Action Quality Assessment with Temporal Parsing Transformer

Yang Bai[1,2*†], Desen Zhou[2*], Songyang Zhang[3], Jian Wang[2], Errui Ding[2], Yu Guan[4], Yang Long[1], and Jingdong Wang[2‡]

[1] Department of Computer Science, Durham University
[2] Department of Computer Vision Technology (VIS), Baidu Inc.
[3] Shanghai AI Laboratory, [4] University of Warwick
{yang.bai,yang.long}@durham.ac.uk,{zhoudesen,wangjingdong}@baidu.com

Abstract. Action Quality Assessment(AQA) is important for action understanding and resolving the task poses unique challenges due to subtle visual differences. Existing state-of-the-art methods typically rely on the holistic video representations for score regression or ranking, which limits the generalization to capture fine-grained intra-class variation. To overcome the above limitation, we propose a temporal parsing transformer to decompose the holistic feature into temporal part-level representations. Specifically, we utilize a set of learnable queries to represent the atomic temporal patterns for a specific action. Our decoding process converts the frame representations to a fixed number of temporally ordered part representations. To obtain the quality score, we adopt the state-of-the-art contrastive regression based on the part representations. Since existing AQA datasets do not provide temporal part-level labels or partitions, we propose two novel loss functions on the cross attention responses of the decoder: a ranking loss to ensure the learnable queries to satisfy the temporal order in cross attention and a sparsity loss to encourage the part representations to be more discriminative. Extensive experiments show that our proposed method outperforms prior work on three public AQA benchmarks by a considerable margin.

Keywords: action quality assessment, temporal parsing transformer, temporal patterns, contrastive regression

1 Introduction

Action quality assessment(AQA), which aims to evaluate how well a specific action is performed, has attracted increasing attention in research community recently [18, 17]. In particular, assessing the action quality accurately has great potential in a wide range of applications such as health care [34] and sports analysis [2, 20, 19, 18].

In contrast to the conventional action recognition tasks [25, 4], AQA poses unique challenges due to the subtle visual differences. Previous works on AQA

* Equal contribution. † Work done when Yang Bai was a research intern at VIS, Baidu.
‡ Corresponding Author.

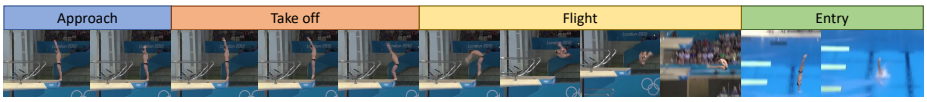

Fig. 1: An action consists of multiple temporally ordered key phases.

either use ranking-based pairwise comparison between test videos[5] or estimate the quality score with regression-based methods[19, 29]. However, these methods typically represent a video with its *holistic representation*, via the global pooling operation over the output of the backbone network(e.g., I3D[4]). Since the videos to be evaluated usually are from the same coarse action category (e.g., diving) in AQA, it's crucial to capture *fine-grained intra-class variation* to estimate more accurate quality scores. Thus, we propose to decompose the holistic feature into more fine-grained temporal part-level representations for AQA.

To achieve this, a promising strategy is to represent the video by using a set of atomic action patterns. For example, a diving action consists of several key phases, such as *approach*, *take off*, *flight*, etc., as illustrated in Fig.1. The fine-grained patterns enable the model to describe the subtle differences, which is expected to improve the assessment of action quality effectively. Nevertheless, it remains challenging to learn such atomic patterns as the existing AQA datasets do not provide temporal part-level labels or partitions.

In this work, we aim to tackle the aforementioned limitations by developing a regression-based action quality assessment strategy, which enables us to leverage the fine-grained atomic action patterns without any explicit part-level supervision. Our key idea is to model the shared atomic temporal patterns, with a set of learnable queries for a specific action category. Similar to the decoding process of transformer applied in natural language modeling[24], we propose a temporal parsing transformer to decode each video into a fixed number of part representations. To obtain quality scores, we adopt the recent state-of-the-art contrastive regression framework[32]. Our decoding mechanism allows the part representations between test video and exemplar video to be implicitly aligned via a shared learnable query. Then, we generate a relative pairwise representation per part and fuse different parts together to perform the final relative score regression.

To learn the atomic action patterns without the part-level labels, we propose two novel loss functions on the cross attention responses of the decoder. Specifically, to ensure the learnable queries satisfy the temporal order in cross attention, we calculate an attention center for each query by weighted summation of the attention responses with their temporal clip orders. Then we adopt a marginal ranking loss on the attention centers to guide the temporal order. Moreover, we propose a sparsity loss for each query's attention distribution to guide the part representations to be more discriminative.

We evaluate our method, named as temporal parsing transformer(TPT), on three public AQA benchmarks: MTL-AQA[18], AQA-7[17] and JIGSAWS[7].

As a result, our method outperforms previous state-of-the-art methods by a considerable margin. The visualization results show that our method is able to extract part-level representations with interpretable semantic meanings. We also provide abundant ablation studies for better understanding.

The main contributions of this paper are three folds:

- We propose a novel temporal parsing transformer to extract fine-grained temporal part-level representations with interpretable semantic meanings, which are optimized with the contrastive regression framework.
- We propose two novel loss functions on the transformer cross attentions to learn the part representations without the part-level labels.
- We achieve the new state-of-the-art on three public AQA benchmarks, namely MTL-AQA, AQA-7 and JIGSAWS.

2 Related Work

2.1 Action quality assessment

In the past years, the field of action quality assessment (AQA) has been repaid developed with a broad range of applications such as health care[34], instructional video analysis[5, 6], sports video analysis[17, 18], and many others[8, 9]. Existing AQA methods can be categorized into two types: regression based methods and ranking based methods.

Regression based methods Mainstream AQA methods formulate the AQA task as a regression task based on reliable score labels, such as scores given by expert judges of sports events. For example, Pirsiavash et al. [20] took the first steps towards applying the learning method to the AQA task and trained a linear SVR model to regress the scores of videos based on handcrafted features. Gordan et al. [8] proposed in their pioneer work the use of skeleton trajectories to solve the problem of quality assessment of gymnastic vaulting movements. Parmar et al. [19] showed that spatiotemporal features from C3D [23] can better encode the temporal representation of videos and significantly improve AQA performance. They also propose a large-scale AQA dataset and explore all-action models to better evaluate the effectiveness of models proposed by the AQA community. Xu et al. [29] proposed learning multi-scale video features by stacked LSTMs followed [19]. Pan et al. [16] proposed using spatial and temporal graphs to model the interactions between joints. Furthermore, they also propose to use I3D [4] as a stronger backbone network to extract spatiotemporal features. Parmar et al. [18] introduced the idea of multi-task learning to improve the model capacity of AQA, and collected AQA datasets with more annotations to support multi-task learning. To diminish the subjectiveness of the action score from human judges, Tang et al. [22] proposed an uncertainty-aware score distribution learning (USDL) framework Recently. However, the video's final score can only provide weak supervision concerning action quality. Because two videos with different low-quality parts are likely to share similar final scores, which means the score couldn't provide discriminative information.

Ranking based methods Another branch formulates AQA task as a ranking problem. Doughty et al. [5] proposed a novel loss function that learns discriminative features when a pair of videos exhibit variance in skill and learns shared features when a pair of videos show comparable skill levels. Doughty et al. [6] used a novel rank-aware loss function to attend to skill-relevant parts of a given video. However, they mainly focus on longer, more ambiguous tasks and only predict overall rankings, limiting AQA to applications requiring some quantitative comparisons. Recently, Yu et al. [32] proposed the Contrastive Regression (CoRe) framework to learn the relative scores by pair-wise comparison, highlighting the differences between videos and guiding the models to learn the key hints for assessment.

2.2 Temporal action parsing

Fine-grained action parsing is also studied in the field of action segmentation or temporal parsing [11, 10, 1, 12, 13, 31]. For example, Zhang et al.[33] proposed Temporal Query Network adopted query-response functionality that allows the query to attend to relevant segments. Dian et al.[21] proposed a temporal parsing method called TransParser that is capable of mining sub-actions from training data without knowing their labels. However, different from the above fields, part-level labels are not available in AQA task. Furthermore, most of the above methods focus more on frame-level feature enhancement, whereas our proposed method extracts part representations with interpretable semantic meanings.

3 Method

In this section, we introduce our temporal parsing transformer with the contrastive regression framework in detail.

3.1 Overview

The input of our network is an action video. We adopt the Inflated 3D ConvNets(I3D)[4] as our backbone, which first applies a sliding window to split the video into T overlapping clips, where each clip contains M consecutive frames. Then, each clip goes through the I3D network, resulting in time series clip level representations $\boldsymbol{V} = \{\boldsymbol{v}_t \in \mathbb{R}^D\}_{t=1}^T$, where D is feature dimension and T is the total number of clips. In our work, we do not explore spatial patterns, hence each clip representation $\boldsymbol{v}_t$ is obtained by average pooling across spatial dimensions. The goal of AQA is to estimate a quality score $\mathbf{s}$ based on the resulting clips representation $\boldsymbol{V}$. In contrastive regression framework, instead of designing a network to directly estimate raw score $\mathbf{s}$, it estimates a relative score between the test video and an exemplar video $\boldsymbol{V}_0$ with known quality score $\mathbf{s}_0$, which is usually sampled from training set. Then, contrastive regression aims to design a network $\mathcal{F}$ that estimates the relative score $\boldsymbol{\Delta s}$:

$$\boldsymbol{\Delta}\mathbf{s} = \mathcal{F}(\boldsymbol{V}, \boldsymbol{V}_0), \tag{1}$$

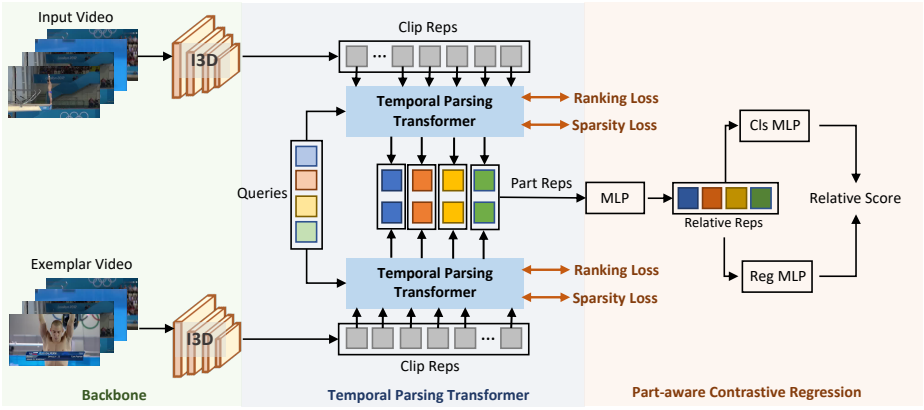

Fig. 2: Overview of our framework. Our temporal parsing transformer converts the clip-level representations into temporal part-level representations. Then the part-aware contrastive regressor first computes part-wise relative representations and then fuses them to estimate the relative score. We adopt the group-aware regression strategy, following[32]. During training, we adopt the ranking loss and sparsity loss on the decoder cross attention maps to guide the part representation learning.

then final score can be obtained by

$$\mathbf{s} = \mathbf{s}_0 + \Delta\mathbf{s}. \tag{2}$$

In our framework, we first adopt a temporal parsing transformer $\mathcal{G}$ to convert the clip level representations $\boldsymbol{V}$ into temporal part level representations, denoted by $\boldsymbol{P} = \{\boldsymbol{p}_k \in \mathbb{R}^d\}_{k=1}^K$, where d is the part feature dimension and K is the number of queries, i.e. temporal atomic patterns. Then for test video and exemplar video, we can have two set of aligned part representations $\boldsymbol{P}$ and $\boldsymbol{P}_0 = \{\boldsymbol{p}_k^0 \in \mathbb{R}^d\}_{k=1}^K$. Our new formulation can be expressed as:

$$\Delta\mathbf{s} = \mathcal{R}(\boldsymbol{P}, \boldsymbol{P}_0). \tag{3}$$

where $\mathcal{R}$ is the relative score regressor, and

$$\boldsymbol{P} = \mathcal{G}(\boldsymbol{V}), \boldsymbol{P}_0 = \mathcal{G}(\boldsymbol{V}_0). \tag{4}$$

An overview of our framework is illustrated in Fig.2. Below we describe the detailed structure of temporal parsing transformer $\mathcal{G}$ and part-aware contrastive regressor $\mathcal{R}$.

3.2 Temporal parsing transformer

Our temporal parsing transformer takes the clip representations as memory and exploits a set of learnable queries to decode part representations. Different from

prevalent DETR architecture[3], our transformer only consists of a decoder module. We found that the encoder module does not provide improvements in our framework; it even hurts the performance. We guess it might because that clip-level self-attention smooths the temporal representations, and our learning strategy cannot decode part presentations in this way without part labels.

We perform slight modifications to the standard DETR decoder. That is, the cross attention block in our decoder has a learnable parameter, temperature, to control the amplification of the inner product. Formally, in the i-th decoder layer, the decoder part feature $\{p_k^{(i)} \in \mathbb{R}^d\}$ and learnable atomic patterns(i.e. query set) $\{q_k \in \mathbb{R}^d\}$ are first summed as a query and then perform cross attention on the embedded clip representation $\{\mathbf{v}_t \in \mathbb{R}^d\}$:

$$\alpha_{k,t} = \frac{\exp^{(\boldsymbol{p}_k^{(i)}+q_k)^T \cdot \boldsymbol{v}_t/\tau}}{\sum_{j=1}^{T} \exp^{(\boldsymbol{p}_k^{(i)}+q_k)^T \cdot \boldsymbol{v}_j/\tau}}, \tag{5}$$

where $\alpha_{k,t}$ indicates the attention value for query k to clip t, $\tau \in \mathbb{R}$ indicates the learnable temperature to enhance the inner product to make the attentions more discriminative. Unlike DETR[3], in our decoder, we do not utilize position embedding of clip id to the memory $\{\boldsymbol{v}_t\}$. We expect our query to represent atomic patterns, instead of spatial anchors, as in the detection task[28, 14]. We found that adding position encoding significantly drops the performance and makes our learning strategy fail, which will be shown in the experiment section.

In our experiments, we only utilize one-head attention in our cross attention blocks. The attention values are normalized across different clips, since our goal is to aggregate clip representations into our part representation. Then the updated part representation $\boldsymbol{p}_k^{(i)'}$ has the following form:

$$\boldsymbol{p}_k^{(i)'} = \sum_{j=1}^{T} \alpha_{k,j} \boldsymbol{v}_j + \boldsymbol{p}_k^{(i)}. \tag{6}$$

We then perform standard FFN and multi-head self-attention on decoder part representations. Similar to DETR[3], our decoder also has a multi-layer structure.

3.3 Part-aware contrastive regression

Our temporal parsing transformer converts the clip representations $\{\boldsymbol{v}_t\}$ into part representations $\{\boldsymbol{p}_k\}$. Given a test video and exemplar video, we can obtain two part representation sets $\{\boldsymbol{p}_k\}$ and $\{\boldsymbol{p}_k^0\}$. One possible way to estimate the relative quality score is to fuse each video's part representations and estimate the relative score. However, since our temporal parsing transformer allows the extracted part representations to be semantically aligned with the query set, we can compute the relative pairwise representation per part and then fuse them

together. Formally, we utilize a multi-layer perceptron(MLP) f_r to generate the relative pairwise representation $\boldsymbol{r}_k \in \mathbb{R}^d$ for k-th part:

$$\boldsymbol{r}_k = f_r(Concat([\boldsymbol{p}_k; \boldsymbol{p}_k^0])). \tag{7}$$

The MLP f_r is shared across different parts. To balance the score distributions across the whole score range, we adopt the group-aware regression strategy to perform relative score estimation[32]. Specifically, it first calculates B relative score intervals based on all possible pairs in training set, where each interval has equal number of pair-samples. Then it generates a one-hot classification label $\{l_n\}$, where l_n indicates whether the ground truth score $\Delta\mathbf{s}$ lies in n-th interval, and a regression target $\gamma_n = \frac{\Delta\mathbf{s} - x_{left}^n}{x_{right}^n - x_{left}^n}$, where x_{left}^n, x_{right}^n denote the left and right boundary of n-th interval. Readers can refer to [32] for more details.

We adopt average pooling[1] on the relative part representations $\{\boldsymbol{r}_k\}$ and then utilize two two-layer MLPs to estimate the classification label $\{l_n\}$ and regression target $\{\gamma_n\}$. Different from [32], we do not utilize tree structure. Since we have obtained fine-grained part-level representations and hence the regression becomes simpler, we found that two-layer MLP works fine.

3.4 Optimization

Since we do not have any part-level labels at hand, it's crucial to design proper loss functions to guide the part representation learning. We have assumed that each coarse action has a set of temporally ordered atomic patterns, which are encoded in our transformer queries. To ensure that our query extracts different part representations, we constrain the attention responses in cross attention blocks for different queries. Specifically, in each cross attention process, we have calculated the normalized attention responses $\{\alpha_{k,t}\}$ by Eq.5, then we compute an attention center $\bar{\alpha}_k$ for k-th query:

$$\bar{\alpha}_k = \sum_{t=1}^{T} t \cdot \alpha_{k,t}, \tag{8}$$

where T is the number of clips and $\sum_{t=1}^{T} \alpha_{k,t} = 1$. Then we adopt two loss functions on the attention centers: ranking loss and sparsity loss.

Ranking loss To encourage that each query attends to different temporal regions, we adopt a ranking loss on the attention centers. We wish our part representations have a consistent temporal order across different videos. To this end,

[1] We note that it might be better to weight each part. However, part weighting does not provide improvements during our practice. We guess that it may be during the self-attention process in the decoder, the relations between parts have already been taken into account.

we define an order on the query index and apply ranking losses to the corresponding attention centers. We exploit the margin ranking loss, which results in the following form:

$$L_{rank} = \sum_{k=1}^{K-1} \max(0, \bar{\alpha}_k - \bar{\alpha}_{k+1} + m) + \max(0, 1 - \bar{\alpha}_1 + m) + \max(0, \bar{\alpha}_K - T + m), \tag{9}$$

where m is the hyper-parameter margin controlling the penalty, the first term guides the attention centers of part k and $k+1$ to keep order: $\bar{\alpha}_k < \bar{\alpha}_{k+1}$. From Eq. 8, we have the range of attention centers: $1 \leq \bar{\alpha}_k \leq T$. To constrain the first and last part where $k = 1$ and $k = K$, we assume there is two virtual centers at boundaries: $\bar{\alpha}_0 = 1$ and $\bar{\alpha}_{K+1} = T$. The last two terms in Eq. 9 constrain the first and last attention centers not collapsed to boundaries.

Sparsity loss To encourage the part representations to be more discriminative, we further propose a sparsity loss on the attention responses. Specifically, for each query, we encourage the attention responses to focus on those clips around the center μ_k, resulting in the following form:

$$L_{sparsity} = \sum_{k=1}^{K} \sum_{t=1}^{T} |t - \bar{\alpha}_k| \cdot \alpha_{k,t} \tag{10}$$

During training, our ranking loss and sparsity loss are applied to the cross attention block in each decoder layer.

Overall training loss In addition to the above auxiliary losses for cross attention, our contrastive regressor $\mathcal{R}$ generates two predictions for the group classification label $\{l_n\}$ and regression target $\{\gamma_n\}$, we follow [32] to utilize the BCE loss on each group and square error on the ground truth regression interval:

$$L_{cls} = -\sum_{n=1}^{N} l_n \log(\tilde{l}_n) + (1 - l_n) \log(1 - \tilde{l}_n) \tag{11}$$

$$L_{reg} = \sum_{n=1}^{N} \mathbb{1}(l_n = 1)(\gamma_n - \tilde{\gamma}_n)^2 \tag{12}$$

where L_{reg} only supervises on the ground truth interval, $\tilde{l}_n$ and $\tilde{\gamma}_n$ are predicted classification probability and regression value. The overall training loss is given by:

$$L_{all} = \lambda_{cls} L_{cls} + \lambda_{reg} L_{reg} + \lambda_{rank} \sum_{i=1}^{L} L^i_{rank} + \lambda_{sparsity} \sum_{i=1}^{L} L^i_{sparsity}, \tag{13}$$

where i indicates layer id and L is the number of decoder layers, λ_{cls}, λ_{reg}, λ_{rank}, $\lambda_{sparsity}$ are hyper-parameter loss weights.

4 Experiment

4.1 Experimental Setup

Datasets We perform experiments on three public benchmarks: MTL-AQA[18], AQA-7[17], and JIGSAWS[7]. See Supplement for more details on datasets.

Evaluation Metrics Following prior work[32], we utilize two metrics in our experiments, the Spearman's rank correlation and relative L2 distance(R-ℓ_2). **Spearman's rank correlation** was adopted as our main evaluation metric to measure the difference between true and predicted scores. The Spearman's rank correlation is defined as follows:

$$\rho = \frac{\sum_i (p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_i (p_i - \bar{p})^2 \sum_i (q_i - \bar{q})^2}} \tag{14}$$

It focuses on the ranking of test samples. In contrast, **relative L2 distance** measures the numerical precision of each sample compared with ground truth. Formally, it's defined as:

$$R\text{-}\ell_2 = \frac{1}{N} \sum_{n=1}^{N} (\frac{|s_n - \hat{s}_n|}{s_{max} - s_{min}})^2 \tag{15}$$

Implementation Details We adopt the I3D backbone pretrained on Kinetics[4] as our local spatial-temporal feature extractor. The Adam optimizer is applied with a learning rate 1×10^{-4} for the backbone and transformer module. The learning rate for the regression head is set to 1×10^{-3}. The feature dimension is set to 512 for the transformer block. We select 10 exemplars for each test sample during the inference stage to align with previous work[32] for fair comparisons. As for the data-preprocessing on AQA-7 and MTL-AQA datasets, we sample 103 frames following previous works for all videos. Since our proposed method requires more fine-grained temporal information, unlike previous work that segmented the sample frames into 10 clips, we segment the frames into 20 overlapping clips each containing 8 continuous frames. As for the JIGSAWS dataset, we uniformly sample 160 frames following [22] and divide them into 20 non-overlapping clips as input of the I3D backbone. We select exemplars from the same difficulty degree on MTL-dataset during the training stage. For AQA-7 and JIGSAWS datasets, all exemplars come from the same coarse classes.

4.2 Comparison to state-of-the-art

We compare our results with state-of-the-art methods on three benchmarks in Tab.1, Tab.2 and Tab.3. Our method outperforms priors works on all three benchmarks under all settings.

On **MTL-AQA dataset**, we evaluated our experiments with two different settings, following prior work[32]. Specifically, the MTL-AQA dataset contains

Table 1: Performance comparison on MTL-AQA dataset. 'w/o DD' means that training and test processes do not utilize difficulty degree labels, 'w/ DD' means experiments utilizing difficulty degree labels.

Method (w/o DD)	Sp. Corr.	R-ℓ_2($\times$100)
Pose+DCT[20]	0.2682	-
C3D-SVR[19]	0.7716	-
C3D-LSTM[19]	0.8489	-
MSCADC-STL[18]	0.8472	-
C3D-AVG-STL[18]	0.8960	-
MSCADC-MTL[18]	0.8612	-
C3D-AVG-MTL[18]	0.9044	-
USDL[22]	0.9066	0.654
CoRe[32]	0.9341	0.365
TSA-Net[27]	0.9422	-
Ours	**0.9451**	**0.3222**
Method (w/ DD)	Sp. Corr	R-ℓ_2($\times$ 100)
USDL[22]	0.9231	0.468
MUSDL[22]	0.9273	0.451
CoRe[32]	0.9512	0.260
Ours	**0.9607**	**0.2378**

the label of difficult degree, and each video's quality score is calculated by the multiplication of the raw score with its difficulty. In the experiment setting 'w/o DD', the training and test processes do not utilize difficulty degree labels. In setting 'w/ DD', we exploit the difficulty label by comparing the test video to the exemplar videos with the same difficulty, and we estimate the raw score, which is multiplied by the difficulty to get the final quality. Our method outperforms existing works under both settings. As shown in Tab. 1, under 'w/ DD', our method achieves a Sp. Corr. of 0.9607, and R-ℓ_2 of 0.2378, outperforms the tree-based CoRe[32]. Note that our model simply utilizes two shallow MLPs to perform contrastive regression instead of the tree structure as in [32]. Our transformer extracts fine-grained part representations, hence the regression becomes easier. Under 'w/o DD', out method achieves 0.9451(Sp. Corr) and 0.3222(R-ℓ_2), outperforms the CoRe and recent TSA-Net[27]. It's worth noting that TSA-Net utilizes an external VOT tracker[35] to extract human locations and then enhance backbone features, which is orthogonal to the main issue of temporal parsing addressed in our work. Consequently, we expect that our method can be further improved by incorporating the attention module as in [27].

On **AQA-7 dataset**, our method achieves state-of-the-art on 5 categories and comparable performance on the rest category, shown in Tab. 2. In particular, on average, our method outperforms CoRe by 3.14 Corr.($\times$100) and TSA-Net by 2.39 Corr.($\times$100), and obtains a very small R-ℓ_2 of 1.68($\times$100), demonstrating the effectiveness of our temporal parsing transformer.

On the smallest **JIGSAW dataset**, we perform 4-fold cross validation for each category, following prior work[32, 22]. Our method achieves an average of 0.89 Corr. and 3.668 R-ℓ_2, achieves new state-of-the-art.

Table 2: Performance comparison on AQA-7 dataset.

Sp. Corr	Diving	Gym Vault	BigSki.	BigSnow.	Sync. 3m	Sync. 10m	Avg. Corr.
Pose+DCT[20]	0.5300	0.1000	-	-	-	-	-
ST-GCN[30]	0.3286	0.5770	0.1681	0.1234	0.6600	0.6483	0.4433
C3D-LSTM[19]	0.6047	0.5636	0.4593	0.5029	0.7912	0.6927	0.6165
C3D-SVR[19]	0.7902	0.6824	0.5209	0.4006	0.5937	0.9120	0.6937
JRG[16]	0.7630	0.7358	0.6006	0.5405	0.9013	0.9254	0.7849
USDL[22]	0.8099	0.7570	0.6538	0.7109	0.9166	0.8878	0.8102
CoRe[32]	0.8824	0.7746	0.7115	0.6624	0.9442	0.9078	0.8401
TSA-Net[27]	0.8379	0.8004	0.6657	0.6962	**0.9493**	0.9334	0.8476
Ours	**0.8969**	**0.8043**	**0.7336**	**0.6965**	0.9456	**0.9545**	**0.8715**
R-ℓ_2(×100)	Diving	Gym Vault	BigSki.	BigSnow.	Sync. 3m	Sync. 10m	Avg. R-ℓ_2
C3D-SVR[19]	1.53	3.12	6.79	7.03	17.84	4.83	6.86
USDL[22]	0.79	2.09	4.82	4.94	0.65	2.14	2.57
CoRe[32]	0.64	1.78	3.67	3.87	0.41	2.35	2.12
Ours	**0.53**	**1.69**	**2.89**	**3.30**	**0.33**	**1.33**	**1.68**

Table 3: Performance comparison on JIGSAW dataset.

Sp. Corr.	S	NP	KT	Avg.
ST-GCN[30]	0.31	0.39	0.58	0.43
TSN[26]	0.34	0.23	0.72	0.46
JRG[16]	0.36	0.54	0.75	0.57
USDL[22]	0.64	0.63	0.61	0.63
MUSDL[22]	0.71	0.69	0.71	0.70
CoRe[32]	0.84	0.86	0.86	0.85
Ours	**0.88**	**0.88**	**0.91**	**0.89**
R-ℓ_2	S	NP	KT	Avg.
CoRe[32]	5.055	5.688	2.927	4.556
Ours	**2.722**	**5.259**	**3.022**	**3.668**

4.3 Ablation Study

In this subsection, we perform ablation studies to evaluate the effectiveness of our proposed model components and designs. All of our ablation studies are performed on MTL-AQA dataset under 'w/ DD' setting. We build a baseline network that directly pool the clip features without transformer, and utilize the resulting holistic representation to perform contrastive regression.

Different model components In this work, we propose a novel temporal parsing transformer(TPT), and exploit the ranking loss(L_{rank}) and sparsity loss($L_{sparsity}$) on cross attention responses to guide the part representation learning. We first perform experiments to show the effectiveness of each design, the results are shown in Tab.4. We can observe that with only TPT, the performance only improves marginally from 0.9498 Corr. to 0.9522 Corr.. With the ranking loss, the performance is significantly improved, demonstrating the importance of temporally ordered supervision strategy. The sparsity loss further improves the performance, showing that the discrimination of parts is also important.

Table 4: Ablation study of different components on MTL-AQA dataset.

Method	TPT	L_{rank}	$L_{sparsity}$	Sp. Corr.	R-ℓ_2
Baseline	×	×	×	0.9498	0.2893
	✓	×	×	0.9522	0.2742
	✓	✓	×	0.9583	0.2444
Ours	✓	✓	✓	**0.9607**	**0.2378**

Table 5: More ablation studies on MTL-AQA dataset.

(a) Different part generation strategies.

Method	Sp. Corr.	R-ℓ_2
Baseline	0.9498	0.2893
Adaptive pooling	0.9509	0.2757
Temporal conv	0.9526	0.2758
TPT(ours)	**0.9607**	**0.2378**

(b) Effect of order guided supervision.

Method	Sp. Corr.	R-ℓ_2
Baseline	0.9498	0.2893
Diversity loss	0.9538	0.2655
Ranking loss(ours)	**0.9607**	**0.2378**

(c) Effect of positional encoding.

Pos. Encode	Memory(clip)	Query(part)	Sp. Corr.	R-ℓ_2
	✓	✓	0.9526	0.2741
	✓	×	0.9532	0.2651
Proposed	×	×	**0.9607**	**0.2378**

(d) Different relative representation generation.

Method	Sp. Corr.	R-ℓ_2
Baseline	0.9498	0.2893
Part-enhanced holistic	0.9578	0.2391
Part-wise relative + AvgPool(ours)	**0.9607**	**0.2378**

Different relative representation generation Since we have obtained part representations from TPT for each video, we may have two options to generate relative representation for contrastive regression. For the first option, we can first fuse the part representations with a pooling operation for each video, then each video takes the part-enhanced holistic representation to estimate the relative score. For the second option, which is our proposed strategy, we first compute a part-wise relative representation and then apply the AvgPool operation over the parts. We compare the results of above options in Tab.5d. We can see that the part-wise strategy outperforms part-enhanced strategy. It's worth noting that the part-enhanced approach also outperforms our baseline network, which implies that each part indeed encodes fine-grained temporal patterns.

Different part generation strategies Our method utilizes the temporal parsing transformer to extract part representations. In this ablation study, we compare our method with the other two baseline part generation strategy, shown in Tab. 5a. The first strategy utilizes the adaptive pooling operation cross temporal

frames to down-sample the origin T clip representation into K part representations. The second strategy replaces the above adaptive pooling with a temporal convolution with stride $\lfloor T/K \rfloor$, resulting in a representation with K size. We found that both strategies introduce minor improvements as they can not capture fine-grained temporal patterns.

Effect of position encoding Different from conventional transformer[3, 24], our transformer decoding process does not rely on the temporal position encoding. We compare the results of different position encoding strategies on the memory(clip) and query(part) in Tab.5c. To embed the position encoding on queries, we add the cosine series embedding of $\lfloor T/K \rfloor \times i$ to i-th learnable query, making the queries have positional guidance uniformly distributed across temporal clips. We keep the ranking loss and sparsity for fair comparisons. From Tab.5c, we can observe that adding position encoding hurts the learning of temporal patterns.

Effect of order guided training strategy Our ranking loss on the attention centers consistently encourages the temporal order of atomic patterns. To verify the importance of such order guided supervision, we replace the ranking loss to a diversity loss following the Associative Embedding[15] to push attention centers: $L_{div} = \sum_{i=1}^{K} \sum_{j=i+1}^{K} \exp^{-\frac{1}{2\sigma^2}(\bar{\alpha}_i - \bar{\alpha}_j)^2}$. Compared with L_{rank}, L_{div} does not encourage the order of queries, but keeps diversity of part representations. As shown in Tab. 5b, the performance significantly drops from 0.9607 Corr. to 0.9538 Corr., demonstrating the effectiveness of our order guided training strategy.

4.4 Visualization results

We provide some visualization results in Fig.3 and Fig.4. Samples are from MTL-AQA dataset trained under 'w/ DD' setting and AQA-7 dataset. In Fig.3, we visualize the clip frames with the highest attention responses in cross attention maps of the last decoder layer. Since each clip consists of multiple frames, we select the middle frame of a clip as representative. We can observe that our transformer can capture semantic temporal patterns with learned queries. In Fig.4, we visualize the cross attention maps. We can observe that the attention responses have a consistent temporal order due to our designed ranking loss, and they are also sparse due to our sparsity loss.

5 Conclusion

In this paper, we propose a novel temporal parsing transformer for action quality assessment. We utilize a set of learnable queries to represent the atomic temporal patterns, and exploit the transformer decoder to convert clip-level representations to part-level representations. To perform quality score regression, we exploit the contrastive regression framework that first computes the relative pairwise representation per part and then fuses them to estimate the relative

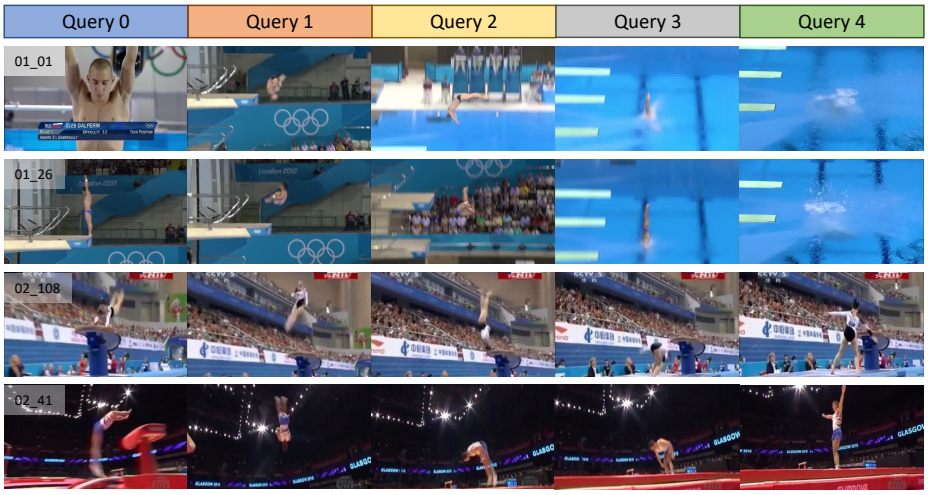

Fig. 3: Visualization of the frames with the highest attention responses in decoder cross attention maps on MTL-AQA and AQA-7 datasets. Each row represents a test video from different representative categories (diving from MTL-AQA, gymnastic vault from AQA-7), whose ID is shown in the left first frame. Different columns correspond to temporally ordered queries. The above results show that our transformer is able to capture semantic temporal patterns with learned queries.

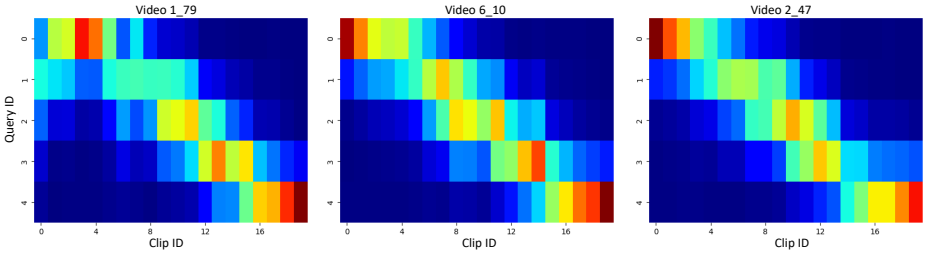

Fig. 4: Visualization of cross attention maps on three video samples from MTL-AQA dataset, where video IDs are shown on the top. In each subfigure, each row indicates one query, and each column indicates one clip. We can observe that the bright grids(with high attention responses) have a consistent temporal order due to ranking loss, and the attention maps are sparse due to our sparsity loss.

score. To learn the atomic patterns without part-level labels, we propose two novel loss functions on cross attention responses to guide the queries to attend to temporally ordered clips. As a result, our method is able to outperform existing state-of-the-art methods by a considerable margin on three public benchmarks. The visualization results show that the learned part representations are semantic meaningful.

References

1. Alayrac, J.B., Laptev, I., Sivic, J., Lacoste-Julien, S.: Joint discovery of object states and manipulation actions. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2127–2136 (2017)
2. Bertasius, G., Soo Park, H., Yu, S.X., Shi, J.: Am i a baller? basketball performance assessment from first-person videos. In: Proceedings of the IEEE international conference on computer vision. pp. 2177–2185 (2017)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
5. Doughty, H., Damen, D., Mayol-Cuevas, W.: Who's better? who's best? pairwise deep ranking for skill determination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6057–6066 (2018)
6. Doughty, H., Mayol-Cuevas, W., Damen, D.: The pros and cons: Rank-aware temporal attention for skill determination in long videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7862–7871 (2019)
7. Gao, Y., Vedula, S.S., Reiley, C.E., Ahmidi, N., Varadarajan, B., Lin, H.C., Tao, L., Zappella, L., Béjar, B., Yuh, D.D., et al.: Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In: MICCAI workshop: M2cai. vol. 3, p. 3 (2014)
8. Gordon, A.S.: Automated video assessment of human performance. In: Proceedings of AI-ED. vol. 2 (1995)
9. Jug, M., Perš, J., Dežman, B., Kovačič, S.: Trajectory based assessment of coordinated human activity. In: International Conference on Computer Vision Systems. pp. 534–543. Springer (2003)
10. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 780–787 (2014)
11. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 156–165 (2017)
12. Lei, P., Todorovic, S.: Temporal deformable residual networks for action segmentation in videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6742–6751 (2018)
13. Li, J., Lei, P., Todorovic, S.: Weakly supervised energy-based learning for action segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6243–6251 (2019)
14. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional detr for fast training convergence. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3651–3660 (2021)
15. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. Advances in neural information processing systems **30** (2017)
16. Pan, J.H., Gao, J., Zheng, W.S.: Action assessment by joint relation graphs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6331–6340 (2019)

17. Parmar, P., Morris, B.: Action quality assessment across multiple actions. In: 2019 IEEE winter conference on applications of computer vision (WACV). pp. 1468–1476. IEEE (2019)
18. Parmar, P., Morris, B.T.: What and how well you performed? a multitask learning approach to action quality assessment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 304–313 (2019)
19. Parmar, P., Tran Morris, B.: Learning to score olympic events. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 20–28 (2017)
20. Pirsiavash, H., Vondrick, C., Torralba, A.: Assessing the quality of actions. In: European Conference on Computer Vision. pp. 556–571. Springer (2014)
21. Shao, D., Zhao, Y., Dai, B., Lin, D.: Intra-and inter-action understanding via temporal action parsing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 730–739 (2020)
22. Tang, Y., Ni, Z., Zhou, J., Zhang, D., Lu, J., Wu, Y., Zhou, J.: Uncertainty-aware score distribution learning for action quality assessment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9839–9848 (2020)
23. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4489–4497 (2015)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
25. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision. pp. 3551–3558 (2013)
26. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L.V.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36. Springer (2016)
27. Wang, S., Yang, D., Zhai, P., Chen, C., Zhang, L.: Tsa-net: Tube self-attention network for action quality assessment. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 4902–4910 (2021)
28. Wang, Y., Zhang, X., Yang, T., Sun, J.: Anchor detr: Query design for transformer-based detector. arXiv preprint arXiv:2109.07107 (2021)
29. Xu, C., Fu, Y., Zhang, B., Chen, Z., Jiang, Y.G., Xue, X.: Learning to score figure skating sport videos. IEEE transactions on circuits and systems for video technology **30**(12), 4578–4590 (2019)
30. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI conference on artificial intelligence (2018)
31. Yi, F., Wen, H., Jiang, T.: Asformer: Transformer for action segmentation. arXiv preprint arXiv:2110.08568 (2021)
32. Yu, X., Rao, Y., Zhao, W., Lu, J., Zhou, J.: Group-aware contrastive regression for action quality assessment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7919–7928 (2021)
33. Zhang, C., Gupta, A., Zisserman, A.: Temporal query networks for fine-grained video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4486–4496 (2021)

34. Zhang, Q., Li, B.: Relative hidden markov models for video-based evaluation of motion skills in surgical training. IEEE transactions on pattern analysis and machine intelligence **37**(6), 1206–1218 (2014)
35. Čehovin Zajc, L.: A modular toolkit for visual tracking performance evaluation. SoftwareX **12**, 100623 (2020). https://doi.org/https://doi.org/10.1016/j.softx.2020.100623