

# ROVER: Reasoning Over Video with Efficient Retrieval

Anonymous ACL submission

## Abstract

Answering questions in natural language based on a given video is commonly referred to as the VideoQA task. When videos are long and questions are complex, multi-step reasoning is often required to integrate visual evidence distributed throughout the video. Efficiently sampling relevant visual evidence from long videos under limited computational budgets remains a key challenge. In this paper, we use *visual tokens* as a measure of sampling cost and propose **ROVER (Reasoning Over Video with Efficient Retrieval)**. ROVER is a tool-augmented framework that first retrieves low-resolution frames containing fewer visual tokens to locate relevant events, and then selectively zooms in by retrieving high-resolution frames with richer visual details. ROVER is trained using a SFT-then-RL recipe, enabling dynamic coordination of low- and high-resolution frame retrieval under a question-dependent visual-token budget. ROVER achieves state-of-the-art performance on 3 out of 4 video reasoning benchmarks, while remaining competitive on four general VideoQA benchmarks. Extensive experiments also empirically show a strong accuracy–efficiency balance.

## 1 Introduction

Reasoning over long videos is a form of VideoQA that requires models to identify sparse yet critical visual evidence distributed across extended videos. This task is constrained by both context length and computational cost. A common approach *uniformly* samples video frames to construct the visual context; however, this strategy (Feng et al., 2025; Li et al., 2025; Chen et al., 2025a,b) exposes a fundamental trade-off: encoding dense frames quickly exhausts token budget, while sparse sampling often misses evidence required for accurate reasoning.

To address this challenge, tool-augmented, agentic pipelines—such as iterative video search and zoom-in—have recently emerged as a promising

alternative. These tool-augmented video agents (TAVAs; He et al. (2025); Ouyang et al. (2025); Zhang et al. (2025a); Fu et al. (2025b)) enable models to acquire visual observations over multiple turns rather than processing the entire video in a single pass. As illustrated in Figure 1, a typical TAVA begins with a fixed number of uniformly sampled video frames, where the number of initial frames is empirically determined based on video length. In subsequent turns, the agent selectively retrieves additional frames from specific temporal segments to refine its understanding.

Despite this flexibility, existing TAVAs remain limited in both efficiency and adaptability. First, they incur a relatively large initial visual context by ingesting a substantial portion of the video before any question-specific reasoning begins, even for simple queries such as “*Who is the man at 05:43?*”. Second, they enforce a hard, fixed budget on subsequent tool interactions, typically in the form of a predefined maximum number of tool calls. Such a fixed interaction budget restricts iterative skimming, localization, and refinement, even when broader exploration is required, as in questions like “*What sequence of evidence leads to the conflict/problem that occurs near the end?*”.

These limitations stem from how modern vision–language models (VLMs; Liu et al. (2023); Bai et al. (2025); Wang et al. (2025b)) consume computation. Specifically, each video frame is partitioned into patches and encoded by a vision backbone into *visual tokens*. The number of visual tokens, together with text tokens, typically determines the computational cost. A large initial visual context therefore leads a high number of tokens consumed during computation, leaving limited capacity for subsequent interactions. As a result, imposing a hard cap on the number of interactions becomes a natural, but restrictive, engineering choice.

Motivated by these observations, we aim to build

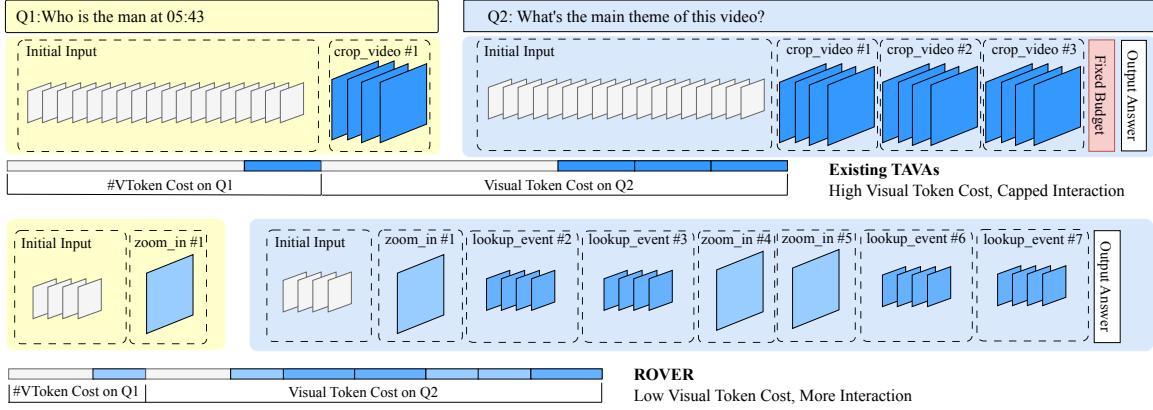


Figure 1: Budget mismatch of visual tokens in existing agentic long-video pipelines. Top: fixed-budget pipelines use a large question-agnostic initial visual context and a hard cap on tool calls, causing overhead on easy queries and rigidity on hard ones. Bottom: ROVER uses minimal initial inputs and a soft, adaptive budget, interleaving low-cost LOOKUP\_EVENT with targeted high-resolution ZOOM\_IN.

an agent that (i) starts from a minimal initial visual context and relies on active exploration to acquire evidence, and (ii) avoids hard-capping exploration by softly regulating evidence acquisition through reward design. A key enabler is a unified visual interface that supports both (i) inexpensive, low-resolution temporal skimming for broad coverage and (ii) targeted high-resolution zoom-in for fine-grained visual details. This unification allows the agent to interleave inexpensive skim actions with selective zoom actions across multiple turns, adaptively allocating visual tokens based on question requirements (Figure 1, bottom).

To this end, we propose the **Reasoning Over Video with Efficient Retrieval (ROVER)**, a tool-augmented video agent under the philosophy of “*explore it yourself*.” ROVER learns to reason by spending *visual-token* budget efficiently for evidence acquisition via a three-stage training pipeline. (1) **Cold-start SFT** initializes tool usage via supervised learning on synthesized tool-calling trajectories. (2) **Tool-Augmented RL** stabilizes tool usage and encourages proactive evidence gathering, shaping a high-recall agent that answers conservatively only after sufficient evidence is collected. (3) **Budget-Efficient RL** prunes unnecessary evidence acquisition without sacrificing accuracy by optimizing a question-dependent budget objective. Concretely, we maintain a per-question buffer that tracks the average visual tokens used by previously successful trajectories, termed the *visual-token baseline*. The policy is then encouraged to solve each question using fewer visual tokens *relative to this baseline*, rather than under a global

penalty, allowing easy questions to consume less budget while preserving capacity for harder ones.

**Contributions.** We make three contributions:

- We formulate long-video reasoning as a budgeted evidence acquisition problem and advocate visual tokens as a unified measure of sampling cost.
- We develop a unified visual interface that supports interleaved interaction for low-resolution temporal skimming and high-resolution zoom-in.
- We design a stable post-training recipe with a question-wise visual-token baseline. We further introduce gated budget shaping to control evidence acquisition. Together, these techniques yield improved accuracy–efficiency trade-offs across diverse long-video reasoning benchmarks, as demonstrated through extensive ablations.

## 2 Related Work

### 2.1 RL-based Multimodal Reasoning in Video

Modern VLMs (Bai et al., 2025; Liu et al., 2023; Wang et al., 2025b) motivate extensions to video (Lin et al., 2024; Cheng et al., 2024; Leng et al., 2025), while RL post-training strengthens multi-step reasoning (DeepSeek-AI, 2025; Shao et al., 2024; Liu et al., 2025; Yuan et al., 2025). In video, reasoning-oriented post-training also improves long-video QA and temporal reasoning, e.g., Video-R1 (Feng et al., 2025), VideoChat-R1 (Li et al., 2025), and LongVILA-R1 (Chen et al., 2025a). However, most approaches assume a *pre-specified* observation budget (fixed sampled frames/clips), facing a coverage–fidelity trade-off in long videos: dense high-resolution inputs ex-

haust the context, while sparse sampling may miss crucial evidence.

## 2.2 Tool-augmented Agentic VLMs

Tool-augmented agentic VLMs replace passive uniform sampling with iterative evidence gathering, often following the ReAct/tool-use paradigm (Yao et al., 2023; Schick et al., 2023; Yang et al., 2023). In long-video understanding, a representative line learns adaptive temporal evidence acquisition policies via multi-turn interaction, including FrameThinker for frame spotlighting, FrameMind (Ge et al., 2025) with frame-interleaved RL reasoning, and LongVT for native tool calling in long videos (He et al., 2025; Yang et al., 2025). Complementary to temporal selection, multi-scale evidence tools enable targeted high-resolution inspection when fine-grained recognition is required, exemplified by LOVE-R1’s adaptive zoom-in and Conan’s progressive multi-scale reasoning over accumulated clues (Fu et al., 2025b; Ouyang et al., 2025). ReWatch-R1 improves complex video reasoning through agentic data synthesis, producing trajectories that better align intermediate reasoning with retrieved evidence (Zhang et al., 2025a). A closely related direction combines tools with RL for video QA and grounding; e.g., VITAL still largely budgets in frames (Zhang et al., 2025b).

Overall, prior methods often constrain evidence acquisition via fixed frame budgets or explicit turn/context caps (Zhang et al., 2025a; Yang et al., 2025; Ge et al., 2025), making it hard to adapt budget across easy versus hard questions under patch-token compute.

## 3 ROVER Framework

Given a video  $V$  and a question  $q$ , we consider a token-budgeted reasoning process where an agent iteratively acquires visual evidence from  $V$  before producing an answer to  $q$ . Specifically, at step  $k$ , conditioned on the interaction history  $h_k = (q, t_{1:k-1}, u_{1:k-1}, o_{1:k-1})$ , the policy  $\pi_\theta$  outputs an internal thinking process  $t_k$  and an acquisition action  $u_k \in \mathcal{U}$ . Executing  $u_k$  returns an observation  $o_k$ . The overall trajectory can be represented as:

$$\tau = (q, (t_1, u_1, o_1), \dots, (t_K, u_K, o_K)), \quad (1)$$

where  $K$  is the number of tool-calling steps before termination, and a valid trajectory must end with an  $u_K$  of OUTPUT\_ANSWER. The final answer  $a$  is parsed from the argument of  $u_K$ .

**Available Tools.** We consider **four** tools in our framework, which would be candidate for each  $u_K$ :

- **LOOKUP\_EVENT**: a *low-resolution temporal sampler* that retrieves a set of frames, each termed as  $I$ , covering a broad temporal span at low resolution, providing inexpensive global context. Specifically, given a time window, it returns the observation  $o_k = \{I_{i_j}^{(r_l)}\}_{j=1}^n$ , i.e.,  $n$  uniformly sampled frames downsampled to resolution  $r_l$ . The visual-token evidence is  $e_k = \text{Enc}(o_k)$ , and the induced cost is  $\text{cost}(e_k) = |e_k|$ .
- **ZOOM\_IN**: a *high-resolution zoom-in* tool that selectively fetches a specific frame. Given a frame index  $i$ , it returns the observation  $o_k = I_i^{(r_h)}$  at resolution  $r_h$ , with  $e_k = \text{Enc}(o_k)$  and  $\text{cost}(e_k) = |e_k|$ .
- **GET\_FRAME\_INDEX** computes the frame index given a timestamp; it returns a short text response (e.g., “the frame index of 05:30 is 420”). We set  $e_k = o_k$  and incur zero visual-token cost, i.e.,  $\text{cost}(e_k) = 0$ .
- **OUTPUT\_ANSWER** terminates evidence acquisition and outputs the final answer with citations; it returns no observation (we set  $e_k = \emptyset$ ) and incurs zero visual-token cost, i.e.,  $\text{cost}(e_k) = 0$ .

For patch-based visual encoders,  $\text{cost}(e_k)$  can be written explicitly as a function of resolution: if an  $r \times r$  frame is tokenized into non-overlapping  $p \times p$  patches, then a single frame yields  $L(r) = (r/p)^2$  visual tokens (up to rounding). Therefore, LOOKUP\_EVENT incurs approximately  $n \cdot L(r_l)$  tokens and ZOOM\_IN incurs  $L(r_h)$  tokens, while GET\_FRAME\_INDEX has  $\text{cost}(e_k) = 0$ . These tools jointly enable mixed-granularity visual interface for evidence acquisition and precise indexing. We define the visual-token cost of a trajectory as:

$$C(\tau) = \sum_{k=1}^K \text{cost}(e_k). \quad (2)$$

## 4 Training Strategy

In this section, we describe the training strategy for instantiating the ROVER framework (Figure 2). We first conduct *Cold-start SFT* to initiate the tool-calling behavior of the base model, followed by *Tool-Augmented RL* to consolidate tool-augmented evidence acquisition behavior, and finally introduce verification-gated and budget-disciplined rewards into *Budget-Efficient RL*.

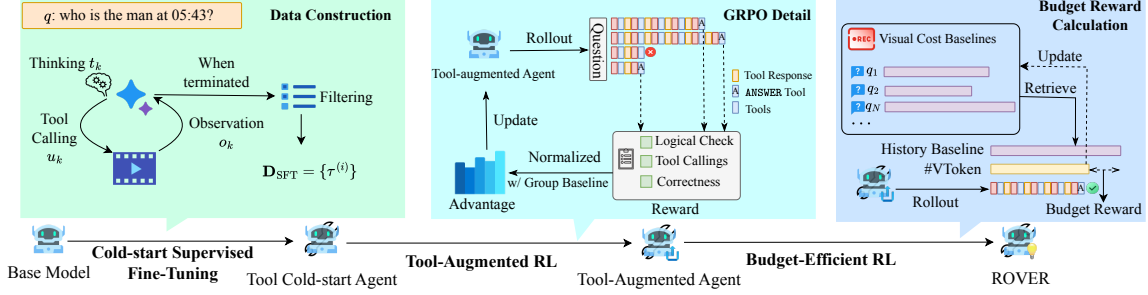


Figure 2: Training pipeline of ROVER. We train with a three-stage recipe: Cold-start SFT  $\rightarrow$  Tool-Augmented RL  $\rightarrow$  Budget-Efficient RL, progressively stabilizing tool use and encouraging accuracy-preserving efficiency via a question-wise visual-token baseline. Invalid tool calls terminate the rollout with zero reward.

#### 4.1 Cold-start Supervised Fine-Tuning

A key challenge in adapting a base VLM to the ROVER framework lies in equipping it with robust multi-step, tool-augmented reasoning capabilities. Though the base model can perform multi-step reasoning to some extent, this behavior may degrade during RL without sufficient supervised fine-tuning on tool-augmented data. Therefore, it is crucial to construct high-quality supervised fine-tuning data that covers diverse evidence acquisition strategies and reasoning paths. We empirically show that Cold-start SFT is a crucial prerequisite for effective reinforcement learning (Section 5).

**Cold-start Data Curation.** We deploy a high-capacity teacher VLM to generate multi-step reasoning trajectories under our tool-augmented agentic framework described in Section 3. We synthesize tool-calling trajectories on a diverse mixture of VideoQA and video reasoning datasets (using only question-answer pairs), and then filter and balance them to obtain high-quality SFT data. Appendix A.2.1 provides dataset, synthesis, and filtering details.

**Filtering Strategy.** To ensure the data quality, we apply the following filtering criteria:

1. *Format Check:* We retain only those trajectories that strictly adhere to the required formatting specifications, including valid thinking blocks and tool-call formats.
2. *Logical Continuity:* Each GET\_FRAME\_INDEX call must be immediately followed by a corresponding ZOOM\_IN or LOOKUP\_EVENT call centered on the retrieved frame index.
3. *Answer Correctness:* Only trajectories that produce a correct final answer are retained.
4. *Tool Calling Deduplication:* We remove sam-

ples with identical tool invocations, such as repeated event lookups or zoom-in with the same parameters, within the same trajectory to avoid redundant steps.

After filtering, we construct a high-quality cold-start dataset comprising diverse and valid evidence-acquisition trajectories. We then fine-tune the base VLM on this dataset to instill the desired tool-augmented reasoning behavior. Detailed implementation settings are provided in Appendix A.1.

#### 4.2 Tool-Augmented RL

After the cold-start stage, the model can adapt to the ROVER framework and produce valid evidence-acquisition trajectories. Nevertheless, its rollout behavior remains unstable: small deviations from the supervised data can result in invalid tool calls or malformed outputs. Therefore, we further consolidate the model’s behavior through Tool-Augmented RL. We construct the RL training set by filtering the cold-start data, as detailed in Appendix A.2.2. At this stage, we employ Group Relative Policy Optimization (GRPO; Shao et al. (2024)) to optimize the policy. Following the standard GRPO practice, we disable the KL divergence term to promote exploration and maintain high learning efficiency. The objective is defined as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\tau_i \sim \pi_{\theta, \text{old}}} \left[ \frac{1}{G} \sum_{i=1}^G \min(r_i A_i, A_{i, \text{clip}}) \right],$$

$$A_{i, \text{clip}} = \text{clip}(r_i, 1 - \epsilon, 1 + \epsilon) A_i.$$

where  $r_i = \frac{\pi_{\theta}(\tau_i | q)}{\pi_{\theta, \text{old}}(\tau_i | q)}$ ,

$$A_i = \frac{R_i - \text{mean}(\{R_1, \dots, R_G\})}{\text{std}(\{R_1, \dots, R_G\}) + \delta}.$$

We employ a sophisticated reward design that encourages valid tool-calling behavior and accu-

rate question answering. Specifically, the reward consists of three components:

**Format Reward**  $R_{\text{format}}$ . A binary reward indicating whether the trajectory follows the correct format, specifically the presence of a valid thinking block, a tool-call format, and an evidence block.

**Accuracy Reward**  $R_{\text{accuracy}}$ . A binary reward that reflects the correctness of the final answer; the reward is set to 0 if the answer is not parsable.

**Tool Reward**  $R_{\text{tool}}$ . A shaping reward that encourages evidence acquisition through diverse tool utilization, comprising a *discounted tool-calling term* and a *tool-diversity term*.

The tool-calling term is defined as:

$$R_{\text{tool-call}} = \sum_{k=1}^{K-1} \gamma^k \cdot \mathbb{I}(u_k \in \{\text{LE}, \text{ZI}\}), \quad (3)$$

where  $k$  indexes the tool-calling step in Equation (1),  $\gamma \in (0, 1)$  is a discount factor, and LE and ZI denote LOOKUP\_EVENT and ZOOM\_IN, respectively. We deliberately design this bounded reward to encourage the model to acquire sufficient evidence but avoid excessive tool calls. Equation (3) corresponds to a *decay* tool reward; in our ablations, we additionally consider (i) a *binary* variant  $R_{\text{tool-call}}^{\text{bin}} = \mathbb{I}(\sum_{k=1}^{K-1} \mathbb{I}(u_k \in \{\text{LE}, \text{ZI}\}) > 0)$ , and (ii) a *linear-proportional* variant  $R_{\text{tool-call}}^{\text{lin}} = \sum_{k=1}^{K-1} \mathbb{I}(u_k \in \{\text{LE}, \text{ZI}\})$ . These functional forms are analogous to the binary and token-delta (linear) shaping choices used in budget rewards (Equation (11)).

The tool-diversity term is defined as the number of distinct tools used during evidence acquisition:

$$R_{\text{tool-div}} = \left| \{u_k\}_{k=1}^{K-1} \right|, \quad (4)$$

where we exclude the terminal OUTPUT\_ANSWER action, since it is always required for a valid trajectory.

We then combine them into a single tool reward:

$$R_{\text{tool}} = \mu_{\text{tool}} R_{\text{tool-call}} + \lambda_{\text{tool-div}} R_{\text{tool-div}}, \quad (5)$$

where  $\mu_{\text{tool}}$  and  $\lambda_{\text{tool-div}}$  are set to balance the tool-calling and tool-diversity terms, respectively.

The final reward at this stage is defined as:

$$R_{\text{total}} = R_{\text{accuracy}} \cdot (R_{\text{format}} + R_{\text{tool}}). \quad (6)$$

All rewards are gated by answer correctness. Because  $R_{\text{accuracy}} = 0$  for incorrect or unparseable

answers, trajectories receive no reward unless they ultimately produce a correct response, regardless of the number or diversity of tool calls. Consequently,  $R_{\text{tool}}$  functions as an accuracy-gated shaping signal: it stabilizes tool use and promotes proactive evidence acquisition while preventing reward hacking on incorrect rollouts.

### 4.3 Budget-Efficient RL

We now describe the final stage, Budget-Efficient RL, which softly regulates visual evidence acquisition without imposing a hard budget. While the earlier stages establish basic tool use and multi-turn interaction behaviors, optimizing task reward alone tends to promote conservative *over-acquisition*: the model can safely improve success by collecting additional, often redundant, observations. Introducing a global cost penalty offers only a coarse solution and risks prematurely limiting exploration for queries that intrinsically require richer evidence.

Accordingly, we propose a question-adaptive, budget-efficient shaping reward that promotes efficiency without sacrificing correctness, while preserving flexibility for more challenging questions.

**Question-wise visual-token baseline.** Let  $C(\tau)$  denote the visual-token cost of a trajectory, as defined in Equation (2). In GRPO, each question  $i$  is associated with a group of rollouts  $\{\tau_g\}_{g=1}^G$ . We maintain a question-wise visual-token baseline  $b_i$ , intended to capture the typical cost of *successful* solutions for question  $i$ . To offer a better estimate of sample-wise visual token costs, we compute the mean cost over correct trajectories within the group. Let  $\mathcal{G}_i^+ = \{g \mid \text{correct}(\tau_g)\}$  and define

$$\bar{C}_i^+ = \frac{1}{|\mathcal{G}_i^+|} \sum_{g \in \mathcal{G}_i^+} C(\tau_g). \quad (7)$$

We use  $b_i$  from the previous iteration to compute the budget-efficient rewards for the current group, and then update  $b_i$  using an EMA:

$$b_i \leftarrow (1 - \alpha) b_i + \alpha \bar{C}_i^+, \quad (8)$$

when  $|\mathcal{G}_i^+| > 0$ , leaving  $b_i$  unchanged otherwise. This baseline naturally adapts to intrinsic difficulty: questions that consistently require more evidence converge to higher baselines.

**Gated budget-efficient reward.** We incorporate budget awareness as a *positive* shaping signal rather

than an explicit penalty. For each rollout  $\tau_g$  of question  $i$ , we augment the task reward with an additional budget reward that is activated only when  $\tau_g$  is *correct* and improves upon the (pre-update) question baseline:

$$R(\tau_g) = R_{\text{task}}(\tau_g) + \lambda R_{\text{bud}}(\tau_g; b_i), \quad (9)$$

$$\begin{aligned} R_{\text{bud}}(\tau_g; b_i) &= I_{g,\text{bud}} \cdot g(C(\tau_g), b_i), \\ I_{g,\text{bud}} &= \mathbb{1}[\text{correct}(\tau_g)] \cdot \mathbb{1}[C(\tau_g) < b_i]. \end{aligned} \quad (10)$$

where  $g(\cdot)$  quantifies the degree of budget improvement. We consider several simple choices for  $g(C, b)$ :

$$\begin{aligned} g_{\text{bin}}(C, b) &= 1, \\ g_{\text{tok}}(C, b) &= b - C, \\ g_{\text{ratio}}(C, b) &= \frac{b - C}{b + \epsilon}, \end{aligned} \quad (11)$$

Unless otherwise specified, we use **Ratio delta** as the default shaping function. Importantly, rollouts that exceed the baseline are *not* explicitly penalized for efficiency; they simply receive no additional budget reward. This avoids bluntly compressing all questions toward a uniform budget and preserves the ability to spend more when additional observation is consistently required for correctness.

**Optimization with GRPO.** We optimize the policy using a GRPO-style objective. For each question, we sample a group of rollouts, compute shaped returns using the pre-update baseline  $b_i$ , update the policy to favor higher-return rollouts relative to the group, and finally update  $b_i$  using the mean cost of correct rollouts in the group. Algorithm 1 summarizes the procedure.

## 5 Experiments

### 5.1 Experimental Setup

**Benchmarks** To evaluate ROVER, we conduct experiments on the widely adopted benchmarks spanning both reasoning-centric and general video understanding tasks. For the reasoning-oriented assessment, we evaluate on Video-Holmes (Cheng et al., 2025a), MMR-V (Zhu et al., 2025), VideoMathQA (Rasheed et al., 2025), and LongVideoReason (Chen et al., 2025a) in Table 1. For the general VideoQA, we evaluate on VideoMME (Fu et al., 2025a), VideoMMM (Cheng et al., 2025b), LongVideoBench (Wu et al., 2024), and MLVU (Zhou et al., 2025) in Table 2. Unless otherwise specified, we follow the official evaluation

---

### Algorithm 1 Budget-Efficient RL with gated question-wise baselines

---

```

1: for each question  $i$  do
2:   Sample a group of rollouts  $\{\tau_g\}_{g=1}^G$  with current policy  $\triangleright$ 
    $G = \text{rollout.n}$ 
3:    $S \leftarrow 0$ ;  $M \leftarrow 0$   $\triangleright S$ : sum of costs over correct rollouts;  $M$ : count
4:   for each rollout  $\tau_g$  do
5:     Compute task reward  $R_{\text{task}}(\tau_g)$  and cost  $C(\tau_g)$ 
6:     if  $\tau_g$  is correct then
7:        $S \leftarrow S + C(\tau_g)$ ;  $M \leftarrow M + 1$ 
8:     end if
9:     if  $\tau_g$  is correct and  $C(\tau_g) < b_i$  then  $\triangleright$  use pre-update  $b_i$ 
10:       $R_{\text{bud}}(\tau_g) \leftarrow g(C(\tau_g), b_i)$ 
11:    else
12:       $R_{\text{bud}}(\tau_g) \leftarrow 0$ 
13:    end if
14:     $R(\tau_g) \leftarrow R_{\text{task}}(\tau_g) + \lambda R_{\text{bud}}(\tau_g)$ 
15:  end for
16:  Update policy with GRPO using  $\{R(\tau_g)\}_{g=1}^G$ 
17:  if  $M > 0$  then
18:     $\bar{C}_i^+ \leftarrow S/M$ 
19:    Update baseline  $b_i \leftarrow (1 - \alpha)b_i + \alpha\bar{C}_i^+$ 
20:  end if
21: end for

```

---

protocols of each benchmark and report the corresponding accuracy or overall score, where higher values indicate better performance.

**Baselines.** We compare ROVER with a set of representative baselines spanning closed-source frontier VLMs, strong open-source video VLMs, and recent tool-augmented long-video agents in Tables 1 and 2. For closed-source VLMs, we include GPT-4o (Hurst et al., 2024) and Gemini-series models, namely Gemini 1.5 Pro (Reid et al., 2024), 2.0 Flash, 2.5 Flash, and 2.5 Pro (Team, 2025), as high-performing references when results are available. For open-source VLMs, we use Qwen2.5-VL-7B-Instruct (Bai et al., 2025) as our base model, and compare against reasoning-oriented post-training methods and long-video models, including Video-R1 (Feng et al., 2025), VideoChat-R1 (Li et al., 2025), VideoRFT (Wang et al., 2025a), Video-MTR (Xie et al., 2025), LongVILA (Chen et al., 2025b), and LongVILA-R1 (Chen et al., 2025a), as well as widely used VideoQA baselines such as VideoLLaVA (Lin et al., 2024), ShareGPT4V (Chen et al., 2024), and Video-XL (Shu et al., 2025). For tool-augmented agents, we compare against FrameThinker (He et al., 2025), ReWatch-R1 (Zhang et al., 2025a), and Conan (Ouyang et al., 2025), which explicitly perform multi-turn evidence acquisition.

For benchmarks where a method does not report

Model	Holmes $\uparrow$	MMR-V $\uparrow$	MathQA $\uparrow$	LVR $\uparrow$	Avg $\uparrow$
<i>Closed-source VLMs</i>					
GPT-4o	42.0	44.0	–	–	–
Gemini 1.5 Pro	41.2	–	–	<b>69.3</b>	–
Gemini 2.0 Flash	30.6	42.6	–	–	–
Gemini 2.5 Flash	–	51.2	–	–	–
Gemini 2.5 Pro	45.0	–	–	–	–
<i>Open-source VLMs</i>					
Qwen2.5-VL-7B-Instruct	27.8	30.1	24.8	64.1	36.7
Video-R1	36.5	36.3	26.0	68.1	<u>41.7</u>
VideoChat-R1	33.0	36.1	22.6	67.2	39.7
VideoRFT	41.4	–	25.0	67.9	–
Video-MTR	35.7	36.5	–	57.3	–
<i>Tool-augmented Agent</i>					
FrameThinker	<u>46.8</u>	<u>45.3</u>	–	<b>76.1</b>	–
ReWatch-R1	37.8	–	<u>30.7</u>	70.5	–
Conan	44.6	42.7	–	72.8	–
<b>ROVER</b>	<b>49.2</b>	<b>46.3</b>	<b>31.4</b>	<u>74.3</u>	<b>50.3</b>

Table 1: Main results on four video reasoning benchmarks (higher is better), along with their average (Avg) when all four benchmark scores are available. Bold and underline denote the best and second-best results in each column; “–” indicates that results are not available.

470 results or is not directly comparable, we mark the  
471 entry as unavailable, denoted by “–”, rather than  
472 introducing additional evaluation assumptions.

473 **Abbreviations.** For compactness, we abbrevi-  
474 ate Video-Holmes as **Holmes**, VideoMathQA as  
475 **MathQA**, and LongVideoReason as **LVR** when  
476 reporting per-benchmark results. In ablation stud-  
477 ies, we use **SFT** to denote cold-start supervised  
478 fine-tuning, **TA-RL** to denote tool-augmented RL,  
479 and **BE-RL** to denote budget-efficient RL.

480 **Implementation Details.** See Appendix A.1.

## 481 5.2 Main Results

482 **Accuracy on video reasoning benchmarks.** Ta-  
483 ble 1 compares ROVER with representative closed-  
484 source VLMs, open-source VLMs, and recent  
485 tool-augmented agents across four challenging  
486 video reasoning benchmarks. ROVER achieves the  
487 best performance among tool-augmented agents  
488 on Holmes, MMR-V, and MathQA, and achieves  
489 the second-best performance on LVR with 74.3.  
490 Compared with our base model Qwen2.5-VL-7B-  
491 Instruct, ROVER yields consistent gains, improv-  
492 ing by +21.4, +16.2, +6.6, and +10.2 points on  
493 Holmes, MMR-V, MathQA, and LVR, respec-  
494 tively, indicating that our mixed-granularity vi-  
495 sual interface together with the three-stage train-  
496 ing pipeline effectively strengthens long-horizon  
497 video reasoning. On LVR, ROVER reaches 74.3  
498 and remains competitive with the strongest tool-

Model	VideoMME(O/L) $\uparrow$	VideoMMMU $\uparrow$	LongVideoBench $\uparrow$	MLVU $\uparrow$	Avg $\uparrow$
<i>Open-source VLMs</i>					
VideoLLaVA	40.4/38.1	–	39.1	30.7	–
ShareGPT4V	43.6/37.9	–	39.7	33.8	–
Video-XL	52.3/48.9	–	50.7	45.6	–
Video-R1	59.3/ <u>50.2</u>	50.3	–	<b>62.5</b>	–
VideoChat-R1	56.9/–	46.4	–	<u>60.5</u>	–
VideoRFT	<u>64.8</u> /–	49.9	–	–	–
Video-MTR	59.3/–	–	–	59.7	–
Qwen2.5-VL-7B-Instruct	53.6/44.7	49.7	48.9	48.4	<u>49.0</u>
<i>Tool-augmented Agent</i>					
FrameThinker	–/47.6	–	<u>52.9</u>	59.1	–
ReWatch-R1	<b>64.9</b> /–	<b>52.2</b>	50.5	55.2	–
<b>ROVER</b>	<b>64.3</b> / <b>52.5</b>	<u>51.8</u>	<b>54.3</b>	56.4	<b>55.2</b>

Table 2: Main results on four long-video understanding / VideoQA benchmarks (higher is better), along with their average (Avg) when all four benchmark scores are available. Bold and underline denote the best and second-best results in each column; “–” indicates that results are not available or not reported.

Variant	#VTokens $\downarrow$	Holmes $\uparrow$	MMR-V $\uparrow$	MathQA $\uparrow$	LVR $\uparrow$	Avg $\uparrow$
Base Model	2.9K	27.8	30.1	24.8	64.1	36.7
SFT	3.6K	41.3	38.4	27.3	70.1	44.3
SFT + BE-RL	3.2K	43.1	46.9	29.4	75.2	48.7
SFT + TA-RL	8.8K	39.1	41.2	25.4	71.2	44.0
ROVER	4.2K	49.2	46.3	31.4	74.3	50.3

Table 3: Stage-level ablations on four video reasoning benchmarks (higher is better) with average visual token usage (lower is better).

499 augmented baseline, FrameThinker at 76.1, sug-  
500 gesting that ROVER can scale to long-context,  
501 multi-step reasoning while maintaining robust per-  
502 formance.

503 **Accuracy on long-video understanding bench-**  
504 **marks.** We further include results on long-video  
505 understanding / VideoQA benchmarks in Ta-  
506 ble 2. ROVER consistently improves over the  
507 Qwen2.5-VL-7B-Instruct base model across all  
508 four benchmarks, with especially large gains on  
509 VideoMME and MLVU. Compared with recent  
510 tool-augmented agents, ROVER remains compet-  
511 itive on LongVideoBench with 54.3, as well as  
512 VideoMME, VideoMMMU, and MLVU.

## 513 5.3 Ablation Studies

514 We summarize stage-level ablations in Table 3 and  
515 use Figures 3 and 4 to illustrate training dynamics.  
516 We also ablate tool reward designs during TA-RL  
517 in Appendix A.3 (Figure 6), and summarize budget  
518 shaping choices in Table 4.

519 **Stage-level ablations.** Table 3 reports both accu-  
520 racy and average visual-token usage (#VTokens),  
521 computed as the average trajectory cost  $\mathbb{E}[C(\tau)]$   
522 (Equation (2)); only LOOKUP\_EVENT and ZOOM\_IN  
523 contribute nonzero visual tokens, while text-only  
524 tools incur zero cost. Cold-start SFT is necessary

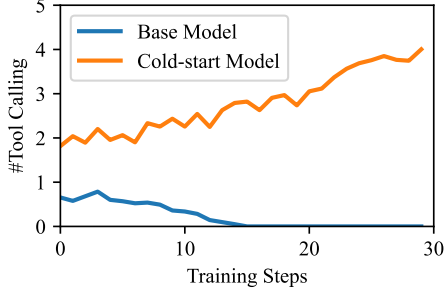


Figure 3: Tool calling behavior in TA-RL: base model vs. cold-start (SFT) model. The base model’s tool calling collapses over training, while the cold-start model sustains and increases tool usage.

for strong tool-augmented reasoning, improving the average score by +7.6 points over the base model. Directly applying BE-RL after SFT further improves accuracy with comparable token usage, while TA-RL increases token usage by encouraging exploration and evidence acquisition. Our ROVER achieves the highest average accuracy while keeping token usage moderate, indicating that BE-RL can prune redundant evidence on top of the exploration behavior learned in TA-RL.

**Impact of cold-start SFT on Tool-Calling.** Figure 3 compares the average number of tool calls during TA-RL when starting from the base vision-language model versus a model that has undergone cold-start SFT. The base model occasionally triggers tool calls at the beginning, but its tool calling decays steadily and collapses to nearly zero as training proceeds. In contrast, the cold-start model begins with substantially higher tool usage and continues to increase tool calling, suggesting that supervised initialization is important for maintaining stable tool-augmented behavior under reinforcement learning.

**Impact of TA-RL on Tool-Calling.** Figure 4 reports the average visual token usage over the course of BE-RL, comparing policies trained with and without the TA-RL stage. Higher visual token usage indicates stronger exploration and evidence acquisition (though higher token usage is not necessarily better under a budget objective). When skipping TA-RL, exploration is brief during the first epoch, where BE-RL primarily establishes per-question baselines and the budget shaping signal is effectively inactive, and then the budget reward quickly compresses token usage from the second epoch onward. This *premature budget contrac-*

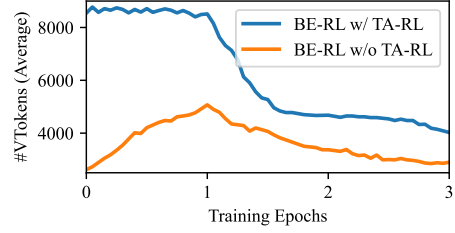


Figure 4: Visual token usage during BE-RL with vs. without TA-RL. TA-RL pretraining prevents premature budget contraction and enables more gradual pruning for efficiency.

Budget Shaping	#VTokens ↓	Holmes ↑	MMR-V ↑	MathQA ↑	LVR ↑	Avg ↑
Before BE-RL	8.8K	39.1	41.2	25.4	71.2	44.2
Ratio Delta (ROVER)	4.2K	49.2	46.3	31.4	74.3	50.3
Token Delta	4.1K	47.3	44.2	31.6	72.1	48.8
Binary Delta	5.9K	46.1	40.4	28.1	69.3	46.0

Table 4: Ablation of budget shaping choices in BE-RL.

tion can reduce exploration headroom and limit further evidence acquisition in later epochs. In contrast, TA-RL establishes stronger exploration capacity and following BE-RL can prune redundant evidence more gradually.

**Budget shaping choices.** Our BE-RL objective uses a baseline-relative shaping term  $g(C, b)$  as described in Section 4. Table 4 compares shaping choices for  $g(C, b)$  in Equation (11) and includes the model before BE-RL as a reference baseline. In our setting, the ratio form  $g_{\text{ratio}}$  offers the most effective shaping, achieving the highest average score while substantially reducing visual token usage relative to the pre-BE-RL policy. Compared with token-delta shaping, ratio-delta shaping yields higher accuracy at similar token usage, while binary-delta shaping tends to spend more tokens with lower overall accuracy.

## 6 Conclusion

We introduced ROVER, a tool-augmented long-video reasoning framework that allocates visual-token budget adaptively via a mixed-granularity interface for temporal skimming and targeted zoom-in. A three-stage post-training recipe stabilizes tool use and encourages budget-efficient evidence acquisition through a budget baseline mechanism. Experiments show that ROVER substantially improves long-video reasoning performance, suggesting that adaptive budget allocation with mixed-granularity evidence acquisition is an effective and scalable approach for long-context, multi-step reasoning.

## 7 Limitation

We use visual-token budget as a controllable, hardware-agnostic proxy for efficiency; however, it does not fully capture end-to-end serving cost under different stacks (e.g., decoding overhead, batching, I/O, and kernel-level performance). A strictly apple-to-apple comparison of visual token cost against all contemporary baselines is also challenging, since visual-token cost depends on input configurations such as frame sampling and effective resolution that are not uniformly reported, and would otherwise require additional assumptions; we therefore emphasize controlled ablations under matched settings to isolate the efficiency gains of our design. Finally, we instantiate the framework with a core set of evidence-acquisition tools and a fixed training recipe; extending to richer toolsets may require additional interface engineering and re-tuning of the reward/budget schedule.

## References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. [Qwen2.5-vl technical report](#). *CoRR*, abs/2502.13923.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2024. [Sharegpt4v: Improving large multi-modal models with better captions](#). In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XVII*, volume 15075 of *Lecture Notes in Computer Science*, pages 370–387. Springer.

Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, Sifei Liu, Hongxu Yin, Yao Lu, and Song Han. 2025a. [Scaling RL to long videos](#). *CoRR*, abs/2507.07966.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Yihui He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. 2025b. [Longvila: Scaling long-context visual language models for long videos](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. 2025a. [Video-holmes: Can MLLM think like holmes for complex video reasoning?](#) *CoRR*, abs/2505.21374.

Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. 2025b. [Video-holmes: Can MLLM think like holmes for complex video reasoning?](#) *CoRR*, abs/2505.21374.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. 2024. [Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms](#). *CoRR*, abs/2406.07476.

DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *CoRR*, abs/2501.12948.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. 2025. [Video-r1: Reinforcing video reasoning in mllms](#). *CoRR*, abs/2503.21776.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, and 2 others. 2025a. [Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 24108–24118. Computer Vision Foundation / IEEE.

Shenghao Fu, Qize Yang, Yuan-Ming Li, Xihan Wei, Xiaohua Xie, and Wei-Shi Zheng. 2025b. [LOVE-R1: advancing long video understanding with an adaptive zoom-in mechanism via multi-step reasoning](#). *CoRR*, abs/2509.24786.

Haonan Ge, Yiwei Wang, Kai-Wei Chang, Hang Wu, and Yujun Cai. 2025. [Famemind: Frame-interleaved video reasoning via reinforcement learning](#). *CoRR*, abs/2509.24008.

Zefeng He, Xiaoye Qu, Yafu Li, Siyuan Huang, Daizong Liu, and Yu Cheng. 2025. [Framethinker: Learning to think with long videos via multi-turn frame spotlighting](#). *CoRR*, abs/2509.24304.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, and 79 others. 2024. [Gpt-4o system card](#). *CoRR*, abs/2410.21276.

Sicong Leng, Jing Wang, Jiayi Li, Hao Zhang, Zhiqiang Hu, Boqiang Zhang, Yuming Jiang, Hang Zhang, Xin Li, Lidong Bing, Deli Zhao, Wei Lu, Yu Rong, Aixin Sun, and Shijian Lu. 2025. [Mmr1: Enhancing multimodal reasoning with variance-aware sampling and open resources](#). *Preprint*, arXiv:2509.21268.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and



814 Yu Rong. 2025. [Vl-cogito: Progressive curriculum](#)  
815 [reinforcement learning for advanced multimodal rea-](#)  
816 [soning](#). *Preprint*, arXiv:2507.22607.

817 Congzhi Zhang, Zhibin Wang, Yinchao Ma, Ji-  
818 awei Peng, Yihan Wang, Qiang Zhou, Jun Song,  
819 and Bo Zheng. 2025a. [Rewatch-r1: Boosting](#)  
820 [complex video reasoning in large vision-language](#)  
821 [models through agentic data synthesis](#). *CoRR*,  
822 [abs/2509.23652](#).

823 Haoji Zhang, Xin Gu, Jiawen Li, Chixiang Ma, Sule  
824 Bai, Chubin Zhang, Bowen Zhang, Zhichao Zhou,  
825 Dongliang He, and Yansong Tang. 2025b. [Think-](#)  
826 [ing with videos: Multimodal tool-augmented rein-](#)  
827 [forcement learning for long video reasoning](#). *CoRR*,  
828 [abs/2508.04416](#).

829 Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan  
830 Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma.  
831 2024. [Llamafactory: Unified efficient fine-tuning](#)  
832 [of 100+ language models](#). In *Proceedings of the*  
833 *62nd Annual Meeting of the Association for Computa-*  
834 *tional Linguistics (Volume 3: System Demonstra-*  
835 *tions)*, Bangkok, Thailand. Association for Computa-  
836 *tional Linguistics*.

837 Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang  
838 Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yong-  
839 ping Xiong, Bo Zhang, Tiejun Huang, and Zheng  
840 Liu. 2025. [MLVU: benchmarking multi-task long](#)  
841 [video understanding](#). In *IEEE/CVF Conference on*  
842 *Computer Vision and Pattern Recognition, CVPR*  
843 *2025, Nashville, TN, USA, June 11-15, 2025*, pages  
844 13691–13701. Computer Vision Foundation / IEEE.

845 Kejian Zhu, Zhuoran Jin, Hongbang Yuan, Jiachun Li,  
846 Shangqing Tu, Pengfei Cao, Yubo Chen, Kang Liu,  
847 and Jun Zhao. 2025. [MMR-V: what’s left unsaid? A](#)  
848 [benchmark for multimodal deep reasoning in videos](#).  
849 *CoRR*, [abs/2506.04141](#).

## A Appendix

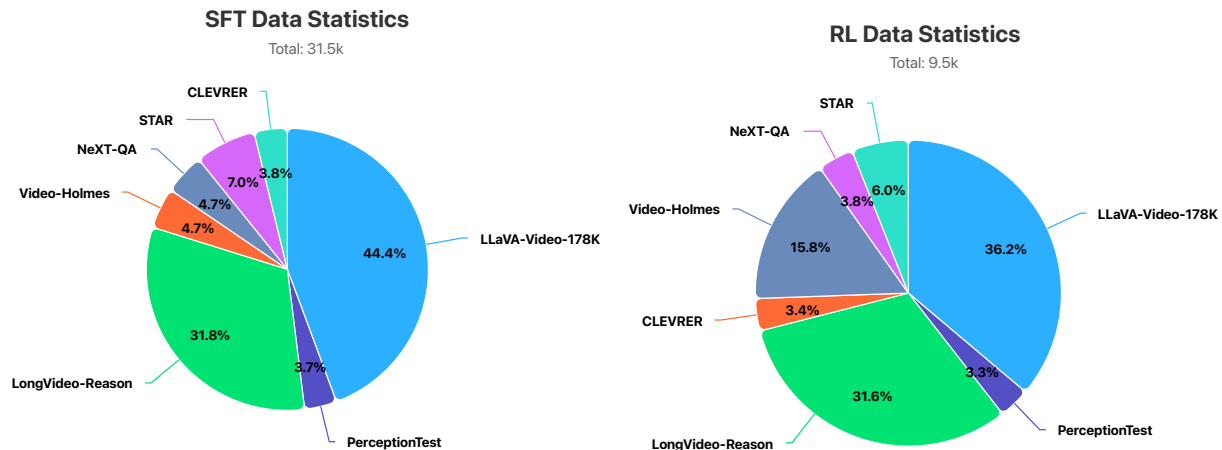


Figure 5: Dataset statistics for cold-start SFT (left) and RL (right) training data.

### A.1 Implementation Details

**Training setup.** We adopt Qwen2.5-VL-7B-Instruct as the base model for its competitive performance in image and video understanding. Cold-start SFT is performed using LLaMAFactory (Zheng et al., 2024) on 32 H100 GPUs with a global batch size of 128 for 3 epochs, using a learning rate of  $1e-5$  with a cosine decay schedule and 0.1 warm-up ratio. Tool-augmented RL is conducted on 64 H100s with a global batch size of 64 and a constant learning rate of  $5e-6$ . Both the tool-augmented and budget-efficient RL stages are trained for 3 epochs and are implemented in VeRL (Sheng et al., 2024). To support progressively longer video inputs, we employ multimodal sequence parallelism (Chen et al., 2025a,b) with a sequence parallel size of 4.

### A.2 Dataset Construction

#### A.2.1 SFT Data Construction

We combine data from 7 existing VideoQA and video reasoning datasets. Specifically, we randomly sample 15K data points from LongVideoReason (Chen et al., 2025a), all 1.7K data points from VideoHolmes (Cheng et al., 2025a), and 30K data points from NExT-QA, STAR, CLEVRER, PerceptionTest, and LLaVA-Video-178K.

**Trajectory Synthesis.** We deploy the strongest open-source VLM Qwen3-VL-235B-A22B-Instruct (Bai et al., 2025) as our base model to generate synthetic tool-calling trajectories for supervised fine-tuning (SFT). We follow the same multi-turn protocol as described in Section 3. For each question, we rollout 8 trajectories with

temperature 1.0, and conduct filtering described below to select high-quality ones.

**Trajectory Filtering.** We apply a combination of criteria to filter out low-quality trajectories. First, we only keep trajectories that lead to the correct final answer. Second, we filter out trajectories that contain redundant tool calls. Specifically, if two consecutive tool calls retrieve video segments with more than 80% temporal overlap, we consider the latter one as redundant and remove the entire trajectory. Additionally, we rebalanced the filtered trajectories according to their length. For each question, we only keep 1 trajectory if multiple ones pass the filtering. In total, we keep 31.5K high-quality trajectories for SFT training. The data statistics are summarized in the left pie chart in Figure 5.

#### A.2.2 RL Data Construction

For the Tool-Augmented RL (TA-RL) and Budget-Aware RL (BE-RL) stages, we further filter the SFT data by running Qwen2.5-VL-7B-Instruct (Bai et al., 2025) to answer each question with 8 repeated trials and temperature 1.0. We filter out questions that are overly easy or hard for the base model. Specifically, only questions that are correctly answered in 3 to 6 out of 8 trials are kept for RL training. We further downsampled the filtered data according to their source datasets to balance the mixture, resulting in 9.5K questions for RL training. The data statistics are summarized in the right pie chart in Figure 5.

### A.3 Impact of Tool Reward Design

Figure 6 ablates tool reward designs during TA-RL and tracks the resulting average visual token us-

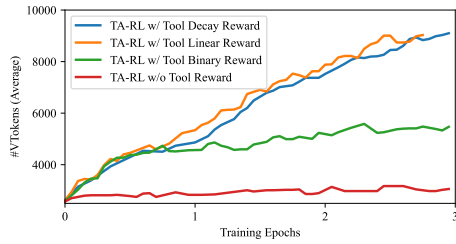


Figure 6: Visual token usage during TA-RL under different tool rewards. Denser shaping (linear/decay) sustains higher exploration than binary or no tool reward.

age. The *decay* form corresponds to Equation (3); the *binary* and *linear* forms correspond to an all-or-nothing indicator and a reward linearly proportional to the number of evidence tool calls, analogous to the binary and token-delta (linear) shaping choices in budget rewards (Equation (11)). Higher visual token usage here indicates stronger exploration and evidence acquisition. Without the tool reward, the policy stays at a low and nearly flat token level, reflecting weak evidence acquisition. Adding a binary tool reward increases token usage to a moderate level but quickly plateaus, suggesting that sparse, all-or-nothing shaping is insufficient to sustain exploration. In contrast, more graded tool rewards (linear or decay) lead to substantially stronger and steadily increasing evidence acquisition over training, indicating that denser shaping better supports intermediate tool decisions under correctness-gated reinforcement learning. Notably, linear and decay rewards achieve similar exploration strength in terms of visual token usage; however, we observe that the decay reward yields a lower sequence clip ratio during training, so we use the decay reward by default.

#### A.4 Qualitative Examples

We show two representative tool-calling conversations for the same multiple-choice question to highlight how RL stages shape interaction length.

**Short (BE-RL; Figure 7).** The policy makes a straight-cut decision from the provided key frames (notably frame 852) and outputs the answer directly without any extra tool calls, reflecting budget control that favors earlier termination once sufficient evidence is deemed available.

**Long (TA-RL; Figure 8).** The policy first zooms in on frame 852 to identify the coach/team, then performs additional verification by trying to sample a very narrow frame range (which even triggers a tool error), and finally zooms in on frame

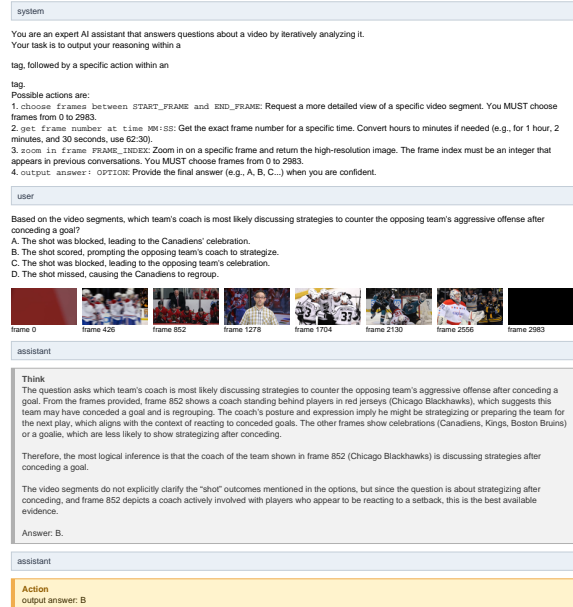


Figure 7: A short, budget-controlled tool-calling example after BE-RL.

1704 to confirm the scoring celebration before answering. This is a typical post TA-RL pattern with redundant tool usage and backtracking (two zooms + one failed tool call) even when an early hypothesis is already available.

955  
956  
957  
958  
959

system

You are an expert AI assistant that answers questions about a video by iteratively analyzing it. Your task is to output your reasoning within a tag, followed by a specific action within an tag.









Possible actions are:

- choose frames between START\_FRAME and END\_FRAME: Request a more detailed view of a specific video segment. You MUST choose frames from 0 to 2983.
- get frame number at time HH:SS: Get the exact frame number for a specific time. Convert hours to minutes if needed (e.g., for 1 hour, 2 minutes, and 30 seconds, use 62:30).
- zoom in frame FRAME\_INDEX: Zoom in on a specific frame and return the high-resolution image. The frame index must be an integer that appears in previous conversations. You MUST choose frames from 0 to 2983.
- output answer : OPTION: Provide the final answer (e.g., A, B, C,...) when you are confident.

user

Based on the video segments, which team's coach is most likely discussing strategies to counter the opposing team's aggressive offense after conceding a goal?

A. The shot was blocked, leading to the Canadiens' celebration.  
 B. The shot scored, prompting the opposing team's coach to strategize.  
 C. The shot was blocked, leading to the opposing team's celebration.  
 D. The shot missed, causing the Canadiens to regroup.











assistant

**Think**  
 The question asks which team's coach is most likely discussing strategies to counter the opposing team's aggressive offense after conceding a goal. The video frames include multiple teams (Canadiens, Blackhawks, Kings, Sharks, Capitals, Bruins), celebrations, and a coach (frame 852) observing his team. To identify the correct answer, I need to find contextual evidence such as a coach's reaction after conceding a goal or a team regrouping. Frame 852 shows a coach with his team sitting on the bench, which may imply a reaction after conceding. To get a clearer understanding, I should zoom in on frame 852 to examine the coach's expression and context more closely.

**Action**  
 zoom in frame 852

user



assistant

**Think**  
 In frame 852, I can see a coach standing behind his players on the bench. The players are wearing red jerseys with the Chicago Blackhawks logo. The coach appears to be observing the game intently, and the players seem focused or possibly frustrated, which could suggest they have recently conceded a goal and are regrouping. This aligns with the scenario in the question about a coach discussing strategies after conceding a goal.

Now, I need to determine which option best fits this context. Option D mentions "The shot missed, causing the Canadiens to regroup," but the team in frame 852 is the Blackhawks, not the Canadiens. Option A and C mention the Canadiens, which does not match the team shown. Option B refers to the opposing team's coach strategizing after a goal is scored, which could potentially fit if the Blackhawks conceded a goal. Since the coach appears to be reacting to a negative situation, I need to confirm if a goal was scored against them.

To verify if the Blackhawks conceded a goal, I should look at frames that show potential scoring moments or celebrations. I will examine frame 1704, which shows the Kings celebrating, suggesting they may have scored. If the Kings scored, the Blackhawks likely conceded, making their coach's reaction in frame 852 relevant.

**Action**  
 choose frames between 1700 and 1708

user


Error: Frame range [1700, 1708] too small for sampling 8 frames.

assistant

**Think**  
 The frame range I requested was too small, so I need to adjust my approach. Instead of trying to sample a narrow range, I will directly examine frame 1704, which likely shows the Kings celebrating a goal. This will help confirm whether the Blackhawks conceded a goal, which is crucial for determining if their coach is strategizing after conceding.

**Action**  
 zoom in frame 1704

user



assistant

**Think**  
 In frame 1704, the Los Angeles Kings players are celebrating a goal, as evidenced by their huddling and expressions of joy. This suggests that the Kings scored, and since the coach in frame 852 is with the Chicago Blackhawks, it is likely that the Blackhawks conceded this goal. Following a conceded goal, it is common for a coach to discuss strategies to counter the opposing team's aggressive offense. Therefore, the coach in frame 852 is most likely strategizing after conceding a goal to the Kings.

Now, I need to match this context to one of the given options. Option B states: "The shot scored, prompting the opposing team's coach to strategize." This fits the scenario, as the Kings scored (opposing team), and the Blackhawks' coach (opposing team's coach from the Kings perspective) is likely strategizing after conceding.

**Action**  
 output answer: B

Figure 8: A long tool-calling example after TA-RL.