REVISOR: Beyond Textual Reflection, Towards Multimodal Introspective Reasoning in Long-Form Video Understanding

Supplementary Material

7. Related Work

Long-Form Video Understanding. In long-form video understanding, the visual inputs are far more complex than those in image-based tasks. Consequently, identifying and distilling only the question-relevant information from large amounts of redundant visual content becomes crucial for improving model performance. Existing approaches can be grouped into three categories: external-model augmentation, agentic methods, and model-internalized selection.

External-model augmentation techniques [21, 29, 37], rely on vision–text similarity models like CLIP for keyframe selection. While efficient, these methods perform static, one-shot selection, limiting their adaptability to complex or evolving queries. Agentic approaches [24, 46] address this limitation through iterative frameworks that dynamically refine frame selection. For example, VideoAgent [44] employs an LLM as a central agent in a state–action–observation loop, whereas VideoTree [47] adopts a hierarchical tree structure to perform coarse-to-fine search, directing computation toward the most relevant segments. More recent model-internalized selection methods [40] embed frame selection directly into the model's reasoning process. GenS [56] trains a lightweight generative sampler to identify query-relevant frames end to end, though the selection and QA stages remain decoupled.

Our proposed REVISOR framework, after reinforcement learning training, can autonomously identify and explore visual content that requires additional careful review during the inference stage. Unlike previous methods that are limited to statically searching for important video segments based solely on the question itself, our approach fully integrates with the reasoning capabilities of MLLMs. This allows it to engage in deeper deliberation and, through comprehensive interaction with its ongoing reasoning, more precisely pinpoint critical visual information, thereby significantly enhancing the model's reasoning ability.

Self-Reflection Mechanism. Incorporating self-reflection into inference has been shown to improve model performance [12]. Self-Refine [28] implements an iterative feedback–revision loop without requiring additional training or external supervision. SCoRe [16] adopts multi-round reinforcement learning to cultivate reflective reasoning skills, yielding substantial gains. GEPA [1] further introduces a Pareto-based reflective prompt-evolution framework that outperforms reinforcement learning in both sample efficiency and overall performance.

Extending self-reflection to multimodal settings likewise yields consistent improvements [15, 54]. VL-Rethinker [42] incorporates a self-verification–and-correction stage via reinforcement learning, substantially enhancing multimodal reasoning. SRPO [41] further strengthens self-reflection and self-correction by curating reflection-oriented supervised data and introducing a reflection-aware reward under group relative policy optimization, significantly boosting complex multimodal reasoning.

However, existing multimodal self-reflection mechanisms still depend on text-based re-evaluation processes. They lack explicit reflective operations over visual information and therefore remain fundamentally text-centric. The REVISOR framework proposed in this paper addresses this limitation by extending text-centric reflection into a truly multimodal reflective process.

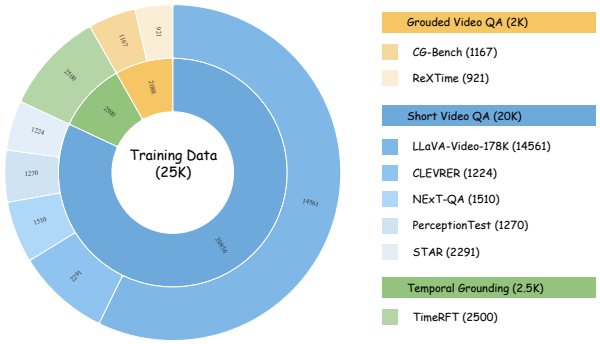

Figure 6. REVISOR framework training dataset composition. The training dataset for the REVISOR framework consists of three tasks: Short Video QA, Temporal Grounding, and Grounded Video QA, totaling 25K training samples. The parenthetical value for each dataset denotes its specific sample contribution.

8. More Experimental Details on REVISOR

In Sec. 8.1, we present the detailed experimental setup for experiments involving the REVISOR framework. Sec. 8.2 outlines the composition of the training data for the REVISOR framework. Sec. 8.3 then presents supplementary experimental results, including comprehensive results on the Temporal Video Grounding task and complete ablation studies on the reward scaling factors λ_1 and λ_2.

8.1. Detailed Experimental Setup

Our experiments utilized Qwen2.5-VL-7B as the base model, which comprises a visual encoder, a merger projec-

Table 5. Comprehensive evaluation of REVISOR on the temporal video grounding task. **Bold** text indicates the best performance.

	Charades-STA				NExT-GQA			
Model	**R@0.3**	**R@0.5**	**R@0.7**	**mIoU**	**R@0.3**	**R@0.5**	**R@0.7**	**mIoU**
Qwen2.5-VL-7B[2]	57.1	33.6	15.5	36.9	31.6	18.1	7.5	20.9
VTimeLLM[13]	55.3	34.3	14.7	34.6	37.9	20.2	9.7	24.4
iMOVE[18]	71.7	51.3	26.1	47.3	-	-	-	-
TimeChat[33]	51.5	32.2	13.4	-	34.1	17.9	6.2	20.6
VideoChat-TPO[53]	58.3	40.2	18.4	38.2	41.2	23.4	8.2	27.7
TVG-R1[6]	70.8	50.5	23.9	46.7	41.7	20.8	10.0	29.3
Ours	**76.5**	**57.3**	**31.8**	**51.4**	**47.6**	**25.5**	**11.9**	**33.2**

Table 6. Complete ablation study of the Dual Attribution Decoupled Reward mechanism. **Bold** fonts highlight the best performance. The row marked in gray represents our base model.

$\lambda_1(R_{final})$	$\lambda_2(R_{causal})$	VideoMME	LongVideoBench	LVBench	MLVU	NExT-GQA
-	-	64.3	56.5	40.2	67.3	20.87
0.3	0.6	64.0	56.0	41.1	68.7	33.9
0.6	0.0	62.2	54.0	40.8	68.3	32.1
0.45	0.45	64.3	57.3	41.2	68.5	33.7
0.5	0.4	64.9	57.1	41.8	69.0	33.4
0.6	0.3	**65.7**	**57.5**	**42.0**	**69.8**	**33.2**
0.8	0.1	64.6	57.6	40.3	68.4	31.6

tor, and a large language model. REVISOR was trained using a single stage of reinforcement learning, eliminating the need for an additional supervised fine-tuning phase. We extended the verl framework to support REVISOR's training. Following the approach in DAPO, we removed the KL regularization term from GRPO.

Key hyperparameters were set as follows: $\lambda_1 = 0.6$, $\lambda_2 = 0.3$, a learning rate of 1×10^{-6}, a batch size of 32, and 8 rollouts. During both training and evaluation, input video tokens were limited to a maximum of 8192, sampled at 1 FPS. For the Review Segment, the sampling rate was increased to 2 FPS, while still adhering to a maximum of 8192 video tokens. Absolute timestamps were displayed in the lower-left corner of each image frame. The entire training phase consisted of 792 optimization steps.

8.2. Detailed Information of Training Data

As detailed in Fig. 6, our training corpus spans three tasks central to video understanding: Short Video QA, Temporal Grounding, and Grounded Video QA. For Short Video QA, we aggregate data from LLaVA-Video-178K, CLEVRER, NExT-QA, PerceptionTest, and STAR, all of which provide short video clips ($\leq$ a few minutes). From these datasets, we sample a balanced set of 20K video–question–answer triplets. The Temporal Grounding task is sourced directly from TimeRFT without further filtering. For Grounded Video QA, we incorporate CG-Bench and ReXTime, from which we randomly select 2K examples.

8.3. Supplementary Experimental Results

Temporal Video Grounding. As summarized in Tab. 5, we conduct an extensive assessment of REVISOR's temporal grounding performance across both the Charades-STA and NExT-GQA benchmarks. On Charades-STA, REVISOR attains an R@0.3 accuracy of 76.5%, surpassing the prior state-of-the-art method iMOVE by a margin of 4.8%. On NExT-GQA, REVISOR reaches an R@0.5 accuracy of 25.5%, outperforming TVG-R1, designed explicitly with reinforcement learning to enhance temporal localization, by 4.3%. Together, these results highlight the robustness and effectiveness of REVISOR, equipped with DADR, in achieving precise temporal video localization.

Ablation Study of DADR Mechanism. We conducted an in-depth analysis of the Dual Attribution Decoupled Reward (DADR) mechanism. As shown in Tab. 6, when $\lambda_2 = 0$, that is, when only the Final Answer Verification Reward is applied, the performance of REVISOR on Video-MME decreases substantially from 65.7% to 62.2%, even falling below the base model. This demonstrates that, without the Causal Segment Sufficiency Reward (CSSR), the model struggles to reliably identify the correct review segment S due to the sparsity of the reward signal.

When λ_2 exceeds λ_1, the model's temporal grounding improves; however, its long-video reasoning capability de-

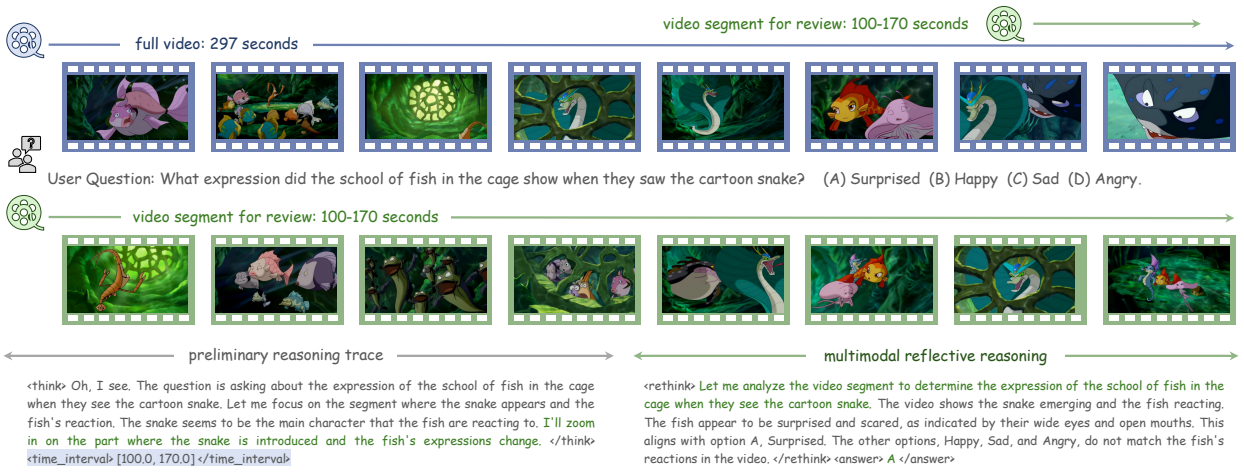

Figure 7. Successful example of the REVISOR framework: achieving more accurate detail capture. By reviewing key video segments, the REVISOR framework accurately identified that the fish in the cage exhibited surprise when they saw the cartoon snake. Without visual reflection, the model mistakenly believed the fish's emotion was anger.

grades. For example, MLVU performance declines from 69.8% to 68.7%, suggesting that the model overemphasizes locating the review segment S while underutilizing it for answer derivation. Consequently, we set $\lambda_1 > \lambda_2$ to encourage accurate localization of S while still promoting strong reasoning based on it.

Nonetheless, if λ_1 is too large and λ_2 too small, the model's ability to locate S deteriorates, harming long-form video understanding. Conversely, when λ_1 is only slightly greater than λ_2, REVISOR identifies S effectively but fails to fully leverage it for reasoning, resulting in strong temporal grounding but weaker long-video comprehension. Empirically, we find that $\lambda_1 = 0.6$ and $\lambda_2 = 0.3$ provide an effective balance, enabling REVISOR to both accurately localize S and utilize it to enhance reasoning performance.

9. Training Qwen2.5-VL-7B with Textual Reflection Mechanism on Video Data

To ensure a fair comparison, we train a Qwen2.5-VL-7B model equipped with a text-based self-reflection mechanism using the datasets listed in Fig. 6. Specifically, unlike REVISOR, the text-reflection model generates only textual output during the reflection phase. Apart from this distinction, all other experimental settings remain identical to those of REVISOR. The learning rate is set to 1×10^{-6}, the total batch size is 32, and the number of rollouts is set to 8. In addition, the total number of input video tokens is capped at 8192. The prompt template is shown in Fig. 11.

10. Case Study of REVISOR Framework

Improved Video Reasoning Capability. The REVISOR framework can significantly enhance the long-form video understanding capabilities of MLLMs. Fig. 7, Fig. 8, and Fig. 9 respectively demonstrate these improvements from three perspectives: more precise detail capture, more comprehensive scene understanding, and more accurate object counting.

Visual Reflection is Even More Important. In this paper, we repeatedly emphasize that, for long-video understanding tasks, visual reflection is more important than textual reflection. In Sec. 5.1, we quantitatively validate this conclusion by monitoring changes in the length of textual reflections during training. Here, we further illustrate this point with a concrete example. As shown in Fig. 10, during training, the textual reflection generated by the REVISOR framework for a given question becomes increasingly concise, while its retrieval of key video segments becomes increasingly accurate. Ultimately, the model precisely identifies the critical 25-second segment within a 300-second video, confirming that respiratory cells could potentially be used in treatment for cardiovascular disease. This phenomenon is not an isolated incident; similar observations were made across virtually all samples.

11. Prompt Templates of REVISOR

The complete prompt template used during the training of the REVISOR framework consists of three primary components: the system prompt, the initial reasoning stage, and the reflective reasoning stage. Fig. 12 illustrates the templates for both the initial reasoning and reflective reasoning stages, while Fig. 13 presents the system prompt template. A key distinction from the plain-text reflection mechanism is that, during the initial reasoning stage, the REVISOR framework identifies the critical video segments that require further examination.

Figure 8. Successful example of the REVISOR framework: achieving more accurate scene understanding. By reviewing key video segments, the REVISOR framework accurately determined that the man was standing on the cliff edge to avoid an oncoming truck. Without this visual reflection, the model erroneously concluded that he was merely taking photos on the cliff.

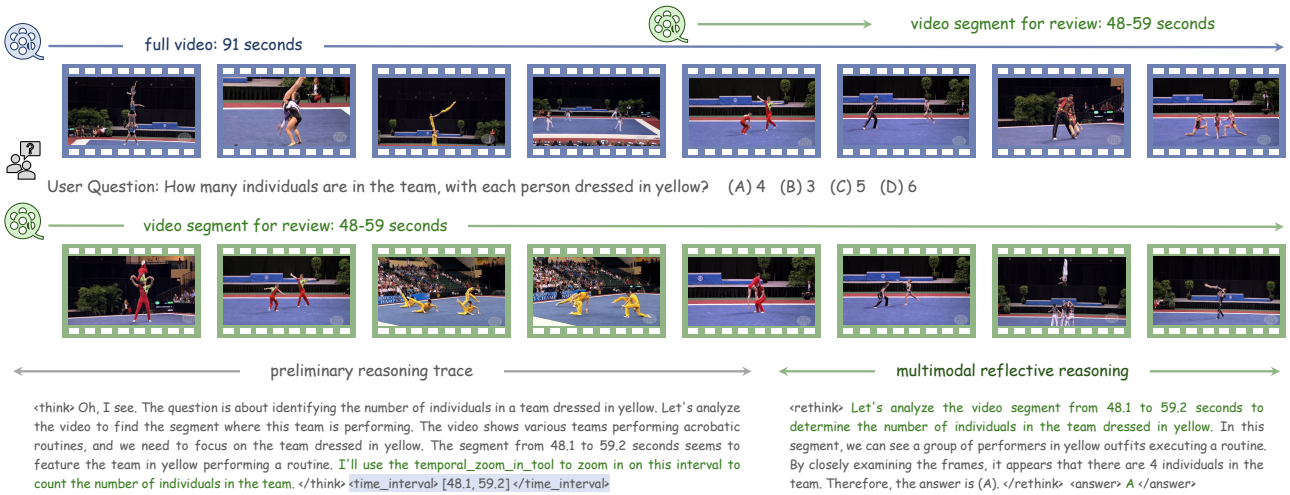

Figure 9. Successful example of the REVISOR framework: achieving more accurate object counting. By reviewing key video segments, the REVISOR framework accurately identified four team members wearing yellow. Without this visual reflection, the model erroneously counted only three.

Figure 10. As training progressed, REVISOR's outputted reflective text paragraphs became increasingly concise, and the video segments it referenced grew more precise. Ultimately, it accurately concluded that respiratory cells could potentially be used in the treatment of cardiovascular disease.

The Complete Prompt Template for Plain-Text Reflection Mechanism

System Prompt:

You are a helpful assistant.

Stage 1: Preliminary Reasoning Phase

User Instruction:

<User Question>. Please think carefully and then provide your answer.

Output Format:

Format strictly as: <think> Your reasoning steps </think> <answer> Your answer </answer>

Stage 2: Reflective Reasoning Phase

User Instruction:

Please rethink the reasoning process above and provide your final answer.

Output Format:

Format strictly as: <rethink> Your rethinking reasoning steps </rethink> <answer> Your final answer </answer>

Figure 11. The complete prompt template used during the training of Qwen2.5-VL-7B equipped with a pure text-based reflection mechanism, covering both the initial reasoning and reflective reasoning stages.

Reasoning Prompt Template for REVISOR Framework Training

Stage 1: Preliminary Reasoning Phase

User Instruction:

<User Question>. Please think first, and then use the temporal_zoom_in_tool to find the video segment that can answer the user's question.

Output Format:

Format strictly as: <think> Your reasoning steps </think> <time_interval>[start_time, end_time] </time_interval>

Stage 2: Reflective Reasoning Phase

User Instruction:

Please refer to the Visual Review segment above, think carefully, and provide your final answer.

Output Format:

Format strictly as: <rethink> Your reasoning steps </rethink> <answer> Your final answer </answer>

Figure 12. Reasoning prompt template for REVISOR framework training

System Prompt Template for REVISOR Framework Training

You are a helpful assistant.

 # Tools

You may call one or more functions to assist with the user query. You are provided with function signatures within <tools></tools> XML tags:

<tools>

```
{
  "type": "function",
  "function": {
    "name": "temporal_zoom_in_tool",
    "description": "Identify the precise time segment in the video that contains
        enough information to answer the question.",
    "parameters": {
      "properties": {
        "interval": {
          "type": "str",
          "description": "The time range to zoom in on, formatted as 'start_time to
              end_time'. Timestamps are in seconds."
        }
      },
      "required": ["interval"]
    }
  }
}
```

</tools>

 # Explanation of "temporal zoom-in"

When you call temporal_zoom_in_tool with <time_interval> [start_time, end_time] </time_interval>, the tool returns a new sequence of denser video frames sampled from the specified time range ([start_time, end_time]) in the original video. This provides higher temporal precision, helping you answer the user's question more accurately.

 # How to call a tool

Return the interval directly in XML tags: <time_interval> [12.3, 28.7] </time_interval>

Figure 13. System prompt template for training the REVISOR framework.