the ability to avoid overfitting. Although the performance of TSA-Net can be improved by increasing the number of TSA modules, it will increase computational cost. To achieve the balance between computational cost and performance, we only take $N_{stack} = 1$ in all subsequent experiments.

4.5 Results on FR-FS Dataset

In FR-FS dataset, we focus on the performance improvement that the TSA module can achieve. Therefore, Plain-Net and TSA-Net are implemented, respectively. The former does not adopt any feature enhancement mechanism, while the latter is equipped with a TSA module. As shown in Table 6, TSA-Net outperforms Plain-Net by 4.33%, which proves the effectiveness of TSA module.

Visualization of Temporal Evolution. A case study is also conducted to further explore the performance of TSA-Net. Two representative videos are selected and the prediction results of each video clip are visualized in Figure 7, . All clip scores are obtained by deleting the temporal pooling operation in Plain-Net and TSA-Net. In the failure case #308-1, both Plain-Net and TSA-Net can detect that the athlete falls in the fourth chip which highlighted in red, but only TSA-Net gets the correct result in the end (0.9673 for Plain-Net and 0.2523 for TSA-Net). The TSA mechanism forces the features in ST-Tube interact with each other in the way of self-attention,

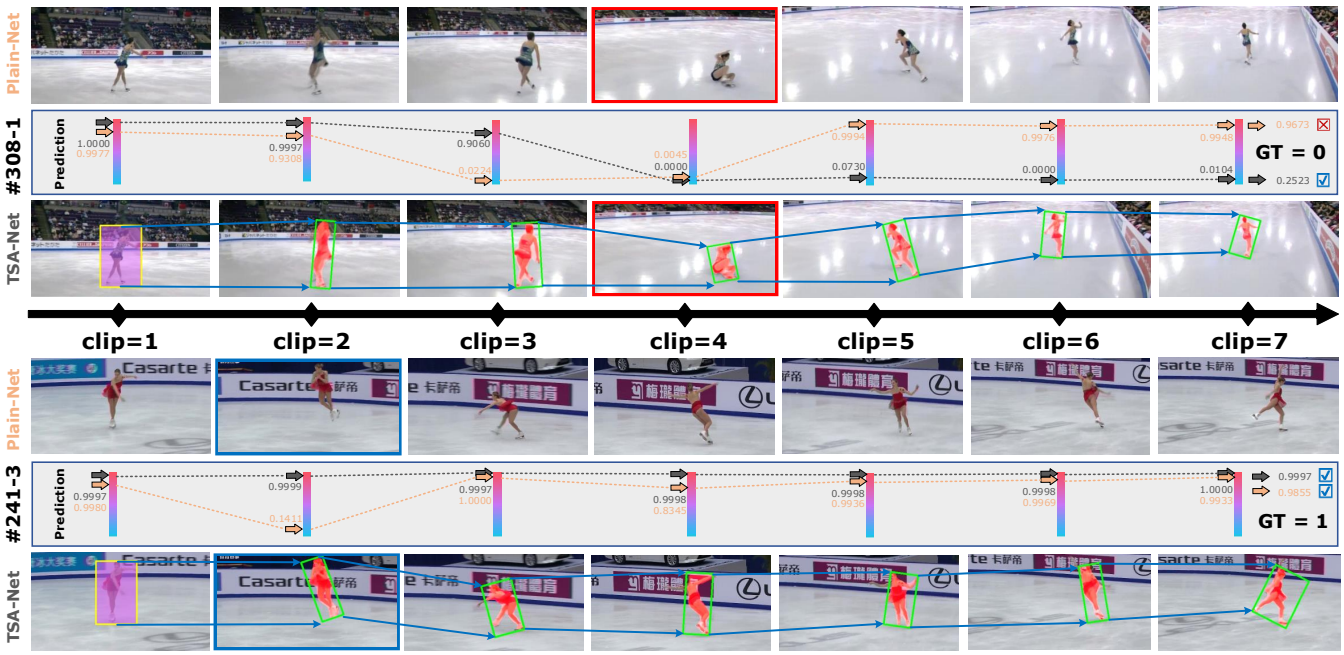

Figure 7: Case study with qualitative results on FR-FS. The failure case #308-1 is above the timeline, while the successful case #241-3 is below the timeline.

which makes TSA-Net regard the standing up and adjusting actions after falling as errors in clip 5 to 7.

It seems that TSA-Net is too strict in fall recognition, but the analysis in the successful case #241-3 has overturned this view. Two models get similar results, except for the second clip (colored in blue), which contains the take-off and rotation phase. Plain-Net has great uncertainty for the stationarity of take-off phase, while TSA-Net can get high confidence results. Based on visual analysis and quantitative analysis, it can be concluded that the TSA module is able to perform feature aggregation effectively and obtain more reasonable and stable prediction results.

4.6 Analysis and Visualization

Reasons for choosing tracking boxes over pose estimation. In sports scenes, high-speed movements of the human body will lead to a series of challenges such as motion blur and self-occlusion, which eventually result in failure cases in pose estimation. Results in Figure 6 show that Alphapose [7] cannot handle these situations properly. The missing of human posture and background audience posture interference will seriously affect the evaluation results. Previous studies in FineGym [31] have come to the same results as ours. Based on these observations, we conclude that methods based on pose estimation are not suitable for AQA in sports scenes. Missing boxes and wrong poses will significantly limit the performance of the AQA model. Therefore, we naturally introduce the VOT tracker into AQA tasks. The proposed TSA-Net achieves significant improvement in AQA-7 and MTL-AQA compared to pose-based methods such as Pose+DCT [27] and ST-GCN [41] as shown in Table 1 and 4. These comparisons show that the TSA mechanism is superior to the posture-based mechanism in capturing key dynamic characteristics of human motion.

Visualization on MTL-AQA and AQA-7. Four cases are visualized in Figure 5. The tracking results generated by SiamMask are very stable and accurate. The final predicted scores are very close to the GT score since the TSA module is adopted. Interestingly, as shown in Figure 5, the VOT tracker can handle various complex situations, such as the disappearance of athletes (#02-32), drastic changes in scale (#056) and synchronous diving (#082). These results show that the tracking strategy perfectly meets the requirements of AQA tasks and verify the effectiveness of the TSA module.

5 CONCLUSION

In this paper, we present a Tube Self-Attention Network (TSA-Net) for action quality assessment, which is able to capture rich spatio-temporal contextual information in human motion. Specifically, a tube self-attention module is introduced to efficiently aggregate contextual information located in ST-Tube generated by SiamMask. Sufficient experiments are performed on three datasets: AQA-7, MTL-AQA, and a dataset proposed by us named FR-FS. Experimental results demonstrate that TSA-Net can capture long-range contextual information and achieve high performance with less computational cost. In the future, an adaptive mechanism of ST-Tube will be explored to avoid the sensitivity of the TSA-Net to the size issue.

ACKNOWLEDGMENTS

This work was supported by Shanghai Municipal Science and Technology Major Project 2021SHZDZX0103 and National Natural Science Foundation of China under Grant 82090052.

REFERENCES

[1] Joao Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4724–4733.

[2] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. 2018. A2-Nets: Double Attention Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*. 350–359.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.

[4] Hazel Doughty, Dima Damen, and Walterio Mayol-Cuevas. 2018. Who's Better? Who's Best? Pairwise Deep Ranking for Skill Determination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6057–6066.

[5] Hazel Doughty, Walterio Mayol-Cuevas, and Dima Damen. 2019. The pros and cons: Rank-aware temporal attention for skill determination in long videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7854–7863.

[6] Qi Fan, Wei Zhuo, Chi Keung Tang, and Yu Wing Tai. 2020. Few-Shot Object Detection With Attention-RPN and Multi-Relation Detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. 2017. RMPE: Regional Multi-person Pose Estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2353–2362.

[8] Christoph Feichtenhofer. 2020. X3D: Expanding Architectures for Efficient Video Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 200–210.

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2016. Convolutional Two-Stream Network Fusion for Video Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1933–1941.

[10] Kai Han, Jianyuan Guo, Chao Zhang, and Mingjian Zhu. 2018. Attribute-Aware Attention Model for Fine-Grained Representation Learning. In *Proceedings of ACM international conference on Multimedia (ACM MM)*. 2040–2048.

[11] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. 2018. Relation Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[12] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. 2020. Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020), 2011–2023.

[13] Zilong Huang, Xinggang Wang, Chang Huang, Yunchao Wei, Lichao Huang, , and Wenyu Liu. 2020. CCNet: Criss-Cross Attention for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1–1.

[14] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2010. 3D Convolutional Neural Networks for Human Action Recognition. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*. 495–502.

[15] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-Scale Video Classification with Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1725–1732.

[16] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The Kinetics Human Action Video Dataset. , Article arXiv:1705.06950 (2017). arXiv:1705.06950

[17] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

[18] Ji Lin, Chuang Gan, and Song Han. 2019. TSM: Temporal Shift Module for Efficient Video Understanding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 7082–7092.

[19] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the IEEE International Conference on Computer Vision (ECCV)*.

[20] Anand Malpani, S. Swaroop Vedula, Chi Chiung Grace Chen, and Gregory D. Hager. 2014. Pairwise Comparison-Based Objective Score for Automated Skill Assessment of Segments in a Surgical Task. In *Information Processing in Computer-Assisted Interventions*. 138–147.

[21] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent Models of Visual Attention. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2204–2212.

[22] Jia-Hui Pan, Jibin Gao, and Wei-Shi Zheng. 2019. Action Assessment by Joint Relation Graphs. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 6330–6339.

[23] Paritosh Parmar and Brendan Tran Morris. 2017. Learning to Score Olympic Events. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 76–84.

[24] Paritosh Parmar and Brendan Tran Morris. 2019. Action quality assessment across multiple actions. In *Winter Conference on Applications of Computer Vision (WACV)*. 1468–1476.

[25] Paritosh Parmar and Brendan Tran Morris. 2019. What and How Well You Performed? A Multitask Learning Approach to Action Quality Assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 304–313.

[26] Adam Paszke, S. Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito, Zeming Lin, Alban Desmaison, L. Antiga, and A. Lerer. 2017. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems Workshops (NIPSW)*.

[27] Hamed Pirsiavash, Antonio Torralba, and Carl Vondrick. 2014. Assessing the Quality of Actions. In *Proceedings of the IEEE International Conference on Computer Vision (ECCV)*. 556–571.

[28] A. Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training.

[29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language Models are Unsupervised Multitask Learners. (2018).

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 39, 6 (2017), 1137–1149.

[31] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. 2020. FineGym: A Hierarchical Video Dataset for Fine-grained Action Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2613–2622.

[32] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[33] Yansong Tang, Zanlin Ni, Jiahuan Zhou, Danyang Zhang, Jiwen Lu, Ying Wu, and Jie Zhou. 2020. Uncertainty-Aware Score Distribution Learning for Action Quality Assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 9836–9845.

[34] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 4489–4497.

[35] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6450–6459.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*. 6000–6010.

[37] Vinay Venkataraman, Ioannis Vlachos, and Pavan Turaga. 2015. Dynamical Regularity for Action Analysis. In *British Machine Vision Virtual Conference (BMVC)*. 67.1–67.12.

[38] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H.S. Torr. 2019. Fast Online Object Tracking and Segmentation: A Unifying Approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1328–1338.

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7794–7803.

[40] Chengming Xu, Yanwei Fu, Bing Zhang, Zitian Chen, Yu-Gang Jiang, and Xiangyang Xue. 2020. Learning to Score Figure Skating Sport Videos. *IEEE Transactions on Circuits and Systems for Video Technology* (2020), 4578–4590.

[41] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 7444–7452.

[42] Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. 2018. Compact Generalized Non-Local Network. In *Advances in Neural Information Processing Systems (NeurIPS)*. 6511–6520.

[43] Dong Zhang, Hanwang Zhang, Jinhui Tang, Meng Wang, Xiansheng Hua, and Qianru Sun. 2020. Feature Pyramid Transformer. In *Proceedings of the IEEE International Conference on Computer Vision (ECCV)*. 323–339.

[44] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. 2018. PSANet: Point-wise Spatial Attention Network for Scene Parsing. In *Proceedings of the IEEE International Conference on Computer Vision (ECCV)*. 270–286.

[45] Aneeq Zia, Yachna Sharma, Vinay Bettadapura, Eric L. Sarin, Thomas Ploetz, Mark A. Clements, and Irfan Essa. 2016. Automated video-based assessment of surgical skills for training and evaluation in medical schools. *International Journal of Computer Assisted Radiology and Surgery* (2016), 1623–1636.

[46] Aneeq Zia, Chi Zhang, Xiaobin Xiong, and Anthony M Jarc. 2017. Temporal clustering of surgical activities in robot-assisted surgery. *International Journal of Computer Assisted Radiology and Surgery* (2017), 1171–1178.