Supplementary Material for
“LOGO: A Long-Form Video Dataset for
Group Action Quality Assessment”

Shiyi Zhang[1,2,3], Wenxun Dai[1], Sujia Wang[1], Xiangwei Shen[1], Jiwen Lu[2,3], Jie Zhou[2,3], Yansong Tang[1,*]
[1] Shenzhen International Graduate School, Tsinghua University
[2] Department of Automation, Tsinghua University
[3] Beijing National Research Center for Information Science and Technology
`{shiyi-zh19@mails.,lujiwen@,jzhou@,tang.yansong@sz.}tsinghua.edu.cn`

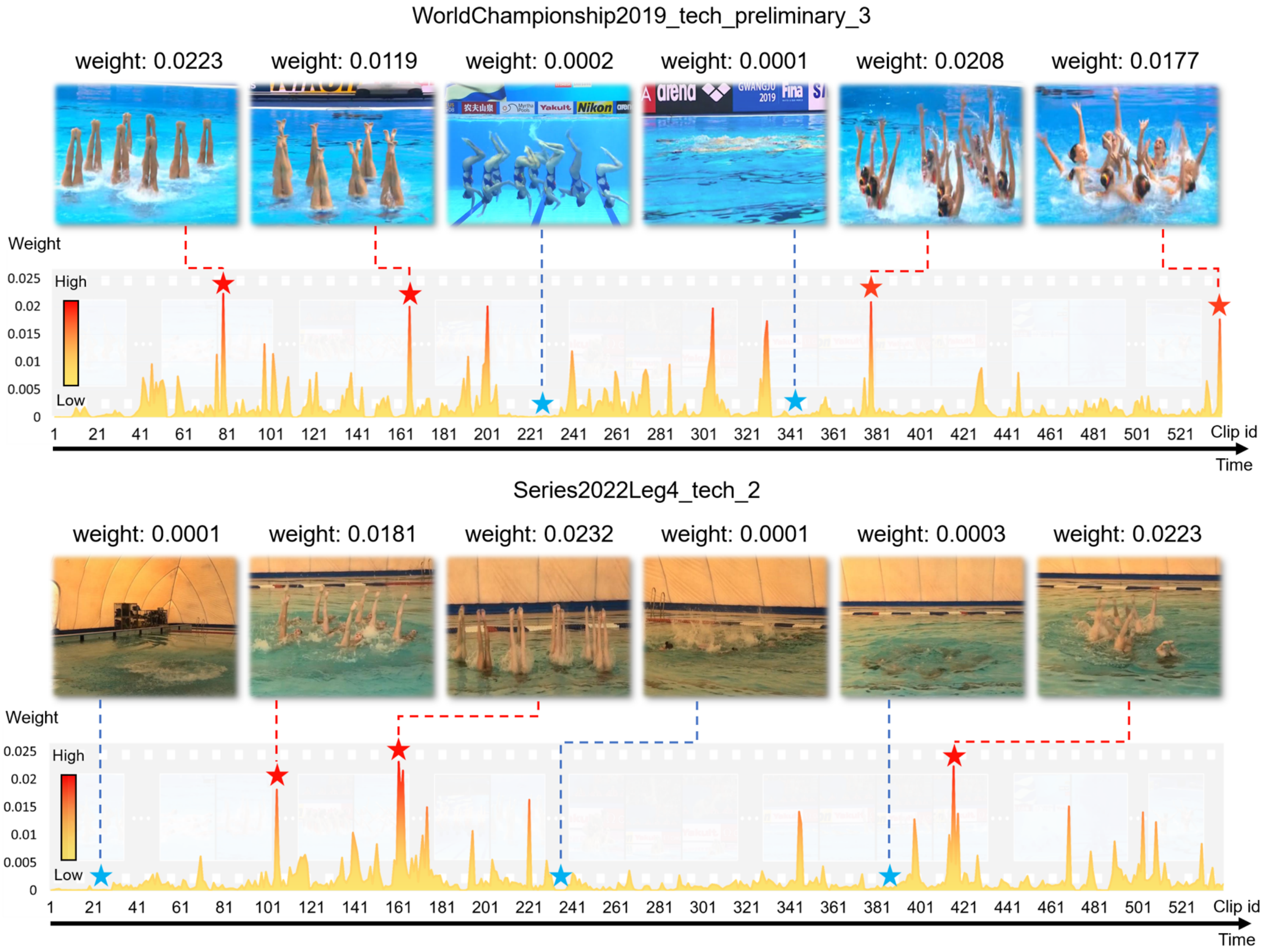

Figure 1. The visualization of the output of our proposed ***GOAT*** in action quality assessment. Our approach can focus on where the athletes perform effective movements with clear formations while it can also ignore the redundant part such as all actors are under-water.

1. Visualization

We provide more visualizations in this section. Figure 1 shows the visualization of the output of the GOAT, which

---

[*] indicates the corresponding author.

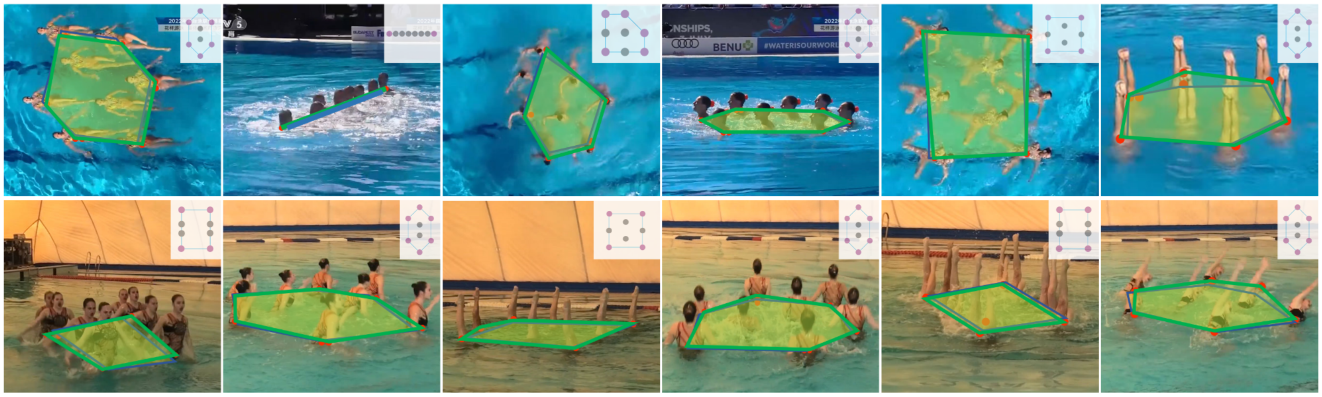

Figure 2. The additional visualizations of the prediction results of our formation detector method. The green polygons represent prediction results and the yellow polygons with blue edges are the ground truth. The results show that our approach can detect the positions of actors and distinguish whether the athlete is the formation vertex or not, which guarantees the reliability of the formation features.

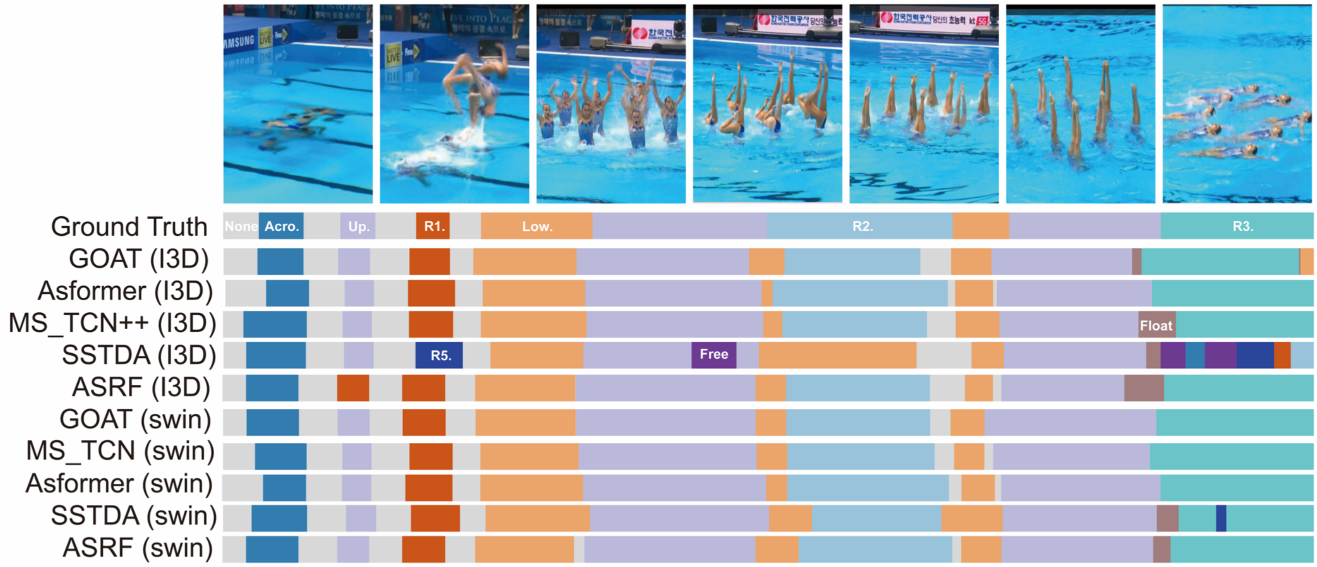

Figure 3. An visualization of our action segmentation experiments. **Acro.**, **Up.** and **Low.** represent *Acrobatic, Upper and Lower movements*. **R1.**, **R2.** and **R3.** represents *Required movements 1, 2 and 3*.

demonstrates that it can focus on where the athletes perform effective movements with clear formations and also ignore the redundant part of the input videos. Figure 2 shows the visualization of the prediction result of our formation detector method. It illustrates that our approach distinguishes whether an athlete is the formation vertex or not, which ensures the reliability of the formation features. Figure 3 shows the results of action segmentation on the LOGO dataset, which demonstrates the challenges our dataset brings.

Table 1. Comparisons of the inference time with existing AQA methods on LOGO.

Method	Inference Time (s)
USDL [6]	40.39
USDL [6]+GOAT	40.40
CoRe [9]	40.39
CoRe [9]+GOAT	40.40
TSA [8]	40.41
TSA [8]+GOAT	40.51

2. Computational Time Costs

In this section, we compare the inference time of the mainstream AQA methods mentioned in our paper with our approaches. All methods have experimented with the same settings. As shown in Table 1, the increase in inference time after the introduction of GOAT is minimal, which proves the efficiency of our proposed GOAT. Most of the operations

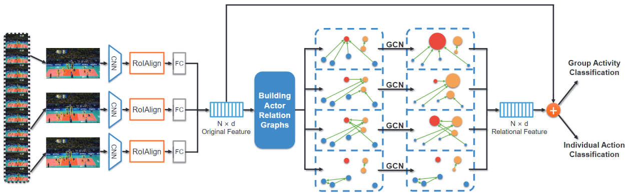
CNN
RoIAlign
FC
N × d
Original Feature
Building Actor Relation Graphs
GCN
N × d
Relational Feature
Group Activity Classification
Individual Action Classification

Figure 4. The pipeline of Actor Relation Graphs for group activity recognition.

Table 2. Comparisons of AQA performance with ARG on LOGO. The higher ρ, the lower R-ℓ_2, the better performance.

Method	$\rho\uparrow$	R-$\ell_2(\times 100)\downarrow$
ARG [7]	0.3549	5.7778
USDL [6]+GOAT	0.4620	4.8739
CoRe [9]+GOAT	0.4935	5.0716
TSA [8]+GOAT	0.4855	5.3943

in GOAT are matrix operations, which can be efficiently completed on the GPU. In the inference stage, most of the computing time is spent on the feature extraction part of the backbone. Because I3D contains a large number of convolutional layers, bringing high computational time costs.

3. Experiments

This section presents some experimental results not presented in our paper. To proves the effectiveness of our temporal fusion strategy, we also use the *Actor Relation Graphs* (ARG) from [7] to directly predict the scores without using the features from the video backbone, which means we only use the spatial information for prediction. Figure 4 shows the pipeline of ARG. Specifically, we replace the linear layer used to predict group activity in the last part of the original pipeline with another linear layer to predict the scores. As shown in Table 2, our method achieves better results than just using spatial information for score prediction.

4. Annotation Details

In this section, we systematically introduce the annotation details of LOGO, which includes the annotation tools and the labeling rules. Labeled video samples are provided in the accompanying folder.

4.1. Annotation Tools

The annotation tools we use are shown in Figure 5. We use COIN annotation tools [5] to annotate the frame-wise action types and the temporal boundaries for actions. We use Labelme to annotate the formation labels, including the coordinates of the formation vertexes and the number of edges. The example in Figure 5 shows an annotation result of a pentagon formation.

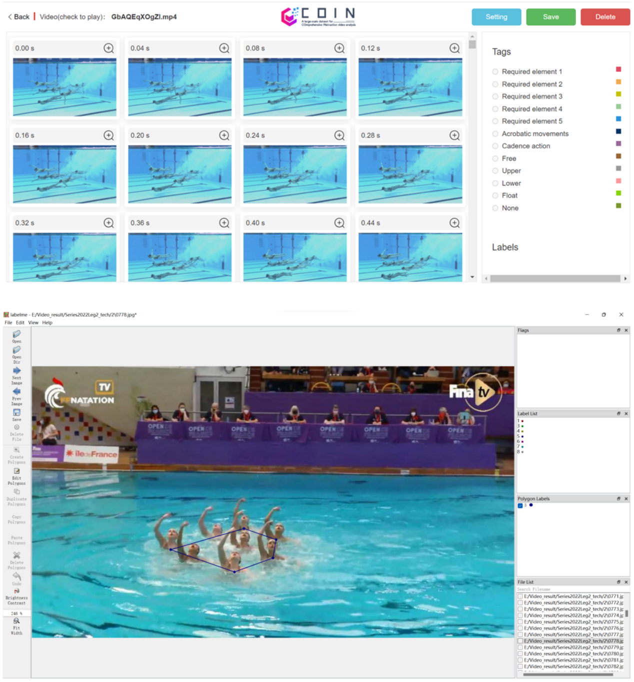

Figure 5. Annotation tools. The top half is COIN annotation tool, and the lower half is Labelme.

4.2. Labeling Rules

Scores. We collect the scores for each video from the official website of FINA, which include three sub-scores, and one final score. All scores have been double-checked.

Actions. We define 12 types of actions under the guidance of professional athletes. We define how the action is performed and its boundaries for each type. For example, for the action type of "*Upper*", it means all athletes perform movements that are performed by the upper body in the scene. We use the first frame when everyone emerges from the water (everyone only needs to emerge a little bit) as the start frame of the annotation, and the first frame before everyone is submerged or enters a floating state as the end frame. To ensure that all annotators have a consistent labeling method, we iteratively update the definitions of all actions and boundaries until the annotation results of the annotators are unified.

Formations. The formation labels consist of the edges and the coordinates of formation vertexes as shown in the lower part of Figure 5. To unify the annotations, we define 17 kinds of formation polygons, which are generated iteratively during the annotation process.

5. Implementation Details

5.1. Action Quality Assessment

The attention block of GOAT has 8 heads and 4 layers. For the SWIN features, we follow [4] to sample frames with a stride of 2 along the temporal axis, and each clip contains 32 frames. We also use warm-up strategy [3] during the training stage. We fix the image size to 224×224. We fix the crop size of RoIAlign to 5×5. The number of "*Free*" events and "*Technical*" events in the training set and testing set are close to 1:1. Our proposed methods were built on the Pytorch toolbox and implemented on a system with A100.

5.2. Action Segmentation

Sampling Strategy. In this session, we conduct action segmentation with GOAT using MS_TCN++ [2] as the baseline. For GOAT, we adopt the same sampling strategy as our AQA experiment, we sample 5406 frames for each video and split them into 540 snippets that contain 16 continuous frames with a stride of 10 frames to extract group features. For I3D, we sample all frames in videos and take 16 contextual frames around each frame. We take 32 contextual frames for each frame to extract SWIN features.

Experimental Setting. The GOAT includes two components: group-aware GCN and temporal-fusion attention. The original temporal-fusion attention acquires group feature to be frame-wise since it asserts sequence length of video embeddings of "*query*", and "*key*" to be the same, which comes with a huge computation cost. To address this, we remove the BatchNorm block and residual block from temporal-fusion attention. Besides, we make the frame-wise feature serve as "*query*" and the video embeddings from group-aware GCN serve as "*key*" and "*value*". We use only one layer of GCN. In the training stage, we split videos into 75 percent for training and 25 percent for evaluation. We implemented and trained these action segmentation methods with the Pytorch toolbox and run on a Linux machine with Nvidia GeForce RTX 3090.

6. Challenges of LOGO

We show current SOTA results on mainstream AQA datasets below to prove the challenges of LOGO. We use CoRe and TPT [1]) on MTL-AQA, AQA-7, FineDiving, and LOGO.

Dataset	MTL-AQA	AQA-7	FineDiving	LOGO
$\rho\uparrow / R{-}\ell_2(\times 100)\downarrow$ **CoRe**	0.9512 / 0.2600	0.8401 / 2.12	0.9061 / 0.3615	0.4712 / 5.4086
$\rho\uparrow / R{-}\ell_2(\times 100)\downarrow$ **TPT**	0.9607 / 0.2378	0.8715 / 1.68	0.9182 / 0.3527	0.4732 / 5.3987

References

[1] Yang Bai, Desen Zhou, Songyang Zhang, Jian Wang, Errui Ding, Yu Guan, Yang Long, and Jingdong Wang. Action quality assessment with temporal parsing transformer. In *ECCV*, pages 422–438, 2022. 4

[2] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *CVPR*, pages 9454–9463, 2020. 4

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[4] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, pages 3202–3211, 2022. 4

[5] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, pages 1207–1216, 2019. 3

[6] Yansong Tang, Zanlin Ni, Jiahuan Zhou, Danyang Zhang, Jiwen Lu, Ying Wu, and Jie Zhou. Uncertainty-aware score distribution learning for action quality assessment. In *CVPR*, pages 9839–9848, 2020. 2, 3

[7] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu. Learning actor relation graphs for group activity recognition. In *CVPR*, pages 9964–9974, 2019. 3

[8] Jinglin Xu, Yongming Rao, Xumin Yu, Guangyi Chen, Jie Zhou, and Jiwen Lu. Finediving: A fine-grained dataset for procedure-aware action quality assessment. In *CVPR*, pages 2949–2958, 2022. 2, 3

[9] Xumin Yu, Yongming Rao, Wenliang Zhao, Jiwen Lu, and Jie Zhou. Group-aware contrastive regression for action quality assessment. In *ICCV*, pages 7919–7928, 2021. 2, 3