TEACHING VIDEO AGENTS WHEN AND WHERE TO LOOK WITH COUNTERFACTUAL INFORMATION GAIN

Anonymous authors
Paper under double-blind review

ABSTRACT

Video agents are increasingly equipped with tools that actively retrieve visual evidence during reasoning. However, video retrieval can be costly, as each retrieved video evidence can introduce thousands of context tokens, and noisy, irrelevant frames can hinder model performance. In this work, we focus on developing efficient video agents that make intelligent tool decisions about *when to look* (whether calling a tool is necessary) and *where to look* (what video interval to retrieve to maximize relevant information). Existing reinforcement learning frameworks for video agents typically use outcome rewards or dense turn-level rewards, which encourage spurious tool calls and struggle to learn stopping conditions and temporal localization. To address this challenge, we introduce Counterfactual Information Gain Policy Optimization (`CIGO`), an RL framework that incentivizes tool calls based on the model's intrinsic information gain compared to directly answering the question. To accurately attribute information gain to the tool calls, we sample two types of counterfactual actions branching from the same reasoning prefix: (1) thinking and answering directly, and (2) issuing a single tool call before answering. By using this counterfactual construction, the model learns the stopping condition, while variance within the tool-call subgroup grounds localization. Extensive experiments across multiple benchmarks demonstrate improved overall performance on complex video reasoning tasks compared to existing agents. Beyond aggregate performance, `CIGO` learns more adaptive tool use and retrieves higher-quality visual evidence than existing video agents.

1 INTRODUCTION

Long video understanding is an important capability for multimodal LLM assistants, due to the prevalence of long video sources in daily life, including lectures Hu et al. (2026), movies Wang et al. (2025b), and camera recordings Grauman et al. (2022). Recent works have developed video agents that can actively retrieve visual evidence on demand during reasoning, leading to improved performance in visual grounding and multimodal reasoning Zeng et al. (2026a); Xie et al. (2025a); Zhang et al. (2026a); Yang et al. (2026a). However, developing scalable long video understanding agents is challenging due to the substantial cost of visual retrieval Qiu et al. (2026b); Wang et al. (2026). Video frames retrieved from tools can add thousands of tokens to agent context windows, making retrieval of irrelevant information particularly costly Chen et al. (2024). While question-critical visual evidence can be helpful to agents, an irrelevant visual retrieval can consume large amounts of an agent's context window and degrade reasoning performance Wu et al. (2025); Sharma et al. (2024). Therefore, agents must learn to request additional visual evidence only when useful.

Video agents engaging in efficient evidence-seeking behavior must make decisions along two coupled dimensions: (1) *When to look*: given the existing visual evidence, the agent must determine whether the current evidence is sufficient to answer a query or if a new retrieval turn is needed, avoiding redundant calls. (2) *Where to look*: if more evidence is needed, the agent must select a video interval that is likely to provide useful information for answering the question. We observe that existing video agents often exhibit retrieval behavior that does not adapt to the evidence available for each question. As illustrated in Figure 1, existing agents may fail to retrieve when additional evidence is needed, or continue to retrieve unnecessarily when current evidence is already sufficient.

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.

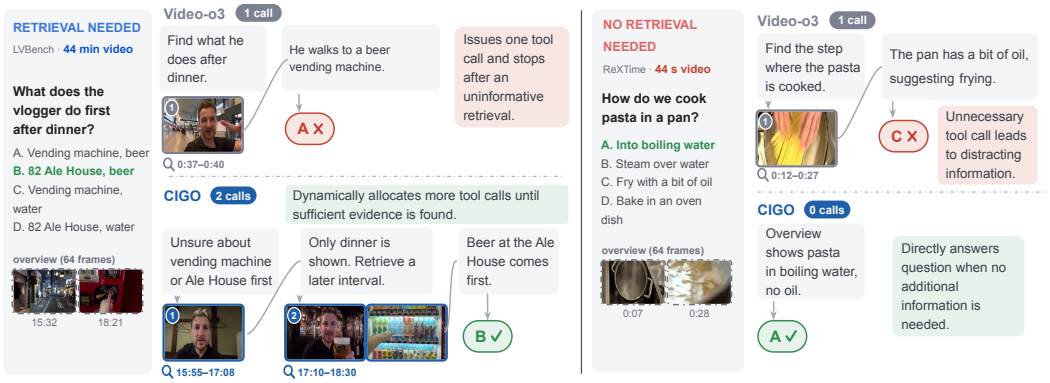


Figure 1: Counterfactual Information Gain Policy Optimization (CIGO) teaches video agents *when* and *where* to retrieve visual evidence. Left: an uninformative retrieval call can lead an agent to answer questions prematurely without enough evidence, whereas a CIGO-trained agent retrieves information until useful evidence is found. Right: when an agent already has sufficient evidence, an unnecessary retrieval can lead to failure, while CIGO-trained agent chooses not to retrieve and answers directly.

Existing methods typically train video agents using reinforcement learning (RL) with verifiable outcome rewards Zeng et al. (2026a) (often with GRPO Shao et al. (2024)) or dense turn-level rewards. Outcome rewards can provide ambiguous credit to retrieval decisions, as the same terminal reward can reward trajectories containing irrelevant tool calls or guesses that happen to be correct without visual grounding. While dense rewards provide finer-grained supervision, these rewards typically evaluate tool calls against external targets (e.g., annotations, LLM judges), rather than directly measuring whether the agent *should* have made the tool call from its current state. What prior objectives have not explicitly measured is the *counterfactual value of a tool call*: whether making one additional video retrieval leads to a better answer than stopping and answering from the current state.

A central challenge in training video agents is, therefore, to develop RL methods that assign credit to retrieval decisions based upon their marginal value to the agent at the current state. To address this challenge, we propose Counterfactual Information Gain Policy Optimization (CIGO), an RL framework that teaches video agents *when to look* and *where to look* during retrieval. Our key idea is to reward a retrieval tool call based on the relative information gain of executing it compared to directly answering the question from the same reasoning prefix. To assess the effect of tool calls, we compare two types of counterfactual actions branching from the same reasoning prefix, sampled from a replay buffer: (1) thinking and answering directly, and (2) issuing a single tool call before answering, teaching the model to decide between continuing to look and stopping. Comparing retrieval against direct answering teaches the model whether further retrieval is worthwhile, while comparing alternative retrieval actions teaches the model how to localize video intervals with useful evidence. In this way, video agents trained with CIGO jointly learn *when* and *where* to look.

Across six video understanding datasets spanning question answering and temporal grounding, our method outperforms competitive baselines. Beyond aggregate performance, we find that our agent learns more adaptive tool-use behavior and retrieves higher-quality visual evidence compared to existing methods. Moreover, we observe that the model develops genuine exploratory retrieval behaviors that optimize confidence before committing to an answer.

- We introduce Counterfactual Information Gain Policy Optimization (CIGO), an RL framework that branches alternative rollouts from the same reasoning prefix to evaluate each retrieval tool call by the marginal value of the visual evidence it provides.
- We present CIGO-8B, a video agent that reasons efficiently with efficient tool retrieval, calling tools only when the available evidence is insufficient and using at most three calls per question.
- Across six video understanding benchmarks, CIGO-8B achieves competitive performance, including the best result on Video-MMMU (65.9%); beyond aggregate performance, it learns more adaptive tool use and retrieves higher-quality visual evidence than existing video agents.

2 Related Work

2.1 Agentic Long Video Understanding

Prior work reduces the cost of long video understanding through hierarchical representations and memory compression (Wang et al., 2025c; Song et al., 2024), or retrieves visual evidence through prompted agents and uncertainty-guided reasoning (Wang et al., 2024; Zhi et al., 2025). Recent methods train agents to interleave reasoning with native visual tool calls (Zhang et al., 2026b; Yang et al., 2026b; Zeng et al., 2026b). Their training signals include annotated retrieval relevance and temporal localization (Xie et al., 2025b; Yang et al., 2026b), navigation efficiency (Qiu et al., 2026b), and rewards for sustained tool interaction (Zhao et al., 2026). We build on this agentic formulation and focus on learning the marginal value of additional visual evidence.

2.2 Credit Assignment for Tool Agents

Outcome-based RL trains agents to interleave reasoning and tool use (Jin et al., 2025), but trajectory-level rewards do not directly identify which calls improve the answer; in group-normalized approaches, trajectory-level rewards can yield zero advantages when all sampled outcomes are identical (Wang et al., 2025a). Prior methods obtain finer-grained credit through Monte Carlo value estimates (Kazemnejad et al., 2024), grouping actions from repeated states (Feng et al., 2026b), or intermediate search supervision (Zheng et al., 2025). Our work similarly seeks action-level credit, with an explicit focus on distinguishing useful retrieval from cases where the available evidence is already sufficient.

2.3 Information Gain and Adaptive Stopping

Among prior work, the most closely related reward-design approaches measure improvements in correct-answer likelihood using the policy itself (Wang et al., 2025a) or a frozen reference model (Dou, 2026). Counterfactual methods compare actual retrieval with random-document replacements (Liang et al., 2026), score next actions with factual versus masked feedback (Kong et al., 2026), or reward evidence-conditioned reasoning (Guo et al., 2026). These approaches motivate our use of model-based information signals to evaluate evidence acquisition. Prior work also trains selective information acquisition. AdaTooler-V modulates tool-use rewards using a question-level benefit score estimated from reference-model accuracy with and without tools (Wang et al., 2026). CaRT teaches termination through counterfactual trajectory pairs and verbal rationales (Liu et al., 2025), while CIGAsk combines information-gain rewards for clarification quality with ambiguity-label supervision for when to ask (Li et al., 2026). Our method, CIGO, differs from these prior methods by directly comparing stopping with one additional retrieval from the same reasoning prefix. This prefix-conditioned, action-level comparison jointly supervises whether further retrieval is useful and, if so, which temporal interval to retrieve, given the evidence already acquired by the agent.

3 Method

3.1 Problem Formulation and Notation

We formulate agentic video reasoning as a Markov Decision Process (MDP). The initial input is $x = (q, V_{\text{init}})$, consisting of a textual query q and a set of uniformly sampled global frames V_{init}; the agent answers q through a multi-turn reasoning process.

At each round t, the policy conditions on the accumulated context and produces a rationale followed by a decision,

$$(R_t, a_t) \sim \pi_\theta(\cdot \mid s_{t-1}), \tag{1}$$

where $s_{t-1} = \{V_{\text{init}}, q, (R_1, a_1, V_1), \ldots, (R_{t-1}, a_{t-1}, V_{t-1})\}$ denotes the full agent context.

The action $a_t = (c_t, u_t)$ pairs a termination decision with a turn output:

$$c_t \in \{\texttt{STOP}, \texttt{LOOK}\}, \qquad u_t \in \begin{cases} \mathcal{Y}, & c_t = \texttt{STOP}, \\ \{(\tau_s, \tau_e) : 0 \le \tau_s < \tau_e \le L\}, & c_t = \texttt{LOOK}, \end{cases} \tag{2}$$

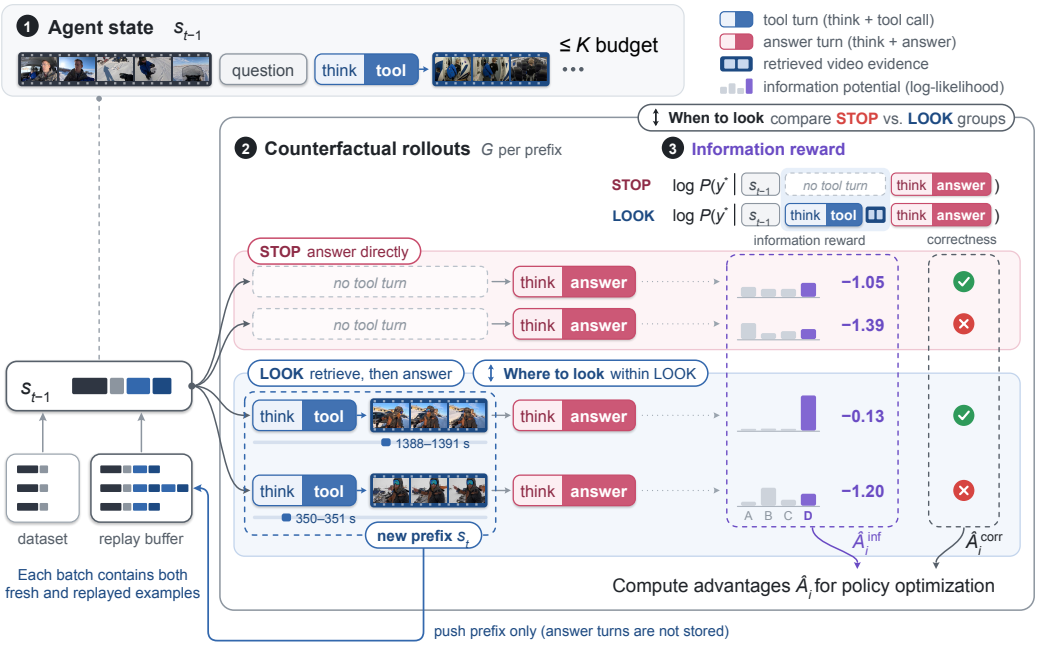


Figure 2: Overview of Counterfactual Information Gain Policy Optimization (CIGO). Given an agent state, CIGO samples G counterfactual rollouts from the same reasoning prefix. `STOP` rollouts answer questions directly, while `LOOK` rollouts retrieve an additional video interval before answering. Each rollout is evaluated using its information potential and answer correctness to compute advantages for policy optimization. K denotes the retrieval turn budget.

where $\mathcal{Y}$ is the answer space and L is the video duration. If $c_t = \texttt{LOOK}$, we invoke the frame retrieval tool $V_t = \textsc{RetrieveClip}(\tau_s, \tau_e)$ and append the turn to the context, $s_t = s_{t-1} \cup (R_t, a_t, V_t)$. Otherwise the process terminates and u_t is returned as the final answer y.

Problem Statement. We study the *tool-budgeted* evidence-retrieval setting, in which the agent aim to maximize answer correctness while issuing as few retrieval turns as necessary. Following Liu et al. (2025), given a budget of K turns and a discount $\gamma \in (0, 1)$ penalizing each additional retrieval round, we optimize the following objective:

$$\max_{\pi_\theta} \mathbb{E}_{(x,y^*)\sim\mathcal{D}} \left[\mathbb{E}_{\tau\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{K} \gamma^t \mathbb{1}\{c_t = \texttt{STOP}\}\, r(u_t, y^*) \right] \right], \tag{3}$$

where γ acts as a decaying factor on the reward accounting for the additional computation expense of the retrieval turns before committing to an answer.

3.2 Counterfactual Rollouts

Motivation. Training a video understanding agent using outcome-only or dense turn-level reward to learn the decision between stopping and continuous is challenging due to the difficulty in measuring the counterfactual effect of an retrieval action versus a tool call to the final corrcetness. To overcome this limitation, we propose to directly measure how different counterfactual actions (i.e., `LOOK` or `STOP`) conditioned on the *exact same prefix* lead to different future outcome, and then retrospectively distill the signal back for the model to learn when would a tool call be most useful and when it is not.

Given a fixed prefix s_{t-1} containing N existing turns of reasoning and tool calls $s_{t-1} = (R_t, a_t, V_t)_N$, we ask the model to explicitly decide between 2 following actions:(1) *Direct-Answer* ($c_t = \texttt{STOP}$), which answers from the current context without further retrieval; or (2) *Retrieve-then-Answer*

($c_t = \texttt{LOOK}$), which issues a single retrieval u_t, receives the retrieved frames V_t, and then answers:

$$\tau^{\texttt{STOP}} = s_{t-1} \oplus \langle \texttt{think} \rangle \oplus \langle \texttt{answer} \rangle, \tag{4}$$

$$\tau^{\texttt{LOOK}} = s_{t-1} \oplus \langle \texttt{think} \rangle \oplus \langle \texttt{tool_call} \rangle \oplus V_t \oplus \langle \texttt{think} \rangle \oplus \langle \texttt{answer} \rangle, \tag{5}$$

3.3 Information Potential Reward

Motivation. Counterfactual rollouts teach the model when to look by comparing the outcomes of stopping and retrieving from the same prefix. However, they do not effectively teach the model where to look. This is because using only correctness reward cannot distinguish different tool call actions that receive the same success or failure, causing collpase issue that the model has no learning signal to learn from. To tackle this issue, our main intuition is that to promote most effective tool call that leads to termination, a tool call should be rewarded by how much more information it provides for answering the question, in addition to whether the final answer is correct. Therefore, we propose to measure the model's confidence in the ground-truth answer, which we call the *information potential*, as a model-intrinsic and efficient proxy for task success.

Counterfactual information gain. For each counterfactual continuation i, we evaluate the information potential at its terminal context $\tilde{s}_i$, i.e., the context immediately before its final <answer> tag:

$$\tilde{s}_i = \begin{cases} s_{t-1} \oplus \langle \texttt{think} \rangle, & \texttt{STOP}, \\ s_{t-1} \oplus \langle \texttt{think} \rangle \oplus \langle \texttt{tool_call} \rangle \oplus \langle \texttt{tool_response} \rangle \oplus \langle \texttt{think} \rangle, & \texttt{LOOK}, \end{cases} \tag{6}$$

where $\langle \texttt{tool_response} \rangle$ contains the retrieved frames. The information gain of continuation i over its prefix is therefore $\Delta\Phi_i = \Phi(\tilde{s}_i) - \Phi(s_{t-1})$. Since the counterfactual construction ensures that all G continuations share s_{t-1}, $\Phi(s_{t-1})$ is constant within the group, and subtracting a constant from every reward leaves the group-normalized advantages (Section 3.4) unchanged. We therefore assign the information reward

$$r_{\text{inf},i} = \Phi(\tilde{s}_i), \tag{7}$$

which yields exactly the same advantages as $\Delta\Phi_i$. We call the resulting information advantage $\hat{A}_{\text{inf},i}$ the *counterfactual information gain* of continuation i. Across subgroups, it compares the terminal information potential increase induced by the tool call action. Within subgroups of $LOOK$, provides fine-grained signal for comparing tool calls (*where* to look), even when their correctness ties.

3.4 Policy Optimization with Short-horizon Rollout

Replay buffer. Training on prefixes at every depth requires states that already contain earlier retrievals. Since regenerating these turns for every update is costly, we store them in a replay buffer $\mathcal{B}$. After a LOOK continuation executes its retrieval, its state s_t (without the answer turn) is pushed into $\mathcal{B}$ if $t < K$ and its group is neither all correct nor all incorrect. Each batch mixes a fraction $\rho = 0.5$ of fresh questions with prefixes replayed from $\mathcal{B}$.

Rewards and advantage estimation. Each continuation i receives an information reward $r_{\text{inf},i} = \Phi(\tilde{s}_i)$, a correctness reward $r_{\text{corr},i}$ ($\mathbb{1}\{y_i = y^*\}$ for MCQ, $\text{IoU}(y_i, y^*)$ for TG), and a format reward $r_{\text{fmt},i} \in \{-1, 0\}$ penalizing over-budget or invalid continuations. Since the first two differ in scale, we normalize them separately over the G continuations of the same prefix:

$$\hat{A}_i = \lambda \underbrace{\frac{r_{\text{inf},i} - \mu(r_{\text{inf}})}{\sigma(r_{\text{inf}})}}_{\hat{A}_{\text{inf},i}} + \underbrace{\frac{r_{\text{corr},i} - \mu(r_{\text{corr}})}{\sigma(r_{\text{corr}})}}_{\hat{A}_{\text{corr},i}} + r_{\text{fmt},i}, \tag{8}$$

where λ weights the counterfactual information gain $\hat{A}_{\text{inf},i}$, and the correctness term anchors optimization to the task metric, since a higher Φ does not guarantee a correct sampled answer.

Policy update. We optimize the clipped GRPO objective with KL regularization (Shao et al., 2024). The optimization objective is in Section A.1.

Inference. At inference, no counterfactual continuations are sampled. Starting from $s_0 = x$, the agent applies the learned decision at every round with a budget of K retrievals until it answers.

4 EXPERIMENT

4.1 EXPERIMENT SETUP

Training Dataset. We construct our RL training dataset by collecting a mixture of data from Video-R1 (short) (Feng et al., 2025), LongVideo-Reason (mid) (Chen et al., 2025b), and CG-Bench(long) (Chen et al., 2025a), and Charades-STA (short) to cover a variety of multiple choice question-answering (MCQ) and temporal grounding (TG) tasks. We apply a two-stage filtering procedure. First, since the Qwen3 base model may have been exposed to benchmark data during training, we evaluate the base model without providing the video and filter questions that this model consistently answers above chance level. Second, motivated by prior analyses of collapsed group advantages during GRPO (Wang et al., 2025a), we perform offline filtering to remove examples for which all eight base model rollouts are either correct or incorrect. The final training dataset mixture contains 4800 samples: 303 from Video-R1, 1449 from LongVideo-Reason, 1608 from CG-Bench, 1440 from Charades-STA. Additional details are in Appendix Table 5.

Reinforcement Learning. We initialize our method on the Qwen3-VL-Thinking-8B model Bai et al. (2025), without any SFT stage. We set the information reward weight to 0.5 and train for 1 epoch with a learning rate of 1×10^{-5}, rollout batch size 32, group size 8, and a 4096 per-token budget. During training, the model receives 64 uniformly sampled initial frames and up to 64 frames per tool call, each at 224*224 resolution. We use a replay buffer capacity of 6400 prefixes with a ttl of 25 to ensure sample diversity in the replay buffer. Training is implemented with the verl framework, and runs on 4xRTX6000 GPUs for 1.5 days. Additional details are in Appendix Table Section B.

Evaluation. We evaluate our proposed method CIGO on 6 video understanding benchmarks for MCQ (Video-MME Fu et al. (2025), MLVU Zhou et al. (2025), LVBench Wang et al. (2025b), LongVideoBench Wu et al. (2024), Video-MMMU Hu et al. (2026)) and TG (Charades) tasks. Following Zhang et al. (2026a), we provide 1024 initial frames at 224*224 resolution for Video-MME, MLVU, LVBench, LongVideoBench, and 256 initial frames at 384*384 resolution for Video-MMMU and Charades-STA. The agent is allowed three tool calls, at most, each returning up to 64 frames. We test Qwen3-VL-8B-Thinking with the same evaluation setting and compare with existing baselines.

4.2 OVERALL PERFORMANCE

Table 1 compares CIGO with existing methods for video understanding. We find that CIGO improves performance over its base model on all 6 benchmarks, with performance gains ranging from +0.5 to +4.8 points. Among the reported open-source methods, CIGO achieves the highest-performing result on Video-MMMU (65.9%) and the second-highest performance on Video-MME (65.5%) and MLVU (69.4%). We note that CIGO performance gains are obtained by training directly from the base model with only 4800 samples, *without* using an SFT training stage and *without* curating large amounts of expensive teacher-distilled reasoning trajectories. In addition, CIGO achieves competitive performance with existing video tool call methods while using a modest retrieval budget at inference time, permitting at most 3 tool calls, in contrast with up to 8 tool calls for Video-o3 and up to 16 tool calls for VideoSEAL. Overall, our findings demonstrate the potential of CIGO to improve video reasoning across diverse tasks while using limited training data and a constrained retrieval budget.

4.3 ABLATION STUDY

We conduct an ablation study to examine the two main components within CIGO: counterfactual training (CF) and the information potential reward (IR). All ablation variants are trained with the same Qwen3-VL-8B-Thinking base model with the same data mixture and training recipe. Ablation results are presented in Table 2.

Effectiveness of Counterfactual Training. To isolate the effect of counterfactual training, we compare CIGO (CF + IR) with GRPO + IR. We find that adding counterfactual training improves

Table 1: **Comparison on long-video question answering and temporal grounding.** All 7B methods build on Qwen2.5-VL-7B-Instruct except VideoMind (Qwen2-VL-7B-Instruct); ParaVT builds on Qwen3-VL-8B-Instruct, VideoSEAL on Qwen3-8B, and Ours on Qwen3-VL-8B-Thinking. N_f is the evaluation setting, written as initial frames + maximum tool calls × frames per call; K means no fixed cap. Where the budget varies by benchmark, we show the long-video setting. * marks scores that we reproduced; others are reported from the source papers. – marks scores that were not reported by source papers. The highest-performing and second highest-performing open-source results are bolded and underlined. Green indicates the CIGO gain over Qwen3-VL-8B-Thinking base model.

Method	SFT	RL	Dense reward	N_f	Video-MME *17 min*	MLVU *15 min*	LVBench *68 min*	LongVideo-Bench *8 s–60 min*	Video-MMMU *8 min*	Charades-STA *30 s*
Proprietary multimodal models										
GPT-4o Hurst et al. (2024)	—	—	—	varies	71.9	64.6	48.9	66.7	61.22	35.7
Gemini-1.5-Pro Team et al. (2024)	—	—	—	1 fps	75.0	61.8	33.1	64.0	53.89	–
Open-source multimodal models										
Qwen3-VL-8B-Thinking Bai et al. (2025)	✓	✓	✗	1024	64.7*	68.9*	40.2*	59.3*	<u>65.1</u>*	50.3*
Video-R1-7B Feng et al. (2026a)	✓	✓	✗	64	61.4	61.6	40.1	52.7	52.4	34.9
VideoChat-R1-7B Li et al. (2025)	✗	✓	✗	32	56.9	57.0	34.3	49.1	54.0	**60.8**
Multi-agent systems										
VideoMind-7B Liu et al. (2026b)	✓	✗	✗	32–150	58.2	64.4	40.8	56.3	–	50.2
ParaVT-8B Yang et al. (2026c)	✓	✓	✗	$64+K\times16$	62.1	65.0	39.8	60.4	–	50.1
VideoSEAL-8B Qiu et al. (2026a)	✗	✓	✗	$0+16\times64$	62.9	68.2	**55.1**	**62.0**	–	–
Tool-call agents										
VITAL-7B Zhang et al. (2026a)	✓	✓	✓	$1024+1\times64$	64.1	–	–	–	54.2	59.9
LongVT-7B Yang et al. (2026a)	✓	✓	✗	$768+4\times128$	–	59.4	41.3	54.7	45.4	27.2
VideoZoomer-7B Ding et al. (2026)	✓	✓	✗	$64+4\times16$	65.2	68.8	41.5	57.7	52.2	–
VideoTemp-o3-7B Liu et al. (2026a)	✓	✓	✗	$1024+2\times64$	64.5	–	43.0	–	53.2	57.8
Video-o3-7B Zeng et al. (2026a)	✓	✓	✗	$768+8\times$crop	**66.5**	**72.1**	<u>47.6</u>	<u>60.5</u>	51.7	<u>60.7</u>
Video-MTR-7B Xie et al. (2025a)	✗	✓	✓	$40+2\times20$	62.7	–	42.3	57.1	–	–
Ours										
Ours (8B)	✗	✓	✓	$1024+3\times64$	<u>65.5</u>	<u>69.4</u>	42.1	60.0	**65.9**	55.1
Δ vs. Qwen3-VL-8B-Thinking					+0.8	+0.5	+1.9	+0.7	+0.8	+4.8

performance over GRPO + IR across all 6 benchmarks and substantially changes the learned tool-use behavior. GRPO + IR converges to an almost fixed one-call policy, with 99.3% of late-training rollouts issuing exactly one tool call. In contrast, CIGO results in the video agent distributing rollouts across zero (63.5%), one (14.1%), two (20.8%), and three (1.6%) tool calls. We find that GRPO + AR exhibits a similar one-call collapse in tool-calling behavior, with 98.6% of rollouts issuing exactly one tool call. In addition, we find that standard GRPO tends to over-retrieve video evidence, with 54.2% of rollouts fully using the three-call budget. Overall, these findings suggest that counterfactual training in CIGO can help video agents learn an adaptive stopping policy and avoid collapsing towards single tool calls or repeated retrieval.

Effectiveness of Information Potential Reward. To isolate the effect of the information potential reward, we compare GRPO + IR with standard GRPO. We find that adding IR improves performance across all 6 benchmarks. We observe the largest gain on temporal grounding, with IR resulting in video agent performance on Charades-STA improving from 36.9 to 53.6 mIoU. GRPO + IR performance also matches or outperforms the annotation-based dense reward baseline GRPO + AR on four of the six benchmarks, without requiring an expensive annotation-based retrieval reward.

4.4 In-depth Analysis

In this section, we study the following questions: *(1) Can our video agent trained with CIGO make better decisions between* `LOOK` *or* `STOP`*?*, and *(2) Does our agent* `LOOK` *at better video locations for video understanding?* For comparison with CIGO in these analyses, we select 3 open-weight baseline models.

Tool Call Decisions. We examine whether the agent's retrieval behavior adapts to the expected benefit of using additional video evidence (whether or not answering the question needs a tool call). To obtain the label for each question's tool call necessity, we measure whether providing

Table 2: **Ablation studies.** Every variant is trained with RL directly from Qwen3-VL-8B-Thinking (no SFT) on the same data mixture and training recipe. *CF*: RL with single-step counterfactual rollouts. *AR*: annotation-based dense reward, computed by the IoU((T_s, T_e), Y^*). *IR*: information potential reward. *Base*: Qwen3-VL-8B-Thinking without tools. **Left:** evaluation results. Evaluation uses 1024 frames at 224^2 on Video-MME, MLVU, LVBench and LongVideoBench, 256 frames at 384^2 on Video-MMMU and Charades-STA, and at most three tool calls of up to 64 frames each. MLVU is the official M-Avg. Italic figures give the average video duration. Best and second-best results among the trained variants are in **bold** and underlined. **Right:** share of last 15 steps training rollouts that make 0, 1, 2 or 3 tool calls.

Method	Video QA (accuracy %)					TG (mIoU %)	Training rollouts (%)			
	Video-MME *17 min*	MLVU *15 min*	LVBench *68 min*	LongVideo Bench *8 s–60 min*	Video-MMMU *8 min*	Charades-STA *30 s*	0 *calls*	1 *call*	2 *calls*	3 *calls*
Base (no tool)	64.7	68.9	40.2	59.3	65.1	50.3	100	–	–	–
Training algorithm										
GRPO	57.5	65.6	28.8	46.5	60.8	36.9	1.3	17.3	27.2	54.2
GRPO + AR	61.4	66.8	38.8	53.6	64.4	52.4	0.8	98.6	0.3	0.3
GRPO + IR	62.1	66.2	39.7	54.1	63.9	53.6	0.3	99.3	0.3	0.1
CF + IR (Ours)	**65.5**	**69.4**	**42.1**	**60.0**	65.9	**55.1**	63.5	14.1	20.8	1.6
Reward formulation										
CT	64.5	68.2	40.8	58.5	**66.3**	52.8	4.4	83.0	5.3	7.3
CF + IR (Ours)	**65.5**	**69.4**	**42.1**	**60.0**	65.9	**55.1**	63.5	14.1	20.8	1.6

Agent	LVBench n=200 Nec.↑ (n=91)	LVBench Unn.↓ (n=109)	LVBench Ratio↑	ReXTime n=200 Nec.↑ (n=54)	ReXTime Unn.↓ (n=146)	ReXTime Ratio↑
Video-o3	1.02	0.98	*1.0*×	1.02	0.96	*1.1*×
VideoTemp-o3	1.41	1.38	*1.0*×	0.18	0.15	*1.2*×
ParaVT	2.81	2.86	*1.0*×	4.13	4.37	*0.9*×
Ours	1.31	1.30	*1.0*×	**0.49**	**0.12**	***4.1***×

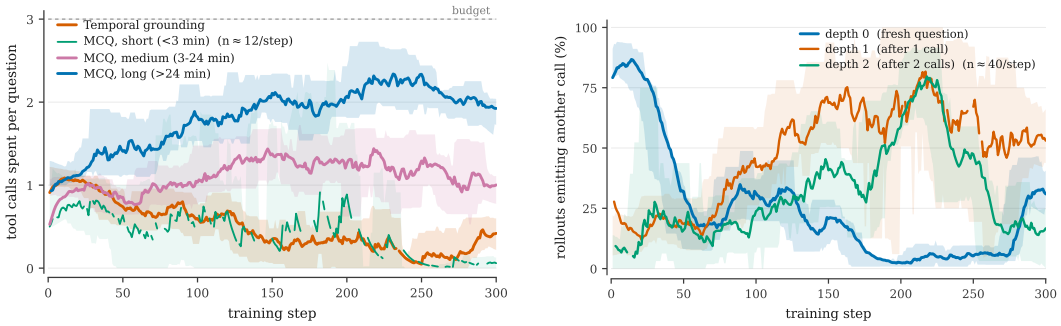
Table 3: **Tool-call allocation by ground-truth evidence benefit.** Mean tool calls per question on the tool-solvable (Nec.) and video-solvable (Unn.) subsets; Ratio = Nec./Unn. Grey ratios indicate that Ratio > 1 is not significant at $p < 0.05$. n: number of questions.

Agent	LVBench n=200 ΔAcc.	LVBench mIoU	LVBench R@0.5	ReXTime n=200 ΔAcc.	ReXTime mIoU	ReXTime R@0.5
Video-o3	−0.0	5.4	5.5	1.9	12.2	11.0
VideoTemp-o3	−0.8	2.1	1.5	0.3	7.7	4.0
ParaVT	−0.2	4.0	3.5	2.1	4.7	2.0
Ours	**1.7**	**8.6**	**7.5**	**2.3**	**29.4**	**22.0**
GT window	*11.4*	*100.0*	*100.0*	*3.4*	*100.0*	*100.0*

Table 4: **Quality of retrieved evidence.** ΔAcc. measures the increase in a reference model's success rate when given the evidence retrieved by each agent. mIoU and R@0.5 measure the overlap between the retrieved window and the ground-truth window. n: number of questions.

ground truth fine-grained evidence can lead to substantial performance improvement by the reference model. We generate 16 rollouts from Qwen3-VL-8B-Thinking both with and without the ground truth evidence and compute the difference in success rate. We, then, split the questions into necessary and unnecessary subsets, based on whether the difference in success rate is greater than 0. As seen in Table 3, we find little separation between these two subsets on LVBench. However, on ReXTime, video agents trained with CIGO make substantially more tool calls on questions that benefit from additional video evidence, compared to questions that do not. Our findings suggest that CIGO can train video agents to adapt tool call decisions to the expected benefit of additional video evidence.

Tool Call Quality. We next study whether CIGO helps video agents learn *where* to retrieve information from videos. To evaluate the quality of retrieved evidence, we make each agent execute a tool call and measure two metrics: (1) the improvement in a reference model's ability to answer questions when given the retrieved evidence and (2) the temporal overlap between the retrieved video interval and the ground-truth evidence window. As shown in Table 4, CIGO results in video agents



(a) Calls spent per question.

(b) Calling decision by depth.

Figure 3: **Tool-calling behavior over training.** (a) calls spent per answered question, by task and Video-MME duration band. (b) the fraction of rollouts emitting another call having already made d. Only depth 0 is purely the policy; deeper rows are replayed from a buffer fed only by questions where the model already called.

that achieve the strongest retrieval quality across all reported metrics on both LVBench and ReXTime. On LVBench, CIGO achieves a +1.7-point accuracy improvement, 8.6 mIoU, and 7.5 R@0.5; on ReXTime, it achieves +2.3 points, 29.4 mIoU, and 22.0 R@0.5. Our results indicate that CIGO can help video agents learn to retrieve video intervals that are more useful for answering questions and better aligned with the relevant ground truth temporal evidence.

Training Dynamics. We examine how the video agent's tool calling behavior evolves during training. As shown in Figure 3, we find that CIGO differentiates retrieval behavior according to task type and video duration. CIGO allocates more tool calls to MCQ examples in longer videos, while exhibiting lower retrieval on short videos. We also observe that the probability of issuing an initial tool call decreases over training. Overall, these training dynamics indicate that CIGO helps agents learn an adaptive retrieval policy, rather than converging to a fixed number of tool calls.

5 Conclusion

We introduced Counterfactual Information Gain Policy Optimization (CIGO), an RL framework for teaching video agents both *when* and *where* to retrieve additional visual evidence during video reasoning. CIGO branches counterfactual rollouts from the same reasoning prefix and rewards each tool call by the information gain that it provides, compared to answering directly. Comparing retrieval against direct answering teaches the agent *when* further retrieval is worthwhile, while comparing alternative retrievals teaches it *where* to look. Experiments across six benchmarks demonstrate that CIGO consistently improves over its base model and achieves competitive results against existing strong video understanding models. Our analyses show that CIGO helps video agents avoid collapsing to fixed tool-use patterns and learn adaptive strategies for retrieval during video reasoning.

AI Use Disclosure

In this work, we used generative AI tools (specifically coding agents) to assist in implementing methods and monitoring experiments. Accordingly, in the submission, we selected the following item from the dropdown menu for AI Use Disclosure: "Yes, for research ideation or execution. Details are described in the paper". Because this category combines research ideation and execution, we clarify that our use was limited to research execution; we did not use generative AI tools for research ideation. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, or interpret results. We note that generating synthetic data sets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. In addition, we used generative AI tools (coding agents) to

help in the figure creation process. We have manually reviewed all AI-assisted work. LLM-generated code was fully verified and tested for correctness by human researchers.

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*, 2025.

Guo Chen, Yicheng Liu, Yifei Huang, Baoqi Pei, Jilan Xu, Yuping He, Tong Lu, Yali Wang, and Limin Wang. Cg-bench: Clue-grounded question answering benchmark for long video understanding. In *International Conference on Learning Representations*, volume 2025, pp. 45647–45682, 2025a.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

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. Scaling rl to long videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Yang Ding, Xin Lai, Yizhen Zhang, Wei Li, Ruihang Chu, and Yujiu Yang. Videozoomer: Reinforcement-learned temporal focusing for long video reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 20087–20111, 2026.

Hao Dou. Cigpo: Contextual information-gain policy optimization for multi-turn evidence-reading llm agents. *arXiv preprint arXiv:2607.16244*, 2026.

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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=a2JTVVvcEl.

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. *Advances in Neural Information Processing Systems*, 38:99114–99137, 2026a.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026b.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18995–19012, 2022.

Xingyu Guo, Wei Chen, Linlin Yang, and Baochang Zhang. Contextual information policy optimization for search agents. *arXiv preprint arXiv:2608.06128*, 2026.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Xiang Yue, Bo Li, Yuanhan Zhang, and Ziwei Liu. Video-MMMU: Evaluating knowledge acquisition from multidisciplinary professional videos. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27798–27828, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1281. URL https://aclanthology.org/2026.acl-long.1281/.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms. *arXiv preprint arXiv:2410.01679*, 2024.

Fanqi Kong, Jiayi Zhang, Mingyi Deng, Chenglin Wu, Yuyu Luo, and Bang Liu. Infopo: Information-driven policy optimization for user-centric agents. *arXiv preprint arXiv:2603.00656*, 2026.

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*, 2025.

Yunxiang Li, Xixin Wu, and Helen Meng. When and how should an agent clarify? cigask: Teaching llms to clarify via counterfactual information gain, 2026. URL `https://arxiv.org/abs/2609.24290`.

Zihan Liang, Yufei Ma, Ben Chen, Zhipeng Qian, Huangyu Dai, Lingtao Mao, Xuxin Zhang, Chenyi Lei, and Wenwu Ou. Ig-search: Step-level information gain rewards for search-augmented reasoning. *arXiv preprint arXiv:2604.15148*, 2026.

Grace Liu, Yuxiao Qu, Jeff Schneider, Aarti Singh, and Aviral Kumar. Cart: Teaching llm agents to know when they know enough. *arXiv preprint arXiv:2510.08517*, 2025.

Wenqi Liu, Yunxiao Wang, Shijie Ma, Meng Liu, Qile Su, Tianke Zhang, Haonan Fan, Changyi Liu, Kaiyu Jiang, Jiankang Chen, et al. Videotemp-o3: Harmonizing temporal grounding and video understanding in agentic thinking-with-videos. *arXiv preprint arXiv:2602.07801*, 2026a.

Ye Liu, Kevin Qinghong Lin, Chang-Wen Chen, and Mike Zheng Shou. Videomind: A chain-of-lora agent for temporal-grounded video reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 57481–57506, 2026b.

Chenhao Qiu, Yechao Zhang, Xin Luo, Shien Song, and Xusheng Liu. Videoseal: Mitigating evidence misalignment in agentic long video understanding by decoupling answer authority. *arXiv preprint arXiv:2605.12571*, 2026a.

Jihao Qiu, Lingxi Xie, Xinyue Huo, Qi Tian, and Qixiang Ye. Longvideo-r1: Smart navigation for low-cost long video understanding. *arXiv preprint arXiv:2602.20913*, 2026b.

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

Aditya Sharma, Michael Saxon, and William Yang Wang. Losing visual needles in image haystacks: Vision language models are easily distracted in short and long contexts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5429–5451, 2024.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18221–18232. IEEE, 2024.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Chaoyang Wang, Kaituo Feng, Dongyang Chen, Zhongyu Wang, Zhixun Li, Sicheng Gao, Meng Meng, Xu Zhou, Manyuan Zhang, Yuzhang Shang, et al. Adatooler-v: Adaptive tool-use for images and videos. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 18058–18072, 2026.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn llm agents. *arXiv e-prints*, pp. arXiv–2510, 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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967. IEEE, 2025b.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pp. 58–76. Springer, 2024.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3282. IEEE, 2025c.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Tsung-Han Wu, Giscard Biamby, Jerome Quenum, Ritwik Gupta, Joseph E Gonzalez, David Chan, et al. Visual haystacks: A vision-centric needle-in-a-haystack benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 64592–64615, 2025.

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

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

Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, et al. Longvt: Incentivizing" thinking with long videos" via native tool calling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33816–33826, 2026a.

Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, et al. Longvt: Incentivizing" thinking with long videos" via native tool calling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33816–33826, 2026b.

Zuhao Yang, Kaichen Zhang, Sudong Wang, Keming Wu, Zhongyu Yang, Bo Li, Xiaojuan Qi, Shijian Lu, Xingxuan Li, and Lidong Bing. Paravt: Taming the tool prior paradox for parallel tool use in agentic video reinforcement learning. *arXiv preprint arXiv:2605.20342*, 2026c.

Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, et al. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. *arXiv preprint arXiv:2601.23224*, 2026a.

Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, et al. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. *arXiv preprint arXiv:2601.23224*, 2026b.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32903–32914, 2026a.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32903–32914, 2026b.

Shitian Zhao, Shaoheng Lin, Ming Li, Haoquan Zhang, Wenshuo Peng, Kaipeng Zhang, and Chen Wei. Pyvision-rl: Forging open agentic vision models via rl. *arXiv preprint arXiv:2602.20739*, 2026.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21816–21841, 2025.

Zhuo Zhi, Qiangqiang Wu, Wenbo Li, Yinchuan Li, Kun Shao, Kaiwen Zhou, et al. Videoagent2: Enhancing the llm-based agent system for long-form video understanding by uncertainty-aware cot. *arXiv preprint arXiv:2504.04471*, 2025.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701. IEEE, 2025.

Table 5: **RL training data composition.** Top: rows per source dataset and each dataset's share of the 4,800 training rows. The candidate pool is the questions that pass the blind filter; from it we keep only questions the policy answered correctly on some rollouts and incorrectly on others, and at most 3 questions per video. Bottom: multiple-choice rows by video length, and the share of each length in the multiple-choice evaluation benchmarks (Video-MME, MLVU, LVBench, LongVideoBench, Video-MMMU).

	LongVideo-Reason	CG-Bench	Video-R1	Charades-STA	Total
Task	MCQ	MCQ	MCQ	TG	–
Selected	1,449	1,608	303	1,440	4,800
Share (%)	30.2	33.5	6.3	30.0	100.0

Video length	<2 min	2–30 min	30–60 min	>1 h	Total
MCQ rows	491	2,061	753	55	3,360
Share, training (%)	14.6	61.3	22.4	1.6	100.0

A Appendix

Our code is available at: `https://anonymous.4open.science/r/CIGO-DB3B`

A.1 Method Details

Policy Optimization. We optimize the policy using the standard GRPO objective applied to these single-step counterfactual rollouts:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(x,y^*)\sim\mathcal{D},\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Bigg(\frac{\pi_\theta(o_{i,t}\mid x,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid x,o_{i,<t})}\hat{A}_{i,t}, \\ \text{clip}\left(\frac{\pi_\theta(o_{i,t}\mid x,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid x,o_{i,<t})},\,1-\epsilon,\,1+\epsilon\right)\hat{A}_{i,t}\Bigg) - \beta\,\mathbb{D}_{\text{KL}}(\pi_\theta\,\|\,\pi_{\text{ref}})\Bigg] \tag{9}$$

B Training Implementation Details

Table 6: **RL training configuration.**

Setting	**Value**
Initial policy / reference model	Qwen3-VL-8B-Thinking
Training questions	4,800
Prompts per step / group size	32 / 8
Optimizer, learning rate	AdamW, 1×10^{-5}
KL loss coefficient	0.01
Clipping range	$\epsilon_{\text{low}} = 0.2$, $\epsilon_{\text{high}} = 0.28$
Rollout temperature	1.0
Max tool calls / tokens per turn	3 / 4,096
Uncertainty reward weight β	0.5
Replayed-prefix share per batch	0.5
Hardware	4 × RTX PRO 6000 (96 GB)

C Additional In-depth Analysis

D Prompt Templates

Figures 4 list every prompt the agent reads, in training and in evaluation. The same system prompt, user question prompt and tool response prompt are used in both; only the forced-answer prompt is

Table 7: **Accuracy on blind and non-blind questions.** A question is *blind* (B) when Qwen3-VL-8B-Thinking answers it correctly from the question text alone, with no video (four no-video samples at temperature 0.6, α=0.01). Every other question is *non-blind* (NB), which means the answer has to come from the video. Each baseline is evaluated under the setting stated in its own paper. Charades-STA reports mIoU (%); all other benchmarks report accuracy (%). MLVU is per-question accuracy rather than the official M-Avg, because M-Avg averages seven task types and cannot be split into the two subsets. The Δ row gives the relative change (%) of Ours over Qwen3-VL-8B-Thinking. Number of questions (NB / B): Video-MME 1939 / 761, MLVU 1522 / 652, LVBench 1347 / 202, LongVideoBench 941 / 396, Video-MMMU 548 / 331, Charades-STA 3324 / 396. Italic figures give the average video duration.

Method	**Video-MME** *17 min*		**MLVU** *15 min*		**LVBench** *68 min*		**LongVideoBench** *8 s–60 min*		**Video-MMMU** *8 min*		**Charades-STA** *30 s*	
	NB	B	NB	B	NB	B	NB	B	NB	B	NB	B
Video-o3-7B	60.5	80.0	64.9	87.4	46.4	63.9	50.3	81.3	38.1	69.5	–	–
VideoTemp-o3-7B	56.6	77.9	63.9	84.1	38.8	58.9	48.5	78.0	32.5	68.0	54.9	63.1
LongVT-7B	–	–	–	–	–	–	–	–	–	–	–	–
Qwen3-VL-8B-Thinking	56.4	85.8	62.2	88.2	36.4	65.4	47.4	87.6	50.4	89.4	49.3	57.9
Ours (8B)	58.4	83.4	62.8	84.8	38.4	61.9	51.2	80.8	51.5	89.7	53.4	68.9
Δ vs. Qwen3-VL-8B-Thinking	+2.0	−2.4	+0.6	−3.4	+2.0	−3.5	+3.8	−6.8	+1.1	+0.3	+4.1	+11.0

specific to evaluation. The system prompt carries the tool schema and the step protocol. The user question prompt follows the overview frames and states their sampling density. Each tool call returns frames wrapped in the tool response prompt, which names the retrieved window, its sampling density, every window retrieved so far, and whether further calls are allowed. When no calls remain, the last line switches to the must-answer sentence. In evaluation, a trajectory that reaches the turn limit without an answer receives the forced-answer prompt once.

System Prompt

You are a video reasoning assistant. You are given a uniformly sampled overview of a video and a question about it. You may retrieve additional frames from any time range before answering.

Tools

You may call one or more functions to assist with the user query.

You are provided with function signatures within <tools></tools> XML tags:
<tools>
{"type": "function", "function": {"name": "get_video_clip_frame", "description": "Inspect a time range of the video closely. Returns up to 64 frames evenly spaced across [start_time, end_time]; the video is stored at 2 frames per second, so a range shorter than 32 seconds returns 2 frames for each of its seconds. A narrow range therefore comes back densely sampled, while a range covering the whole video returns the same overview you already hold.", "parameters": {"type": "object", "properties": {"start_time": {"type": "number", "description": "The start time (in seconds) of the video segment, keeping two decimal places."}, "end_time": {"type": "number", "description": "The end time (in seconds) of the video segment, keeping two decimal places."}}, "required": ["start_time", "end_time"]}}}
</tools>

For each function call, return a json object with function name and arguments within ***<tool_call></tool_call>*** XML tags:
<tool_call>
{"name": <function-name>, "arguments": <args-json-object>}
</tool_call>

Procedure

Each step is one ***<think> </think>*** block followed by exactly one ***<tool_call> </tool_call>*** or one ***<answer> </answer>***.

If the observations you currently hold do not yet show a visual detail that is essential to answering, name the time range that would contain that detail and call get_video_clip_frame on it to observe that range in fine detail.

i.e. ***<think>*** ... ***</think><tool_call>*** ... ***</tool_call>***

Once you believe you have observed all the video clues that help you answer the question, and are able to integrate the existing clues to give the correct answer, put the correct answer within the ***<answer>*** and ***</answer>*** tags and stop.

i.e. ***<think>*** ... ***</think><answer>*** ... ***</answer>***

All timestamps refer to the full video, lie within its duration, and are written to two decimal places.

The user message states a tool budget. This is the maximum number of tool calls allowed for the entire question. You must not exceed it: once the remaining budget reaches zero you must answer.

Answer format

You must output exactly one ***<answer> </answer>*** block and it must be the last thing you produce.

For a multiple-choice question, put only the option letter inside the answer tags, e.g. ***<answer>***C***</answer>***.

For a temporal grounding question, put only the start and end time in seconds inside the answer tags, e.g. ***<answer>***The event happens in 24.30 - 30.42 seconds.***</answer>***.

Otherwise follow exactly the answer format requested in the question.

Figure 4: **System prompt and user question prompt**, shared by training and evaluation. The `# Tools` block and the ***<tool_call>*** format follow the Qwen3-VL chat template.

User Question Prompt

<video> *({total_frames} overview frames, each preceded by its timestamp token <{t} seconds> and stamped with it in the pixels)*
The frames above are the global context of the video: the full video uniformly sampled at {sample_fps:.2f} fps, {total_frames} frames spanning 0.00s to {max_duration:.2f}s, one frame every {frame_interval:.2f}s. An event shorter than about {frame_interval:.2f}s can fall entirely between two of these frames.
User Question:
{question}

Question Text for Multiple-Choice and Temporal Grounding

Multiple choice. {question}
A. {option A}
B. {option B}
...
Temporal grounding. Please find the visual event described by a sentence in the video, determining its starting and ending times. The format should be: 'The event happens in the start time - end time'. For example, The event 'person turn a light on' happens in the 24.30 - 30.42 seconds. Now I will give you the textual sentence: {query}. Please return its start time and end time.

Tool Call Emitted by the Model

<think>*your reasoning, naming the missing detail and where it should be*</think>
<tool_call>
`{"name": "get_video_clip_frame", "arguments": {"start_time": {start}, "end_time": {end}}}`
</tool_call>

Tool Response Prompt (calls remain)

<tool_response>
<video> *({n_frames} frames from {start_time} to {end_time}, each preceded by its absolute timestamp token and stamped with it in the pixels)*
Fine-grained Observation {obs_idx} - {start_time:.2f}s to {end_time:.2f}s, {n_frames} frames, one every {clip_interval:.2f}s.
Retrieved so far: {retrieved_windows}
Next step: You may answer or call another tool in this round.
</tool_response>

Tool Response Prompt (no calls remain)

<tool_response>
<video> *(as above)*
Fine-grained Observation {obs_idx} - {start_time:.2f}s to {end_time:.2f}s, {n_frames} frames, one every {clip_interval:.2f}s.
Retrieved so far: {retrieved_windows}
Next step: You must answer in this round; no calls remain.
</tool_response>