FineDiving: A Fine-grained Dataset for Procedure-aware Action Quality Assessment *Supplementary Materials*

A. More Experiment Settings

A.1. The Criterion of Selecting Exemplars

For our approach TSA (w/ DN), we selected exemplars from the training set based on the action types. In training, for each instance, we randomly selected an instance as an exemplar from the rest of the training instances with the same action type. In inference, we adopted the multi-exemplar voting strategy, i.e., randomly selecting 10 instances as 10 exemplars from the training instances with the same action type. For our approach TSA (w/o DN), we randomly selected exemplars from the training set in both training and inference.

A.2. The Number of Step Transitions

In FineDiving, the 60% action types (e.g., 307C and 614B) consist of 3 sub-action types, corresponding to 3 steps; 30% action types (e.g., 5251B and 6241B) consist of 4 sub-action types, corresponding to 4 steps; 10% action types (e.g., 5152B and 5156B) consist of 4 sub-action types, corresponding to 5 steps. We provided full annotations of 5 possible steps to support future research on utilizing fine-grained annotations for AQA. In experiments, our approach keeps L constant and equals 2, indicating segmenting a dive action into 3 steps, i.e., 2 step transitions. The reasons are as follows. A dive action consists of *take-off*, *flight*, and *entry*. If *flight* is accomplished by one sub-action type, a dive action is divided into 3 steps, e.g., 307C containing 3 steps: Reverse, 3.5 Soms.Tuck, and Entry. If *flight* is accomplished by two sub-action types successively, a dive action is divided into 4 steps, e.g., 6241B containing 4 steps: Arm.Back, 0.5 Twist, 2 Soms.Pike, and Entry. If *flight* is accomplished by two sub-action types and one sub-action type is interspersed in another, a dive action is divided into 5 steps, e.g., 5152B containing 5 steps: Forward, 2.5 Soms.Pike, 1 Twist, 2.5 Soms.Pike, and Entry, where “1 Twist” is interspersed in “2.5 Soms.Pike”. We see that two cases of *flight* accomplished by two sub-action types can be seen as two special cases of *flight* accomplished by one sub-action type. Therefore, $L=2$ makes sense.

B. More Ablation Study

We summarize hyper-parameters for training as follows: the number of frames in each step is fixed into 5 before being fed into Multi-head Cross-Attention (MCA); all configurations are based on transformer decoder with 3 layers and 8 heads.

Table 1. Effects of the number of frames in each step in the procedure-aware cross-attention model.

L_{step}	AIoU@		ρ	R-$\ell_2(\times 100)$
	0.5	0.75		
3	76.49	24.17	0.9081	0.4003
5	**82.51**	**34.31**	**0.9203**	**0.3420**
8	80.65	33.37	0.9198	0.3501
10	77.97	30.44	0.9174	0.3588
15	77.96	30.32	0.9149	0.3675

To investigate the effects of the number of frames in each step (denoted as L_{step}) on the performance of action quality assessment, we conduct several experiments on the FineDiving dataset. Table 1 summarizes the performance with different L_{step} including 3, 5, 8, 10 and 15. We observe that when L_{step} increases from 3 to 5, the action quality assessment performance achieves 1.22% and 0.0583 improvements respectively on Spearman’s rank correlation and Relative ℓ_2-distance. When L_{step} is bigger than 8, the performance tends to be flat with a slight decrease. Specifically, compared with the performance of L_{step}=5, that of L_{step}=8 degrades 0.05% and 0.0081 on Spearman’s rank correlation and Relative ℓ_2-distance, respectively. The experimental results illustrate that the number of frames in each step is not proportional to the action quality assessment performance, since each step containing fewer frames cannot make full use of intra-step information while each step containing too many frames may introduce some noisy information.

Table 2. Effects of the number of transformer decoder layers.

R	AIoU@		ρ	R-$\ell_2(\times 100)$
	0.5	0.75		
1	81.71	33.24	0.9168	0.3612
3	**82.51**	**34.31**	**0.9203**	**0.3420**
5	79.83	31.84	0.9202	0.3483
10	79.57	31.78	0.9171	0.3534

To explore the effect of the number of transformer decoder layers (denoted as R) on the performance of action quality assessment, we conduct several experiments on the FineDiving dataset. Table 2 summarizes the performance with different R, namely 1, 3, 5, and 10. It can be seen that when R equals 3, the action quality assessment performance reaches the peaks (namely 0.9203 and 0.3420) on

Spearman's rank correlation and Relative ℓ_2-distance, respectively. The performance tends to be flat with a slight decrease when $R > 5$, since too many transformer decoder layers may lead to overfitting for each step containing 5 frames (L_{step}=5).

C. Code

We provide the FineDiving dataset and code of our approach[1], including the training and inference phases.

D. The FineDiving Dataset Details

We will release the FineDiving dataset to promote future research on action quality assessment.

D.1. Descriptions of Action and Sub-action Types

Table 3 shows the detailed descriptions of action and sub-action types in FineDiving mentioned in the lexicon in subsection 3.1. Dataset Construction. We see that a combination of the sub-action types from three phases (namely take-off, flight, and entry) generates an action type. Specifically, the take-off phase is annotated by one of six sub-action types which are "Forward, Back, Reverse, Inward, Armstand Forward, Armstand Back, and Armstand Reverse". The entry phase is annotated by the sub-action type "Entry". The flight phase is labeled by one or two sub-action types describing the somersault process, where the number of sub-action types depends on the somersault process whether containing the twist or not.

As shown in Table 3, if the somersault process contains the twist, the flight phase is annotated by two sub-action types, where one sub-action type annotates the number of twist turns in the somersault process and another is annotated the number of somersault turns in the somersault process. The former sub-action type is a part of the latter sub-action type and does not be performed independently without the latter sub-action type. For different action types, the former sub-action type (twist) may occur at different locations in the somersault process. When the twist is performed at the beginning of the somersault process, we first annotate the number of twist turns and then annotate the number of somersault turns, such as the action types 5231D and 6245D in Table 3. When the twist is performed at the middle of the somersault process, we first annotate the number of somersault turns, then annotate the number of twist turns, and finally annotate the number of somersault turns, such as 5132D and 5172B in Table 3. Note that, the first and the last annotated the same somersault process but are separated by the twist. If the somersault process does not contain the twist, the flight phase is annotated by one sub-action type, that is, the number of somersault turns, such as 205B and 626C in Table 3.

D.2. Annotation Tool

In the fine-grained annotation stage, the durations of sub-action types are different for different action instances, which may cost a huge workload to label the FineDiving dataset with a conventional annotation tool. To improve the annotation efficiency, we utilize a publically available annotation toolbox [1] (mentioned in Annotation in subsection 3.1. Dataset Construction) to generate the frame-wise labels for various sub-action types, which can ensure high efficiency, accuracy, and consistency of our annotation results. Figure 1 shows an example interface of the annotation tool, which annotates the frames extracted from an action instance.

References

[1] 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. 2

[1] https://github.com/xujinglin/FineDiving

Table 3. The detailed descriptions of action and sub-action types.

Action Type	Sub-action Type			Action Type	Sub-action Type				
	Take-off	Flight	Entry		Take-off	Flight			Entry
101B	Forward	0.5 Som.Pike	Entry	612B	Arm.Fwd	1 Som.Pike			Entry
103B	Forward	1.5 Soms.Pike		614B	Arm.Fwd	2 Soms.Pike			
105B	Forward	2.5 Soms.Pike		626B	Arm.Back	3 Soms.Pike			
107B	Forward	3.5 Soms.Pike		626C	Arm.Back	3 Soms.Tuck			
109B	Forward	4.5 Soms.Pike		636C	Arm.Reverse	3 Soms.Tuck			
107C	Forward	3.5 Soms.Tuck		5231D	Back	0.5 Twist	1.5 Soms.Pike		Entry
109C	Forward	4.5 Soms.Tuck		5233D	Back	1.5 Twists	1.5 Soms.Pike		
201A	Back	0.5 Som.Straight		5235D	Back	2.5 Twists	1.5 Soms.Pike		
201B	Back	0.5 Som.Pike		5237D	Back	3.5 Twists	1.5 Soms.Pike		
201C	Back	0.5 Som.Tuck		5251B	Back	0.5 Twist	2.5 Soms.Pike		
205B	Back	2.5 Soms.Pike		5253B	Back	1.5 Twists	2.5 Soms.Pike		
207B	Back	3.5 Soms.Pike		5255B	Back	2.5 Twists	2.5 Soms.Pike		
205C	Back	2.5 Soms.Tuck		5331D	Reverse	0.5 Twist	1.5 Soms.Pike		
207C	Back	3.5 Soms.Tuck		5335D	Reverse	2.5 Twists	1.5 Soms.Pike		
301B	Reverse	0.5 Som.Pike		5337D	Reverse	3.5 Twists	1.5 Soms.Pike		
305B	Reverse	2.5 Soms.Pike		5353B	Reverse	1.5 Twists	2.5 Soms.Pike		
303C	Reverse	1.5 Soms.Tuck		5355B	Reverse	2.5 Twists	2.5 Soms.Pike		
305C	Reverse	2.5 Soms.Tuck		6142D	Arm.Fwd	1 Twist	2 Soms.Pike		
307C	Reverse	3.5 Soms.Tuck		6241B	Arm.Back	0.5 Twist	2 Soms.Pike		
401B	Inward	0.5 Som.Pike		6243D	Arm.Back	1.5 Twists	2 Soms.Pike		
403B	Inward	1.5 Soms.Pike		6245D	Arm.Back	2.5 Twists	2 Soms.Pike		
405B	Inward	2.5 Soms.Pike		5132D	Forward	1.5 Soms.Pike	1 Twist	1.5 Soms.Pike	Entry
407B	Inward	3.5 Soms.Pike		5152B	Forward	2.5 Soms.Pike	1 Twist	2.5 Soms.Pike	
405C	Inward	2.5 Soms.Tuck		5154B	Forward	2.5 Soms.Pike	2 Twists	2.5 Soms.Pike	
407C	Inward	3.5 Soms.Tuck		5156B	Forward	2.5 Soms.Pike	3 Twists	2.5 Soms.Pike	
409C	Inward	4.5 Soms.Tuck		5172B	Forward	3.5 Soms.Pike	1 Twist	3.5 Soms.Pike	

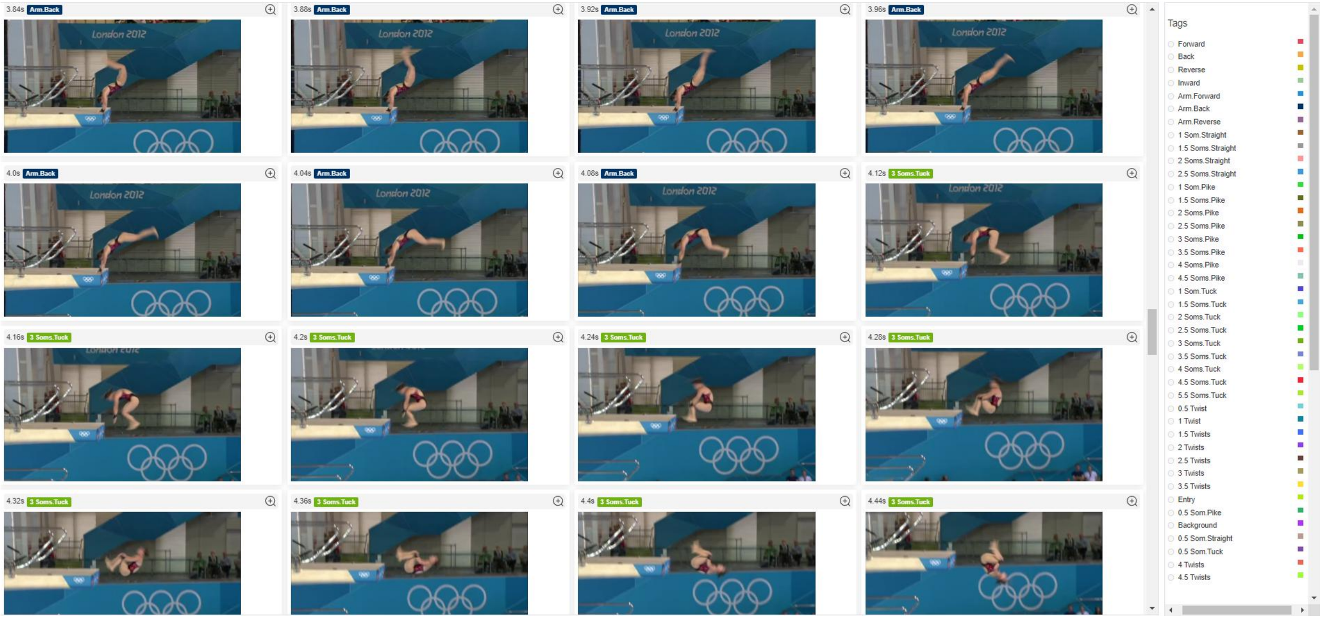

Figure 1. The interface for sub-action type annotation tool. The left part represents the video frames to be annotated by the pre-defined sub-action types. The right part shows the pre-defined sub-action types using different colors.