GETR: GUIDED EXPLORATION TRAJECTORIES FOR LONG-VIDEO REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video reasoning requires finding and combining evidence that may appear throughout an hour-long video. An answer-only reward does not tell the policy whether it inspected the supporting evidence, because the same answer may come from partial or irrelevant observations, or from a guess. Giving the model a tool to look closer does not help by itself, because an untrained model does not know where to look: Qwen3-VL-8B with a crop tool is less accurate than with uniform frames. We propose Guided Exploration Trajectories for Reasoning (GETR), which rewards a long-video policy for looking at the correct evidence. The policy starts from hints, candidate segments retrieved offline from text summaries of the video, and is rewarded for where its first crop falls and for whether its final crop covers the labeled evidence. An IoGT coverage reward measures whether its final crop contains the labeled evidence, and we report how often any crop reaches that evidence. We train the model with GRPO on 1,668 question–answer–evidence triplets, without SFT or labeled tool calls, whereas LongVT and EVA fine-tune on 263K and 21K supervised samples, respectively, including distilled tool-call trajectories. Across four long-video benchmarks (Video-MME-Long, LVBench, HERBench-Lite, and CG-Bench), GETR improves the Qwen3-VL-8B base model by 3.6 points on average, and by 4.4 on which the same model given the crop tool without training instead loses 5.8 points. The gains are largest where the evidence is short relative to the video (CG-Bench +6.3, LVBench +5.4).

1 INTRODUCTION

Long-video reasoning (Mangalam et al., 2023; Zhang et al., 2025a; Yeo et al., 2025) requires answering questions over extended temporal context. Since processing every frame of a long video at high resolution is often computationally expensive, many works rely on retrieval, sampling, or tool-based inspection to select a subset of temporal regions for analysis. In these systems, solving the problem correctly depends not only on the model's ability to reason over observed content, but also on whether the selected regions contain the evidence needed to support the answer. Answer accuracy does not reveal this: it measures the final prediction, not the process that produced it. An agent may answer correctly after inspecting only partial or irrelevant regions, while another may gather the necessary evidence yet fail at the final prediction step. Recent evaluation of video reasoning traces finds that the dominant failure of current models is temporal localization rather than the reasoning aspect (Nagrani et al., 2025).

Long videos make this problem acute for training. An agent trained with RL on an answer-only reward learns where to look correctly only if rewards for the correct answers are correlated with the correctness of the inspected evidence. In an hour-long video, the evidence for a question may last only a few seconds, so an untrained policy rarely finds it. Correct answers, by contrast, remain available without finding it: from the uniformly sampled frames the model sees first, by chance, or from subtitles, which VideoMME provides as an evaluation setting and which prior work, including LongVT, uses on it (Fu et al., 2025; Yang et al., 2025). An answer-only reward cannot tell these answers apart from answers grounded in the evidence, so it reinforces them just the same; Giving the model a tool to look closer does not solve this on its own: in our experiments, a strong video LLM given a crop tool without training becomes less accurate, not more (Sec. 4.1). Knowing where to look has to be learned, and we ask whether a reward on where the policy looked can teach it. Existing methods decide where to look in one of three ways. Single-pass models, including RL-trained

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

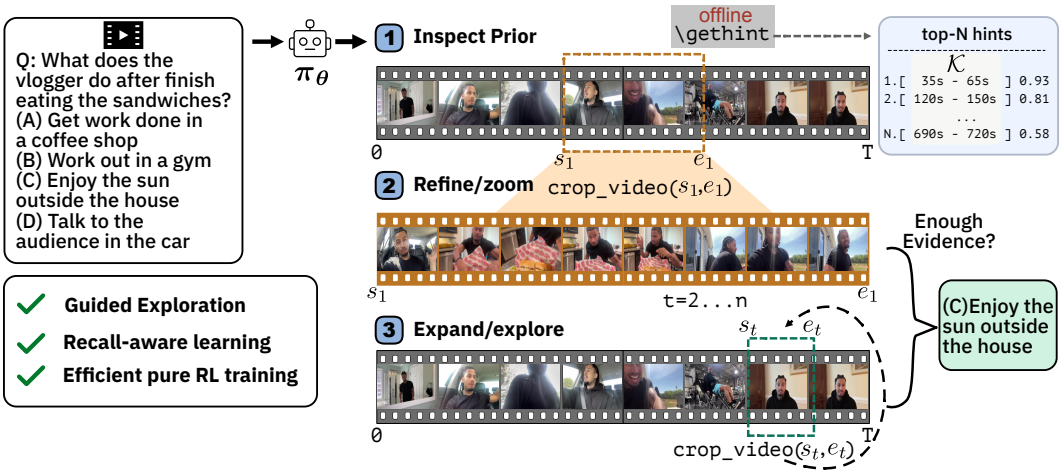


Figure 1: **Guided Exploration Trajectories for Reasoning** . Given a long video and a multiple-choice question, the agent first inspects candidate temporal windows given by `\gethint` (step 1), then iteratively *refines/zooms* into promising regions (step 2) and *expands/explores* beyond the initial candidates when needed (step 3) via `crop_video` (s_t, e_t) calls. The loop terminates once enough evidence exists to answer.

reasoners such as Video-R1 and VideoChat-R1 (Feng et al., 2025b; Li et al., 2025b; Wang et al., 2025a), reason over a fixed set of sampled frames. Training-free frame selection (Liu et al., 2025; Tang et al., 2025) and retrieval-based methods (Luo et al., 2024; Yeo et al., 2025) choose frames or segments by similarity to the question, but choose once and cannot revise that choice. Tool-using agents such as LongVT, EVA, FrameThinker and VITAL (Yang et al., 2025; Zhang et al., 2026; He et al., 2025; Zhang et al., 2025a) can inspect the video repeatedly, but first learn how to use the tool from supervised trajectories LongVT and EVA fine-tune on 263K and 21K samples, including distilled tool-call trajectories and LongVT then rewards the IoU of its crops with the evidence. Video-RTS (Wang et al., 2025c) learns without a supervised stage, but from the answer alone.

Figure 2 shows where tool-using agents look on LVBench. LongVT-RL places 95% of its localized crops in the first tenth of the video, whereas the annotated evidence is spread across it, and 54% of EVA's crops span more than half of the video. Looking in the right region in aggregate is not the same as reaching the evidence for a particular question.

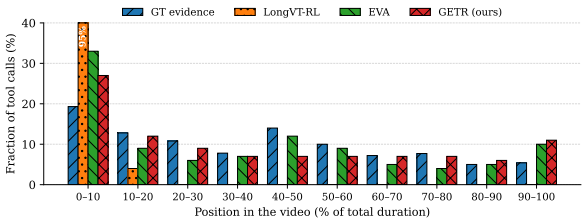


Figure 2: **Where tool calls land vs. where the evidence is** (LVBench, 1,549 questions). Share of localized `\cropvideo` calls (windows shorter than half the video) whose midpoint falls in each tenth of the video, and share of annotated evidence spans per tenth. Total-variation distance to the evidence distribution: LongVT-RL 0.77, EVA 0.18, GETR 0.14. The y-axis is capped at 40% (LongVT-RL's first bin: 95%); 54% of EVA's calls span more than half of the video and are not shown.

To address this, we propose Guided Exploration Trajectories for Reasoning (GETR), which rewards a long-video policy for where it looked. Before searching, the policy can call `\gethint`, which returns up to four candidate windows, the hints. To find them, a VLM writes a short text summary of each 30-second chunk of the video, once and offline, and the chunks whose summaries best match the question become the hints. The policy sees only the time spans of the windows, in random order and without scores; it never sees the summaries. The hints give the search a place to start but do not restrict it: the policy can check a hint, move to another, or crop any other part of the video with `\cropvideo`. Besides the answer, training rewards two decisions: how closely the first crop matches the labeled evidence, and whether the final crop covers it. Coverage is measured by Intersection-over-Ground-

Truth (IoGT), the fraction of the evidence span the crop contains; unlike IoU, it does not penalize a crop for including context around the evidence.

We evaluate one set of trained weights, from Qwen3-VL-8B, on LVBench (Wang et al., 2025b), VideoMME-Long (Fu et al., 2025), CG-Bench (Chen et al., 2024) and HERBench-Lite (Ben-Ami et al., 2025), without per-benchmark tuning. Our contributions are threefold. First, GETR learns where to look with RL alone, on 1,668 question–answer–evidence triplets, without SFT or tool-call traces. Second, it improves over the same model with uniform frames by 4.4 points on the three benchmarks that require the video (3.6 on all four) (Sec. 4.1). Third, we measure for every trajectory whether a crop reached the annotated evidence. On CG-Bench, a crop reaches the evidence for 16.8% of questions after training, against 7.1% before, and GETR answers 55.0% of those questions correctly, against 44.6% of the others (Sec. 4.2).

2 Related Work

Video MLLMs and frame selection. Video MLLMs reason over a fixed set of sampled frames (Bai et al., 2025b; Zhu et al., 2025; Zhang et al., 2024b; Bai et al., 2025a), so their answers depend on which segments happen to be observed; EgoSchema's temporal certificates (Mangalam et al., 2023) show that long evidence defeats such sampling. Training-free frame selection (Liu et al., 2025; Tang et al., 2025) and retrieval over video indices (Ren et al., 2026; Luo et al., 2024; Yeo et al., 2025; Zhang et al., 2025b) choose better frames, but in one step, before the model has seen any of them. `\gethint` plays a similar role in GETR, except that its candidates only seed the search: after each crop, the policy can issue another `\cropvideo` call based on what it has seen. **Tool-using video agents trained with RL.** Recent agents inspect long videos through repeated tool calls, trained with SFT and RL (Yang et al., 2025; He et al., 2025) or organized around planning, reflection, and memory (Zhang et al., 2026; Wang et al., 2025d; Zuo et al., 2025). Closest to ours are policies trained with RL and grounding-aware rewards: Video-MTR (Xie et al., 2025) rewards each turn by the relevance of its frames to annotated key segments, VideoTemp-o3 (Liu et al., 2026) uses a thresholded IoU reward, TimeSearch-R (Pan et al., 2025) rewards whether the searched frames alone suffice to answer, and Video-RTS (Wang et al., 2025c) learns tool use without a supervised stage. GETR differs in rewarding where the search ends by coverage of the evidence rather than by IoU, in adding a reward on where the search starts, and in evaluating, for every trajectory, whether the evidence was reached. **Temporal grounding and active perception.** Temporal grounding is scored by IoU (Lei et al., 2021; Clark et al., 2026) because over-reaching is an error there; an inspecting agent only needs to see enough evidence to answer, so we score coverage of the evidence span instead (Sec. 3.2). Choosing the next window from what has already been seen is active perception (Bajcsy, 1988; Jayaraman & Grauman, 2018) along the time axis. CG-Bench (Chen et al., 2024) and MINERVA (Nagrani et al., 2025) use annotated evidence to evaluate models; we use it as a training signal. Appendix A.1 discusses these lines of work in more detail, together with tool use and RL for reasoning.

3 Method

We present GETR, a Guided Exploration Trajectories for Reasoning framework. Given a video $\mathcal{V}$ of duration T and a question q, the goal is to predict the correct answer $\hat{a}$ from the video. GETR has three components. First, each video is divided into 30-second chunks, and each chunk is summarized in text once, offline using VLM. The summaries are compared with the question, and the time spans of the best-matching chunks become the hints: a set $\mathcal{K}$ of up to four candidate windows (Sec. 3.1). The policy sees only these windows, never the summaries. Second, the policy works with two tools: `\gethint` returns $\mathcal{K}$ as an unordered set without similarity scores, and `\cropvideo` returns timestamped frames from any window the policy requests. No window is imposed: every crop, including the first, may fall on a candidate, inside one, or anywhere else in the video. Third, the policy is trained with rewards for answer correctness, output format, whether its final crop covers the labeled evidence, and how well its first crop overlaps the labeled evidence (Sec. 3.2). Figure 1 gives an overview and Table 6 lists the notation.. The policy π_θ defines a distribution over trajectories conditioned on $(\mathcal{V}, q)$, from which we sample

$$\tau \sim \pi_\theta(\cdot \mid \mathcal{V}, q), \qquad \hat{a} = \phi(\tau), \tag{1}$$

where τ interleaves reasoning, tool calls, tool results and a final answer, and f extracts the answer from the final `<answer>` block from which ϕ extracts the answer.

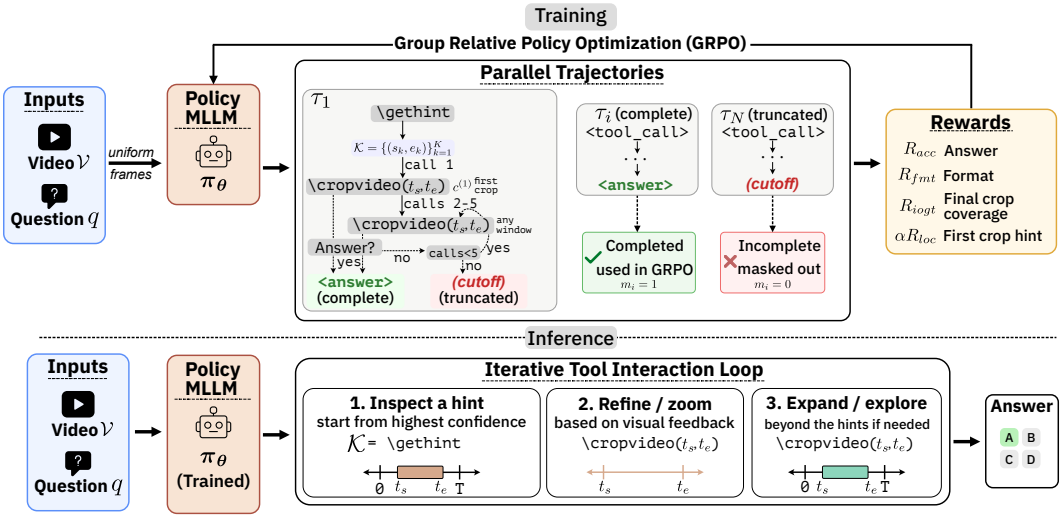


Figure 3: **Overview of GETR.** The policy generates N parallel trajectories per prompt. Each begins with `\gethint`, returning candidate windows $\mathcal{K}$ derived from timestamp-aligned local summaries. Each `\cropvideo` call may target a candidate window, a part of one, or any other interval of the video; no window is imposed at any step. Completed trajectories are scored by four rewards: answer correctness r_{acc}, format r_{fmt}, trajectory-level coverage r_{cov}, and localization Prior r_{loc} and used in the GRPO update; truncated trajectories are masked out and excluded from the gradient. *(Inference, bottom.)* The trained policy retrieves candidate windows via `\gethint`, once at the start, then issues a sequence of `\cropvideo` calls across the video until sufficient evidence is gathered and a final answer is produced.

3.1 Video Tools for Exploration

The two tools separate *where to look* (`\gethint`) from *what is there* (`\cropvideo`). Before any tool call, the policy receives a uniform pass over the whole video; frame counts and resolutions are given in Appendix A.7.

"Where in Video" with `\gethint`. Each video $\mathcal{V}$ of duration T is partitioned into $C = \lceil T/30 \rceil$ non-overlapping 30-second chunks, with start time $t_s^i = 30(i-1)$ and end time $t_e^i = \min(30i, T)$. For each chunk we sample $n_c = 30$ frames at equal intervals, and Qwen3-VL-8B-Instruct (Bai et al., 2025a) writes a short text summary σ_i, typically two to four sentences, of the main actions, objects and scene changes (examples in Appendix A.5). This gives timestamp-aligned summaries $\mathcal{S}_\mathcal{V} = \{(t_s^i, t_e^i, \sigma_i)\}_{i=1}^{C}$; because each summary carries its chunk's timestamps, a match in text maps back to a video segment. The summaries are computed once per video and reused across all questions. Given a question q, a retrieval query q' is formed from the question-side input available in each setting. During training (open-ended LongVT-Parquet split, Sec. 4) q' is the question text itself; at evaluation on multiple-choice benchmarks q' is the question concatenated with all answer options, so that q' contains competing hypotheses without revealing the correct one. In both cases q' and the summary text σ_i of each entry of $\mathcal{S}_\mathcal{V}$ are encoded by the same text-embedding model $e(\cdot)$, all-MiniLM-L6-v2 (Reimers & Gurevych, 2019), and compared with cosine similarity,

$$s_i = \tfrac{1}{2}\big(\cos(e(q'), e(\sigma_i)) + 1\big) \in [0, 1], \tag{2}$$

and chunks are selected by Eq. 3. The policy never sees q', σ_i or the scores; it receives only the selected timestamp windows. The summaries come from the same 8B model and are unreliable on the details questions ask about, such as counts, colours and on-screen text; a policy that could read them would learn to answer from text without looking.

$$\mathcal{I} = \big\{i : s_i \geq \max(\delta_{\text{abs}},\ \delta_{\text{frac}} \cdot \max_j s_j)\big\} \cap \text{top}_{K_{\max}}(s), \tag{3}$$

where $\text{top}_{K_{\max}}(s)$ denotes the indices of the $K_{\max}$ highest-scoring chunks, $K_{\max} = 4$ caps the number of candidates, $\delta_{\text{frac}} = 0.6$ retains chunks scoring at least 60% of the top chunk score, $\delta_{\text{abs}} = 0$

is an absolute score floor, and at least one candidate is always retained. Each selected chunk already has an associated timestamp window (t_s^i, t_e^i). We therefore represent the selected candidates as $\mathcal{K} = \{(\hat{s}_k, \hat{e}_k)\}_{k=1}^{K}$, $K \leq K_{\max}$, where $(\hat{s}_k, \hat{e}_k)$ is the timestamp window of the k-th selected chunk. The policy receives the selected candidates $\mathcal{K}$ through `\gethint` (more details in Appendix A.5). When invoked, `\gethint` returns the windows in random order, under the header "unordered candidate windows" and without scores, so the policy must check candidates visually rather than trust a ranking computed from text. The hints remain available throughout the search (Appendix A.10). Because the summaries and scores are cached, `\gethint` only looks up and formats saved results; it does not run another model. We next define the video-inspection tool that turns a chosen timestamp window into visual observations.

What is there: `\cropvideo`. With 512 uniform frames, evidence that spans the median 0.5% of an LVBench sample (Appendix A.3) receives about 2.6 frames, and the gain from inspecting it closer grows with video length (Figure 6). We adopt `\cropvideo` from prior work (Yang et al., 2025; Zhang et al., 2026; He et al., 2025). Given a window $c = (t_s, t_e)$ with $0 \leq t_s < t_e \leq T$, it returns $n = 8$ frames sampled uniformly from that interval, in temporal order, each with its absolute timestamp:

$$\textsc{crop}(c) = \big((\mathrm{str}(t_1), f_1), \ldots, (\mathrm{str}(t_n), f_n)\big), \qquad t_s \leq t_1 < \cdots < t_n \leq t_e, \tag{4}$$

where f_i is the frame at time t_i and $\mathrm{str}(t_i) =$ <round$(t_i, 1)$ `seconds`>, e.g. <`34.3 seconds`>. Following evidence that explicit time tokens improve temporal grounding (Meinardus et al., 2024), the timestamp lets the policy name its next window in seconds; we use the format Qwen3-VL already uses for video, so the backbone learns no new convention, and one decimal resolves positions well within a 30-second hint. Eight frames over a 30-second window are about twice as dense in time as the evaluation-time initial pass on LVBench, but the frames returned by `\cropvideo` have half its pixel budget (25,088 vs. 50,176 pixels): a crop adds temporal density, not spatial detail. The returned frames are fed back into the trajectory, and the policy decides whether to refine, move or answer. A trajectory may contain up to five tool calls; the policy always has a further turn to answer after its last tool result, and a sixth tool call ends the trajectory without an answer. At evaluation, a trajectory that ends without an answer gets one further turn in which it must answer (Sec. 3.2).

Why the hints are a prior and not a target. `\gethint` narrows the search from the whole video to a few windows, which makes the graded first-crop reward learnable. Because GRPO's gradient comes from differences among the N rollouts of a prompt, this reward is informative only if sibling rollouts start in different places, and returning the candidates unordered and unscored ensures that they do. Over the first five training steps, 90% of prompt groups show variation in R_{loc} (mean 0.13, zero in 63% of rollouts). When the reward instead indicates overlap with a ranked candidate list, only 8% do (mean 0.98, zero in 2%), because nearly every rollout satisfies it. Rewarding overlap with the evidence rather than starting at a hint forces the policy to refine the hints. Over training, the first crop's IoU with the evidence rises from 0.10 to 0.32, and the share of first crops that reproduce a hint window falls from 99% to 28%.

3.2 Reward Design

Besides answer correctness R_{acc} and output format R_{fmt}, the standard terms of prior tool-using agents (Yang et al., 2025; Zhang et al., 2026) (Appendix A.6), the reward scores where the policy looked: its final crop and its first.

Final crop: IoGT coverage reward. Let $c^\star(\tau)$ be the last `\cropvideo` window of trajectory τ and $g = [g_s, g_e]$ the labeled evidence segment. We reward the final crop by its Intersection-over-Ground-Truth (IoGT):

$$\mathrm{IoGT}(c, g) = \frac{|c \cap g|}{|g|}, \qquad R_{\mathrm{iogt}}(\tau) = \mathbb{1}[\mathrm{IoGT}(c^\star(\tau), g) > 0.5]\,. \tag{5}$$

Because IoGT divides by the duration of the evidence, a crop that contains the evidence is rewarded even if it also shows context around it. IoU, the metric when exact boundaries are the target (Lei et al., 2021; Clark et al., 2026) and the reward of earlier tool-using methods (Li et al., 2025b; Yang et al., 2025), would penalize that context, although the policy only needs to see the evidence to answer. Crops between the first and the last are not scored, so the policy can explore before it commits.

Algorithm 1 GETR training

Require: videos $\{\mathcal{V}\}$, training tuples $\{(q, a, g)\}$, policy π_θ, rollouts N
1: **Offline, once per video:** split $\mathcal{V}$ into 30-s chunks, summarize each with Qwen3-VL-8B-Instruct, and embed the summaries
2: **for** each training tuple $(\mathcal{V}, q, a, g)$ **do**
3: select the hints $\mathcal{K}$ by Eq. (3)
4: **for** $i = 1, \ldots, N$ **do**
5: roll out: uniform pass, then up to 5 tool calls
6: m_i by Eq. (8); R_i by Eq. (7)
7: **end for**
8: $A_{i,t} \leftarrow \big(R_i - \mu_{\mathcal{G}(i)}\big)/\big(\sigma_{\mathcal{G}(i)} + \varepsilon\big)\, M_{i,t}^{\text{resp}}\, m_i$
9: update θ with Eq. (9)
10: **end for**

First crop: localization reward. The evidence is short relative to the video, so R_{iogt} is rarely earned early in training. We add graded reward on the first crop $c^{(1)}(\tau)$, its IoU with labeled evidence:

$$R_{\text{loc}}(\tau) = \text{IoU}\big(c^{(1)}(\tau), g\big) = \frac{|c^{(1)}(\tau) \cap g|}{|c^{(1)}(\tau) \cup g|}. \tag{6}$$

R_{loc} is not thresholded and does not take the hints as its target, so a first crop that only partly overlaps the evidence still earns signal. Unlike for the final crop, IoU's penalty on surrounding context is intended here: copying a 30-second hint window around a 16-second evidence span earns at most $16/30 \approx 0.53$, while a first crop that narrows onto the span earns up to 1. The term is paid only when `\gethint` precedes the first `\cropvideo` call, a condition met in every rollout of our runs. R_{loc} scores where the search starts and R_{iogt} where it ends; both are 0 for a trajectory without a crop. The stotal reward is

$$R = R_{\text{acc}} + R_{\text{fmt}} + R_{\text{iogt}} + \alpha R_{\text{loc}}, \qquad \alpha = 0.2. \tag{7}$$

Excluding incomplete rollouts. GRPO (Shao et al., 2024) normalizes each reward against the N rollouts of the same prompt. A rollout cut off by the response-length limit would be penalized only for running out of tokens, so, following Mini-o3 (Lai et al., 2025), we exclude it from the update. In the runs reported here the same exclusion applied to any rollout that did not end in exactly one closed `<answer>` block, so a trajectory that stopped after a tool result received neither reward nor gradient; Sec. 4.3 reports the effect of this choice and the evaluation-time answer turn we use because of it. Let R_i be the reward of rollout i, $\mathcal{G}(i)$ its prompt group, and

$$m_i = \begin{cases} 0, & \text{if rollout } i \text{ reached the response-length limit or ended without a closed answer,} \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

3.3 Training Objective

With $r_{i,t}(\theta) = \pi_\theta(o_{i,t} \mid \cdot)/\pi_{\theta_{\text{old}}}(o_{i,t} \mid \cdot)$, we maximize the clipped GRPO objective with a KL penalty, averaged over the tokens of each response and then over responses (Algorithm 1):

$$\begin{aligned} \mathcal{J}(\theta) &= \mathbb{E}_{(\mathcal{V},q)}\, \mathbb{E}_{\tau_i \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{N} \sum_{i=1}^{N} \frac{m_i}{|o_i|} \sum_{t=1}^{|o_i|} \mathcal{J}_{i,t}(\theta) \right], \\ \mathcal{J}_{i,t}(\theta) &= \min(r_{i,t}(\theta) A_{i,t}, \text{clip}(r_{i,t}(\theta), 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}})\, A_{i,t}) \\ &\quad - \beta D_{\text{KL}}^{(i,t)}[\pi_\theta \,\|\, \pi_{\text{ref}}]\,. \end{aligned} \tag{9}$$

The factor m_i removes truncated rollouts from both terms. The KL term keeps the policy close to the initial model π_{ref}, uses the low-variance estimator of Shao et al. (2024) with $\beta = 10^{-3}$, and is not folded into the reward. We use $\epsilon_{\text{clip}} = 0.2$ and one PPO epoch per batch.

Table 1: Comparison with prior video-reasoning systems and controlled Qwen3-VL-8B baselines (accuracy, %). All Qwen3-VL-8B rows use 512 initial frames; crops return up to 8 frames and the clue-annotated CG-Bench questions. Other rows use their own backbones, frame budgets and settings and are listed for reference: † re-evaluated by us with each method's released checkpoint; ‡ the RL-only ablation reported of Yang et al. (2025) (Video-MME with subtitles, LVBench test); * Qwen off-the-shelf model baselines reproduced by us. Frames: input frames per question. Bold marks the best Qwen3-VL-8B score; –: not available.

Method	Backbone	Tool	Frames	VideoMME	LVBench	HERBench	CGBench	Avg.
Closed-Source Models								
GPT-5.4-mini Singh et al. (2025)	GPT-5.4-mini	✗	64	60.6	40.9	32.3	43.6	44.4
VideoAgent (Fan et al., 2024)	GPT-4	✓	87	49.0	29.3	–	–	–
DVD (Zhang et al., 2025b)	OpenAI-o3	✓	–	67.3	74.2	–	–	–
Open-Source Baselines								
Qwen2.5-VL-7B*	Qwen2.5-VL-7B	✗	512	63.3	40.5	34.5	35.0	43.3
Qwen3-VL-8B (Text only)*	Qwen3-VL-8B	✗	–	60.1	23.2	25.2	28.2	34.2
Qwen3-VL-8B (uniform)*	Qwen3-VL-8B	✗	512	66.9	47.4	38.1	40.1	48.1
Adaptive Agent (SFT-based)								
LongVT-SFT† (Yang et al., 2025)	Qwen2.5-7B + SFT	✓	512	48.4	40.5	31.5	35.0	38.9
EVA-SFT (Zhang et al., 2026)	Qwen2.5-7B + SFT	✓	26.6*	45.8	26.5	–	–	–
FrameThinker (He et al., 2025)	Qwen2.5-7B + SFT + RL	✓	24.1	47.6	36.6	–	–	–
LongVT-RL† (Yang et al., 2025)	Qwen2.5-7B + SFT + RL	✓	512	48.2	39.6	31.6	35.4	38.7
EVA† (Zhang et al., 2026)	Qwen2.5-7B + SFT + RL	✓	adaptive	48.4	43.3	32.5	39.3	40.9
Reasoning-based								
LongVT-7B (RL only)‡ (Yang et al., 2025)	Qwen2.5-VL-7B + RL	✓	–	52.7	37.1	–	–	–
Video-R1† (Feng et al., 2025b)	Qwen2.5-7B + RL	✗	32	51.3	38.0	36.0	35.1	40.1
VideoChat-R1† (Li et al., 2025b)	Qwen2.5-7B + RL	✗	32	51.9	39.1	32.5	37.0	40.1
Video-MTR§ (Xie et al., 2025)	Qwen2.5-VL-7B + RL	✓	80	–	42.3	–	–	–
Ours								
Qwen3-VL-8B (`\cropvideo`)*	Qwen3-VL-8B	✓	512	24.3	46.0	27.0	35.1	33.1
Qwen3-VL-8B (`\cropvideo` + `\gethint`)*	Qwen3-VL-8B	✓	512	51.8	38.9	26.4	32.2	37.3
+ GETR	**Qwen3-VL-8B + RL**	✓	**512**	**68.1**	**52.8**	**39.6**	**46.4**	**51.7**

4 EXPERIMENTS

Datasets. We train on the RL split of LongVT-Parquet (Yang et al., 2025): 1,668 open-ended questions on HACS and Ego4D-NAQ videos, each with an answer and an annotated evidence interval, and no tool-call traces. We evaluate on four benchmarks: LVBench (Wang et al., 2025b) (1,549 MCQs, avg. 68 min) and VideoMME-Long with subtitles (Fu et al., 2025) (900 MCQs, 30–60 min), mostly single-evidence; the clue-annotated subset of CG-Bench (Chen et al., 2024) (3,000 MCQs, 10–80 min); and HERBench-Lite (Ben-Ami et al., 2025) (2,000 MCQs, avg. 3 min), where every question requires at least three evidence cues. A single GETR model, Qwen3-VL-8B-Instruct trained with GRPO and no SFT, is evaluated on all four without per-benchmark tuning. Appendix A.2 gives dataset details, training settings, baselines and metrics.

4.1 MAIN RESULTS

Table 1 reports accuracy on the four benchmarks. All Qwen3-VL-8B rows share the same 512 uniformly sampled frames, judge and prompts; rows with `\cropvideo` additionally receive 8 frames per call at half the pixel budget. Rows 5 and 6 show how much each benchmark depends on the video. Text alone reaches already 60.1 on VideoMME-Long, compared to the 66.9 with uniform frames, because the subtitles carry most answers; on LVBench, HERBench-Lite and CG-Bench it reaches 23.2, 25.2 and 28.2, far below the 47.4, 38.1 and 40.1 of uniform frames. VideoMME-Long therefore says little about where a model looks. Given the crop tool but no training, the model falls below uniform frames on all three video benchmarks (LVBench 46.0 vs. 47.4, HERBench-Lite 27.0 vs. 38.1, CG-Bench 35.1 vs. 40.1): it can request segments but does not know which to request. Adding the hints does not help the untrained model either: 38.9 on LVBench and 26.4 on HERBench-Lite, below both uniform frames and the crop tool alone, because it follows the tool protocol poorly: 72.8% of its LVBench trajectories end without a closed <answer> block, and the judge scores the option named in the unstructured text. GETR recovers this loss and turns it into a gain, reaching 46.3 on Avg. and 51.7 on the four-benchmark mean (uniform frames: 41.9 and 48.1). The gain falls where a crop can add what the initial pass lacks. It is largest on CG-Bench (+6.3, 46.4 vs. 40.1) and LVBench (+5.4, 52.8 vs. 47.4), whose videos have median lengths of 25 and 61

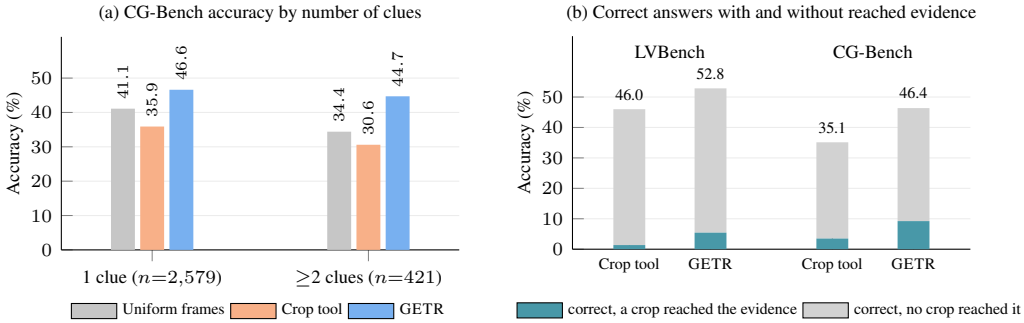


Figure 4: (a) CG-Bench accuracy for questions annotated with one clue and with two or more. (b) Accuracy split into correct answers for which some crop reached the evidence (IoGT > 0.5) and correct answers for which none did; totals above the bars. Crop tool: the untrained model given the crop tool.

minutes and evidence spans of a few seconds; against untrained hints + crop the LVBench gain is +13.9. It is smallest where the input already holds the answer: on VideoMME-Long (+1.2, 68.1 vs. 66.9) the subtitles carry most answers, and on HERBench-Lite (+1.5, 39.6 vs. 38.1) the 3-minute videos are already sampled every 0.35 s at twice the pixel budget of a crop. Sec. 4.4 shows the same trend within benchmarks. The prior-work rows in Table 1 use other backbones or closed models, other frame budgets, and supervised stages of up to 263K samples, so we do not rank against them; uniform-frame Qwen3-VL-8B already scores above most of them on LVBench and VideoMME-Long. The comparison that holds across rows is the gain of each tool agent over its own uniform-frame backbone: GETR improves on all four benchmarks, whereas LongVT-RL and EVA, trained with supervised tool traces, score below Qwen2.5-VL-7B with uniform frames (63.3 / 40.5 / 34.5 / 35.0) on three and two benchmarks respectively.

4.2 Evidence Grounding

Table 2 reports how often a crop reaches the annotated evidence on CG-Bench (Hit@IoGT: any crop; Recall@IoGT: final crop); Table 5 in Appendix A.8 gives the same measures on LVBench. The first two rows are the untrained Qwen3-VL-8B given the crop tool (`\cropvideo`), and given hints (`\gethint`) and the crop tool. Training raises Hit@IoGT from 7.1% to 16.8% and accuracy from 35.1% to 46.4% (the CG-Bench accuracies are reported also in Table 1). Coverage stays low in absolute terms because CG-Bench evidence spans a median 0.6% of the video, against 7.5% in the training data (Appendix A.3). **Accuracy hides grounding.** No crop reached the evidence for 80% of the correct answers of GETR. These answers are not necessarily guesses: for 94% of the questions whose evidence no crop reached, at least one of the 512 uniformly sampled frames lies inside the annotated span, so accuracy alone cannot show whether the policy used the evidence. **Reaching the evidence still helps.** A crop of GETR reached the evidence for 504 of the 3,000 questions (the 16.8% Hit@IoGT); on these it is correct 55.0% of the time, against 44.6% on the other 2,496, which together give its 46.4% overall. Figure 4b splits accuracy this way for both policies: 5.7 of the 11.3-point gain of GETR over the untrained crop policy (46.4 vs. 35.1) comes from the questions a crop reached.

Table 2: Evidence coverage on CG-Bench (%). Hit: any crop; Recall: final crop; both at IoGT > 0.5. Acc. for reference

Model	Acc.	Hit	Recall
Qwen3-VL-8B			
+ `\cropvideo`	35.1	7.1	6.5
+ `\gethint`	32.2	12.1	10.2
+ GETR	**46.4**	**16.8**	**16.7**

4.3 Ablations

First-crop reward. Scored against the hint windows, R_{loc} is met almost from the first step (mean 0.99 over steps 1–10; Fig. 5, dotted), so it barely varies within a rollout group and R_{iogt} carries the grounding signal. A variant whose R_{loc} is instead the IoU of the first crop with the annotated evidence (trained with four hints instead of two) keeps a graded start reward (0.27–0.31) and reaches higher final-crop coverage in training (0.60 vs. 0.25; Fig. 5). Its evaluation is confounded by a training-mask error and is reported in Appendix A.3. **Hint quality at inference.** Table 3 replaces the hints of GETR at inference on CG-Bench and LVBench subset, so accuracies differ from Table 1. The conditions are no hints, one or all hints replaced by random windows of the same length, the model's own hints, and the annotated evidence windows as hints (oracle). Hint quality matters, but modestly. From no

Table 3: Hints replaced at inference (accuracy, %; CG-Bench 600 and LVBench 500 questions; random conditions: mean of three seeds).

Hints	CG	LVB
None	43.0	53.2
All random	44.0	54.1
One random	45.6	54.9
GETR (own hints)	45.2	55.8
Oracle (evidence)	**47.8**	**60.2**

Table 4: Pilot LongVT-backbone CG-Bench component study (%). Time: frame timestamps in the crop response; Tool: share of questions with a tool call; Recall: final-crop Recall@IoGT.

Policy	Acc.	Tool	Recall
LongVT-RL ($R_{\text{acc}} + R_{\text{fmt}}$)	31.0	89.8	18.4
+ R_{iogt}	34.0	95.2	29.3
+ R_{iogt} + Time	35.6	97.8	43.1
+ R_{iogt} + Time + R_{loc}	**38.0**	100.0	**46.7**

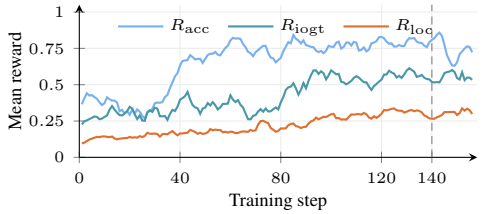


Figure 5: Five-step mean training rewards of GETR with the evidence-anchored R_{loc}; dashed line: evaluated step 140.

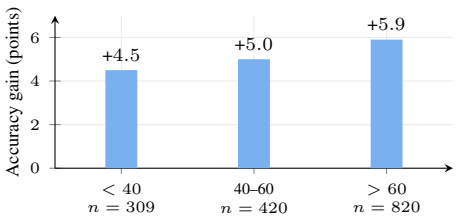


Figure 6: LVBench: GETR minus uniform-frame accuracy (points) by video duration (minutes).

hints to oracle hints, accuracy rises by 4.8 points on CG-Bench and 7.0 on LVBench. The model's own hints sit 2.2 and 2.6 points above no hints, leaving a gap of 2.6 and 4.4 points to oracle hints for better retrieval to close. Replacing all hints with random windows keeps about half of the benefit of real hints on CG-Bench (1.0 of 2.2 points) and a third on LVBench (0.9 of 2.6). Subset intervals are about ± 4 points, so only the no-hint versus oracle difference is clearly outside noise. **Rewards on a LongVT backbone.** Table 4 reports a pilot component study with LongVT-RL as the policy on CG-Bench: accuracy and final-crop Recall@IoGT rise together as R_{iogt}, the frame timestamps and R_{loc} are added.

4.4 Analysis

Accuracy by video length. Figure 6 plots the paired LVBench gain of GETR by video duration. The gain is positive in every bin and largest for videos longer than 60 minutes (+5.9 points, 95% CI 2.3–9.1, $n = 820$); for 40–60 minutes it is +5.0 (CI 0.5–9.8) and for videos under 40 minutes +4.5 (CI -0.7–9.7). On CG-Bench the gain is +8.3 for 20–40-minute videos (CI 6.0–10.8, $n = 1{,}655$) and +6.4 for 40–60 minutes (CI 2.2–11.2, $n = 456$). **Questions with several clues.** CG-Bench annotates 421 of its 3,000 questions with two or more clue intervals. The gain of GETR over uniform frames roughly doubles on these questions: +5.5 points with one clue (46.6 vs. 41.1) and +10.3 with two or more (44.7 vs. 34.4; 95% CIs 44.6–48.6 and 39.7–49.4; Figure 4a). This does not come from covering every clue: some crop reaches all annotated clues for only 8.8% of the multi-clue questions (16.3% with one clue). Appendix A.4 partitions the errors of GETR: on CG-Bench, 65.6% occur when neither a hint nor a crop reaches the clue, so most remaining errors come from where the policy looks rather than from how it reads the evidence.

5 Conclusion

We presented GETR, a guided exploration method that teaches a long-video policy where to look with RL alone, without supervised fine-tuning or tool-call traces. Hints retrieved offline from text summaries give the policy a few places to start, and two rewards score where it looked: its first crop for how precisely it selects the evidence, and its final crop for covering it. A crop tool alone does not help an untrained model; with this training, the same model and tool improve over uniform sampling, most where short evidence is spread through long videos. The method has limits that point to future work. The coverage reward scores one evidence span at a time, so questions with several scattered segments would benefit from rewarding their union. Coverage training improves where the policy looks, not how it reasons once the evidence is found. The hints depend on how well the question matches visual summaries, so they miss questions about spoken content, and search adds little when the uniform frames or the transcript already contain the answer; transcript-aware summaries and a policy that learns when not to search are natural next steps.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs, propose or refine hypotheses, and edit a research paper to improve readability. We did not use generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, or interpret results. We reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

Dan Ben-Ami, Gabriele Serussi, Kobi Cohen, and Chaim Baskin. Herbench: A benchmark for multi-evidence integration in video question answering. *arXiv preprint arXiv:2512.14870*, 2025.

Guo Chen, Yicheng Liu, Yifei Huang, Yuping He, Baoqi Pei, Jilan Xu, Yali Wang, Tong Lu, and Limin Wang. Cg-bench: Clue-grounded question answering benchmark for long video understanding. *arXiv preprint arXiv:2412.12075*, 2024.

Yi Chen, Yuying Ge, Rui Wang, Yixiao Ge, Lu Qiu, Ying Shan, and Xihui Liu. Exploring the effect of reinforcement learning on video understanding: Insights from seed-bench-r1. *arXiv preprint arXiv:2503.24376*, 2025.

Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, et al. Molmo2: Open weights and data for vision-language models with video understanding and grounding. *arXiv preprint arXiv:2601.10611*, 2026.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *ECCV*, 2024.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025a.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025b.

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, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *CVPR*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Zefeng He, Xiaoye Qu, Yafu Li, Siyuan Huang, Daizong Liu, and Yu Cheng. Framethinker: Learning to think with long videos via multi-turn frame spotlighting. *arXiv preprint arXiv:2509.24304*, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Dinesh Jayaraman and Kristen Grauman. Learning to look around: Intelligently exploring unseen environments for unknown tasks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1238–1247. IEEE, 2018.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. *arXiv preprint arXiv:2509.07969*, 2025.

Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, 2025a.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025b.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3318–3327. IEEE, 2025.

Wenqi Liu, Yunxiao Wang, Shijie Ma, Meng Liu, Qile Su, Tianke Zhang, Haonan Fan, Changyi Liu, Kaiyu Jiang, Jiankang Chen, Kaiyu Tang, Bin Wen, Fan Yang, Tingting Gao, Han Li, Yinwei Wei, and Xuemeng Song. Videotemp-o3: Harmonizing temporal grounding and video understanding in agentic thinking-with-videos. *arXiv preprint arXiv:2602.07801*, 2026.

Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, et al. Video-rag: Visually-aligned retrieval-augmented long video comprehension. *arXiv preprint arXiv:2411.13093*, 2024.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023.

Boris Meinardus, Hector Rodriguez, Anil Batra, Anna Rohrbach, and Marcus Rohrbach. Chrono: A simple blueprint for representing time in mllms. *arXiv preprint arXiv:2406.18113*, 2024.

Arsha Nagrani, Sachit Menon, Ahmet Iscen, Shyamal Buch, Ramin Mehran, Nilpa Jha, Anja Hauth, Yukun Zhu, Carl Vondrick, Mikhail Sirotenko, Cordelia Schmid, and Tobias Weyand. Minerva: Evaluating complex video reasoning. *arXiv preprint arXiv:2505.00681*, 2025.

Junwen Pan, Qizhe Zhang, Rui Zhang, Ming Lu, Xin Wan, Yuan Zhang, Chang Liu, and Qi She. Timesearch-r: Adaptive temporal search for long-form video understanding via self-verification reinforcement learning. *arXiv preprint arXiv:2511.05489*, 2025.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3982–3992. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. Videorag: Retrieval-augmented generation with extreme long-context videos. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2390–2401, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29118–29128, 2025.

Shijian Wang, Jiarui Jin, Xingjian Wang, Linxin Song, Runhao Fu, Hecheng Wang, Zongyuan Ge, Yuan Lu, and Xuelian Cheng. Video-thinker: Sparking “thinking with videos" via reinforcement learning. *arXiv preprint arXiv:2510.23473*, 2025a.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *ICCV*, 2025b.

Ziyang Wang, Jaehong Yoon, Shoubin Yu, Md Mohaiminul Islam, Gedas Bertasius, and Mohit Bansal. Video-rts: Rethinking reinforcement learning and test-time scaling for efficient and enhanced video reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025c.

Ziyang Wang, Honglu Zhou, Shijie Wang, Junnan Li, Caiming Xiong, Silvio Savarese, Mohit Bansal, Michael S. Ryoo, and Juan Carlos Niebles. Active video perception: Iterative evidence seeking for agentic long video understanding. *arXiv preprint arXiv:2512.05774*, 2025d.

Yuan Xie, Tianshui Chen, Zheng Ge, and Lionel Ni. Video-mtr: Reinforced multi-turn reasoning for long video understanding. *arXiv preprint arXiv:2508.20478*, 2025.

Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, and Lidong Bing. Longvt: Incentivizing "thinking with long videos" via native tool calling. *arXiv preprint arXiv:2511.20785*, 2025.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. *arXiv preprint arXiv:2512.02425*, 2025.

Haoji Zhang, Xin Gu, Jiawen Li, Chixiang Ma, Sule Bai, Chubin Zhang, Bowen Zhang, Zhichao Zhou, Dongliang He, and Yansong Tang. Thinking with videos: Multimodal tool-augmented reinforcement learning for long video reasoning. *arXiv preprint arXiv:2508.04416*, 2025a.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*, 2024a.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. *arXiv preprint arXiv:2505.18079*, 2025b.

Yaolun Zhang, Xiaogeng Liu, and Chaowei Xiao. Metaagent: Automatically constructing multi-agent systems based on finite state machines. *arXiv preprint arXiv:2507.22606*, 2025c.

Yaolun Zhang, Ruohui Wang, Jiahao Wang, Yepeng Tang, Xuanyu Zheng, Haonan Duan, Hao Lu, Hanming Deng, and Lewei Lu. Eva: Efficient reinforcement learning for end-to-end video agent. *arXiv preprint arXiv:2603.22918*, 2026.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024b.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Jialong Zuo, Yongtai Deng, Lingdong Kong, Jingkang Yang, Rui Jin, Yiwei Zhang, Nong Sang, Liang Pan, Ziwei Liu, and Changxin Gao. Videolucy: Deep memory backtracking for long video understanding. *arXiv preprint arXiv:2510.12422*, 2025.

A APPENDIX

This appendix gives the details needed to reproduce GETR and the analyses that support the main results. Appendix A.1 extends the related work of Sec. 2. Appendices A.2–A.10 cover the method and setup: training and evaluation data and the supervision budgets of prior agents (A.2), how the hints are built, with example chunk summaries and retrieval queries (A.5), the answer and format rewards with the judge prompt (A.6), hyperparameters, frame budgets, baselines and metrics (A.7), and the inference procedure (A.10). Appendices A.3, A.4 and A.8 give additional results: why evidence coverage learned in training transfers to HERBench-Lite but less to CG-Bench and LVBench, a partition of GETR's errors by whether a hint or a crop reached the evidence, and the LVBench counterpart of Appendices A.11 and A.12 show example trajectories, including successes and one case for each failure type. Annotated evidence intervals are used only in the training rewards and in these analyses; the policy never sees them. Table 6 collects the notation.

A.1 EXTENDED RELATED WORK

This section expands on Sec. 2.

Video MLLMs. Multimodal large language models extend image-based reasoning to video by processing sampled frames, and show strong visual reasoning and instruction following across VideoQA benchmarks (Bai et al., 2025b; Zhu et al., 2025; Zhang et al., 2024b; Bai et al., 2025a). Their inference interface is passive: the model reasons over a fixed set of uniformly sampled frames, so its predictions depend on which segments happen to be observed rather than on deliberate inspection of relevant content. EgoSchema (Mangalam et al., 2023) makes this concrete with the temporal certificate, the minimum footage needed to answer a question, and shows that long certificates defeat frame sampling. This dependence on sampling motivates methods that inspect long videos adaptively, driven by the query.

Frame selection and retrieval. Training-free frame selection chooses which frames a fixed model sees. BOLT (Liu et al., 2025) scores frames by query–frame similarity and samples them by inverse transform sampling; AKS (Tang et al., 2025) selects keyframes by trading off relevance against temporal coverage of the video. Both improve over uniform sampling without training. AKS's *coverage* is spread over the timeline, whereas the coverage we reward is coverage of the labeled evidence. Retrieval methods turn videos into searchable data and reason over the selected segments. VideoRAG-style methods build textual or multimodal retrieval indices (Ren et al., 2026; Luo et al., 2024), while scene- or memory-based approaches such as WorldMM (Yeo et al., 2025) introduce more structured representations for long-video search. DVD (Zhang et al., 2025b) builds a video database at several levels of detail and provides Global Browse, Clip Search, and Frame Inspect tools, which lets an LLM search a long video without processing every frame; building such a database is costly and becomes harder for large video collections. Frame selection and most retrieval methods choose what the model sees in one step. `\gethint` provides candidates in the same way, but in GETR they only seed a search that the policy continues crop by crop.

Tool-using video agents. A growing body of work treats long-video understanding as an interactive task in which models inspect video content through repeated tool calls. LongVT (Yang et al., 2025) adds tools for selecting times in a video and trains them in three stages: SFT, RL, and RFT. FrameThinker (He et al., 2025) learns active frame selection with SFT followed by GRPO. EVA (Zhang et al., 2026) plans before viewing and repeats a summarize–plan–act–reflect cycle; AVP (Wang et al., 2025d) repeats a plan–observe–reflect cycle, with main results using closed models; and VideoLucy (Zuo et al., 2025) uses a layered memory that lets it return to earlier evidence. These results show that choosing what to inspect can work better than sampling frames at fixed intervals.

Tool use and RL for reasoning. Beyond video, language models perform better on interactive tasks when they can call tools while reasoning (Qin et al., 2023; Feng et al., 2025a; Li et al., 2025a), and groups of agents and automatically built workflows can solve more complex tasks (Zhang et al., 2025c; 2024a). RL and preference-based training are now widely used to improve reasoning models (Guo et al., 2025; Jaech et al., 2024; Shao et al., 2024; Ethayarajh et al., 2024). For video,

RL has been used to find events in time, understand changes in space and time, and support several rounds of interaction (Feng et al., 2025b; Li et al., 2025b; Chen et al., 2025; Wang et al., 2025a).

RL with grounding-aware rewards. This line of work is moving quickly, and several recent methods are close to ours. Video-MTR (Xie et al., 2025) trains a multi-turn segment-selection policy with RL alone, rewarding each turn by the relevance of its frames to annotated key segments, gated by answer correctness. VideoTemp-o3 (Liu et al., 2026) rewards grounding with an IoU term that is penalized below a threshold to discourage arbitrary localizations. TimeSearch-R (Pan et al., 2025) adds a reward for whether the searched frames alone suffice to answer. Video-RTS (Wang et al., 2025c) shows that a video reasoning policy can learn to use tools with RL, without an initial supervised stage and with much less training data than supervised methods. GETR belongs to this family but differs in its training signal. Rather than imitating tool traces and rewarding the answer or the IoU of a predicted span, it rewards where the policy looked: where its first crop falls and whether its final crop covers the labeled evidence. It is trained without a supervised stage, and for every trajectory we report whether the evidence was reached.

Temporal grounding, active perception, and evidence. Temporal grounding predicts the start and end of a queried moment and is evaluated by IoU (Lei et al., 2021; Clark et al., 2026), because over-reaching is an error there. An inspecting agent has a different goal, which is to see enough evidence to answer, so we score coverage of the evidence span instead (Sec. 3.2). Choosing the next window from what has already been seen is an instance of active perception (Bajcsy, 1988; Jayaraman & Grauman, 2018); GETR applies it along the time axis, with the hints as the starting point. The same notion of evidence underlies clue-grounded evaluation: CG-Bench (Chen et al., 2024) annotates the intervals that support each answer, and MINERVA (Nagrani et al., 2025) finds that most failures of current models are temporal-localization errors. We use this evidence as a training signal rather than only for evaluation.

A.2 Model and training

Datasets. *Training.* We use the RL split of the LongVT-Parquet dataset (Yang et al., 2025), 1,668 examples drawn from HACS and Ego4D-NAQ. Each example is a tuple of a video, an open-ended question that requires evidence from a specific time, its answer, and the annotated evidence interval used by R_{iogt} and R_{loc}. Training answers are scored by the judge in Appendix A.6. No tool-call traces are used. By comparison, prior tool-using agents fine-tune before RL. LongVT fine-tunes on 247.9K samples, including 19.2K distilled tool-call trajectories, and adds 15.4K self-distilled trajectories for rejection-sampling fine-tuning (Yang et al., 2025). EVA fine-tunes on 10K distilled agent trajectories and 11K preference-labelled ones (Zhang et al., 2026). *Evaluation.* LVBench (Wang et al., 2025b) (103 videos, avg. 68 min; 1,549 MCQs) and VideoMME-Long with subtitles (Fu et al., 2025) (300 videos, 30–60 min; 900 MCQs) test long-form reasoning with mostly single-evidence questions. CG-Bench (Chen et al., 2024) (1,219 videos, 10–80 min; 12,129 QA pairs) adds clue-grounded evaluation, which checks whether answers are supported by annotated temporal clues rather than language priors. We evaluate on its 3,000 clue-annotated questions. HERBench-Lite (Ben-Ami et al., 2025) is the most direct test of multi-evidence aggregation: every question requires fusing at least three non-overlapping evidence cues from distinct segments. We use the released Lite split (2,000 five-way MCQs, avg. 3 min).

A.3 Coverage transfer

In training, the final crop of GETR covers the evidence in 25% of rollouts. At evaluation, the final crop reaches the evidence for 76% of the 480 HERBench-Lite questions with clue timestamps, 16.7% of CG-Bench questions and 11.6% of LVBench questions. The difference is the size of the target relative to the video. In the training data, videos have a median length of 3.4 minutes and the evidence spans a median 7.5% of the video. The evidence spans a median 3.0% of the video on HERBench-Lite (3-minute videos), 0.6% on CG-Bench (25 minutes) and 0.5% on LVBench (61 minutes). Where the evidence is a comparable fraction of the video, as on HERBench-Lite, the coverage learned in training transfers. Where it is ten times smaller, as on CG-Bench and LVBench, a crop of the same length rarely overlaps more than half of it.

A.4 FAILURE CASES

Figure 7 partitions the wrong answers of GETR (LVBench: 731; CG-Bench: 1,609). On LVBench, neither a hint nor a crop reaches the clue in 82.5% of the errors, a hint contained the clue but no crop reached it in 4.0%, a crop reached the clue and the answer is still wrong in 6.2%, and 7.3% end without a closed answer. On CG-Bench the shares are 65.6%, 8.3%, 12.1% and 14.0%. On both benchmarks, most errors occur because the search never reaches the evidence, not because the policy misreads it once found. Appendix A.12 gives one trajectory for each category (F3–F5).



Figure 7: Shares of the wrong answers of GETR by failure type (LVBench: 731; CG-Bench: 1,609). Ground-truth clue windows are used for analysis only.

A.5 TIMESTAMP-ALIGNED SUMMARIES

Timestamp-Aligned Summary Retrieval Examples

```
1. qid=14, video_uid=3ZpmN3-2-Zo
Question: In the video, what color is the vehicle driving from the left in the
     animation?
%
- Candidate 1: [600.0, 630.0]
  Summary: The video begins with a toddler walking down a hallway toward the camera,
     followed by several children and adults in pajamas.
The scene then shifts to a living room where the children, still in pajamas, are
     playing behind a white sheet, which they pull down to reveal themselves, causing
     the adults to laugh and cover their mouths in surprise.
%
- Candidate 2: [0.0, 30.0]
  Summary: The video segment begins with a close-up of flames, then transitions to a
     festive, animated living room scene featuring a fireplace, a snowman, and a
     Christmas tree.
The camera zooms in on a snow globe containing a decorated house, which then opens to
     reveal the exterior of the house, where a snowman and an RV appear.
The scene shifts to a family of seven, dressed in matching green holiday sweaters and
     elf hats, posing in front of their decorated home.
The video concludes with a small white dog wearing an elf hat jumping onto a pile of
     presents near the Christmas tree.
%
- Candidate 3: [240.0, 270.0]
  Summary: The video segment begins in a dimly lit room, where a child is seen moving
     around on a bed.
As the lighting improves, the child, wearing festive pajamas, is shown smiling and
     looking at the camera while being gently interacted with by an adult.
The child remains on the bed, displaying happy expressions throughout the clip.
%
- Candidate 4: [2700.0, 2730.0]
  Summary: The video segment features a woman in a plaid shirt speaking directly to the
      camera in a room with a lit Christmas tree in the background.
She gestures with her hands, including showing her wrist tattoo, and at one point holds
      up a photograph of a group of people.
The segment concludes with a black screen displaying the text "the end." followed by a
     final screen showing the text "One Life Live It" over the photograph.
%
2. qid=38, video_uid=BV1s4411b75P
Question: In the video, how many people are flying in wing suits when landing from high
      altitude to an island in the ocean?
Answer: 3
%
- Candidate 1: [0.0, 30.0]
  Summary: The video segment opens with aerial shots of three wingsuit flyers
     descending over a body of water with distinctive heart-shaped islands, one flyer
     leaving a white smoke trail.
The perspective shifts to first-person views from the flyers, showing their faces and
     the sky, before returning to wide shots of the group flying in formation over the
     islands.
```

```
The scene then transitions to a different location, showing a single wingsuit flyer
     navigating close to a steep, rocky mountain cliff face, with a valley and town
     visible below.
%
- Candidate 2: [60.0, 90.0]
  Summary: The video segment shows a person engaged in wingsuit flying over mountainous
      terrain.
Initially, the perspective is from a first-person view as the flyer glides over a
     valley with forests and a town below, then transitions to a third-person view
     showing the flyer navigating steep, snow-dusted cliffs.
The flyer, wearing a red and black suit with a helmet and goggles, is seen preparing
     and launching from a high vantage point, with spectators visible in the background.

%
- Candidate 3: [270.0, 300.0]
  Summary: The video segment shows a person in a black wingsuit, with a GoPro camera
     mounted on their helmet, flying through the sky.
They are being carried or towed by another person in a yellow and pink wingsuit, who is
      visible in the background.
The camera perspective remains fixed on the first person, capturing their flight and
     the surrounding landscape below.
%
- Candidate 4: [180.0, 210.0]
  Summary: The video segment shows a person in a wingsuit preparing for and executing a
      wingsuit flight from a high mountain ledge.
The sequence begins with a view of a valley town below, followed by the flyer launching
      off the cliff, gliding through the air with the rugged mountain terrain and
     distant snow-capped peaks visible.
The perspective shifts between a first-person view from the flyer's helmet and a third-
     person view showing the flyer's body in flight, concluding with the flyer landing
     on a grassy slope.
```

Retrieval query q'. Training (LongVT-Parquet, open-ended): q' is the question text, e.g.

> q' = "After the man in a pink shirt and white cap drops his piece of paper at the starting platform, what does he wave to start the kayak race?"

Evaluation (multiple choice): q' is the question concatenated with all options, e.g.

> q' = "What instruments do the prince and princess play respectively during the performance? (A) The prince plays the guitar and the princess plays the drum set. (B) The prince plays the piano and the princess plays the violin. (C) The prince plays the violin and the princess plays the piano. (D) The prince plays the drum set and the princess plays the guitar."

A.6 Standard Accuracy and Format Rewards

We use the standard answer-correctness and output-format rewards

$$R_{\text{acc}} = \begin{cases} \mathbb{1}[\hat{a} = a^\star], & \text{multiple choice,} \\ J(q, \hat{a}, a^\star), & \text{open-ended,} \end{cases} \qquad R_{\text{fmt}} = \mathbb{1}[\text{valid trajectory format}], \tag{10}$$

where $J \in \{0, 1\}$ is the strict judge below. A valid trajectory alternates valid `<tool_call>` block and ends with a valid `<answer>` block.

Open-Ended Answer Correctness Prompt

```
You are a strict evaluator assessing answer correctness.
You must output 1 for fully correct answers and 0 for any other case.
You will receive the question, the ground truth answer, and the model prediction.
%
# Input
Question:
```
{question}
```
%
Ground Truth Answer:
```
```
```

```
{answer}
```
Model Prediction:
```
{prediction}
```
%
Evaluation Rules
- For multiple-choice questions: Score 1 if the predicted answer matches the ground
    truth answer, it can be directly in option letters or the content of the options.
- For open-ended questions:
  * Score 1 if the prediction matches the answer semantically, it can be in different
    format.
  * Score 0 for partially correct answers or answers with extra incorrect information,
    even if the reasoning process is correct.
- Ignore minor differences in formatting, capitalization, or spacing since the model
    may explain in a different way.
- Treat numerical answers as correct if they match within reasonable precision
- For questions requiring units, both value and unit must be correct
%
Strict Output format
1 or 0
```

### A.7 Experimental Setup

**Model and training.** GETR uses Qwen3-VL-8B-Instruct (Bai et al., 2025a) as its base model and trains it with GRPO, without supervised fine-tuning. We use the LongVT-RL settings (Yang et al., 2025): a global prompt batch of 16, 16 rollouts per prompt, and a learning rate of $10^{-6}$. Each rollout may make up to five tool calls within six assistant turns and 8,192 response tokens. During training, the initial uniform pass contains up to 256 frames sampled at about 1 fps, at 25,088 pixels per frame; crops return 8 frames at the same 25,088 pixels. At evaluation, the initial pass contains 512 frames at 50,176 pixels and crops return 8 frames at 25,088 pixels, so crop frames have half the pixel budget of the initial frames. We validate every 20 steps on 114 held-out questions and select the checkpoint with the highest validation accuracy; a moving-average plateau criterion was in place, but all runs reached the three-day wall-clock limit first. Training uses one node with eight H100 GPUs; Appendix A.9 lists the remaining settings.

**Baselines.** The controlled comparisons use the same Qwen3-VL-8B-Instruct backbone, benchmark inputs, and answer scorer. Text only receives the question, options, and available subtitles. Uniform sampling adds the initial video frames. The untrained crop row permits `\cropvideo`; the untrained hints-and-crop row also permits `\gethint`. These tool rows use different prompts and generation limits, so the table reports their observed behavior rather than attributing every difference to tool access alone.

**Metrics.** We report Qwen2.5-72B-Instruct judge-scored multiple-choice accuracy on all four benchmarks. On LVBench and the clue-annotated CG-Bench subset, we also report temporal evidence coverage: Hit@IoGT is the share of questions for which any valid crop covers more than half of an annotated evidence interval; Recall@IoGT applies the same test to the final valid crop. The latter is the target of the terminal coverage reward. A correct answer without a reaching crop does not establish that the policy inspected the annotated clue.

### A.8 Evidence Coverage on LVBench

Table 5 gives the LVBench counterpart of Table 2, with the same measures and thresholds.

Table 5: Accuracy and evidence coverage on LVBench (%). Hit: any crop; Recall: final crop; both at IoGT $> 0.5$.

| Model | Acc. | Hit | Recall |
|---|---|---|---|
| Crop, no training | 46.0 | 2.1 | 2.1 |
| Hints + crop, no training | 38.9 | 9.0 | 8.5 |
| GETR | **52.8** | **12.5** | **11.6** |
```

A.9 Training and Inference Details

We use **Qwen3-VL-8B-Instruct** (Bai et al., 2025a) as the base model and train it directly with GRPO rather than performing supervised fine-tuning on human- or model-generated tool traces. The localization prior module reuses the same Qwen3-VL-8B-Instruct to produce timestamp-aligned chunk summaries offline, which are then encoded with a text embedding model and compared with the retrieval query. The summaries are computed once per video and cached. Unless noted otherwise, the initial video observation uses up to 256 frames at `max_pixels`=25,088, and each `crop_video` call returns up to 8 frames at the same resolution. We allow up to five tool calls per trajectory, followed by one turn to answer (six assistant turns), with a maximum response length of 8,192 tokens. Training is performed on a single 8×H100 node with global prompt batch size 16, 16 sampled responses per prompt, one PPO epoch, policy learning rate 10^{-6}, and KL coefficient 10^{-3}.

A.10 Inference

At test time, the policy begins by calling `get_hint()` to retrieve a unordered set of candidate temporal windows $\mathcal{K} = \{(t_s^k, t_e^k, c_k)\}_{k=1}^{K}$. It then issues an initial `crop_video`(p_s, p_e) call, typically starting from the highest-confidence candidate. The policy then enters an iterative interaction loop. At each step, based on the retrieved frames and their timestamps, it may call `crop_video` again with a narrower interval to refine the current window, a wider or shifted interval to inspect nearby context, or an interval from another localization prior candidate. The exploration is not restricted to $\mathcal{K}$, and the policy may inspect arbitrary temporal regions if needed. This process continues until sufficient evidence is gathered, at which point the trajectory terminates with a final answer $\hat{a} \in \mathcal{A}$ in the `<answer>` block, or until the turn or token budget is exhausted. No ground-truth temporal annotations are available at test time; the policy relies entirely on the learned exploration strategy.

Table 6: Notation.

$\mathcal{V}$, T	video and its duration (s)
q, $a^\star$, $\hat{a}$	question, ground-truth answer, predicted answer
q'	retrieval query built from q (Sec. 3.1)
$C = \lceil T/30 \rceil + 1$	number of 30-s chunks
(t_s^i, t_e^i), σ_i	start/end of chunk i, its text summary
$e(\cdot)$, s_i	text-embedding model; rescaled similarity of chunk i to q'
$\mathcal{K} = \{(\hat{s}_k, \hat{e}_k)\}_{k=1}^{K}$	hint windows and confidences returned by `get_hint()`; $K \le K_{\max} = 4$
τ_{abs}, $\tau_{\text{frac}} = 0.6$	absolute / relative score thresholds (Eq. 3)
$c = (t_s, t_e)$, $c^{(1)}$, $c^\star$	a crop window; first crop; final crop of a trajectory
$g = [g_s, g_e]$	ground-truth evidence segment
$n = 8$	frames returned per `crop_video` call
τ, π_θ, $N = 16$	trajectory, policy, rollouts per prompt
$R_{\text{acc}}, R_{\text{fmt}}, R_{\text{iogt}}, R_{\text{loc}}, \alpha = 0.2$	reward terms and prior weight (Eq. 6)
m_i, $M_{i,t}^{\text{resp}}$	completion mask; response-token mask
μ_g, σ_g, $A_{i,t}^{\text{final}}$	group statistics over complete rollouts; masked advantage
$\epsilon_{\text{clip}} = 0.2$, $\beta = 10^{-3}$	PPO clip range; KL coefficient (Eq. 9)

Question: What instruments do the prince and princess play respectively during the performance? **Correct:** (D) The prince plays the drum set and the princess plays the guitar. **Before RL (Qwen3-VL-8B-Instruct, same tools):**

```
User:      [question + options A-D]
Assistant: <tool_call> get_hint() </tool_call>
Tool:      Candidate windows: 1.[90-120s]  2.[2430-2460s]
           3.[1140-1170s]  4.[1200-1230s]
Assistant: <tool_call> crop_video(90,120) </tool_call>
Tool:      [8 frames from 90-120s]
Assistant: <tool_call> crop_video(180,210) </tool_call>
Tool:      [8 frames from 180-210s]
Assistant: [trajectory ends without a final <answer> block]
```

Incomplete trajectory ($m_i = 0$), masked out of the gradient (Eq. equation 8). **After RL (GETR):**

```
User:      [same question + options]
Assistant: <tool_call> get_hint() </tool_call>
Tool:      [same 4 candidate windows]
Assistant: <tool_call> crop_video(90,120) </tool_call>
Tool:      [8 frames from 90-120s]
Assistant: <tool_call> crop_video(180,210) </tool_call>
Tool:      [8 frames from 180-210s]
Assistant: The frames show the prince at a drum set holding drumsticks
           and the princess playing a guitar. <answer> (D) </answer>
```

Outcome: correct ($R_{\text{acc}} = 1$, $R_{\text{fmt}} = 1$).

Figure 8: The same question before and after RL. The untrained policy calls the tools but never commits to an answer; GETR reaches the evidence and answers.

A.11 Qualitative Example from an Earlier Checkpoint

Figure 9 shows a trajectory from an earlier GETR checkpoint: the policy requests ranked hint windows, inspects the top candidate, and answers from the observed segment. Section A.12 gives five trajectories of the reported checkpoint.

A.12 Audited Example Trajectories

The five cases below come from the GETR evaluation. They show the executed tool calls in order and retain the model's answer. The ground-truth evidence window was *not* shown to the policy; it is included here to explain the measured IoGT category. Each valid crop returned eight frames. The selected frames in Figure 10 are drawn from those returned crops.



Figure 10: Frames returned to the policy in the final crops of LVBench-2387 (about 7217 s), CG-Bench-48 (about 147 s), and CG-Bench-4928 (about 489 s). The first two answers are correct; the third is wrong despite reaching the annotated clue.

Success: LVBench-2387, search beyond both hints

Question: What is the last major event? Correct option: (C) the prince and the mermaid hold a wedding.

1. `get_hint()` returns ranked windows 3150–3180 s and 540–570 s.
2. `crop_video(3150,3180)` returns the mermaid and merman. The assistant says this "does not depict the final event" and decides to inspect a later segment.
3. `crop_video(7200,7230)` returns the wedding scene. The assistant answers (C), which the judge marks correct.

Annotation for analysis: 7177–7225 s. The first crop misses; the final crop covers 52.1% of the annotated span (IoGT), so Recall@IoGT is one.

Success: CG-Bench-48, refine a nearby hint

Question: How many holes are punched in the golden square piece of wood? Correct option: (G) one.

1. `get_hint()` returns ranked windows 30–60 s and 120–150 s.
2. `crop_video(30,60)` misses the annotated evidence.
3. `crop_video(130,160)` inspects the drill press. The assistant answers "(G) one," which the judge marks correct.

Annotation for analysis: 145–170 s. The final crop reaches the clue with IoGT 0.60.

F3: CG-Bench-6662, hints and search both miss

Question: After the discussion of intermediate Markdown features, which proposed topics are incorrect? Correct option: (F) items 1, 2, and 3.

1. `get_hint()` returns 30–60 s and 0–30 s.
2. The assistant executes `crop_video` on 30–60 s, 60–90 s, and 90–120 s.
3. It concludes that only music is incorrect and answers (D) "Only 2"; the judge marks the answer wrong.

Annotation for analysis: 1630–1639 s. Neither hint nor any executed crop overlaps the clue, giving Hit@IoGT and Recall@IoGT of zero.

F4: CG-Bench-3894, useful hint skipped

Question: What words appear on the tablecloth? Correct option: (G) MUMU'S KITCHEN.

1. `get_hint()` returns 330–360 s first and 240–270 s second.
2. The assistant executes `crop_video(330,360)` and then `crop_video(360,390)`. It never inspects the second hint.
3. It answers (A) MOMO'S KITCHEN; the judge marks the answer wrong.

Annotation for analysis: 252.7–263.7 s. The second hint contains the entire clue, but both executed crops miss it.

F5: CG-Bench-4928, evidence reached but answer wrong

Question: How many times does Li Yongzhou add ingredients to the stew? Correct option: (D) three.

1. `get_hint()` returns 420–450 s and 510–540 s.
2. The assistant executes `crop_video` on 420–450 s, 450–480 s, and 480–510 s.
3. It counts nine additions and answers (C) nine; the judge marks the answer wrong.

Annotation for analysis: 488–491 s. The final crop fully contains the clue (IoGT 1.0), so both Hit@IoGT and Recall@IoGT are one despite the wrong answer.

Question What does the vlogger do after finish eating the sandwiches? (A) Get work done in a coffee shop (B) Work out in a gym (C) Enjoy the sun outside the house (D) Talk to the audience in the car



GT: C (accuracy) **Relevant Segment** [600-630] (recall)

Output

```
<question> What does the vlogger do after finish eating the
sandwiches? (A) .. (B) .. (C) .. (D) .. </question>

<tool_call> get_hint () </tool_call>

<tool_result> Candidate windows ranked by relevance:
1. [600.0s, 630.0s]
2. [0.0s, 30.0s]
3. [570.0s, 600.0s]
4. [270.0s, 300.0s] </tool_result>

<tool_call> crop_video(start_time=600.0, end_time=630.0)
</tool_call>

<answer> C </answer>
```

Question What are the people holding the yellow ropes trying to do at the place that has trees with blooms? (A) They are trying to lasso the dog (B) They are trying to lasso the cat (C) They are trying to lasso the Panda (D) They are trying to lasso the yak



GT: D (accuracy) **Relevant Segment** [4180-4188] (recall)

Output

```
<question> What are the people holding the yellow ropes trying
to do at the place that has trees with blooms?  (A) .. (B) ..
(C) .. (D) .. </question>

<tool_call> get_hint () </tool_call>

<tool_result> Candidate windows ranked by relevance:
1. [4170.0s, 4200.0s]
2. [4200.0s, 4230.0s]
3. [4230.0s, 4260.0s]
4. [4140.0s, 4170.0s] </tool_result>

<tool_call> crop_video(start_time=4230.0, end_time=4260.0)
</tool_call>

<tool_call> crop_video(start_time=4260.0, end_time=4290.0)
</tool_call>

<tool_call> crop_video(start_time=4180.0, end_time=4230.0)
</tool_call>

<answer> D </answer>
```

Question What is the last major event? (A) The merman kills the witch (B) The prince and another girl fall in love (C) The prince and the mermaid hold a wedding (D) The mermaid dies and transforms into bubbles



GT: C (accuracy) **Relevant Segment** [7177-7225] (recall)

Output

```
<question> What is the last major event?
(A) .. (B) .. (C) .. (D) .. </question>

<tool_call> get_hint () </tool_call>

<tool_result> Candidate windows ranked by relevance:
1. [3150.0s, 3180.0s]
2. [540.0s, 570.0s]
3. [900.0s, 930.0s]
4. [6750.0s, 6780.0s] </tool_result>

<tool_call> crop_video(start_time=3150.0, end_time=3180.0)
</tool_call>

<tool_call> crop_video(start_time=7130.0, end_time=7160.0)
</tool_call>

<tool_call> crop_video(start_time=7200.0, end_time=7230.0)
</tool_call>

<answer> C </answer>
```

Figure 9: Example trajectory from an earlier GETR checkpoint, showing the question, candidate windows, crop call, and answer.