Diving-R1: Empowering Multimodal LLMs with Traceable Progressive Reasoning for Interpretable Diving Action Quality Assessment

Kaishen Yuan[1], **Yuting Zhang**[1], **Bohao Xing**[2], **Wenshuo Chen**[1], **Yang Yang**[1], **Shaofeng Liang**[1], **Zitong Yu**[3], **Yutao Yue**[1,4*]
[1] The Hong Kong University of Science and Technology (Guangzhou)
[2] Lappeenranta-Lahti University of Technology LUT [3] Great Bay University
[4] Institute of Deep Perception Technology, JITRI
kyuan156@connect.hkust-gz.edu.cn, yutaoyue@hkust-gz.edu.cn

Abstract

Action Quality Assessment (AQA) for diving aims to quantify the execution quality of diving sequences and has long been a prominent research topic. However, existing diving AQA methods typically regress solely the final diving score, leading to poor interpretability. To address this challenge, we propose Diving-R1, the first attempt to leverage a Multimodal Large Language Model (MLLM) for interpretable diving AQA, enabling the integration of comprehensive information, thereby supporting progressive reasoning and traceable assessment. We construct a novel dataset, DivingThink, as the foundation, which contains progressively structured reasoning chains that follow judging logic and explicitly describe the execution quality of each diving sub-action, providing clear evidence for score determination. Furthermore, we establish a new benchmark, DivingInterp, which evaluates the performance of MLLMs on interpretable diving AQA from multiple complementary perspectives, including score estimation, textual similarity, and reasoning quality. Additionally, we carefully design a three-stage training paradigm, combined with a dedicated composite reward function, which gradually relaxes the annotation requirements for the training corpus at each stage to incrementally enrich the diversity of the training data, achieving continuous improvement. Extensive experiments demonstrate the superiority of Diving-R1 over existing MLLMs in interpretable diving AQA. The dataset and code are available at https://github.com/yuankaishen2001/Diving-R1.

1 Introduction

Understanding and analyzing competitive sports videos has long been a prominent research topic in computer vision [30, 31]. Action Quality Assessment (AQA) is one of the key tasks, aiming to analyze, judge, and quantitatively score an athlete's execution quality in action sequences from videos, with diving being a representative and widely studied sporting discipline [36, 54]. Accordingly, diving AQA has attracted increasing attention, as such systems can not only assist divers in performing simulated evaluations and developing more targeted training plans during practice [20], but also alleviate controversies caused by judges' subjectivity in official competitions, thereby improving the objectivity and consistency of scoring to some extent [40].

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

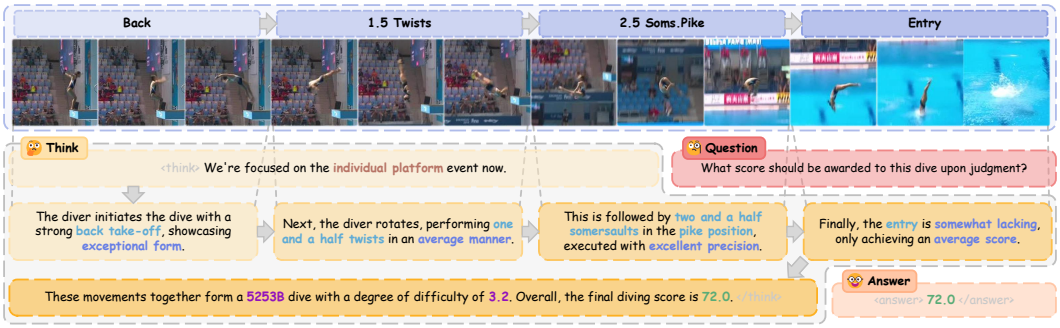


Figure 1: An example sample from DivingThink dataset, featuring a progressively structured reasoning chain: it first identifies the **event type**, then sequentially describes each **sub-action** and assesses its **execution quality**, followed by encoding the **overall action type** and specifying the corresponding **difficulty degree**, and finally integrates all the above information to produce the **final diving score**, enabling traceable and interpretable diving AQA.

With the rapid advancement of deep learning, a large number of diving AQA methods have emerged and achieved impressive performance; however, most of them typically regress a final diving score directly from the input video [33, 46, 35, 15, 27, 24], which is highly opaque and offers poor interpretability. There are also approaches that attempt to parse diving action segments for estimating the score to achieve procedure-aware assessment [40, 2, 8, 49, 23, 39, 43, 41, 42], yet they can only implicitly assign a reference score to each segment (such as *take-off*, *flight*, and *entry*), leaving the absolute quality of each sub-action (such as *somersaults* and *twists*) inaccessible. The root cause is that in official competitions, diving judges only provide an overall score for the entire action rather than fine-grained scores for individual sub-actions, leading to a lack of sub-action-level score annotations and making explicit supervision infeasible. Additionally, few existing methods thoroughly consider other crucial factors closely related to scoring in diving competitions during score estimation, including action type, difficulty degree, and event category. These limitations severely hinder the practical applicability of diving AQA systems, as divers may receive a score but struggle to clearly understand how it was derived or which parts were executed well or poorly, underscoring the urgent need for a solution that can incorporate comprehensive information, transparently assess sub-actions, and provide procedural interpretability.

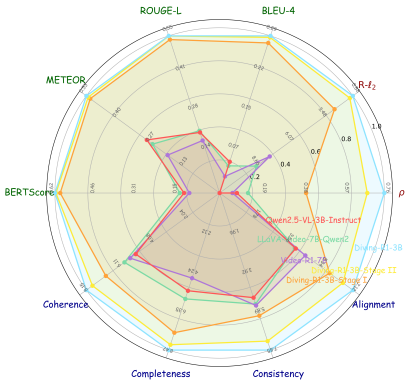


Figure 2: Diving-R1 demonstrates superior performance compared to existing video-capable MLLMs on the DivingInterp benchmark across multiple aspects, including score estimation, textual similarity, and reasoning quality, and shows continuous improvement across consecutive stages.

Recently, Multimodal Large Language Models (MLLMs) have demonstrated impressive capabilities across multiple tasks [37, 16] such as video understanding [52, 47, 13] and visual question answering [12, 6, 1, 45]. In particular, Reinforcement Fine-Tuning (RFT) with Verifiable Rewards, exemplified by DeepSeek-R1 [9], has further unlocked the reasoning potential of MLLMs, enabling them to tackle complex problems through step-by-step reasoning that more closely resembles human thought processes [7, 14, 5, 53, 21, 10]. These advances have opened new opportunities for interpretable and traceable diving AQA. An MLLM that goes beyond simple score regression and is capable of, like a judge, first attending to the details and execution quality of each sub-action in a diving sequence, and then producing a final score based on action type and difficulty, with progressively structured reasoning ability for diving AQA, is highly worth exploring.

Based on the observations above, we propose **Diving-R1**, the first attempt at building an MLLM capable of progressive reasoning, providing traceable and transparent assessments, and incorporating comprehensive event information for interpretable diving AQA. To support this, we construct a novel dataset, **DivingThink**, which contains exhaustive reasoning chains. Specifically, the reasoning process in DivingThink is structured in a progressive manner: it first identifies the event type (*i.e.*, *individual* or *synchronized*, *platform* or *springboard*); then sequentially evaluates the execution quality of each sub-action in the diving sequence using a three-level grading system (*excellent*,

average, *poor*) and provides detailed commentary, where the grades are estimated from the statistical distribution of query-exemplar contrasts for the same sub-action; next, it reports the overall action type and corresponding difficulty degree; finally, it integrates all the above information to produce the final diving score. An example sample from DivingThink is shown in Figure 1. Building on the constructed dataset, we further introduce a new benchmark, **DivingInterp**, which evaluates MLLMs on interpretable diving AQA from multiple perspectives, including score estimation, textual similarity, and reasoning quality. Additionally, we carefully design a three-stage training paradigm that combines Supervised Fine-Tuning (SFT) and RFT and incorporates customized composite rewards focusing on score precision, ranking, temporal order, and action type. By gradually relaxing the annotation requirements for training, the coverage of the training corpus during the RFT stage is also expanded beyond DivingThink, thereby significantly increasing data diversity. Extensive experiments demonstrate that the proposed Diving-R1 exhibits superior performance compared to existing MLLMs on DivingInterp in terms of event information recognition, progressive logical reasoning, and transparent score estimation, as intuitively illustrated in Figure 2, and represents a key step toward the practical deployment of reliable diving AQA systems. Our contributions can be summarized as follows:

- We propose Diving-R1, the first attempt at building an MLLM capable of progressive reasoning, integrating comprehensive information, and providing traceable assessments for interpretable diving AQA.
- We construct a novel dataset, DivingThink, which contains progressively structured reasoning chains and enables explicit evaluation of sub-actions, upon which we establish a new benchmark, DivingInterp.
- We carefully design a three-stage training paradigm and customize a dedicated composite reward function, gradually enriching the diversity of the training corpus at each stage to achieve continuous improvement.
- Extensive experiments demonstrate that the proposed Diving-R1 surpasses existing MLLMs from multiple perspectives, including score estimation, textual similarity, and reasoning quality.

2 Related Work

Action Quality Assessment for Diving. Diving AQA focuses on analyzing and scoring the execution quality of diving sequences in videos, and with the rapid advancement of deep learning, numerous methods have been developed for this task in recent years. In the early stage, Li *et al.* [15] proposed an end-to-end framework based on a fragment-based 3D convolutional neural network, integrating ranking loss with MSE to optimize both score prediction and ranking consistency. Yu *et al.* [46] proposed the contrastive regression framework, which learns relative scores between video pairs using a group-aware regression tree to improve assessment performance. Although these methods achieve promising results, they often directly regress the final diving score from the video, which greatly limits interpretability; therefore, some approaches attempt to parse diving action segments to enable process-aware assessment. Xu *et al.* [40] proposed a framework that segments videos into consecutive action steps using a temporal segmentation attention module and employs procedure-aware cross-attention along with fine-grained contrastive regression to capture semantic, spatial, and temporal correspondences. Bai *et al.* [2] proposed a temporal parsing transformer that decomposes holistic video features into temporal part-level representations using learnable queries, and employs contrastive regression along with ranking and sparsity losses to predict action quality scores. Xu *et al.* [42] proposed Uni-FineParser, a human-centric fine-grained action parsing framework that focuses on foreground action regions and aligns their semantic, temporal, and spatial structures using refined spatio-temporal annotations to learn more credible representations. However, these methods still fail to provide a traceable and transparent scoring process, which greatly hinders the practical application of diving AQA systems. Wu *et al.* [38] proposed HieroAction, a pioneering work on interpretable AQA that integrates stepwise action reasoning with hierarchical policy learning to enhance hierarchical action understanding and temporal alignment. In contrast, our proposed Diving-R1 offers a possibility to comprehensively integrate information, explicitly evaluate the execution quality of sub-actions throughout the process, and perform step-by-step reasoning, greatly enhancing scoring transparency and thus moving toward interpretable diving AQA.

Multimodal Large Language Models. In recent years, MLLMs have witnessed rapid advances [37, 4, 16]. Supervised Fine-Tuning (SFT) on large-scale instruction-following data has substantially

improved their ability to comprehend complex multimodal inputs, enabling remarkable performance in tasks such as video understanding [52, 47, 13, 17, 22, 18] and visual question answering [12, 6, 1, 45, 55, 51]. To further enhance model capabilities, Reinforcement Fine-Tuning (RFT)-based post-training has increasingly become a dominant paradigm [48]. A representative example is DeepSeek-R1 [9], which adopts Group Relative Policy Optimization (GRPO) within the Reinforcement Learning with Verifiable Rewards (RLVR) framework [11, 34, 9] and has demonstrated substantial gains in mathematical reasoning and code generation. Inspired by these successes, similar ideas have quickly expanded into the multimodal domain, leading to methods such as Reason-RFT [32], Visual-RFT [21], Video-R1 [7], and VideoChat-R1 [14]. These developments open up new opportunities for advancing diving AQA from direct score regression toward structured and interpretable reasoning. However, diving assessment involves highly specialized domain knowledge, and existing models, lacking sufficient domain-specific data, struggle to perform effectively. To address this gap, we introduce DivingThink, which establishes a progressive reasoning chain that enables models to operate like professional judges: systematically attending to each key factor contributing to the final score and deriving the assessment step by step. Furthermore, we construct a new benchmark, DivingInterp, which provides comprehensive support for research on interpretable diving AQA.

3 Methodology

In this section, we introduce the key components involved in building interpretable diving AQA. We begin with the construction of the DivingThink dataset, which contains traceable, progressively structured reasoning chains, serving as the foundation for achieving interpretability. We then present the newly established benchmark, DivingInterp, capable of evaluating the performance of MLLMs from multiple complementary dimensions. Finally, we describe Diving-R1, built upon a carefully designed three-stage training paradigm that gradually incorporates more training corpus and enhances model capabilities, ultimately leading to superior performance.

3.1 DivingThink

DivingThink is grounded in FineDiving [41], which provides annotations for 3,000 diving videos, including action types and their corresponding difficulty degrees, sub-action types, transition points, and final diving scores. However, due to inherent data gaps in official competitions, it is infeasible to annotate the execution quality of individual sub-actions, which restricts fine-grained assessment at the sub-action level. Inspired by TSA [40], which obtains relative scores through comparisons between query and exemplar sub-actions to calculate the final score, we innovatively extend this strategy to assign appropriate quality levels to each sub-action. Specifically, we first group samples with the same action type into a set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ and select one as the query q, while the remaining form the exemplar set $\mathcal{E} = \mathcal{S} \setminus \{q\}$. For each sub-action a_j in the set, the relative score of the query with respect to all exemplars is computed as:

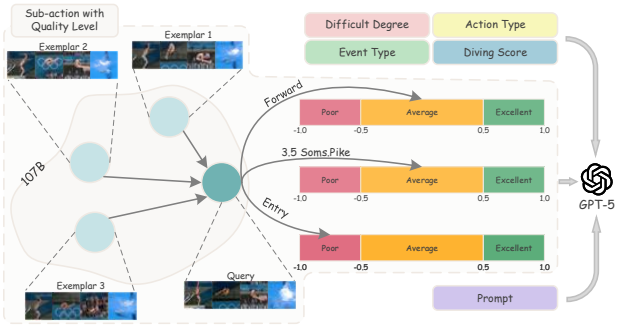


Figure 3: Overview of the construction process of the DivingThink dataset, where the execution quality levels of each sub-action are first estimated through the statistical distribution of query-exemplar contrasts, and then, together with the event type, action type, difficulty degree, and diving score, are fed into GPT-5 guided by carefully designed prompts to generate reasoning chains.

$$r_{q,a_j} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathrm{TSA}(q_{a_j}, e_{a_j}). \tag{1}$$

The same procedure is then applied to every sample in the set, yielding the collection of relative scores for sub-action a_j across all samples: $\mathcal{R}_{a_j} = \{r_{s_1,a_j}, r_{s_2,a_j}, \ldots, r_{s_n,a_j}\}$. Next, the relative score collection is standardized using Z-score normalization, based on the mean μ_{a_j} and standard deviation σ_{a_j}: $\mathcal{Z}_{s_i,a_j} = (r_{s_i,a_j} - \mu_{a_j})/\sigma_{a_j}$. We divide the quality into three levels and estimate them based on the standardized scores, with values below -0.5 assigned *poor*, between -0.5 and 0.5 *average*,

and above 0.5 *excellent*. It is important to note that forcibly dividing scores into three levels may be unreasonable when the distribution variance is very small, so we compute the coefficient of variation for each sub-action score collection as $\mathrm{CV}_{a_j} = \sigma_{a_j}/\mu_{a_j}$, and if $\mathrm{CV}_{a_j} < 0.1$, manual inspection and dedicated evaluation are performed and all samples in the set are assigned an appropriate unified grade for that sub-action. It is worth noting that because the original TSA can only be applied to the three coarse-grained phases of take-off, flight, and entry, we adaptively modify and retrain it to enable the above fine-grained sub-action quality grading process. The rationale behind the Z-score boundary and CV threshold used in this process is detailed in the Appendix.

We align the estimated quality level of each sub-action for every sample with its corresponding sub-action sequence while also supplementing FineDiving with missing competition metadata annotations, including event type (individual or synchronized) and apparatus (springboard or platform). These enriched annotations, together with the action type, difficulty degree, and final score, are fed into GPT-5, which, under carefully designed prompts, is guided to construct a comprehensive, transparent, and progressively structured reasoning chain capable of fine-grained sub-action quality assessment, with the entire process shown in Figure 3. We further craft and randomly assign diverse question templates such as 'What score should be awarded to this dive upon judgment?' to each reasoning chain, forming the initial version of the DivingThink dataset. Finally, three experts with professional diving expertise thoroughly examine the generated content against the corresponding diving videos and structured annotations, and appropriately revise the flagged content with cross-checking among experts to ensure precision and technical soundness, producing the final dataset. Figure 1 provides an example sample of DivingThink, and Table 1 compares the types of annotations available in DivingThink with those in existing Diving datasets, while the prompt used to drive GPT-5, the complete list of question templates, additional samples, and details of the expert revision process are provided in Appendix.

Table 1: Comparison of the constructed DivingThink dataset with existing Diving datasets in terms of the number of samples and available annotation types, where DivingThink is the first dataset to attempt grading sub-action quality and providing progressively structured reasoning chains.

Dataset	Number of samples	Diving Score	Action Type	Sub-action Type	Sub-action Quality	Event Type	Reasoning Chain
MIT Dive [29]	159	✓	✗	✗	✗	✗	✗
UNLV Dive [28]	370	✓	✗	✗	✗	✗	✗
AQA-7-Diving [26]	549	✓	✗	✗	✗	✗	✗
MTL-AQA [27]	1412	✓	✗	✓	✗	✗	✗
FineDiving [40]	3000	✓	✓	✓	✗	✗	✗
DivingThink	3000	✓	✓	✓	✓	✓	✓
DivingSupp	941	✓	✓	✗	✗	✓	✗

3.2 DivingInterp

Based on the constructed DivingThink dataset, we establish a new benchmark, DivingInterp, which follows the test split of FineDiving and contains 749 samples, and aims to comprehensively evaluate the performance of MLLMs in interpretable diving AQA across multiple complementary dimensions. Specifically, we randomly sample one question template and pair it with the diving video as the model input. The generated response is then evaluated from three aspects: First, for score estimation, we measure its ranking ability and numerical deviation using Spearman's rank correlation (ρ) and Relative ℓ_2-distance (R-ℓ_2). Second, for textual quality, we compute the similarity between the generated response and the ground truth using Natural Language Generation (NLG) metrics including BLEU-4, ROUGE-L, METEOR, and BERTScore. Third, to assess reasoning quality beyond what the above metrics can capture, we introduce a GPT-assisted evaluation that assigns scores from 0 to 10 based on the coherence of the reasoning chain, the completeness of the reasoning steps, the consistency between reasoning and conclusion, and the alignment of key elements. Together, these three groups of metrics demonstrate clearly delineated functions and strong complementarity, yielding a thorough and reliable evaluation of MLLMs' capabilities. The specific definitions and details of each metric can be found in the Appendix.

3.3 Diving-R1

To empower MLLMs with traceable, progressive reasoning capabilities for interpretable diving AQA, we adopt a three-stage training paradigm: the first stage employs SFT, followed by two stages of RFT. During training, we gradually relax the reliance on annotations to continuously enrich the diversity of the training corpus, thereby steadily enhancing model performance, ultimately producing Diving-R1. Figure 4 provides an overview of the training paradigm of Diving-R1.

Stage I. Aiming to equip MLLMs with basic reasoning capabilities, we first perform SFT on the training split of the constructed **DivingThink** dataset, serving as a cold-start initialization.

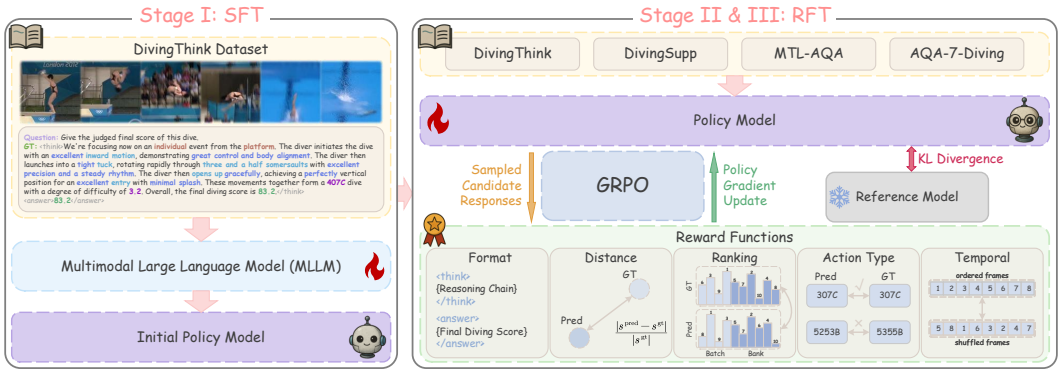


Figure 4: Three-stage training paradigm of Diving-R1, where the first stage is SFT and the following two stages are RFT guided by a carefully designed composite reward function, including the Format Reward, Distance Reward, Ranking Reward, Action-type Reward, and Temporal Reward. The annotation requirements for the training corpus are gradually relaxed across stages, thereby progressively enriching data diversity and achieving continuous improvement for interpretable diving AQA.

The optimization objective is to maximize the likelihood of generating the target response tokens, expressed as:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(\mathbf{T}_{\text{Q}},\mathbf{T}_{\text{V}},\mathbf{T}_{\text{A}})\sim\mathcal{D}} \sum_{t=1}^{T} \log \pi_\theta(y_t \mid \mathbf{T}_{\text{Q}}, \mathbf{T}_{\text{V}}, y_{<t}), \tag{2}$$

where $\pi_\theta(y_t \mid \cdot)$ denotes the conditional probability of generating the t-th token y_t given the randomly sampled question $\mathbf{T}_{\text{Q}}$, the video features $\mathbf{T}_{\text{V}}$ aligned to the textual embedding space, and the previously generated tokens $y_{<t}$. $\mathbf{T}_{\text{A}}$ denotes the reference annotation.

Stage II. After Stage I, the model acquires preliminary reasoning capabilities but often falls into rigid, pattern-matching behaviors induced by SFT, lacking flexibility. To overcome this limitation, we further conduct RFT using Group Relative Policy Optimization (GRPO) under the Reinforcement Learning with Verifiable Rewards (RLVR) framework, encouraging the model to freely explore and internalize more effective and generalizable reasoning strategies to achieve improved interpretable diving AQA. Specifically, the optimization objective can be expressed as:

$$\max_{\pi_\theta} \mathbb{E}_{\mathbf{A}\sim\pi_\theta(\mathbf{Q})} \left[R_{\text{RLVR}}(\mathbf{Q}, \mathbf{A})\right] = \left[R(\mathbf{Q}, \mathbf{A}) - \beta \text{KL}\left[\pi_\theta(\mathbf{A} \mid \mathbf{Q}) \,\|\, \pi_{\text{ref}}(\mathbf{A} \mid \mathbf{Q})\right]\right], \tag{3}$$

where π_θ and π_{ref} are respectively the policy model and the reference model. R is the verifiable reward function, while the KL divergence term penalizes deviations from the reference policy, ensuring both correctness and consistency with prior knowledge. The hyperparameter β balances reward maximization and policy regularization.

Without relying on an additional critic model, GRPO directly compares the relative advantages of responses within a group. Specifically, for a given question $\mathbf{Q}$, GRPO first samples G candidate responses $\{o_1, o_2, \ldots, o_G\}$ from the current policy $\pi_{\theta_{\text{old}}}$, and then assigns corresponding rewards $\{r_1, r_2, \ldots, r_G\}$ to them. By normalizing the rewards using their mean and standard deviation, the relative advantages of each response are computed, based on which the policy is updated to reinforce actions with positive advantages and reduce the probability of less effective ones.

The verifiable reward function R in this stage consists of five components: the Format Reward R_F, the Distance Reward R_D, the Ranking Reward R_R, the Action-type Reward R_A, and the Temporal Reward R_T, which will be described in detail below.

Following DeepSeek-R1 [9], the Format Reward aims to encourage the model to generate responses in the predefined `<think> <answer>` format, assigning 1 if the format is correct and 0 otherwise.

The Distance Reward is used to measure the difference between the predicted final diving score s^{pred} and the ground-truth score s^{gt}, computed as follows:

$$R_D = 1 - \frac{|s^{\text{pred}} - s^{\text{gt}}|}{|s^{\text{gt}}|}. \tag{4}$$

Beyond assessing the accuracy of individual score predictions, another core objective of the diving AQA task is ranking, meaning whether the predicted score order aligns with the ground-truth

order. To capture this aspect, we design a dedicated Ranking Reward to measure the quality of candidate rankings. Specifically, for a batch of size N, each sample produces G candidate scores, resulting in $P = N \times G$ predicted values and their corresponding ground-truth scores, denoted as $\text{Batch}^{\text{pred}} = \{s_1^{\text{pred}}, s_2^{\text{pred}}, \ldots, s_P^{\text{pred}}\}$ and $\text{Batch}^{\text{gt}} = \{s_1^{\text{gt}}, s_2^{\text{gt}}, \ldots, s_P^{\text{gt}}\}$. An intuitive idea is to directly compute the Ranking Reward on the predicted and ground-truth sequences of the current batch, but due to the high variability of data across different batches, doing so at the batch level inevitably leads to unstable training. Therefore, we additionally construct two first-in-first-out score banks of fixed length Q for candidate and ground-truth scores to ensure stability, denoted as $\text{Bank}^{\text{pred}} = \{s_1^{\text{pred}(B)}, s_2^{\text{pred}(B)}, \ldots, s_Q^{\text{pred}(B)}\}$ and $\text{Bank}^{\text{gt}} = \{s_1^{\text{gt}(B)}, s_2^{\text{gt}(B)}, \ldots, s_Q^{\text{gt}(B)}\}$. For each group in the previous batches, we update the banks with only the candidate having the highest R_D (*i.e.*, the most accurate prediction) and its corresponding ground-truth score, rather than adding all candidates, to prevent low-quality predictions from contaminating the ranking and to ensure a cleaner and more stable bank distribution. Overall, we merge the scores from the current batch with the corresponding bank, and compute the Ranking Reward based on the combined set using an approximately differentiable Kendall-style pairwise rank correlation surrogate, taking the i-th element as an example, as follows:

$$R_R = \frac{1}{P+Q-G} \sum_{j \neq i} \text{softsign}\Big((s_i^{\text{pred}} - s_j^{\text{pred}}) \cdot (s_i^{\text{gt}} - s_j^{\text{gt}})\Big), s_i^{\text{pred}}, s_j^{\text{pred}} \in [\text{Batch}^{\text{pred}}; \text{Bank}^{\text{pred}}], s_i^{\text{gt}}, s_j^{\text{gt}} \in [\text{Batch}^{\text{gt}}; \text{Bank}^{\text{gt}}], \quad (5)$$

where $\text{softsign}(x) = \frac{x}{1+|x|}$, which is used to smoothly map values into $[-1, 1]$.

Additionally, during the progressive reasoning process, the model first describes individual sub-actions sequentially and then combines them into an overall action type. Since different action types have varying difficulty degrees, which serve as coefficients directly influencing the final diving score, correctly identifying the action type is also crucial. Therefore, we introduce the Action-type Reward to encourage correct predictions, where the predicted action type a^{pred} is extracted from the model's response using regular expressions and compared with the ground-truth action type a^{gt}, setting $R_A = 0.5$ if they match and 0 otherwise.

Finally, to enhance the model's temporal perception of the input diving videos, following [7], we also consider a Temporal Reward. Specifically, an additional group of candidate responses is generated by shuffling the frame order of the input video, and R_T is set to 0.3 only if the mean R_D of the correctly ordered group exceeds that of the shuffled group; otherwise, it is set to 0. This way, the reward is given only when the current reasoning strategy demonstrates reliance on temporal information, encouraging the model to perceive temporal structure.

In summary, it can be observed that at this stage, the training corpus annotations do not require a full reasoning chain, but only the final diving scores and overall action type. Therefore, in addition to the **DivingThink** dataset used in Stage I, we further incorporate two other datasets to expand the diversity of the training corpus: (1) **MTL-AQA** [27], where we extend the original annotations containing sub-action types to include overall action type encodings; and (2) **DivingSupp**, a supplementary dataset we collect, constructed from recent diving event videos (including the Olympics and World Cup), which provides 941 samples annotated with event types, action types, and diving scores, as shown in Table 1.

Stage III. After Stage II, the model has acquired a strong ability to recognize action types. Building on this, we further broaden the coverage of the training corpus by incorporating the **AQA-7-Diving** [26] dataset, which contains only diving score annotations. We then continue to conduct RFT using the same reward functions as in Stage II, except with the Action-type Reward R_A removed, to obtain the final Diving-R1. All training samples used across the three stages are disjoint from DivingInterp at the video level. Details of the datasets involved in training are provided in the Appendix.

4 Experiments

4.1 Implementation Details

We adopt the efficient Qwen2.5-VL-3B-Instruct [1] as the base MLLM for training, given the limited availability of four NVIDIA RTX A6000 (48 GB) GPUs. During training, Stage I is trained for 15 epochs, while Stage II and Stage III are trained for 2000 iterations each. All stages employ the AdamW optimizer with a learning rate of 1×10^{-6} and a weight decay of 0.01. We also apply gradient

Table 2: Quantitative comparison of the proposed Diving-R1 with representative Instruction MLLMs and Reasoning MLLMs on the DivingInterp benchmark from three perspectives, score estimation, textual similarity, and reasoning quality, where lower R-ℓ_2 indicates better performance, and higher values are better for all other metrics.

Method	SCORE ESTIMATION		TEXTUAL SIMILARITY				REASONING QUALITY			
	ρ	R-ℓ_2($\times$100)	BLEU-4	ROUGE-L	METEOR	BERTScore	Coherence	Completeness	Consistency	Alignment
Instruction MLLMs										
InternVL3-8B [55]	0.1421	17.7387	0.0579	0.2174	0.2797	0.1445	5.37	4.01	5.65	3.54
LLaVA-Video-7B-Qwen2 [52]	0.1316	8.3895	0.0503	0.2178	0.2690	0.1493	5.82	5.71	5.55	4.02
LLaVA-OneVision-Qwen2-7B-OV [12]	0.1182	11.3856	0.0365	0.1885	0.2660	0.1297	4.57	5.32	5.37	4.13
MiniCPM-V-4.5-8B [45]	0.0473	20.8327	0.0622	0.2034	0.2770	0.1455	4.24	5.61	3.97	3.56
Qwen3-VL-8B-Instruct [44]	0.1266	8.7676	0.0628	0.2199	0.3033	0.1567	5.56	5.49	5.72	4.21
VideoLLaMA3-7B [47]	0.0572	9.9369	0.0781	0.2456	0.2657	0.2170	5.85	4.72	4.70	4.30
Reasoning MLLMs										
LongVILA-R1-7B [5]	0.1402	12.6088	0.0294	0.1732	0.1960	0.1285	4.31	5.11	5.17	3.68
Video-R1-7B [7]	0.0596	7.3562	0.0307	0.1838	0.2085	0.1135	5.47	4.59	5.62	4.58
VideoChat-R1-7B [14]	0.0734	8.1924	0.0332	0.2082	0.2126	0.1707	5.29	4.67	5.07	4.62
Qwen2.5-VL-3B-Instruct (Baseline) [1]	0.0798	11.2556	0.0578	0.2121	0.2920	0.1331	5.13	5.27	5.23	4.07
Diving-R1-3B-Stage I	0.4003	2.3063	0.2778	0.5372	0.5189	0.5969	6.96	7.53	6.13	5.88
Diving-R1-3B-Stage II	0.6835	1.1164	0.2881	0.5496	0.5272	0.6129	7.78	8.18	7.40	6.59
Diving-R1-3B	**0.7621**	**0.8841**	**0.2909**	**0.5502**	**0.5344**	**0.6167**	**8.15**	**8.47**	**7.85**	**7.12**

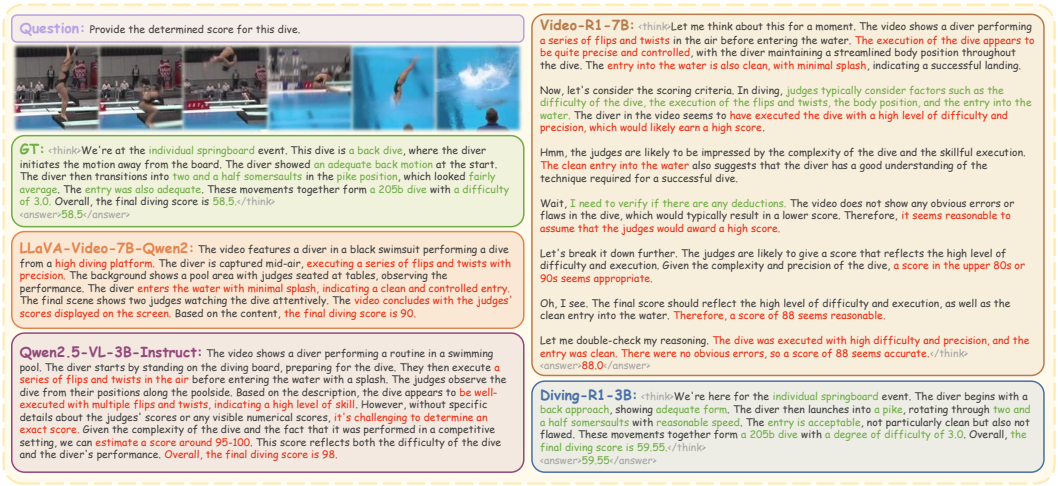


Figure 5: Qualitative comparison of the responses generated by the proposed Diving-R1 and representative MLLMs for interpretable diving AQA, with the full version including comparisons with more MLLMs provided in the Appendix.

clipping with a maximum norm of 5 to ensure training stability. For input videos, we uniformly sample 32 frames. For Ranking Reward R_R, the Bank size Q is set to 512. In the GRPO algorithm, the hyperparameter β in the KL divergence term is set to 0.04, and the candidate group size G is 8. During inference, we follow the official decoding configuration of the Qwen2.5-VL demo, where the temperature is set to 0.1 and top-p to 0.001.

4.2 Quantitative comparison

We compared our approach with existing popular video-capable MLLMs, including Instruction MLLMs and Reasoning MLLMs, on the benchmark DivingInterp, evaluating score estimation, textual similarity, and reasoning quality, with the results shown in Table 2. It can be observed that existing MLLMs nearly fail across the board on interpretable diving AQA, struggling to both evaluate the quality of dives and provide logically coherent reasoning explanations. In contrast, our Diving-R1-3B-Stage I, after SFT on the DivingThink dataset that contains traceable, progressive reasoning chains for score estimation, achieved a significant performance improvement compared to the baseline model, Qwen2.5-VL-3B-Instruct. However, although it performs well on textual similarity metrics, its performance on score estimation and reasoning quality is still unsatisfactory because SFT is essentially a rigid, pattern-matching behavior that only mimics surface patterns and cannot capture the deep causal connections between the reasoning chains and the final diving scores. Further RFT, through a reward mechanism, guides the model to explore more effective reasoning paths and, along with progressively enriched training data in Stages II and III, achieves continuous performance improvement, ultimately enabling Diving-R1 to attain superior performance across all evaluation dimensions. Additionally, for the GPT-assisted reasoning-quality evaluation, we further

Table 3: Ablation analysis on the key elements within the progressive reasoning chain that are closely related to the final diving score.

Diving Score	Action Type	Difficulty Degree	Event Type	Sub-action Type	Sub-action Quality	SCORE ESTIMATION		TEXTUAL SIMILARITY				REASONING QUALITY			
						ρ	R-$\ell_2(\times 100)$	BLEU-4	ROUGE-L	METEOR	BERTScore	Coherence	Completeness	Consistency	Alignment
✓						0.5044	1.7375	0.1025	0.2746	0.3454	0.2318	5.19	5.39	5.27	4.28
✓	✓					0.5461	1.5851	0.1786	0.3941	0.3464	0.4391	5.72	5.56	5.49	4.47
✓	✓	✓				0.5741	1.4607	0.1897	0.4426	0.3908	0.4784	6.03	5.97	5.76	5.06
✓	✓	✓	✓			0.6025	1.3381	0.2050	0.4583	0.4077	0.4893	6.26	6.45	6.02	5.21
✓	✓	✓	✓	✓		0.6738	1.1606	0.2620	0.5267	0.4689	0.5704	7.37	7.04	6.27	6.09
✓	✓	✓	✓	✓	✓	0.7621	0.8841	0.2909	0.5502	0.5344	0.6167	8.15	8.47	7.85	7.12

Table 4: Ablation analysis on the reward functions used during the RFT stage.

R_F	R_D	R_T	R_R	R_A	SCORE ESTIMATION		TEXTUAL SIMILARITY				REASONING QUALITY			
					ρ	R-$\ell_2(\times 100)$	BLEU-4	ROUGE-L	METEOR	BERTScore	Coherence	Completeness	Consistency	Alignment
✓	✓				0.6307	1.2386	0.2831	0.5450	0.5209	0.6100	7.59	8.07	7.22	6.28
✓	✓	✓			0.6544	1.1916	0.2888	0.5437	0.5305	0.6103	7.93	8.31	7.57	6.64
✓	✓	✓	✓		0.7412	0.9426	0.2859	0.5449	0.5322	0.6116	8.06	8.38	7.78	6.91
✓	✓	✓	✓	✓	0.7621	0.8841	0.2909	0.5502	0.5344	0.6167	8.15	8.47	7.85	7.12

assess its consistency with human expert judgments on sampled cases, observing high agreement and correlation, which supports its reliability, with details provided in the Appendix.

4.3 Qualitative analysis

We conducted qualitative analysis and comparison of the responses generated by different MLLMs, with several representative examples shown in Figure 5. It can be observed that Instruction MLLMs, such as LLaVA-Video-7B-Qwen2 and Qwen2.5-VL-3B-Instruct, often struggle to accurately recognize the action types and sub-action processes in diving videos. Instead, they tend to give vague descriptions such as 'a series of flips and twists', making it difficult to reasonably assess the corresponding degree of difficulty. In addition, they lack a correct understanding of the execution quality, frequently misjudging it as 'clean and controlled entry' or 'a high level of skill'. Such issues ultimately lead to overestimated scores and large deviations. The Reasoning MLLM Video-R1-7B demonstrates a certain level of reasoning and reflection ability. During generation, it performs self-verification and confirmation, and is able to correctly mention several key factors that influence the final diving score. However, it still struggles with correctly judging the action types and execution quality, eventually leading to an incorrect final score. In contrast, our proposed Diving-R1-3B adopts a progressive reasoning strategy, first identifying the event type of the dive, then sequentially recognizing each sub-action in the diving process and evaluating their execution quality, subsequently integrating the sub-actions to infer the overall action type and degree of difficulty, thereby precisely estimating the final score with minimal prediction error. For more examples of responses from various MLLMs, please refer to the Appendix.

4.4 Ablation Study

Progressive reasoning chain. In conducting interpretable diving AQA, the progressive reasoning chains in the DivingThink dataset incorporate key information closely related to the final diving scores in a gradual manner, including the event type, action type and its corresponding degree of difficulty, sub-action type, and descriptions of sub-action execution quality. We performed an ablation analysis on these components to investigate their effectiveness, with the results shown in Table 3. Specifically, we started from an initial setting that included only the diving scores and gradually incorporated the aforementioned key information into the training corpus, while keeping all other training conditions unchanged. The results demonstrate that as the information in the reasoning chain becomes progressively richer, the model's performance improves accordingly. In particular, introducing the descriptions of sub-action execution quality brings a significant performance boost, indicating that focusing solely on action types within the reasoning chain is insufficient, while explicitly decomposing and evaluating sub-actions plays a crucial positive role in accurately estimating the final diving scores. **Reward functions.** During RFT in Stages II and III, the composite reward functions play a crucial guiding role for the model in exploring reasoning paths. Building on the basic format reward R_F and distance reward R_D, we sequentially introduced the temporal reward R_T, ranking reward R_R, and action-type reward R_A to investigate their specific effects, with the results shown in Table 4. It can be observed that the introduction of R_T effectively enhances the coherence and consistency of reasoning, as it encourages the model to perceive temporal dependencies; the addition of R_R significantly improves the Spearman's rank correlation, benefiting from its positive reinforcement for correct ranking; and the explicit guidance provided by R_A for action type recognition further strengthens the capability of the model in interpretable diving AQA.

In addition to the above analyses, we also conduct ablation studies to disentangle the effects of RFT and data expansion and to examine key design choices, with details in the Appendix.

5 Conclusion and Future Work

In this paper, we propose Diving-R1, the first MLLM designed for interpretable diving AQA that is capable of progressive reasoning, providing traceable assessments, and integrating comprehensive information. We construct DivingThink, a dataset containing structured reasoning chains that explicitly analyze the execution quality of sub-actions, serving as a foundation for interpretable diving AQA, and establish a new benchmark, DivingInterp, to evaluate the performance of MLLMs from multiple complementary perspectives. Furthermore, we carefully design a three-stage training paradigm combining SFT and RFT, equipped with dedicated composite reward functions, and gradually relax the annotation requirements for the training corpus at each stage, successively enriching the coverage and diversity of the training data to achieve continuous performance improvement. Extensive experiments demonstrate the superiority of the proposed Diving-R1 compared to existing MLLMs across multiple aspects, including score estimation, textual similarity, and reasoning quality. Future work envisions moving beyond video inputs alone by incorporating additional modalities, such as motion information, to help the model better capture and assess actions in diving sequences.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62676345), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2026A1515011579), the HKUST-HKUST(GZ) 1+1+1 Joint Funding Program (Grant No. C_2025_031), and the Guangzhou-HKUST(GZ) Joint Funding Program (Grant No. 2023A03J0008), Education Bureau of Guangzhou Municipality. This work was also supported by Jiangsu Industrial Technology Research Institute (JITRI) and Wuxi National High-Tech District (WND).

References

[1] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] Y. Bai, D. Zhou, S. Zhang, J. Wang, E. Ding, Y. Guan, Y. Long, and J. Wang. Action quality assessment with temporal parsing transformer. In *European conference on computer vision*, pages 422–438. Springer, 2022.

[3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[4] D. Caffagni, F. Cocchi, L. Barsellotti, N. Moratelli, S. Sarto, L. Baraldi, M. Cornia, and R. Cucchiara. The revolution of multimodal large language models: a survey. *arXiv preprint arXiv:2402.12451*, 2024.

[5] Y. Chen, W. Huang, B. Shi, Q. Hu, H. Ye, L. Zhu, Z. Liu, P. Molchanov, J. Kautz, X. Qi, et al. Scaling rl to long videos. *arXiv preprint arXiv:2507.07966*, 2025.

[6] W. Dai, J. Li, D. Li, A. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, and S. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

[7] K. Feng, K. Gong, B. Li, Z. Guo, Y. Wang, T. Peng, J. Wu, X. Zhang, B. Wang, and X. Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

[8] K. Gedamu, Y. Ji, Y. Yang, J. Shao, and H. T. Shen. Fine-grained spatio-temporal parsing network for action quality assessment. *IEEE Transactions on Image Processing*, 32:6386–6400, 2023.

[9] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[10] W. Huang, B. Jia, Z. Zhai, S. Cao, Z. Ye, F. Zhao, Z. Xu, Y. Hu, and S. Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.

[11] N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, S. Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[12] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[13] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[14] X. Li, Z. Yan, D. Meng, L. Dong, X. Zeng, Y. He, Y. Wang, Y. Qiao, Y. Wang, and L. Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

[15] Y. Li, X. Chai, and X. Chen. End-to-end learning for action quality assessment. In *Pacific Rim Conference on Multimedia*, pages 125–134. Springer, 2018.

[16] Y. Li, Z. Lai, W. Bao, Z. Tan, A. Dao, K. Sui, J. Shen, D. Liu, H. Liu, and Y. Kong. Visual large language models for generalized and specialized applications. *arXiv preprint arXiv:2501.02765*, 2025.

[17] Y. Li, C. Wang, and J. Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024.

[18] B. Lin, Y. Ye, B. Zhu, J. Cui, M. Ning, P. Jin, and L. Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5971–5984, 2024.

[19] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[20] J. Liu, H. Wang, K. Stawarz, S. Li, Y. Fu, and H. Liu. Vision-based human action quality assessment: A systematic review. *Expert Systems with Applications*, 263:125642, 2025.

[21] Z. Liu, Z. Sun, Y. Zang, X. Dong, Y. Cao, H. Duan, D. Lin, and J. Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025.

[22] M. Maaz, H. Rasheed, S. Khan, and F. Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602, 2024.

[23] A. Majeedi, V. R. Gajjala, S. S. S. N. GNVV, and Y. Li. Rica^ 2: Rubric-informed, calibrated assessment of actions. In *European Conference on Computer Vision*, pages 143–161. Springer, 2024.

[24] J.-H. Pan, J. Gao, and W.-S. Zheng. Action assessment by joint relation graphs. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6331–6340, 2019.

[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[26] P. Parmar and B. Morris. Action quality assessment across multiple actions. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1468–1476. IEEE, 2019.

[27] P. Parmar and B. T. Morris. 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*, 2019.

[28] P. Parmar and B. Tran Morris. Learning to score olympic events. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 20–28, 2017.

[29] H. Pirsiavash, C. Vondrick, and A. Torralba. Assessing the quality of actions. In *European conference on computer vision*, pages 556–571. Springer, 2014.

[30] K. Rangasamy, M. A. As'ari, N. A. Rahmad, N. F. Ghazali, and S. Ismail. Deep learning in sport video analysis: a review. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 18(4):1926–1933, 2020.

[31] H.-C. Shih. A survey of content-aware video analysis for sports. *IEEE Transactions on circuits and systems for video technology*, 28(5):1212–1231, 2017.

[32] H. Tan, Y. Ji, X. Hao, M. Lin, P. Wang, Z. Wang, and S. Zhang. Reason-rft: Reinforcement fine-tuning for visual reasoning. *arXiv preprint arXiv:2503.20752*, 2025.

[33] Y. Tang, Z. Ni, J. Zhou, D. Zhang, J. Lu, Y. Wu, and J. Zhou. Uncertainty-aware score distribution learning for action quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9839–9848, 2020.

[34] K. Team, A. Du, B. Gao, B. Xing, C. Jiang, C. Chen, C. Li, C. Xiao, C. Du, C. Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[35] S. Wang, D. Yang, P. Zhai, C. Chen, and L. Zhang. Tsa-net: Tube self-attention network for action quality assessment. In *Proceedings of the 29th ACM international conference on multimedia*, pages 4902–4910, 2021.

[36] S. Wang, D. Yang, P. Zhai, Q. Yu, T. Suo, Z. Sun, K. Li, and L. Zhang. A survey of video-based action quality assessment. In *2021 International conference on networking systems of AI (INSAI)*, pages 1–9. IEEE, 2021.

[37] J. Wu, W. Gan, Z. Chen, S. Wan, and P. S. Yu. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256. IEEE, 2023.

[38] J. Wu, X. Gu, Z. Li, Y. Jin, Y. Diao, Z. Li, Z. Song, X. Zhang, and Z. Fan. Hieroaction: Hierarchically guided vlm for fine-grained action analysis. *arXiv preprint arXiv:2508.16942*, 2025.

[39] H. Xu, X. Ke, Y. Li, R. Xu, H. Wu, X. Lin, and W. Guo. Vision-language action knowledge learning for semantic-aware action quality assessment. In *European Conference on Computer Vision*, pages 423–440. Springer, 2024.

[40] J. Xu, Y. Rao, X. Yu, G. Chen, J. Zhou, and J. Lu. Finediving: A fine-grained dataset for procedure-aware action quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2949–2958, 2022.

[41] J. Xu, Y. Rao, J. Zhou, and J. Lu. Procedure-aware action quality assessment: Datasets and performance evaluation. *International Journal of Computer Vision*, 132(12):6069–6090, 2024.

[42] J. Xu, S. Yin, and Y. Peng. Human-centric fine-grained action quality assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[43] J. Xu, S. Yin, G. Zhao, Z. Wang, and Y. Peng. Fineparser: A fine-grained spatio-temporal action parser for human-centric action quality assessment. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 14628–14637, 2024.

[44] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[45] T. Yu, Z. Wang, C. Wang, F. Huang, W. Ma, Z. He, T. Cai, W. Chen, Y. Huang, Y. Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. *arXiv preprint arXiv:2509.18154*, 2025.

[46] X. Yu, Y. Rao, W. Zhao, J. Lu, and J. Zhou. Group-aware contrastive regression for action quality assessment. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7919–7928, 2021.

[47] B. Zhang, K. Li, Z. Cheng, Z. Hu, Y. Yuan, G. Chen, S. Leng, Y. Jiang, H. Zhang, X. Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

[48] K. Zhang, Y. Zuo, B. He, Y. Sun, R. Liu, C. Jiang, Y. Fan, K. Tian, G. Jia, P. Li, et al. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025.

[49] S. Zhang, S. Bai, G. Chen, L. Chen, J. Lu, J. Wang, and Y. Tang. Narrative action evaluation with prompt-guided multimodal interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18430–18439, 2024.

[50] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[51] Y. Zhang, H. Lu, Q. Hu, Y. Wang, K. Yuan, X. Liu, and K. Wu. Period-llm: Extending the periodic capability of multimodal large language model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29237–29247, 2025.

[52] Y. Zhang, J. Wu, W. Li, B. Li, Z. Ma, Z. Liu, and C. Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

[53] Y. Zhang, K. Yuan, H. Lu, Y. Yue, J. Chen, and K. Wu. Medtvt-r1: A multimodal llm empowering medical reasoning and diagnosis. *arXiv preprint arXiv:2506.18512*, 2025.

[54] K. Zhou, R. Cai, L. Wang, H. P. Shum, and X. Liang. A comprehensive survey of action quality assessment: Method and benchmark. *arXiv preprint arXiv:2412.11149*, 2024.

[55] J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, H. Tian, Y. Duan, W. Su, J. Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A DivingThink

In constructing the DivingThink dataset, after aligning the estimated quality level of each sub-action for every sample with its corresponding sub-action sequence, we feed it together with other key elements including event type, action type, difficulty degree, and final diving score into GPT-5, using a carefully designed prompt to guide the generation of a comprehensive, transparent, and progressively structured reasoning chain for interpretable diving AQA, with the full prompt shown in Table 5. The action types involved and their corresponding decomposed sub-action sequences in the DivingThink dataset are detailed in Table 9. We further craft and randomly assign diverse question templates for each generated reasoning chain, as listed in Table 6. Some additional sample examples of the DivingThink dataset can be found in Figures 6 and 7.

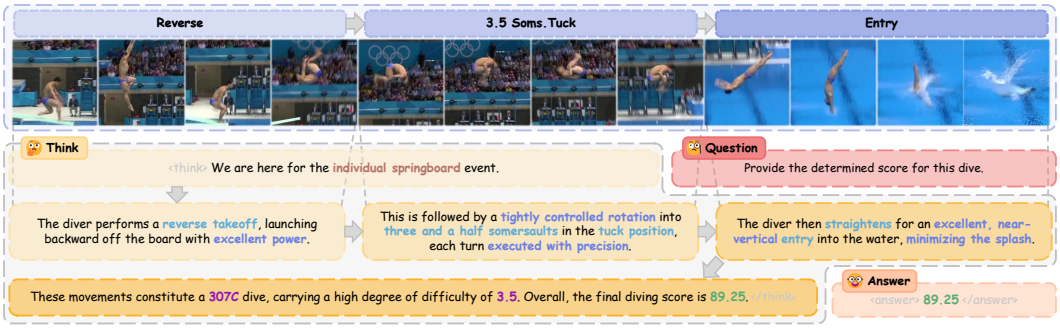


Figure 6: An example sample from the DivingThink dataset, featuring a progressively structured reasoning chain for a 307C dive (Reverse → 3.5 Soms.Tuck → Entry).



Figure 7: An example sample from the DivingThink dataset, featuring a progressively structured reasoning chain for a 5353B dive (Reverse → 1.5 Twists → 2.5 Soms.Pike → Entry).

B Datasets

Besides the DivingThink dataset, it also involves AQA-7 [26], MTL-AQA [27], FineDiving [40], and DivingSupp, which will be introduced in detail below.

AQA-7 [26] is a dataset for AQA, containing 1,189 video samples from seven Olympic events: Single Diving 10m Platform (370), Gymnastic Vault (176), Big Air Skiing (175), Big Air Snowboarding (206), Synchronized Diving 3m Springboard (88), Synchronized Diving 10m Platform (91), and Trampoline (83), along with corresponding quality scores from professional judges, of which we use the 549 samples related to diving.

MTL-AQA [27] is a dataset that supports multi-task diving AQA, containing 1,412 diving video samples from 16 different international competitions, covering various event types such as 3m springboard and 10m platform, as well as individual and synchronized diving for both men and women. Each sample is provided with three types of annotations, including diving quality scores from expert judges, fine-grained action decomposition (covering pose type, rotation style, number of somersaults and twists, etc.), and descriptive commentary generated by professional commentators; here, we further encode the fine-grained action decomposition into overall action type for training.

Table 5: The carefully designed prompt for guiding GPT-5 to integrate the complete information of the dive and generate a comprehensive, transparent, and progressively structured reasoning chain.

Here is the complete information of the dive:
<Event> <Action type> <Difficulty> <Final score> <Sub actions with quality>.
You are asked to generate commentary in English based on the dive information provided above, following this structure.
Use a calm, natural tone; avoid overly dramatic or exaggerated language, and avoid filler words or habitual phrases like 'Alright' or 'Tonight'.
Also, ensure to use singular or plural forms for the diver(s) correctly based on the event type (Individual vs Synchronized).

1. The commentary must begin inside <think> ... </think>.
(a) Begin by naturally describing the competition type, making it clear whether it is an Individual or Synchronized event, and whether it takes place from the springboard or platform. Phrase the introduction in a smooth, conversational way so that each commentary opening feels varied and authentic.
(b) Describe the dive by narrating the sequence of sub-actions in natural, human-like language. For each sub-action:
- Explain what the diver does physically (e.g., rotation, tuck, body posture, approach, or entry).
- Convey the quality of execution in rich, descriptive terms (e.g., smooth, controlled, elegant, slightly rushed, unstable, precise, balanced, steady, fluid, hesitant, tentative). Do not rigidly stick to the adjectives shown in the input field 'Sub actions with quality'; instead, integrate them naturally, fluently, and with varied expression.
- Avoid just stating the action name; instead, describe it as if narrating live: 'They launch into a tight tuck, spinning rapidly with control' instead of 'Tuck'.
- Make the narration flow naturally, mentioning the order and rhythm of the movements.
(c) Conclude this section by summarizing the overall dive (e.g., 'These movements together form a 407c dive with a degree of difficulty of 3.2'), and include a single fixed sentence stating the final score: 'The final diving score is {score}.'

2. After <think> ... </think>, output the final diving score inside <answer> ... </answer>: '<answer> {score} </answer>'.

The <think> tag must appear at the very start of the line with no leading spaces or line breaks.
The </think> tag must appear immediately after the content, with no extra line breaks.
The <answer> tag must appear immediately after the last </think> content, with no leading spaces or line breaks.
Ensure that the <think> and <answer> sections strictly follow this clean structure without extra characters or formatting.

Strictly follow this structure, making the <think> section vivid, natural, and like a live sports commentary, and do not just repeat the action type names. Ensure that the <answer> section always uses the fixed template with the actual score dynamically inserted.

Table 6: The complete list of question templates in the DivingThink dataset.

- "What is the judged score for this dive?"
- "How is this dive measured and scored?"
- "What score should be awarded to this dive upon judgment?"
- "How high is this dive judged to score?"
- "What score would you judge this dive to deserve?"
- "What does this dive score when judged?"
- "Give the judged final score of this dive."
- "Provide the determined score for this dive."
- "State the judged score of this dive."
- "Report the final diving score as judged."

FineDiving [40] is a dataset that provides fine-grained action procedure annotations for diving AQA, containing 3,000 diving video samples covering 52 action types, 29 sub-action types, and 23 difficulty degrees, with precise temporal markings of action boundaries and step transition points, enabling detailed action decomposition, and serving as the foundation of DivingThink for interpretable diving AQA.

DivingSupp is a dataset we meticulously collect that provides an effective further expansion of the diversity of the training corpus, sourced competition videos from the past two years of Olympics and World Cups, covering various event types such as 3m springboard and 10m platform, as well as individual and synchronized diving for both men and women, among which we annotate 941 samples with event types, action types, and diving scores.

C Sensitivity Analysis for Sub-action Quality Grading

To make the grading of sub-action execution quality more reasonable and as consistent as possible with expert perspectives, we first conducted preliminary sensitivity analysis before large-scale generation. Specifically, for the Z-score boundaries, we sampled 300 representative examples covering all

action types, and asked experts to label the execution quality of each sub-action in these samples as *poor* / *average* / *excellent*. We then applied ± 0.3, ± 0.4, ± 0.5, ± 0.6, and ± 0.7 as the Z-score boundaries for grading, and measured the percent agreement with the expert annotations, obtaining 80.7%, 94.9%, 96.4%, 95.2%, and 81.5%, respectively, which shows that sub-action quality grading remains relatively stable when the boundary lies within ± 0.4 to ± 0.6, whereas more extreme settings substantially reduce agreement with expert judgments, so we adopted ± 0.5 as the practical boundary.

For the CV threshold, we considered 170 candidate sub-action groups in total, applied thresholds of 0.05, 0.1, 0.15, and 0.2, which selected 2, 13, 25, and 39 groups, respectively, and then asked experts to make a binary judgment on whether the execution quality within each selected group showed no significant difference, yielding expert-confirmed proportions of 100%, 100%, 52%, and 33.3%, respectively. We therefore selected 0.1 as the CV threshold, considering that it provides broader coverage than 0.05 while avoiding the many groups with significant quality differences introduced by looser thresholds such as 0.15 and 0.2.

D Expert Revision

To ensure the precision and technical soundness of the progressively structured reasoning chains in the DivingThink dataset, after obtaining the preliminary GPT-5-generated version, we conducted expert review and revision.

Specifically, three experts with professional diving expertise first independently reviewed each chain against the corresponding diving video and structured annotations, and assigned each chain one of three labels: no revision, minor revision, or major revision; the inter-expert agreement yielded a Fleiss' κ of 0.91, which further increased to 0.97 after collapsing the labels into revision needed versus no revision, indicating very high consistency.

Following the independent audit, all samples flagged by at least one expert as requiring revision entered a controlled revision process: one expert first revised the chain, a second cross-checked it, and a third intervened only when disagreements remained and adjudication was needed. Through a detailed examination of the revision cases, we observed that minor revisions mainly involved calibrating the wording of sub-action execution quality, such as avoiding overly absolute descriptions or correcting slight over- or under-estimation. By contrast, major revisions were concentrated in a small number of complex five-step action types, such as 5132D and 5152B, where misdescription is more likely because the second and fourth steps are easily mistaken for two independent sub-actions, although they in fact form a single continuous movement sequence, with the third step embedded and interleaved within it.

In the final statistics, 95.5% of the samples required no revision, 3.3% underwent minor revision, and only 1.2% underwent major revision, demonstrating that our data construction pipeline and carefully designed prompts effectively ensure the reliability of the generated progressively structured reasoning chains to a high degree in practice, with only a very small fraction requiring human intervention.

E Evaluation metrics

In the newly established benchmark DivingInterp, we evaluate the performance of MLLMs on interpretable diving AQA from three complementary dimensions, including score estimation, textual similarity, and reasoning quality, as detailed below.

For score estimation, we use Spearman's rank correlation (ρ) to measure the ranking ability of MLLMs, which is defined as:

$$\rho = \frac{\sum_{i=1}^{N}(r_i^{\text{gt}} - \overline{r^{\text{gt}}})(r_i^{\text{pred}} - \overline{r^{\text{pred}}})}{\sqrt{\sum_{i=1}^{N}(r_i^{\text{gt}} - \overline{r^{\text{gt}}})^2 \sum_{i=1}^{N}(r_i^{\text{pred}} - \overline{r^{\text{pred}}})^2}}, \tag{6}$$

where $\mathbf{r}^{\mathbf{gt}}$ and $\mathbf{r}^{\mathbf{pred}}$ denote the rankings of the ground-truth and predicted final diving scores, respectively. We also employ Relative ℓ_2-distance (R-ℓ_2) to assess the numerical deviation of MLLMs' predictions. Given the highest and lowest scores of an action, specifically $s_{\max}$ and $s_{\min}$, R-ℓ_2 can

be calculated as:

$$\text{R-}\ell_2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|s_i^{\text{gt}} - s_i^{\text{pred}}|}{s_{\max} - s_{\min}}\right), \tag{7}$$

where s_i^{gt} and s_i^{pred} denote the ground-truth and predicted scores for the i-th sample, respectively.

For textual similarity, we adopt commonly used metrics in Natural Language Generation (NLG), including BLEU-4 [25], ROUGE-L [19], METEOR [3], and BERTScore [50], to compute how closely MLLMs' generated responses match the ground truth, capturing both lexical overlap and semantic relevance.

For reasoning quality, which goes beyond what the above metrics can capture, we additionally introduce GPT-assisted evaluation to assign scores from 0 to 10 to generated responses from MLLMs based on the coherence of the reasoning chain, the completeness of the reasoning steps, the consistency between reasoning and conclusion, and the alignment of key elements.

The prompt for evaluating coherence is:

> Evaluate the ***coherence*** of the reasoning sequence in the Predicted reasoning text by assessing whether *the described movements form a physically plausible and temporally consistent sequence* (e.g., takeoff → rotation → entry), whether *the causal links between actions are logical and continuous with no contradictions or missing transitions*, and whether *the predicted reasoning remains aligned with the general context of the ground truth reasoning* (exact wording match is not required). ONLY OUTPUT A NUMBER BETWEEN 0 AND 10 (decimals allowed). Predicted reasoning text: `{pred}`. Ground truth reasoning text: `{gt}`.

The prompt for evaluating completeness is:

> Evaluate the ***completeness*** of the reasoning in the Predicted reasoning text for a diving action description by checking whether it covers all key components of a standard dive, including the *event type* (individual or synchronized), *apparatus* (springboard or platform), *takeoff phase*, *mid-air twists/somersaults and body position* (tuck, pike), *water entry and control or splash quality*, *dive code* (e.g., 107B), *degree of difficulty*, and *overall flow* to ensure the reasoning is continuous and no major steps are missing. ONLY OUTPUT A NUMBER BETWEEN 0 AND 10 (decimals allowed). Predicted reasoning text: `{pred}`.

The prompt for evaluating consistency is:

> Evaluate the ***consistency*** between the reasoning and the final score in the Predicted reasoning text for a diving description by assessing whether *the described action qualities (takeoff, rotation, entry) support the given final score*, whether *the score level (high, medium, low) is reasonable based on the described performance quality*, whether *any contradictions exist between the reasoning and the score* (e.g., many mistakes but a very high score), and whether *the reasoning logically justifies the stated final score*. ONLY OUTPUT A NUMBER BETWEEN 0 AND 10 (decimals allowed). Predicted reasoning text: `{pred}`.

The prompt for evaluating alignment is:

> Evaluate the ***alignment*** of key elements between the Predicted reasoning text and the Ground Truth reasoning text in a diving description by determining whether the predicted reasoning corresponds to the ground truth across *event type* (individual or synchronized), *apparatus* (springboard or platform), *takeoff and approach quality*, *mid-air movements including the presence and number of twists or somersaults and body position* (tuck, pike, straight), and *their execution quality*, *entry description and quality*, *dive identification* such as the *dive code* (e.g., 107B), *degree of difficulty*, and *the overall alignment of action flow and performance quality*. ONLY OUTPUT A NUMBER BETWEEN 0 AND 10 (decimals allowed). Predicted reasoning text: `{pred}`. Ground truth reasoning text: `{gt}`.

F Reliability of GPT-assisted Evaluation

When evaluating reasoning quality on the DivingInterp benchmark, we introduce GPT-assisted evaluation, which raises a concern about whether it can reliably substitute for expert human judgment in such a specialized sports domain as diving. To directly examine this concern, we sampled 150 test cases and invited three experts to independently score the four dimensions on a 0-10 scale under the same scoring instructions as GPT-assisted evaluation; the inter-expert agreement is high, with an overall ICC of 0.85, and the mean expert scores show strong correlation with the corresponding GPT-assisted scores, with Spearman's ρ of 0.83 / 0.82 / 0.87 / 0.89 for coherence / completeness / consistency / alignment, respectively. These results suggest that the relative judgments are reasonably stable between GPT-assisted evaluation and human experts, benefiting from the explicitly specified

and clearly defined criteria; therefore, compared with extremely costly human scoring, GPT-assisted evaluation can be reliably used as a practical and scalable trade-off.

G Qualitative Analysis

Figure 8 presents a comprehensive qualitative analysis and comparison of the responses generated by popular video-capable MLLMs and Diving-R1-3B. The results lead to observations consistent with those in Section 4.3, namely that although some models can mention certain key factors affecting the final diving score, existing MLLMs struggle to accurately identify the action types and sub-action sequences in diving videos, instead giving vague descriptions, and lack the ability to correctly understand the quality of execution, which ultimately results in incorrect scoring and large deviations, thereby highlighting the superiority of Diving-R1-3B in interpretable diving AQA.



Figure 8: Comprehensive qualitative comparison of responses generated by the proposed Diving-R1 and existing video-capable MLLMs for interpretable diving AQA.

H Additional Ablation Studies

H.1 Disentangling RFT and Data Expansion

To investigate the effects of the optimization strategy and data expansion, as shown in Table 7, we conducted two disentangled controlled ablations as follows.

(1) **Fixed-data RFT.** We repeated Stage II using RFT on DivingThink only, *i.e.*, with exactly the same training corpus as Stage I SFT and without introducing DivingSupp or MTL-AQA, and it can be observed that this setting still improves performance substantially over Stage I. This shows that, even with the training corpus fixed, the proposed RFT paradigm still brings a clear gain over SFT by encouraging more effective reasoning rather than rigid SFT-style pattern matching. Meanwhile, the original Stage II still outperforms this fixed-data RFT variant, indicating that broader data coverage provides additional benefit beyond the optimization effect.

Table 7: Ablation analysis results for disentangling the effects of RFT and data expansion, including a fixed-data RFT variant using DivingThink only and an SFT-only variant continued on the enlarged corpus.

Method	SCORE ESTIMATION		TEXTUAL SIMILARITY				REASONING QUALITY			
	ρ	R-$\ell_2(\times 100)$	BLEU-4	ROUGE-L	METEOR	BERTScore	Coherence	Completeness	Consistency	Alignment
Qwen2.5-VL-3B-Instruct (Baseline)	0.0798	11.2556	0.0578	0.2121	0.2920	0.1331	5.13	5.27	5.23	4.07
Diving-R1-3B-Stage I	0.4003	2.3063	0.2778	0.5372	0.5189	0.5969	6.96	7.53	6.13	5.88
Diving-R1-3B-Stage II	0.6835	1.1164	0.2881	0.5496	0.5272	0.6129	7.78	8.18	7.40	6.59
Diving-R1-3B-Stage I + continued SFT (enlarged corpus)	0.3684	2.6427	0.2489	0.5116	0.4867	0.5824	6.47	6.72	5.79	5.26
Diving-R1-3B-Stage II (DivingThink only)	0.6057	1.4092	0.2833	0.5441	0.5238	0.6065	7.42	7.87	7.01	6.31

Table 8: Ablation analysis results of key design choices, including the Ranking Reward bank size (Q), bank inclusion criterion, and sampled frame number in Diving-R1. Default settings are in gray.

Settings	SCORE ESTIMATION		TEXTUAL SIMILARITY				REASONING QUALITY			
	ρ	R-$\ell_2(\times 100)$	BLEU-4	ROUGE-L	METEOR	BERTScore	Coherence	Completeness	Consistency	Alignment
Bank Size										
0	0.6662	1.2017	0.2859	0.5453	0.5308	0.6101	7.92	8.29	7.51	6.62
64	0.7108	1.1035	0.2876	0.5483	0.5342	0.6129	8.08	8.40	7.69	6.96
256	0.7316	0.9399	0.2901	0.5495	0.5337	0.6157	8.10	8.43	7.76	7.07
512	0.7621	0.8841	0.2909	0.5502	0.5344	0.6167	8.15	8.47	7.85	7.12
768	0.7623	0.8845	0.2914	0.5496	0.5351	0.6158	8.17	8.44	7.88	7.10
Bank Inclusion Criterion										
All	0.6587	1.1532	0.2876	0.5459	0.5301	0.6098	7.89	8.28	7.46	6.53
Highest only	0.7621	0.8841	0.2909	0.5502	0.5344	0.6167	8.15	8.47	7.85	7.12
Number of Sampled Frames										
8	0.3218	2.5361	0.2016	0.4437	0.4218	0.5024	6.92	7.19	5.89	4.59
16	0.5363	1.5806	0.2478	0.5021	0.4796	0.5653	7.61	7.84	6.90	5.97
32	0.7621	0.8841	0.2909	0.5502	0.5344	0.6167	8.15	8.47	7.85	7.12
48	0.7652	0.8815	0.2933	0.5478	0.5371	0.6152	8.18	8.44	7.91	7.07

(2) **Added data without RFT.** We further constructed an SFT-only variant by continuing Stage I with SFT on the enlarged corpus (DivingThink + DivingSupp + MTL-AQA), *i.e.*, the same data configuration as Stage II but without RFT. We find that this setting not only fails to improve over Stage I, but even causes noticeable degradation, especially in textual similarity and reasoning quality. We speculate that this is mainly because the additional data provide only action types and final diving scores rather than full reasoning chains; thus, naively mixing them into SFT introduces incompatible supervision signals and disrupts the reasoning format learned in Stage I. In contrast, RFT can effectively exploit such weakly annotated data through verifiable rewards, which is exactly the advantage of our stage-wise design: it relaxes annotation requirements while still turning broader data coverage into performance gains.

Taken together, these two disentangled controlled ablations suggest that both factors matter: RFT contributes independently even on a fixed dataset, while additional data are beneficial only when combined with the proposed RFT paradigm, rather than naively added through SFT.

H.2 Ablations on Key Design Choices

We also performed ablation analysis on several key design choices in the proposed Diving-R1, including the bank size Q in the Ranking Reward, the bank inclusion criterion, and the number of sampled video frames, as shown in Table 8.

For the bank size, we consider $Q \in \{0, 64, 256, 512, 768\}$. When $Q = 0$, the reward degenerates to batch-level ranking only, which is less stable, and performance drops substantially. Introducing a bank already brings clear gains at $Q = 64$, and performance continues to improve as Q increases, before largely saturating around $Q = 512$. Increasing the bank size further to 768 yields only marginal change, suggesting that $Q = 512$ already provides a sufficiently rich and stable ranking reference, with a potential risk of redundancy or staleness in larger banks.

For the bank inclusion criterion, we compare the adopted "highest-only" bank update with including all candidates. The latter consistently degrades performance, likely because low-quality candidates contaminate the bank distribution and make the pairwise ranking signal noisier.

For the number of video frames, we try using 8, 16, 32, and 48 uniformly sampled frames, and performance improves substantially from 8 to 32 frames, showing that denser temporal coverage is important for capturing critical moments in fast-paced diving sequences, while the gain from 32 to 48 frames is only marginal, suggesting that 32 frames already preserve most key temporal cues and provide a favorable trade-off between performance and efficiency, given that videos in DivingThink contain 104 frames on average.

Table 9: The list of action types and their corresponding decomposed sub-action sequences involved in the DivingThink dataset.

Action Type	Sub-action Type			Action Type	Sub-action Type				
	Take-off	Flight	Entry		Take-off	Flight			Entry
101B	Forward	0.5 Som.Pike		612B	Arm.Fwd	1 Som.Pike			
103B	Forward	1.5 Soms.Pike		614B	Arm.Fwd	2 Soms.Pike			
105B	Forward	2.5 Soms.Pike		626B	Arm.Back	3 Soms.Pike			Entry
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		
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	Entry	5331D	Reverse	0.5 Twist	1.5 Soms.Pike		Entry
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	
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	Entry
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	

I Computational Overhead

We analyze the computational overhead of Diving-R1 during both training and inference. All three training stages are conducted on four NVIDIA RTX A6000 (48 GB) GPUs. Specifically, Stage I requires 12.26 hours with a peak per-GPU memory usage of 44.08 GB, while Stage II takes 18.52 hours with 47.79 GB, and Stage III takes 19.13 hours with 47.98 GB. During inference, Diving-R1 operates on a single NVIDIA RTX A6000 GPU, requiring an average of 1.679 seconds per video and a peak GPU memory usage of 10.49 GB, indicating that Diving-R1 can complete inference within a few seconds per video with a relatively modest memory footprint and demonstrating its potential for practical deployment.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to the Abstract and the Introduction (Section 1).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations and future directions in the Conclusion and Future Work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not involve theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer to the Methodology and Implementation Details in Sections 3 and 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide sufficient instructions for reproduction in Section 4.1. The dataset and code will be released on GitHub, as mentioned in the abstract.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to Sections 3.1, 3.2, 3.3, and 4.1 for details on the data, benchmark, training/testing process, and hyperparameter settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following previous studies, the statistical significance of the experiments is typically not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Section 4.1, which describes that all experiments in the paper were conducted on four NVIDIA RTX A6000 48GB GPUs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics and confirm that the research described in this paper complies with the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to Section 1, which illustrates the potential positive societal impacts of the research presented in this paper, such as assisting diving training and improving the transparency and consistency of scoring.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper does not introduce any assets that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the resources used in this paper are publicly available, with sources clearly cited, and all relevant license agreements and terms of use strictly followed.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide detailed descriptions of the newly introduced assets in Sections 3.1, 3.2, and 4.1, and, as stated in the Abstract, these assets will be made publicly available on GitHub.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or human-subject studies. The expert review in dataset construction is used only for technical annotation and quality control, and does not collect personal, behavioral, or sensitive information from human participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or human-subject studies; therefore, IRB approval or equivalent review is not applicable.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Please refer to Sections 3.1, 3.2, and 3.3, where the usage of LLMs/MLLMs in dataset construction, benchmark evaluation, and model training is described.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.