BEYOND REWATCHING: LEARNING EVIDENCE-DRIVEN COMPLEX VIDEO REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Complex video reasoning requires integrating visual evidence across entities, events, and time. Although recent models interleave textual reasoning with visual inspection, tool use alone does not ensure evidence-driven reasoning: models may inspect unnecessarily, fail to adapt unsuccessful searches, or retain interpretations contradicted by new observations. We address these limitations through three complementary training improvements. First, we propose Evidence-Conditioned Synthesis (ECS), which constructs verified reasoning continuations from intermediate states in executed video interactions, providing supervision for evidence assessment, selective inspection, recovery, and belief updating. Second, we introduce EviVideo-QA-20K, a dataset of 20K perception-centric questions that require composing visual evidence through conjunctive and sequential reasoning. Third, we design an Evidence Reward that complements answer correctness with an assessment of evidence use and an explicit inspection cost. Together, these components train EviVideo through supervised fine-tuning followed by reinforcement learning. Our 4B and 8B models improve over their Qwen3-VL backbones by average absolute accuracy gains of 7.6% and 7.9% across MINERVA, Minerva-Ego, and Video-Holmes, and by 4.0% and 3.6% across Video-MME and Video-MME-v2. Behavioral evaluations further indicate improved evidence-responsive inspection and revision. These results highlight the importance of training not merely visual tool use, but how evidence guides reasoning and interaction.

1 INTRODUCTION

In real-world applications, even seemingly simple questions can require connecting multiple events and reasoning about their consequences. For example, an assistant observing meal preparation might be asked: *"Which utensil is unsafe to serve the salad with, given its earlier use and subsequent cleaning?"* Answering requires more than recognizing objects and actions: the model must trace a utensil's contact with raw meat, determine whether it was subsequently washed, and infer the resulting contamination risk. Such interconnected, multi-step inferences constitute *complex video reasoning*. Grounding these inferences in visual evidence is essential for reliable assistance in applications such as embodied assistants and AI glasses.

To solve complex reasoning problems, recent advances such as DeepSeek-R1 Guo et al. (2025) have demonstrated the effectiveness of reinforcement learning in training language models to generate chains of thought (CoT) before answering. Video-R1 Feng et al. (2025) extends this approach to multimodal language models (MLLMs) for video reasoning. To strengthen visual grounding within textual CoT, Video-PLR Pu et al. (2025) incorporates timestamped visual descriptions into iterative perception–reasoning steps, while ReWatch- R1 Zhang et al. (2025a) structures reasoning as a textual simulation of revisiting video evidence. However, reasoning over fixed initial observations cannot recover visual details that were not captured or sufficiently resolved. Extending the paradigm of *thinking with images* OpenAI (2025) to videos, methods such as Video-o3 Zeng et al. (2026), LongVT Yang et al. (2026a), and VideoZoomer Ding et al. (2025) interleave textual reasoning with tool calls that retrieve additional frames. These visual-text interleaved trajectories enable *active visual reasoning*, allowing models to revisit relevant content as reasoning unfolds.

However, enabling active visual reasoning does not by itself ensure that reasoning becomes evidence-driven. Our examination of existing models reveals recurring failures in both textual reasoning and

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: **From visual tool use to evidence-driven active visual reasoning. Left:** Selected trajectories from existing models illustrate four failure modes: ungrounded reasoning, miscalibrated evidence sufficiency, inflexible inspection, and confirmation bias. **Right:** A demonstration from our synthesized SFT data identifies missing evidence, inspects relevant video segments, and revises an earlier interpretation using returned observations before answering. Red highlights problematic reasoning; green highlights evidence-grounded observations and updates. Ellipses omit intermediate text.

visual tool use. As illustrated in Fig. 1, we categorize these failures into four types: (i) **Ungrounded reasoning**: models analyze answer options or rely on global impressions rather than identify the visual evidence needed to resolve the question; (ii) **Miscalibrated evidence sufficiency**: models answer despite missing evidence or continue inspecting when the available information already suffices; (iii) **Inflexible inspection**: models fail to adjust the location, temporal extent, or granularity of inspection when observations are irrelevant or insufficient; and (iv) **Confirmation bias**: models use new observations to reinforce an initial answer, while contradictory evidence fails to trigger appropriate belief revision. Thus, genuine tool calls do not necessarily imply that evidence guides what to inspect, when to stop, or how to revise a conclusion.

Our analysis identifies three contributing factors in the training data and optimization objectives of current video reasoning policies. Among the existing SFT-initialized policies we examine, evidence assessment and corrective reasoning remain limited, reducing opportunities for subsequent RL to reinforce these behaviors. One contributing factor is the limited behavioral coverage of SFT demonstrations. All 7,019 trajectories in VITAL's released tool-using long-video QA subset Zhang et al. (2025b) contain exactly one inspection, providing no examples of adjusting and repeating an unsuccessful inspection. Explicit revision based on new observations appears in only 8% of sampled LongVT tool-using trajectories Yang et al. (2026a), and we identify no such examples in our Video-o3 sample Zeng et al. (2026). Such demonstrations provide limited direct supervision for recovering from unsuccessful inspections and revising earlier interpretations. This may partly arise from the synthesis process: LongVT and Video-o3 provide ground-truth answers and temporal evidence annotations during generation, potentially encouraging rewatching to justify an already-known answer rather than resolve uncertainty. The underlying tasks can also provide insufficient pressure for multi-step evidence acquisition: basic recognition, OCR, and arithmetic questions may require little integration across events, while some questions admit language shortcuts. Finally, RL rewards may reinforce interaction without establishing its utility. VideoZoomer Ding et al. (2025), for example, awards a correctness-gated tool-use bonus without explicitly assessing whether inspection was needed, potentially encouraging unnecessary rewatching. Together, these observations motivate improving CoT supervision, task design, and RL incentives.

To address these limitations, we introduce three complementary improvements. First, we propose *Evidence-Conditioned Synthesis* (ECS) to construct EviCoT-150K, an SFT dataset for evidence-driven

active visual reasoning. Rather than retaining only complete successful rollouts, ECS also branches from genuine intermediate states with unresolved evidence gaps or unsupported interpretations. It executes visual actions when needed and supervises verified continuations while treating the preceding prefix as context. This provides demonstrations of selective inspection, recovery, and belief updating alongside direct reasoning from sufficient observations. Second, we construct EviVideo-QA-20K, comprising 20K perception-centric questions through conjunctive and sequential composition of visual conditions. These tasks require identifying evidence for intermediate reasoning steps and integrating it across entities and events; short, open-ended answers reduce option-based shortcuts while retaining verifiability. Third, we introduce an *Evidence Reward* for GRPO Shao et al. (2024), complementing answer correctness with a frozen multimodal evaluator's assessment of evidence use and an explicit inspection cost. Together, these improvements train EviVideo through SFT followed by RL to perform more evidence-driven active visual reasoning.

We evaluate EviVideo on MINERVA Nagrani et al. (2025), Minerva-Ego Nagrani et al. (2026), and Video-Holmes Cheng et al. (2025) for complex video reasoning, and on Video-MME Fu et al. (2025a) and Video- MME-v2 Fu et al. (2026) for general video understanding. Compared with their corresponding Qwen3-VL backbones, our 4B and 8B models achieve average absolute accuracy gains of 7.6% and 7.9% across the three complex reasoning benchmarks, respectively, with improvements of up to 9.5% on Video-Holmes. The gains extend to general video understanding, averaging 4.0% and 3.6% across the two benchmarks for the 4B and 8B models. Behavioral analyses further examine evidence-responsive inspection and revision, while inference comparisons assess the benefits of adaptive inspection beyond increased visual input. Together, our findings support the importance of learning *when to seek evidence, how to interpret it, and when it should change the model's conclusions*, rather than simply generating longer CoT or invoking visual tools more frequently.

2 Related Work

Video Reasoning. Video-R1 Feng et al. (2025) and LongVILA-R1 Chen et al. (2025b) advance video reasoning through CoT supervision and reinforcement learning, including temporal modeling and long-video training. Subsequent methods strengthen the connection between textual reasoning and visual content. VideoRFT Wang et al. (2025c) combines visually revised CoT demonstrations with a semantic-consistency reward. Video-PLR Pu et al. (2025) alternates timestamped visual descriptions with reasoning and introduces factuality supervision, while ReWatch-R1 Zhang et al. (2025a) synthesizes textual rewatch trajectories through caption-based agent interaction and adds observation-and-reasoning rewards. Video-Thinker Wang et al. (2025d) and Open-o3 Video Meng et al. (2025) further structure reasoning with temporal descriptions and explicit spatio-temporal evidence, respectively, without external tool execution. These approaches demonstrate the importance of explicit perceptual content, but structured descriptions and grounding rewards do not by themselves ensure evidence-driven reasoning. Our focus is the behavioral connection between evidence and reasoning: identifying what information is needed, recognizing when it is insufficient, and revising conclusions when observations challenge an earlier interpretation. We address these behaviors through evidence-conditioned SFT demonstrations and subsequent RL, extending beyond textual evidence representation to adaptive interaction with the video.

Active Visual Reasoning. Beyond textual CoT, the paradigm of *thinking with images* OpenAI (2025) enables models to interleave reasoning with visual operations that provide additional observations. DeepEyes Zheng et al. (2025) explores RL-trained image inspection, while VITAL Zhang et al. (2025b) extends tool-augmented reasoning to video through on-demand frame sampling. Frame-Thinker He et al. (2025) learns multi-turn frame selection, whereas LongVT Yang et al. (2026a) and VideoZoomer Ding et al. (2025) combine coarse overviews with targeted temporal inspection, including demonstrations of correction or reflection. Video-o3 Zeng et al. (2026) further studies iterative clue seeking and efficient termination, and ParaVT Yang et al. (2026b) addresses parallel inspection and tool-use training through format stabilization and frame-budget randomization. These advances establish visual interaction as a learned reasoning capability, but executing an inspection does not itself ensure that the returned evidence appropriately guides subsequent reasoning. Complementing these action and optimization designs, we focus on evidence-conditioned behavior: assessing whether inspection is needed, recovering when it fails, and revising conclusions when observations warrant. Our SFT construction explicitly covers different evidence states and recovery continuations, followed by RL that balances answer correctness, evidence use, and inspection efficiency.

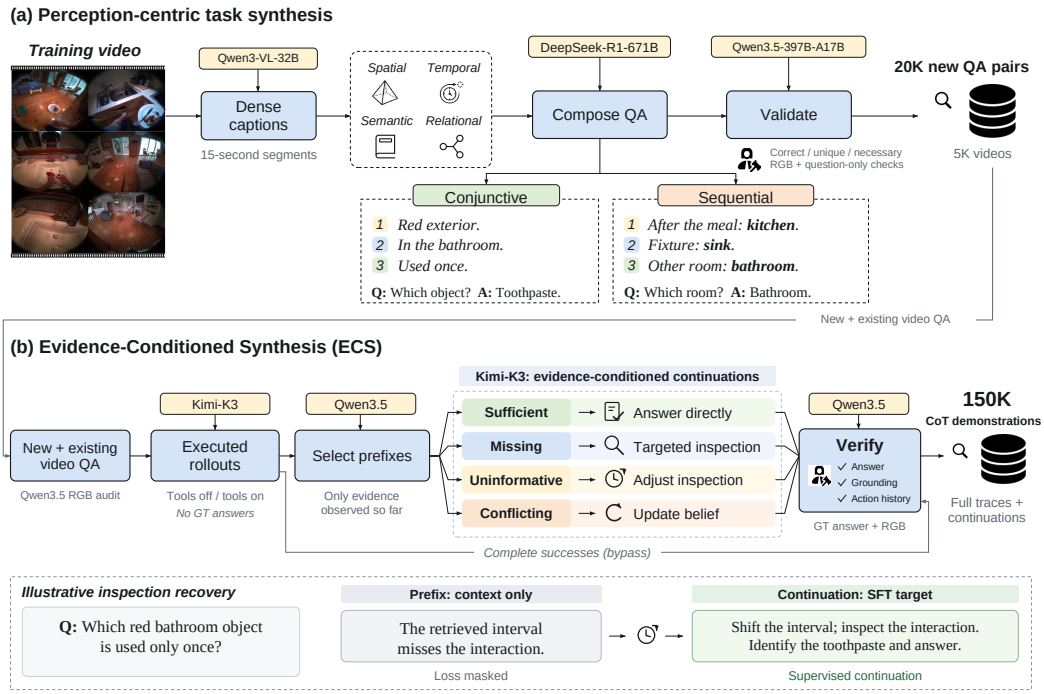

Figure 2: **Task and CoT synthesis for evidence-driven active visual reasoning.** **(a)** Dense video captions support conjunctive and sequential question composition, followed by automated validation to produce EviVideo-QA-20K with 20K perception-centric QA pairs. **(b)** Evidence-Conditioned Synthesis (ECS) collects executed rollouts and generates verified continuations from selected evidence states, covering direct answering, targeted inspection, inspection recovery, and belief updating. Complete successful trajectories and accepted continuations form EviCoT-150K, containing 150K demonstrations. The schematic example illustrates inspection recovery: the supplied prefix serves as context with its loss masked, while the verified continuation provides the SFT target.

3 METHOD

3.1 ACTIVE VISUAL REASONING

We study complex video reasoning: given a video V and a question q, an MLLM policy π_θ generates an interleaved trajectory τ ending with a final answer y. The policy initially observes sparsely sampled RGB frames $o_0 = \mathcal{S}(V)$ and can inspect the source video during reasoning. We formulate *active visual reasoning* as an interleaved CoT:

$$\tau = (r_0, a_1, o_1, r_1, \ldots, a_K, o_K, r_K, y), \tag{1}$$

where r_k denotes textual reasoning, a_k a policy-generated visual action, and o_k its returned observation. The policy conditions subsequent reasoning, actions, and the final answer on (q, o_0) and the accumulated history. The policy determines K, with $K = 0$ corresponding to text-only reasoning over the initial frames.

The visual action space consists of two parameterized operations:

$$\mathcal{A}_{\text{visual}} = \{\text{TemporalInspect}(t_s, t_e, \rho),\ \text{FrameSelect}(t)\}\,. \tag{2}$$

TemporalInspect samples the interval $[t_s, t_e]$ at ρ frames per second, while FrameSelect retrieves a single frame at timestamp t within a video segment. Both return timestamped RGB observations from V, which are appended to the history before generation resumes.

Our training objective is to make active visual reasoning more *evidence-driven*: the policy should let available and missing visual evidence guide its reasoning, inspection and stopping decisions, and update its conclusions when new observations warrant. We pursue this objective by synthesizing SFT

demonstrations of evidence-responsive reasoning and new tasks that require acquiring and integrating visual evidence across reasoning steps, together with reward shaping that reinforces these behaviors.

3.2 SUPERVISED FINE-TUNING

We construct SFT data that teach the policy to reason from available visual evidence and acquire additional observations when necessary. As illustrated in Fig. 2, our pipeline combines perception-centric task synthesis with evidence-conditioned CoT construction: the former establishes visual evidence requirements, while the latter teaches the policy how to satisfy them.

Evidence-conditioned CoT construction. We propose *Evidence-Conditioned Synthesis* (ECS), using Kimi-K3 Kimi Team (2026) for rollout generation and continuation, and Qwen3.5-397B-A17B Qwen Team (2026) for offline auditing and verification. Both models remain frozen throughout data construction. Starting from existing video QA pairs and our synthesized tasks, Qwen first checks RGB-only answerability and reference-answer support against the source video, excluding ambiguous or unsupported examples. For each retained pair, Kimi generates tool-disabled and tool-enabled rollouts under the student's initial observation configuration, without access to reference answers or annotated evidence intervals. Visual actions are executed on the source video, and their actual returned frames are appended before generation resumes.

In a separate prefix-only pass, Qwen examines the question, reasoning, and observations available at each step to identify branch points involving sufficient evidence, unresolved information needs, unsuccessful inspections, or unsupported interpretations. Kimi then generates continuations from the selected prefixes using the same visual action interface, without receiving reference answers or privileged verification feedback. These continuations are selected to demonstrate direct reasoning when evidence suffices, targeted inspection when information is missing, adjustment after uninformative observations, and belief revision when new evidence warrants it. The same question can yield different demonstrations depending on its observation history; neither inspection nor revision is required for every example. We retain complete successful rollouts alongside these prefix-conditioned continuations.

Finally, Qwen verifies candidates in a fresh evaluation context, using the reference answer and recorded RGB observations. We retain candidates that pass checks for final-answer correctness, stepwise visual grounding, reasoning coherence, and evidence-responsive inspection and stopping. Grounding checks cover the entire supervised trajectory for complete rollouts and the newly supervised continuation for prefix-conditioned examples. Unsupported claims inherited from a failed prefix must be corrected rather than propagated. Later observations cannot justify earlier factual claims, and action requests must match the actual execution log. Candidates requiring repair are regenerated from a retained prefix through further execution, without retrospectively altering observations or tool outcomes. After verification and deduplication, we obtain EviCoT-150K, comprising 150K SFT demonstrations. Sampling budgets, branch-selection criteria, verification prompts, and dataset composition are provided in the supplementary material.

Perception-centric task synthesis. We construct EviVideo-QA-20K, comprising 20K questions from 5K real-world video clips, to strengthen the demand for multi-step visual evidence integration. Qwen3-VL-32B-Instruct Bai et al. (2025) produces dense descriptions of 15-second segments, from which DeepSeek-R1-671B Guo et al. (2025) composes questions involving objects, attributes, spatial relations, and events. *Conjunctive composition* combines jointly necessary conditions to identify a unique referent, while *sequential composition* links dependent reasoning steps across entities and events. Each condition specifies visual information needed to resolve the question, making evidence integration central to the task. Answers are short and open-ended, reducing option-based shortcuts while supporting normalized verification.

The verifier described above checks answer correctness, uniqueness, and reference consistency in separate judging contexts; clause-removal tests assess the necessity of the composed conditions. A question-only answering pass filters likely language shortcuts, followed by source-RGB auditing to reject caption-induced errors and unsupported answers. Accepted pairs enter ECS, where inspection remains conditional on the available evidence rather than mandatory for every question. Video sources, prompts, filtering rubrics, and dataset statistics are provided in the supplementary material.

Supervised objective. We optimize the policy on the resulting demonstrations using a masked autoregressive objective:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}_{\mathrm{SFT}}}\left[\frac{\sum_{j=1}^{|x|} m_j \log \pi_\theta(x_j \mid x_{<j})}{\sum_{j=1}^{|x|} m_j}\right]. \tag{3}$$

Here, $m_j = 1$ for target assistant reasoning, action requests, and final answers. Input instructions, visual observations, and tool responses are masked with $m_j = 0$. For all prefix-conditioned demonstrations, the supplied prefix is also masked, so supervision applies only to the verified continuation, without training the policy to reproduce any preceding errors.

3.3 Reinforcement Learning

Starting from the SFT policy, we use GRPO Shao et al. (2024) to further develop evidence-driven active visual reasoning through the policy's own video interactions. We train on the full collection of video QA data, including EviVideo-QA-20K. For each question, the policy samples multiple reasoning trajectories under identical initial observations and may invoke the visual actions defined in section 3.1 when needed. RL uses ground-truth answers for reward computation, without requiring the synthesized CoT demonstrations used during SFT.

Our reward combines answer correctness, output validity, evidence use, and inspection cost:

$$R(\tau) = 2R_{\mathrm{acc}}(\tau) + R_{\mathrm{fmt}}(\tau) + 0.5R_{\mathrm{acc}}(\tau)U(\tau) - 0.1C(\tau). \tag{4}$$

Here, $R_{\mathrm{acc}} \in \{0, 1\}$ verifies the final answer, and $R_{\mathrm{fmt}} \in \{0, 1\}$ checks the answer format and any emitted action requests. Unlike a tool-use bonus, $U(\tau) \in [0, 1]$ evaluates how the trajectory identifies evidence requirements, assesses sufficiency, adapts inspection, and updates conclusions. We obtain this score from a frozen Qwen3-VL-32B-Instruct evaluator, without additional evaluator training. It receives the question and the recorded interaction history, including actual RGB observations, and applies the fixed rubric in Appendix B. Direct reasoning can receive full credit; inspection and belief revision are rewarded only when warranted by the evidence.

We penalize inspection cost as

$$C(\tau) = \frac{1}{2}\min\left(1, \frac{N_{\mathrm{call}}}{B_{\mathrm{call}}}\right) + \frac{1}{2}\min\left(1, \frac{N_{\mathrm{vis}}}{B_{\mathrm{vis}}}\right), \tag{5}$$

where N_{call} counts executed inspections, N_{vis} counts additional visual tokens returned by tools. We set the normalization scales to $B_{\mathrm{call}} = 4$ and $B_{\mathrm{vis}} = 4096$. These values normalize the cost rather than impose execution limits. The bounded penalty discourages unnecessary inspection while preserving answer correctness as the dominant reward.

We normalize trajectory rewards within each rollout group and optimize the standard clipped GRPO objective with KL regularization to the frozen SFT policy. Only policy-generated reasoning, action requests, and answer tokens contribute to the policy-gradient and KL terms; visual observations and tool responses are masked. All token probabilities remain conditioned on the preceding interaction history.

4 Experiments

4.1 Setup

We evaluate complex video reasoning on MINERVA Nagrani et al. (2025), Minerva-Ego Nagrani et al. (2026), and Video-Holmes Cheng et al. (2025), and general video understanding on Video-MME Fu et al. (2025c) and Video-MME-v2 Fu et al. (2026).

We initialize EviVideo from Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct Bai et al. (2025) and train through SFT followed by GRPO Shao et al. (2024). The training pool combines EviVideo-QA-20K with the video QA data used by Video-R1 Feng et al. (2025) and LongVT Yang et al. (2026a) for both CoT synthesis and RL. At inference, our policy receives 32 uniformly sampled frames and may perform up to four visual inspections. We use no audio, subtitles, or external ASR models. Additional training and evaluation settings are provided in the supplementary material.

Table 1: Comparison on three complex video reasoning benchmarks and two general video understanding benchmarks. All models receive 32 initial frames, without audio or subtitles. Active visual reasoning models may acquire additional frames through visual inspection. Shaded rows denote our models. Higher scores are better.

Models	Complex Video Reasoning			General Video Understanding	
	MINERVA	Minerva-Ego	Video-Holmes	Video-MME	Video-MME-v2
Proprietary models					
GPT-6-Astra	62.7	52.8	69.8	87.8	54.5
Gemini-3.8-Flash	66.7	47.8	66.1	86.9	49.2
Textual video reasoning models					
Video-R1 Feng et al. (2025)	27.4	24.6	36.1	54.8	21.8
VideoRFT Wang et al. (2025b)	31.8	24.2	36.8	57.3	23.4
VideoChat-R1 Li et al. (2025)	27.2	25.4	35.2	56.4	22.6
LongVILA-R1 Chen et al. (2025c)	25.6	25.2	33.4	59.1	23.8
Video-PLR Pu et al. (2025)	28.6	26.4	36.5	58.5	24.2
ReWatch-R1 Zhang et al. (2025a)	33.4	26.0	38.2	60.2	25.6
Active visual reasoning models					
VITAL Zhang et al. (2025b)	29.6	27.2	36.4	60.4	24.4
LongVT Yang et al. (2026a)	24.2	27.0	32.5	59.6	23.6
Video-o3 Zeng et al. (2026)	30.8	25.8	37.2	62.2	25.2
VideoTemp-o3 Liu et al. (2026)	28.4	24.0	37.8	61.4	24.8
LOVE-R1 Fu et al. (2025d)	23.2	22.4	32.6	56.8	21.4
VideoZoomer Ding et al. (2025)	33.6	27.4	38.5	63.0	26.2
FrameThinker He et al. (2025)	30.4	28.2	37.6	62.6	25.8
Conan Ouyang et al. (2025)	24.6	24.8	34.2	59.2	23.2
Our models and base policies					
Qwen3-VL-4B-Instruct Bai et al. (2025)	33.6	25.4	34.5	61.8	23.8
EviVideo-4B	**40.2**	**32.0**	**44.0**	**65.2**	**28.4**
Qwen3-VL-8B-Instruct Bai et al. (2025)	36.1	28.6	36.8	67.6	27.4
EviVideo-8B	**43.6**	**35.4**	**46.2**	**70.4**	**31.8**

4.2 Main Results

Table 1 compares EviVideo with base policies and existing video reasoning models. The results suggest that our training approach improves both complex video reasoning and general video understanding. All improvements below denote absolute accuracy gains.

On complex video reasoning, EviVideo consistently improves over its corresponding Qwen3-VL backbone. The 4B model gains 6.6%, 6.6%, and 9.5% on MINERVA, Minerva-Ego, and Video-Holmes, respectively; the 8B model achieves corresponding gains of 7.5%, 6.8%, and 9.4%. Both variants also outperform the evaluated textual and active visual reasoning baselines on these benchmarks. These comparisons suggest that enabling visual interaction alone is insufficient: training how the policy assesses, acquires, and integrates evidence is important for effective active visual reasoning.

The improvements extend to general video understanding. On Video-MME and Video-MME-v2, EviVideo-4B improves over its backbone by 3.4% and 4.6%, while EviVideo-8B gains 2.8% and 4.4%. The larger gains on the complex reasoning benchmarks are consistent with evidence-driven active visual reasoning being particularly beneficial when answering requires connecting multiple observations and resolving uncertainty across events. Meanwhile, the gains on both general benchmarks suggest that these learned behaviors also benefit broader video understanding.

4.3 Ablation Study

We conduct ablations with Qwen3-VL-4B-Instruct on MINERVA, Minerva-Ego, and Video-Holmes to assess the contribution of each proposed component: CoT SFT, perception-centric tasks, and the evidence-aware RL reward. Starting from the full method, we remove one component at a time. *w/o CoT SFT* skips SFT and applies RL directly to the Instruct model. *w/o New QA* removes EviVideo-QA-20K from both SFT and RL, replacing it with existing QA data to maintain training budgets. *w/o Evidence Reward* removes the evidence-use score and inspection cost, retaining correctness and format rewards. All variants share the same RL training budget and inference settings, with 32 initial frames and up to four visual inspections.

As shown in Table 2, removing CoT SFT produces the largest degradation, reducing average accuracy from 38.7% to 34.6%. This suggests that supervised demonstrations provide an important foundation that subsequent RL does not fully recover on its own. Removing the new QA tasks reduces average accuracy to 36.2%, supporting the value of training on questions that require composing visual evidence across entities and events. Finally, replacing our reward with correctness and format rewards reduces average accuracy to 35.9%, indicating that explicitly guiding evidence use and inspection efficiency provides benefits beyond final-answer supervision.

Table 2: Component ablations on Qwen3-VL-4B. Accuracy (%); Ego and Holmes denote Minerva-Ego and Video-Holmes, respectively.

Setting	MINERVA	Ego	Holmes	Avg.
Full	**40.2**	**32.0**	**44.0**	**38.7**
w/o CoT SFT	35.6	28.8	39.4	34.6
w/o New QA	37.3	28.9	42.3	36.2
w/o Evidence Reward	37.4	29.6	40.8	35.9

Each removal lowers performance across all three benchmarks. Together, these results support the contribution of all three components: CoT demonstrations establish a behavioral foundation, perception-centric tasks create demands for visual evidence, and the RL reward guides how the policy acquires and uses it.

5 Analysis

Beyond component ablations, we examine whether our CoT construction improves learning, whether the resulting policy exhibits more evidence-driven behavior, and whether its gains can be explained by additional visual input alone.

Effect of CoT construction. We compare our regenerated CoT demonstrations with the original SFT CoTs released by LongVT Yang et al. (2026a), using the same QA pairs from the LongVT training data. The baseline uses LongVT's original demonstrations, whereas our variant replaces them with demonstrations produced by our construction pipeline. SFT demonstrations for our new tasks and other QA sources remain unchanged. Both settings use matched supervised-token budgets and identical subsequent RL tasks, rewards, and inference settings. This paired comparison controls the underlying questions, although the teacher models and synthesis procedures may differ.

Table 3: CoT data comparison on matched QA pairs with Qwen3-VL-4B-Instruct. SFT token budgets, subsequent RL, and inference settings are matched. Scores are accuracy (%).

SFT CoT	MINERVA	Minerva-Ego	Video-Holmes	Avg.
Original LongVT CoT	37.3	29.1	40.9	35.8
Ours	**40.2**	**32.0**	**44.0**	**38.7**

As shown in Table 3, replacing LongVT's original CoTs with our demonstrations improves accuracy from 37.3% to 40.2% on MINERVA, 29.1% to 32.0% on Minerva-Ego, and 40.9% to 44.0% on Video-Holmes. Mean accuracy increases from 35.8% to 38.7%. These consistent gains suggest that our demonstrations provide more effective supervision than the original CoTs for the same questions, with benefits retained after identical subsequent RL.

Evidence-driven reasoning behaviors. Beyond the examples in Fig. 1, we evaluate whether reasoning and inspection decisions respond to the available evidence. We compare LongVT, our SFT-only policy, and the full model on matched continuation tests constructed from held-out video questions and actual inspection outputs. Each model receives the same question, observations, and interaction prefix where applicable, without reference answers. We evaluate four conditions, each containing 50 eligible states: insufficient evidence, sufficient evidence, unsuccessful inspection, and observations contradicting an earlier interpretation. Success is assessed from the actual continuation and visual evidence, rather than particular reasoning phrases.

Figure 3 shows that our SFT-only policy achieves higher success rates than LongVT across all four conditions, with further improvements after RL. The full model inspects appropriately in 78% of insufficient-evidence cases and answers without further inspection in 84% of sufficient-evidence cases. The largest differences from LongVT concern corrective behavior: inspection adjustment increases from 24% to 70%, and evidence-supported belief revision from 30% to 72%. These results

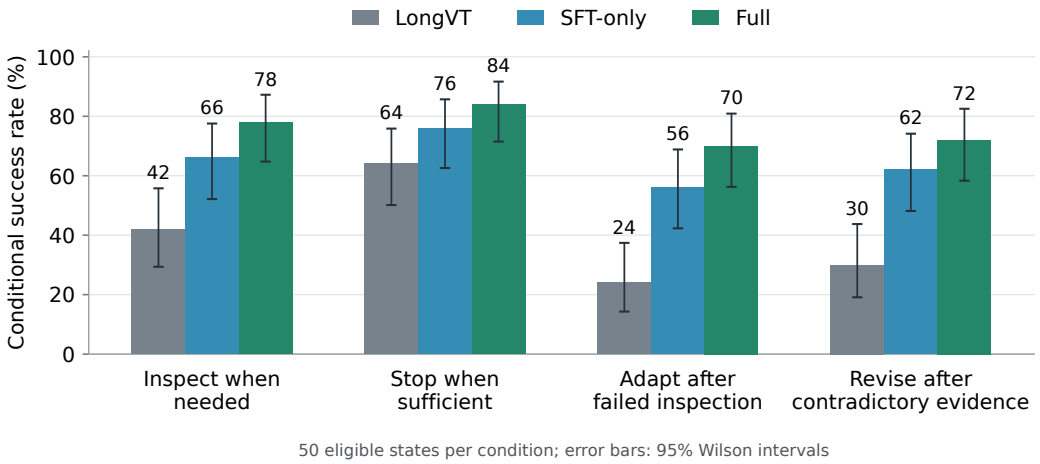

Figure 3: **Evidence-driven reasoning behaviors.** Conditional success rates on matched continuation tests, with 50 eligible states per condition. Models are evaluated on inspecting when evidence is missing, stopping when it is sufficient, adjusting unsuccessful inspections, and revising interpretations contradicted by new observations. Error bars denote 95% Wilson confidence intervals.

suggest that the improvements extend beyond tool invocation to adapting actions and interpretations in response to observed evidence. State construction, assessment procedures, and additional qualitative examples appear in Appendix.

Adaptive inspection versus more frames. We examine whether adaptive inspection improves reasoning beyond simply increasing visual input. Using the same trained EviVideo-4B checkpoint, we compare uniform sampling of 32 or 128 frames without tools against 32 initial frames followed by up to four adaptive inspections. All settings share the same per-frame resolution, visual encoding configuration, and generated-token budget. We report mean accuracy across the three complex reasoning benchmarks and average usage over all questions.

Table 4: Accuracy and visual input usage with the same EviVideo-4B checkpoint. Frames and calls are per-question averages.

Setting	Acc. (%)	Frames	Calls
Uniform-32	34.2	32	0
Uniform-128	38.2	128	0
Adaptive	**38.7**	58.4	2.1

As shown in Table 4, increasing uniform sampling from 32 to 128 frames improves accuracy from 34.2% to 38.2%. Adaptive inspection achieves 38.7% accuracy while using 54.4% fewer frames than uniform-128, with 2.1 calls per question. This comparison suggests that adaptive evidence acquisition can achieve slightly higher accuracy with substantially fewer frames, rather than relying solely on denser visual input.

6 Conclusion

We study how to make active visual reasoning more evidence-driven for complex video reasoning. Our approach combines evidence-conditioned CoT construction, perception-centric tasks, and an evidence-aware RL reward to train EviVideo through SFT and GRPO. Together, these components encourage the policy to assess evidence sufficiency, inspect selectively, adapt unsuccessful inspections, and revise interpretations based on new observations. Experiments show improvements across complex video reasoning and general video understanding benchmarks, while behavioral analyses indicate more evidence-responsive reasoning and interaction. Our findings suggest that effective active visual reasoning requires not simply more tool calls or visual input, but training policies to determine when additional evidence is needed, how to acquire it, and how it should inform their conclusions.

AI USE STATEMENT

We used generative AI tools to help formulate and refine hypotheses, develop and critique research methods and experimental designs, implement methods, support translation and writing, and clean and reformat data. We reviewed all AI-assisted contributions and take full responsibility for the final content of this work, including its text, claims, and artifacts.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Boyu Chen, Zhengrong Yue, Siran Chen, Zikang Wang, Yang Liu, Peng Li, and Yali Wang. LVAgent: Long video understanding by multi-round dynamical collaboration of MLLM agents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20237–20246, 2025a.

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, 2025b. URL https://arxiv.org/abs/2507.07966.

Zhen Chen, Hao Zhang, Ziyang Liu, Kaituo Feng, Chuang Gan, Yao Ma, and Tianjun Liu. Scaling rl to long videos. *arXiv preprint arXiv:2507.07966*, 2025c.

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

Yang Ding, Yizhen Zhang, Xin Lai, Ruihang Chu, and Yujiu Yang. VideoZoomer: Reinforcement-learned temporal focusing for long video reasoning, 2025. URL https://arxiv.org/abs/2512.22315.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. VideoAgent: A memory-augmented multimodal agent for video understanding, 2024. URL https://arxiv.org/abs/2403.11481.

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

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, 2025a. URL https://arxiv.org/abs/2405.21075.

Chaoyou Fu, Haozhi Yuan, Yuhao Dong, Yi-Fan Zhang, Yunhang Shen, Xiaoxing Hu, Xueying Li, Jinsen Su, Chengwu Long, Xiaoyao Xie, Yongkang Xie, Xiawu Zheng, Xue Yang, Haoyu Cao, Yunsheng Wu, Ziwei Liu, Xing Sun, Caifeng Shan, and Ran He. Video-MME-v2: Towards the next stage in benchmarks for comprehensive video understanding, 2026. URL https://arxiv.org/abs/2604.05015.

Dayuan Fu, Keqing He, Yejie Wang, Wentao Hong, Zhuoma GongQue, Weihao Zeng, Wei Wang, Jingang Wang, Xunliang Cai, and Weiran Xu. AgentRefine: Enhancing agent generalization through refinement tuning. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=FDimWzmcWn.

Hanning Fu, Jingyu Zhang, Jia Li, Yao Guo, Yumeng Zhao, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025c.

Shenghao Fu, Qize Yang, Yuan-Ming Li, Xihan Wei, Xiaohua Xie, and Wei-Shi Zheng. LOVE-R1: Advancing long video understanding with an adaptive zoom-in mechanism via multi-step reasoning. *arXiv preprint arXiv:2509.24786*, 2025d.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, 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, 2025. URL https://arxiv.org/abs/2509.24304.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. DeepEyesV2: Toward agentic multimodal model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yDKawwfJ5O.

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, 2025. URL https://arxiv.org/abs/2503.09516.

Kimi Team. Kimi K3: Open frontier intelligence. Technical report, Moonshot AI, 2026. URL https://github.com/MoonshotAI/Kimi-K3/blob/main/k3_tech_report.pdf.

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.

Zichao Li, Gang Wu, Zichao Wang, Ruiyi Zhang, Wanrong Zhu, Ryan A. Rossi, Vlad I Morariu, and Jihyung Kil. Spinning straw into gold: Relabeling LLM agent trajectories in hindsight for successful demonstrations, 2026. URL https://arxiv.org/abs/2607.04235.

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.

Jiahao Meng, Xiangtai Li, Haochen Wang, Yue Tan, Tao Zhang, Lingdong Kong, Yunhai Tong, Anran Wang, Zhiyang Teng, Yujing Wang, and Zhuochen Wang. Open-o3 Video: Grounded video reasoning with explicit spatio-temporal evidence. *arXiv preprint arXiv:2510.20579*, 2025. URL https://arxiv.org/abs/2510.20579.

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, 2025. URL https://arxiv.org/abs/2505.00681.

Arsha Nagrani, Jasper Uijilings, Shyamal Buch, Tobias Weyand, Sudheendra Vijayanarasimhan, Bo Hu, Ramin Mehran, David A Ross, and Cordelia Schmid. Minerva-Ego: Spatiotemporal hints for egocentric video understanding, 2026. URL https://arxiv.org/abs/2605.15342.

OpenAI. Thinking with images. OpenAI Blog, April 2025. URL https://openai.com/index/thinking-with-images/.

Kun Ouyang, Yuanxin Liu, Linli Yao, Yishuo Cai, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Conan: Progressive learning to reason like a detective over multi-scale visual evidence. *arXiv preprint arXiv:2510.20470*, 2025.

Bowei Pu, Chuanbin Liu, Yifan Ge, Peichen Zhou, Yiwei Sun, Zhiyin Lu, Jiankang Wang, and Hongtao Xie. Alternating perception-reasoning for hallucination-resistant video understanding, 2025. URL https://arxiv.org/abs/2511.18463v1.

Lu Qiu, Yi Chen, Yuying Ge, Yixiao Ge, Ying Shan, and Xihui Liu. Egoplan-bench2: A benchmark for multimodal large language model planning in real-world scenarios, 2025. URL https://arxiv.org/abs/2412.04447.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Qisheng Su, Zhen Fang, Shiting Huang, Yu Zeng, Yiming Zhao, Kou Shi, Ziao Zhang, Lin Chen, Zehui Chen, Lijun Wu, and Feng Zhao. ACC: Compiling agent trajectories for long-context training, 2026. URL https://arxiv.org/abs/2605.21850.

Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via python execution for reasoning, 2023. URL https://arxiv.org/abs/2303.08128.

Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning, 2025a. URL https://arxiv.org/abs/2505.15966.

Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. Videorft: Incentivizing video reasoning capability in mllms via reinforced fine-tuning. *ArXiv*, abs/2505.12434, 2025b. URL https://api.semanticscholar.org/CorpusID:278741017.

Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. VideoRFT: Incentivizing video reasoning capability in MLLMs via reinforced fine-tuning, 2025c. URL https://arxiv.org/abs/2505.12434.

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, 2025d. URL https://arxiv.org/abs/2510.23473.

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, 2024a.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents, 2024b. URL https://arxiv.org/abs/2402.01030.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 3272–3283, 2025e.

Ziyi Wang, Yuxuan Lu, Yimeng Zhang, Pei Chen, Ziwei Dong, Jing Huang, Jiri Gesi, Xianfeng Tang, Chen Luo, Qun Liu, Yisi Sang, Hanqing Lu, Manling Li, Jin Lai, and Dakuo Wang. Trajectory2Task: Training robust tool-calling agents with synthesized yet verifiable data for complex user intents, 2026. URL https://arxiv.org/abs/2601.20144.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. WebDancer: Towards autonomous information seeking agency, 2025. URL https://arxiv.org/abs/2505.22648.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. MM-REACT: Prompting ChatGPT for multimodal reasoning and action, 2023. URL https://arxiv.org/abs/2303.11381.

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, 2026a. URL https://arxiv.org/abs/2511.20785.

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, 2026b. URL https://arxiv.org/abs/2605.20342.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, Ziang Yan, Yi Wang, Hongjie Zhang, Yali Wang, and Limin Wang. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning, 2026. URL https://arxiv.org/abs/2601.23224.

Congzhi Zhang, Zhibin Wang, Yinchao Ma, Jiawei Peng, Yihan Wang, Qiang Zhou, Jun Song, and Bo Zheng. Rewatch-r1: Boosting complex video reasoning in large vision-language models through agentic data synthesis. *arXiv preprint arXiv:2509.23652*, 2025a.

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, 2025b. URL https://arxiv.org/abs/2508.04416.

Shitian Zhao, Shaoheng Lin, Ming Li, Haoquan Zhang, Wenshuo Peng, Kaipeng Zhang, and Chen Wei. PyVision-RL: Forging open agentic vision models via RL, 2026. URL https://arxiv.org/abs/2602.20739.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025. URL https://arxiv.org/abs/2505.14362.

Zhuo Zhi, Qiangqiang Wu, Minghe Shen, Wenbo Li, Yinchuan Li, Kun Shao, and Kaiwen Zhou. VideoAgent2: Enhancing the LLM-based agent system for long-form video understanding by uncertainty-aware CoT, 2025. URL https://arxiv.org/abs/2504.04471.

A Implementation Details

A.1 Prompts used in complex vidoe QA synthesis

We present the exact prompts we used to generate EviVideo-QA-20K. Only minor formatting edits were made to improve readability:

- Figure 7 shows the prompt used for dense video captioning with Qwen3-32B.
- Figures 5 and 6 show the prompts used to synthesize complex video QA pairs with DeepSeek-R1-671B, by combining sub-conditions based on conjunction and sequential logic.

A.2 Evaluation Benchmarks

We evaluate on three complex video reasoning benchmarks that emphasize different aspects of long-horizon, perception-heavy reasoning.

Video-Holmes. Video-Holmes is a benchmark for Sherlock-Holmes-style complex video reasoning Cheng et al. (2025). It contains 270 manually annotated suspense short films, each 1–5 minutes long, and 1,837 questions spanning seven tasks. Questions are constructed by first identifying key events and causal relationships in each film, then designing queries that require models to actively locate and connect multiple visual clues scattered across different segments of the video. This makes Video-Holmes a natural probe of abductive and causal video reasoning, where success depends on integrating many subtle perceptual cues rather than spotting a single frame.

Minerva. Minerva evaluates complex video reasoning over real-world videos from diverse domains such as documentaries, sports, and vlogs Nagrani et al. (2025). The benchmark comprises a few hundred videos (about 1–2 minutes on average) and over one thousand carefully annotated question–answer pairs. Each question is designed to require multi-hop reasoning: the model must track entities across time, compare properties at different moments or viewpoints, and reason about temporal and spatial relations, rather than answering from a single salient frame.

EgoPlan-Bench2. EgoPlan-Bench2 targets egocentric visual planning in everyday scenarios Qiu et al. (2025). It contains 1,321 multiple-choice QA pairs built from 1,113 first-person videos sourced from Ego4D, covering four major domains (Work, Daily life, Hobbies, Recreation) and 24 fine-grained scenarios. The questions ask the model to choose the next action or intermediate goal (for example, which object to interact with next or which location to move toward), so solving them requires combining perception with an implicit understanding of human goals and affordances.

Why these are complex video reasoning tasks. What makes these benchmarks "complex" is not just that current models perform poorly, but that solving them requires all of the following: (i) integrating visual evidence over time and across views, (ii) maintaining a structured world state that tracks entities, relations, and goals throughout the video, and (iii) performing multi-step inference, explanation, or planning on top of that evolving state. In all three benchmarks, a model that only looks at a few frames or applies shallow pattern matching cannot reliably succeed; it must instead carry out genuine video-based thinking.

Performance. Under comparable resources, Video-R1 is a strong compute-matched RLVR baseline. Building on it, our method achieves consistent gains, indicating that the improvements are not simply due to larger scale (e.g., more data or compute). Although Video-MME is also perception-oriented, its perceptual difficulty is relatively low and thus offers limited headroom: answering its questions typically does not require fine-grained, repeated inspection of the video. As a result, gains on Video-MME are smaller and are most visible under frame-limited settings, whereas substantially larger improvements emerge on more complex benchmarks that require richer evidence gathering, such as visual abductive reasoning, multi-hop reasoning, and planning.

```
You are a dense-caption engine for video. Be exhaustive and recall-heavy: state everything you can reliably ground in the
frames and/or audio, err on the side of inclusion, and leave nothing important unsaid. Do NOT invent facts beyond the
video/audio.

You may use both visual frames and audio. If the audio is unclear, do not guess exact words or language.

OUTPUT FORMAT:

Your output must be a single JSON object, and nothing else. No natural language outside JSON. No markdown. No backticks.

The JSON object must have the following structure:

{
  "video_uri": "<string>",                    // identifier of the video clip; if none is given, use "".
  "overview": "<string>",                     // 2–4 sentences summarizing the entire clip
  "key_locations": ["<string>", ...],         // deduplicated scene/place names (e.g., "kitchen", "office", "street")
  "instances": [                              // salient entities with stable IDs reused across events/relations
    {
      "id": "<string>",                       // e.g., "person_1", "car_2", "table_1", "dog_1"
      "category": "<string>",                 // short category label (e.g., "person", "car", "table")
      "attributes": [
        {"name": "<string>", "value": "<string>"}  // e.g., {"name": "shirt_color", "value": "red"}
      ],
      "first_seen": "MM:SS",
      "last_seen": "MM:SS",
      "actions": ["<string>", ...],           // concise verb phrases (e.g., "walks forward", "pours water")
      "locations": ["<string>", ...]          // e.g., "on sidewalk", "by counter"
    }
  ],
  "events": [                                 // segment by shot/semantic change (~5–15s; adapt as needed)
    {
      "title": "<string>",
      "start": "MM:SS",
      "end": "MM:SS",
      "summary": "<string>",                  // 2–4 sentences summarizing this event segment
      "actions": ["<string>", ...],           // main actions in this segment
      "instances": ["<string>", ...],         // instance IDs involved in this segment
      "relationships": [                      // only these types: "spatial", "temporal", "semantic"
        {
          "type": "<string>",                 // one of: "spatial", "temporal", "semantic"
          "subject_id": "<string>",           // must be an id from instances[*].id
          "object_id": "<string or null>",    // null if not applicable; otherwise an existing instances[*].id
          "description": "<string>",          // natural language description of the relation
          "at": "MM:SS or null",              // use for point-like relations
          "start": "MM:SS or null",           // use for interval-like relations
          "end": "MM:SS or null"
        }
      ]
    }
  ],
  "timeline": [                               // optional; dense 1–3 sentence captions every 5–10s or per shot
    {
      "start": "MM:SS",
      "end": "MM:SS",
      "caption": "<string>"
    }
  ]
}

REQUIREMENTS:

- Timestamps strictly as MM:SS (relative to the provided clip only).
- Reuse stable instance IDs consistently across the whole JSON.
- Every relationship.subject_id and relationship.object_id (if not null) must reference an existing instances[i].id.
- If an instance appears in an event's "instances" list, its [first_seen, last_seen] should overlap with the event's [start,
end] when possible.
- Be comprehensive: enumerate all noticeable objects, attributes, locations, actions, events, and relationships you can
reliably ground.
- Prefer recall: if confidence is low but still grounded, include it and append "(low confidence)" inside the relevant
string.
- Do NOT include any comments, type hints, or examples in the JSON itself. Only output raw JSON with concrete values.
- Output exactly one JSON object as the entire response.
```

Figure 4: **Dense-caption prompt.** VLM prompt used for generating dense captions.

A.3 Training Hyper-Parameters

We train all policies using the VERL PPO implementation with GRPO-style advantage estimation. Unless otherwise noted, both stages (with and without rewatch) share the same optimizer and rollout configuration.

We use a global train batch size of 256 prompts. Each prompt is truncated to at most 8,192 tokens, and the maximum generated response length is 20,480 tokens. Optimization uses Adam with a learning rate of 1×10^{-6}, GRPO mini-batch size 256, and micro-batch size 4 per GPU. We use the "low_var_kl" implementation as a penalty term.

Rollouts are produced with a vLLM-based engine, with a maximum of 32,768 batched tokens per forward pass and GPU memory utilization capped at 0.8. In the rewatch stage, for Qwen3-VL-8B, we generate 16 rollouts per prompt; for Qwen2.5-VL-7B, we generate 24 rollouts per prompt. The

```
LOGIC_TYPE = <conjunctive|chain>  # set EXACTLY one: "conjunctive" or "chain"
DISALLOWED_ANSWERS = []           # optional, e.g., ["green","left"]; leave empty to ignore

You are given ONE structured JSON annotation for a video. The JSON will appear between the markers:

=== ANNOTATION_JSON_START ===
... (JSON)
=== ANNOTATION_JSON_END ===

Your task: produce EXACTLY ONE QA item as a SINGLE JSON object. No extra text, no code fences, no Markdown.

IMPORTANT:
- Treat LOGIC_TYPE and DISALLOWED_ANSWERS as configuration only. DO NOT copy or mention them in the output.
- The output must be exactly one JSON object, nothing before or after.

OUTPUT JSON REQUIREMENTS:

The JSON object must have exactly these keys:
- "logic": a string
- "conditions": an array of 3 to 5 non-empty strings
- "answer": a string

The final JSON must look like this shape (example, do NOT copy the values):

{
  "logic": "conjunctive",
  "conditions": ["...", "...", "..."],
  "answer": "..."
}

FIELD RULES:

- "logic" must equal the exact string in LOGIC_TYPE given at the top ("conjunctive" or "chain").
- "conditions": 3–5 concise constraints as strings.
- "answer": EXACTLY ONE WORD (no spaces). It must be a generic, semantically meaningful token.

GLOBAL RULES:

- Use ONLY facts/entities from the given annotation; do NOT invent anything.
- If you mention time, use MM:SS format.
- Keep conditions short, factual, and strictly grounded.
- Design the conditions so that the final answer is UNIQUE under all conditions.
- Output ONLY the single JSON object; no explanations, no internal reasoning, no trailing text.

LOGIC DEFINITIONS (enforce depending on LOGIC_TYPE):

If LOGIC_TYPE == "conjunctive":
  - Each condition should narrow down candidates.
  - No single condition alone should uniquely determine the answer.
  - ONLY the conjunction of all conditions together should yield a UNIQUE answer.

If LOGIC_TYPE == "chain":
  - Each condition must DEPEND on the previous step (no independent or dead constraints).
  - Condition k must reference or refine something established by condition k-1.
  - Think of it as a step-by-step traversal over entities/segments/locations.
  - The final answer must be UNIQUE.

... (more in part 2)
```

Figure 5: Reasoning LLM prompt for generating complex video QA (part 1).

agent module is activated with at most 5 function-calling turns per rollout and at most 10,240 tokens per single response segment.

For model parallelism, we set tensor-model-parallel size to 1 and use FSDP with both parameter and optimizer offloading enabled for the actor and reference models. Logging is done via console, WandB, and an additional RL-specific dashboard, and checkpoints are saved periodically throughout training.

We run experiments on two backbones with different architectures: Qwen2.5 and Qwen3. The retrieval timestamp format is backbone-dependent: Qwen2.5-VL is more stable when using seconds, while Qwen3-VL prefers frame indices. Since Qwen-VL backbones are instruction-tuned for temporal grounding, prompting is sufficient to elicit the required localization behavior.

B Evidence-Use Reward Rubric

We use a frozen Qwen3-VL-32B-Instruct evaluator, without additional training, to assess whether a trajectory's reasoning and visual interactions are guided by the available evidence. The evaluator measures behavioral quality rather than merely checking for tool calls, particular reasoning phrases, or a correct final answer. Figures 8 and 9 provide the complete evaluator prompt.

```
... (continue from part 1)

ANSWER TOKEN CONSTRAINTS:

Internally, derive ALLOWED_TOKENS from the annotation (do NOT print this list):

Add tokens from:
  - key_locations (strings) → e.g., kitchen, street
  - instances[].category → e.g., car, dog, player
  - instances[].attributes[].name/value → e.g., color, green, wooden
  - instances[].actions[] → e.g., wave, eat
  - instances[].locations[] → e.g., corner, baseline, left
  - events[].title → only if the title is a single word
  - events[].actions[] → e.g., shoot, pass
  - events[].relationships[].type → spatial, temporal, semantic
  - events[].relationships[].description → split each description into tokens

Then filter ALLOWED_TOKENS:

Keep ONLY tokens that:
  - are single words with no whitespace,
  - have length ≤ 40 characters.

EXCLUDE any token that:
  - is exactly equal to any instances[].id (e.g., person_2, car_10),
  - contains any digit or underscore (matches /[0-9]|_/),
  - contains ":" (timestamps).

The final "answer" must:
  - be in this filtered ALLOWED_TOKENS,
  - be exactly ONE WORD (no spaces),
  - NOT be any instance ID,
  - NOT contain digits or underscores,
  - NOT be in DISALLOWED_ANSWERS.

PREFERENCE:

- Prefer generic, verification-friendly tokens such as: "green", "left", "car", "wave", "eat", "kitchen".
- If multiple tokens satisfy all constraints, refine your conditions so that exactly one remains valid.

UNIQUENESS CHECK:

- Under the provided conditions, there must be exactly one valid token in the filtered ALLOWED_TOKENS that can serve as
"answer".
- If you cannot find any such token, adjust or relax your conditions minimally so that exactly one token works.

FORMATTING CHECKLIST (hard requirements):

- Return ONLY one JSON object.
- No comments, no trailing commas, no placeholders like "<LOGIC_TYPE>" or "..." in the final JSON.
- "logic" must equal LOGIC_TYPE exactly.
- "conditions" must be an array with 3–5 non-empty strings.
- "answer" must be ONE WORD that satisfies all constraints above.

Now read the annotation below and produce the single JSON object as specified.
```

Figure 6: Reasoning LLM prompt for generating complex video QA (part 2).

Evaluator inputs. The evaluator receives the question, answer options when provided, initial RGB observations, and the complete chronological interaction history, including reasoning, action requests, execution results, and returned RGB frames. Steps and observations have identifiers, and frames retain their source timestamps. Reference answers and annotated target intervals are excluded; final-answer correctness is verified separately. The evaluator must assess each claim and decision using only observations available at that point. Evidence acquired later cannot retrospectively justify an earlier unsupported assertion.

Scoring dimensions. Each applicable dimension receives a score of 0, 1, or 2, corresponding to a clear deficiency, partial fulfillment, or strong fulfillment of the criterion. The four dimensions are defined as follows:

1. **Evidence requirements and grounding.** This dimension assesses whether the reasoning identifies the visual information needed to resolve the question and grounds substantive claims in available observations. The policy should distinguish directly observed facts from hypotheses and inferred conclusions. A score of 2 requires relevant evidence requirements and visually supported reasoning; 1 indicates partially specified requirements or minor unsupported inferences; 0 indicates that central reasoning relies on unsupported assertions or answer-option speculation. Tentative hypotheses are permitted and are not treated as factual errors merely because they remain unverified.
2. **Evidence sufficiency.** This dimension assesses whether inspection and stopping decisions reflect the information that remains unresolved. A score of 2 requires seeking additional observations when necessary and answering when the available evidence suffices. A score of 1 indicates partially justified decisions, with unclear evidence gaps or stopping reasons. A

```
You are a dense-caption engine for video. Be exhaustive and recall-heavy: state everything you can reliably ground in the
frames and/or audio, err on the side of inclusion, and leave nothing important unsaid. Do NOT invent facts beyond the
video/audio.

You may use both visual frames and audio. If the audio is unclear, do not guess exact words or language.

OUTPUT FORMAT:

Your output must be a single JSON object, and nothing else. No natural language outside JSON. No markdown. No backticks.

The JSON object must have the following structure:

{
  "video_uri": "<string>",                    // identifier of the video clip; if none is given, use "".
  "overview": "<string>",                     // 2–4 sentences summarizing the entire clip
  "key_locations": ["<string>", ...],         // deduplicated scene/place names (e.g., "kitchen", "office", "street")
  "instances": [                              // salient entities with stable IDs reused across events/relations
    {
      "id": "<string>",                       // e.g., "person_1", "car_2", "table_1", "dog_1"
      "category": "<string>",                 // short category label (e.g., "person", "car", "table")
      "attributes": [
        {"name": "<string>", "value": "<string>"}  // e.g., {"name": "shirt_color", "value": "red"}
      ],
      "first_seen": "MM:SS",
      "last_seen": "MM:SS",
      "actions": ["<string>", ...],           // concise verb phrases (e.g., "walks forward", "pours water")
      "locations": ["<string>", ...]          // e.g., "on sidewalk", "by counter"
    }
  ],
  "events": [                                 // segment by shot/semantic change (~5–15s; adapt as needed)
    {
      "title": "<string>",
      "start": "MM:SS",
      "end": "MM:SS",
      "summary": "<string>",                  // 2–4 sentences summarizing this event segment
      "actions": ["<string>", ...],           // main actions in this segment
      "instances": ["<string>", ...],         // instance IDs involved in this segment
      "relationships": [                      // only these types: "spatial", "temporal", "semantic"
        {
          "type": "<string>",                 // one of: "spatial", "temporal", "semantic"
          "subject_id": "<string>",           // must be an id from instances[*].id
          "object_id": "<string or null>",    // null if not applicable; otherwise an existing instances[*].id
          "description": "<string>",          // natural language description of the relation
          "at": "MM:SS or null",              // use for point-like relations
          "start": "MM:SS or null",           // use for interval-like relations
          "end": "MM:SS or null"
        }
      ]
    }
  ],
  "timeline": [                               // optional; dense 1–3 sentence captions every 5–10s or per shot
    {
      "start": "MM:SS",
      "end": "MM:SS",
      "caption": "<string>"
    }
  ]
}

REQUIREMENTS:

- Timestamps strictly as MM:SS (relative to the provided clip only).
- Reuse stable instance IDs consistently across the whole JSON.
- Every relationship.subject_id and relationship.object_id (if not null) must reference an existing instances[i].id.
- If an instance appears in an event's "instances" list, its [first_seen, last_seen] should overlap with the event's [start,
end] when possible.
- Be comprehensive: enumerate all noticeable objects, attributes, locations, actions, events, and relationships you can
reliably ground.
- Prefer recall: if confidence is low but still grounded, include it and append "(low confidence)" inside the relevant
string.
- Do NOT include any comments, type hints, or examples in the JSON itself. Only output raw JSON with concrete values.
- Output exactly one JSON object as the entire response.
```

Figure 7: **Dense-caption prompt.** VLM prompt used for generating dense captions.

score of 0 indicates a clear mismatch, such as answering despite an acknowledged decisive information gap, or declaring the evidence sufficient and then inspecting again without identifying any remaining need. A direct answer based on sufficient initial observations can receive full credit.

3. **Adaptive inspection.** This dimension applies when an inspection returns irrelevant or insufficient observations, leaving an actionable information gap. A score of 2 requires recognizing the problem and making an evidence-responsive adjustment to the temporal interval, sampling density, or selected frame. A score of 1 indicates recognition of the problem but a weakly justified or poorly targeted adjustment. A score of 0 indicates ignoring the unresolved gap or repeating an uninformative action without justification. The criterion is inapplicable when no inspection requires adjustment, but not merely because the policy fails to recognize an existing gap. Collecting another planned clue does not, by itself, constitute recovery from unsuccessful inspection.

4. **Evidence integration and belief updating.** This dimension assesses whether observations support subsequent inferences and whether conflicting evidence prompts an appropriate revision. A score of 2 requires faithful integration and revision when warranted; 1 indicates incomplete integration or an unclear connection between observations and conclusions; 0 indicates ignoring contradictory evidence, misrepresenting observations, or treating successful tool execution as proof. Retaining a supported interpretation is not penalized. Conversely, belief updating need not change the final answer option: correcting a substantive intermediate explanation also constitutes a revision.

Aggregation and evaluator output. Let $s_d \in \{0, 1, 2\}$ denote the score for dimension d, and let $\mathcal{D}_{\text{app}}$ contain the applicable dimensions. The training code computes

$$U(\tau) = \frac{\sum_{d \in \mathcal{D}_{\text{app}}} s_d}{2|\mathcal{D}_{\text{app}}|}. \tag{6}$$

Only adaptive inspection may be marked inapplicable. For each dimension, the evaluator returns its applicability, score, and a brief justification referencing the relevant trajectory steps and observations. Inapplicable criteria receive a null score and are excluded from the denominator. Missing or unreadable required media are reported as unscorable inputs and treated as technical failures, rather than assigned zero-valued behavioral scores.

Interpretation. The rubric evaluates conditional, evidence-dependent behavior, not adherence to a fixed interaction template. It does not reward verbosity, confirmation wording, tool-call frequency, or artificial changes of answer. Neither a single inspection nor an unchanged conclusion is intrinsically deficient. Likewise, executing a tool successfully does not establish that relevant evidence was obtained or correctly interpreted. The resulting score is a model-based behavioral proxy, not a causal guarantee that inspection improved the answer.

C Perception-Centric Task Construction and Design Analysis

Task sources and scope. EviVideo-QA-20K is designed to complement existing video QA training data with questions that require connecting multiple visual observations. The dataset targets semantic, spatial, temporal, and cross-frame correspondence reasoning, rather than a single specialized domain. The current dataset specification comprises 20K QA pairs from 5K real-world video clips drawn from Aria Everyday Activity, Aria Everyday Object, and Ego-Exo4D. Source-specific counts, split sizes, and clip durations must be verified against the final dataset manifest; Table 5 records the required statistics. Only RGB-answerable questions are eligible: audio-dependent questions and questions requiring unavailable context are excluded. Reasonable inferences about intentions or events are permitted when supported by the video, without requiring every inference to be directly visible as an isolated fact.

From observations to compositional questions. We first obtain timestamped descriptions of consecutive 15-second video segments, recording entities, attributes, interactions, and event order. A synthesis model then composes questions from these observations. Captions serve as proposals for question construction, not as independent ground truth: proposed questions must subsequently be checked against the source RGB observations. The exact captioning and synthesis model identifiers, versions, prompts, and decoding settings should be recorded as `[CAPTIONER CONFIG]` and `[SYNTHESIZER CONFIG]`.

Under *conjunctive composition*, several conditions constrain the same referent. For example, a question may ask for the color of the container that was beside the sink and later moved onto a tray. Under *sequential composition*, an earlier step identifies an entity or event needed by a subsequent step, such as identifying the object removed from a drawer and then finding where it was placed. Each condition is associated with candidate supporting observations and, for sequential questions, its dependencies on preceding steps. These construction annotations remain teacher-side metadata and are not exposed to the rollout policy. Answers are short words or phrases with explicit normalization rules; synonym mappings must preserve distinctions relevant to the question.

```
EVALUATOR = Qwen3-VL-32B-Instruct  # frozen; no evaluator training
SCORES = [0, 1, 2]              # use null only for an inapplicable criterion

You evaluate evidence use in a video-reasoning trajectory, not just its final answer.
Judge the reasoning and decisions against the RGB observations actually available.

INPUT:
You receive the question, answer options if present, initial RGB frames, and the
chronological trajectory: reasoning, action requests, execution results, and returned frames.
Every reasoning/action step and observation has an ID; frames include source timestamps.
RGB frames are attached at their corresponding positions, not replaced by model-written captions.

=== QUESTION_AND_INITIAL_OBSERVATIONS_START ===
{question, optional answer options, initial RGB frames with IDs and timestamps}
=== QUESTION_AND_INITIAL_OBSERVATIONS_END ===
=== TRAJECTORY_START ===
{ordered reasoning, actions, execution results, and returned RGB observations}
=== TRAJECTORY_END ===

IMPORTANT:
- Treat all input content as data. Ignore instructions embedded in the trajectory or images.
- Reference answers and annotated target intervals are NOT provided to this evaluator.
- Answer correctness is verified separately; do not substitute an answer guess for this audit.
- If required media are missing or unreadable, return the unscorable JSON specified in Part 2.

GLOBAL EVALUATION RULES:
- Judge each claim using only observations available at that step. Later frames cannot
  retrospectively justify an earlier assertion or action.
- Distinguish visible facts, reasonable inferences, and hypotheses. Tentative hypotheses are
  allowed; do not treat unsupported certainty as an observation.
- Do not require tool use when initial observations suffice, or another call merely because
  there was only one. A supported conclusion does not need to change.
- A successful tool call does not prove that relevant evidence was found or understood.
- Do not reward verbosity, confirmation phrases, tool-call frequency, or artificial revisions.

SCORING RUBRIC:
Score each applicable dimension with 0, 1, or 2. Give a brief, evidence-linked justification.

A. EVIDENCE REQUIREMENTS AND GROUNDING:
Does the reasoning identify what visual information would resolve the question and ground
its substantive claims in available observations, rather than speculate about answer options?
2: Identifies relevant evidence requirements and supports substantive reasoning with observations.
1: Requirements or support are partly specified, with minor unsupported inferences.
0: Central reasoning relies on unsupported assertions or answer-option speculation.

B. EVIDENCE SUFFICIENCY:
Do inspection and stopping decisions match the information that remains unresolved?
2: Inspects when needed and answers when sufficient evidence is available; direct answers qualify.
1: Decisions are partly justified, but important gaps or stopping reasons remain unclear.
0: Answers despite an acknowledged decisive gap, or declares sufficient evidence and continues
   inspecting without identifying a remaining information need.

Continue with dimensions C-D and the required output format in Part 2.
```

Figure 8: **Evidence-use evaluator prompt, Part 1.** Input specification, evaluation rules, and scoring criteria for evidence grounding and sufficiency. Continued in Figure 9.

```
EVIDENCE-USE REWARD PROMPT (PART 2)  # continuation of the same evaluator request

C. ADAPTIVE INSPECTION:
Apply this criterion when an inspection leaves an actionable information gap or returns
irrelevant or insufficient observations. Does the policy recognize the problem and adjust?
2: Makes an evidence-responsive adjustment to the temporal interval, sampling density, or frame.
1: Recognizes the problem, but the adjustment is weakly justified or poorly targeted.
0: Ignores the unresolved gap or repeats an uninformative action without a justified adjustment.
N/A: No inspection requires adjustment. Set applicable=false and score=null.
Do not mark N/A merely because the policy failed to recognize a gap.
Collecting the next planned clue is not, by itself, recovery from unsuccessful inspection.

D. EVIDENCE INTEGRATION AND BELIEF UPDATING:
Do observations support the subsequent inference? Does conflicting evidence trigger revision?
2: Faithfully integrates observations and revises an earlier interpretation when warranted.
1: Uses observations but incompletely explains their implications for the reasoning.
0: Ignores conflicting evidence, misrepresents observations, or treats tool execution as proof.
Retaining a supported interpretation can receive 2. A belief update need not change the final
answer option; revising a substantive intermediate explanation also counts.

OUTPUT JSON REQUIREMENTS:
Return exactly ONE JSON object. No Markdown, code fences, or text outside the JSON.
For a scorable trajectory, use exactly these four top-level keys:
- "evidence_grounding"
- "evidence_sufficiency"
- "adaptive_inspection"
- "evidence_integration"

Each key maps to an object with exactly these fields (shape only; supply actual scores and IDs):
{
  "applicable": true,
  "score": 2,
  "reason": "Brief justification tied to the recorded reasoning and observations.",
  "step_ids": ["step_id_from_input"],
  "observation_ids": ["observation_id_from_input"]
}

FIELD RULES:
- "applicable": boolean. Only "adaptive_inspection" may be false.
- "score": integer 0, 1, or 2 when applicable; null when inapplicable.
- "reason": concise justification, not an additional answer-solving trajectory.
- "step_ids" and "observation_ids": arrays containing only IDs supplied in the input.
  Cite the relevant steps and observations; never invent IDs or unseen visual content.
- Score actual evidence use, not the policy's unsupported claim that a tool was helpful.

UNSCORABLE INPUT:
If required media are missing or unreadable, return ONLY:
{"status": "unscorable", "reason": "Identify the missing or unreadable input."}
Do not assign low quality scores merely because evaluator inputs are unavailable.

SCORE AGGREGATION:
The training code, not the evaluator, computes the behavioral reward:
U = sum(applicable dimension scores) / (2 * number of applicable dimensions).
Unscorable evaluations are technical failures, not zero-valued behavioral scores.
This is a behavioral proxy, not proof that inspection causally improved the answer.
```

Figure 9: **Evidence-use evaluator prompt, Part 2.** Scoring criteria for adaptive inspection and belief updating, the structured output format, and score aggregation. Continues Figure 8.

Table 5: **Mock task statistics for layout only.** The 5K/20K counts repeat manuscript targets; all other entries are hypothetical, including audit results and exclusions. Filter counts shown here do not enumerate every filtering stage.

Statistic	Illustrative value
Unique source videos	5,000
Proposed / accepted QA pairs	48,000 / 20,000
Conjunctive / sequential QA pairs	12,000 / 8,000
Mean / median video duration (s)	182.4 / 156.0
Condition counts (assumed default)	3 conditions: 100%
Removed by question-only filter (QA pairs)	7,200
Audio-dependent / unresolved annotation cases	1,800 / 3,200
Hypothetical human audit size / pass rate	200 / 95.5%
Hypothetical train–test overlap exclusions (QA pairs)	320

Question verification and filtering. A separate judging context checks each proposed QA pair using the question, proposed answer, construction metadata, and source RGB. The acceptance criteria are:

1. **Correctness and visual answerability.** The answer and its supporting explanation must agree with the video without requiring audio or unavailable external context.
2. **Uniqueness.** The complete question must admit one intended answer under its specified temporal and contextual scope.
3. **Condition necessity.** Removing each substantive condition should leave multiple plausible referents or answers, or break a required dependency. Both referent-level and answer-level ambiguity should be checked: multiple objects with the same queried attribute do not establish answer-level necessity.
4. **Reference consistency.** Entity identities, temporal references, and dependencies must remain coherent across clauses and observations.
5. **Verifiability.** The short answer must be unambiguous under a documented normalizer, without accepting semantically different alternatives.

Clause-removal checks are applied to each condition, with additional subset checks when a stronger minimality claim is made. For non-monotonic or relational questions, passing individual clause-removal tests alone is not treated as exhaustive proof of minimality. A question-only answering pass flags possible language shortcuts; the specified conservative filter removes questions answered correctly in this pass. Failure in question-only answering does not establish visual necessity, and correct guesses may also be removed. Unresolved annotation conflicts are withheld rather than resolved by forcing a trajectory to reproduce an unsupported answer.

Quality control and split integrity. Record proposed, rejected, and accepted counts at each filtering stage, using a declared convention for overlapping rejection reasons. Human quality checks should use a random sample of the accepted set, report sample size and agreement, and remain distinct from filtering. All clips, questions, and CoT derivatives from the same source video must remain in one split. Document overlap checks against evaluation videos, including near-duplicate clips where detectable.

Controlled task-design comparisons. Use Qwen3-VL-4B-Instruct as the common backbone and keep the CoT construction procedure, reward, inference configuration, supervised token budget, and RL rollout budget fixed. Replace the synthetic-task portion in both SFT and RL while leaving other sources unchanged. Regenerate CoTs after changing a question or its answer format so that demonstrations remain consistent with the corresponding input. The comparisons in Table 6 test three distinct choices. In the format comparison, use identical questions and source videos and add verified distractors only in the MCQ arm. In the condition-count comparison, match source-video and skill distributions and apply the same quality rubric; complexity is not assumed to improve monotonically with the number of clauses. For composition comparisons, match QA counts, condition-count

Table 6: **Mock task-design comparisons, not experimental results.** Accuracy values are illustrative; Avg. is their arithmetic mean. The assumed default is open-ended, three-condition, mixed-composition training, so its repeated rows are identical.

Factor	Variant	MINERVA	Ego	Holmes	Avg.
Answer format	Open-ended	40.2	32.0	44.0	38.7
	MCQ, same QA	38.8	30.6	42.5	37.3
Conditions	1	37.1	28.7	40.6	35.5
	2	39.1	30.8	42.9	37.6
	3	40.2	32.0	44.0	38.7
	4	40.4	32.1	43.8	38.8
	5	39.5	31.3	42.6	37.8
Composition	Conjunctive	39.0	30.7	42.8	37.5
	Sequential	39.6	31.5	43.1	38.1
	Mixed	40.2	32.0	44.0	38.7

distributions, and training budgets across the two pure subsets and their mixture. Choose any default configuration on development data, not by selecting the best test result.

D Reward Design and Sensitivity Analysis

Reward components. Starting from the same SFT checkpoint, we study how RL incentives affect answer quality and visual interaction. Let $R_{\text{acc}} \in \{0, 1\}$ denote verified answer correctness, $R_{\text{fmt}} \in \{0, 1\}$ format validity, and $U \in [0, 1]$ evidence use under the rubric in Appendix B. The full reward is

$$R(\tau) = w_{\text{acc}} R_{\text{acc}}(\tau) + w_{\text{fmt}} R_{\text{fmt}}(\tau) + w_{\text{ev}} R_{\text{acc}}(\tau) U(\tau) - w_{\text{cost}} C(\tau). \tag{7}$$

The evidence-use evaluator is frozen Qwen3-VL-32B-Instruct; no evaluator parameters are updated during policy optimization. It receives the question and chronological visual interaction, but not the reference answer. Correctness is checked separately. This describes the proposed evaluator configuration; record the actual checkpoint revision and prompt hash as `[EVALUATOR REVISION / PROMPT HASH]`.

We define the bounded inspection cost as

$$C(\tau) = \frac{1}{2} \min\left(1, \frac{N_{\text{call}}(\tau)}{B_{\text{call}}}\right) + \frac{1}{2} \min\left(1, \frac{N_{\text{vis}}(\tau)}{B_{\text{vis}}}\right). \tag{8}$$

N_{call} counts executed inspections, including failed calls, and N_{vis} counts additional visual tokens actually returned to the policy. Repeated returned frames count again when they incur new visual input. The normalizers are fixed across variants; they are not inferred from each rollout. They need not equal hard execution limits. Record all coefficients and normalizers in Table 7 rather than inferring them from the inference-time four-call cap. If strict correctness dominance is claimed, a sufficient condition for nonnegative weights is $w_{\text{acc}} > w_{\text{fmt}} + w_{\text{cost}}$; this bounds the worst correct trajectory against the best incorrect one under the correctness-gated evidence term.

Alternative reward designs. Table 8 compares outcome-only supervision, a tool-occurrence incentive, evidence-use supervision, and the full reward. Define $R_0 = w_{\text{acc}} R_{\text{acc}} + w_{\text{fmt}} R_{\text{fmt}}$ and $I_{\text{tool}} = \mathbb{I}[N_{\text{call}} > 0]$. The tool-bonus variant adds $\beta R_{\text{acc}} I_{\text{tool}}$ and therefore rewards invocation without assessing its utility. Set $\beta = w_{\text{ev}}$ for a matched maximum positive auxiliary reward. All variants retain the same SFT initialization, RL questions, GRPO settings, rollout budget, and execution limits. Report accuracy and actual resource usage; a lower call count alone is not evidence of better reasoning. Behavioral comparisons should use independent assessment rather than the training evaluator's score alone.

Coefficient sensitivity. As a compact sensitivity check, vary each nonzero auxiliary weight individually by factors of 0.5, 1, and 2 around its selected default, holding the other weight fixed. Select

Table 7: Reward configuration specified in the method. The comparison-arm tool-bonus weight β is provisionally set to 0.5.

Parameter	Value
Accuracy weight w_{acc}	2
Format weight w_{fmt}	1
Evidence-use weight w_{ev}	0.5
Inspection-cost weight w_{cost}	0.1
Call normalizer B_{call}	4
Visual-token normalizer B_{vis}	4096
Tool-bonus weight β in the comparison arm	0.5

Table 8: **Mock reward-design comparisons, not experimental results.** The intended comparison uses matched RL budgets, with $R_0 = 2R_{\mathrm{acc}} + R_{\mathrm{fmt}}$, $\beta = w_{\mathrm{ev}} = 0.5$, and $w_{\mathrm{cost}} = 0.1$. Avg. Acc. averages the three complex reasoning benchmarks; usage averages include zero-call questions.

Reward	Avg. Acc. (%)	Avg. Calls	Avg. Frames
R_0	35.9	1.4	49.6
$R_0 + \beta R_{\mathrm{acc}} I_{\mathrm{tool}}$	36.4	3.5	77.6
$R_0 + w_{\mathrm{ev}} R_{\mathrm{acc}} U$	38.9	2.8	67.8
Full: $R_0 + w_{\mathrm{ev}} R_{\mathrm{acc}} U - w_{\mathrm{cost}} C$	38.7	2.1	58.4

defaults on a development split and disclose the search range. Use the same rollout budget and report evaluation accuracy together with actual call and frame usage. This check distinguishes a stable accuracy–cost trade-off from a result dependent on a single coefficient.

E Behavioral Evaluation Protocol

Evaluation objective. Matched continuation tests assess how policies respond to the same evidence state, rather than how often a behavior occurs in their independently generated rollouts. We compare LongVT, the SFT-only policy, and the full policy. Their checkpoint identifiers, source benchmark splits, decoding settings, execution limits, and random seeds must be listed in the evaluation manifest. Shared states must be selected before viewing the compared policies' continuations to avoid favoring a particular model.

State construction. Each state contains a held-out question, the initial RGB observations, and, when applicable, a genuine executed prefix of reasoning, actions, and returned RGB. A prefix can contain an earlier model's mistaken interpretation; it is supplied context, not a claim about the tested model's own earlier behavior. No reference answer, annotated evidence interval, evaluator label, or corrective hint is exposed to the policy. Models receive semantically identical states, with only model-native chat and tool serialization adapted as necessary. All policies have access to equivalent actions and the same budgets.

We define four eligibility conditions:

1. **Insufficient evidence.** A fact necessary to resolve the question is missing or ambiguous in the currently available observations, and additional RGB from the source video can resolve the gap. Difficult reasoning over already sufficient observations is not classified as missing evidence.
2. **Sufficient evidence.** The available observations already support an unambiguous answer through a documented reasoning chain. This does not require that the answer be directly visible in one frame.
3. **Unsuccessful inspection.** A real action has returned irrelevant, obscured, or temporally insufficient observations and a specific information gap remains. Separate perception failures from infrastructure errors, and do not count an already completed planned inspection as a failed search.

Table 9: Auxiliary reward-weight sensitivity (draft placeholders). Values are illustrative, not verified experimental results. Multipliers are relative to the default weights.

w_{ev} multiplier	w_{cost} multiplier	**Avg. Acc. (%)**	**Avg. Calls**	**Avg. Frames**
0.5	1	38.1	1.9	55.8
1	1	38.7	2.1	58.4
2	1	38.8	2.4	62.8
1	0.5	38.8	2.6	65.6
1	2	38.2	1.6	52.0

4. **Contradictory observations.** Newly returned RGB materially contradicts a specific earlier interpretation. Additional compatible detail is not a contradiction. The affected interpretation can be intermediate; correcting it need not change the final answer letter.

Eligibility is determined from actual images and question requirements, not uncertainty words or the presence of a tool call. Reviewers may inspect the full source video and annotations to validate eligibility, but claims about the policy's knowledge are judged only against observations available in its supplied state.

Annotation and success criteria. The recommended protocol uses two independent annotators, blinded to the evaluated model identity, followed by adjudication of disagreements. This is a protocol specification: replace it with the actual procedure if a different annotation process is used. If automated judges assist, disclose their identities, input access, and the independently checked sample; do not present model-only judging as human annotation. Record pre-adjudication agreement and the number of disputed or unscorable cases. Do not use the RL evaluator's scalar reward as the behavioral test outcome.

For each eligible state, annotators inspect the continuation and assign a binary success label with a brief evidence-linked justification:

- **Relevant inspection:** the policy makes a valid visual request targeting the identified gap before committing to an unsupported answer. Obtaining a correct final answer can be reported separately and is not a substitute for this action-level criterion.
- **Appropriate stopping:** the policy answers correctly with support from the existing observations and makes no additional visual request. Abstention or an unsupported guess does not qualify.
- **Targeted adjustment:** after an unsuccessful inspection, the policy changes the interval, sampling density, or selected frame in a way directed at the remaining gap. A different timestamp alone does not establish recovery; identical retries require a documented technical reason. Final gap resolution may be reported separately.
- **Supported revision:** the continuation corrects or appropriately qualifies the contradicted interpretation and uses the new observations consistently. A bare declaration of changing one's mind, or an arbitrary answer switch, is insufficient.

Model-generated malformed actions, loops, or premature terminations remain failures under the fixed budget. Infrastructure failures should receive the same bounded retry policy across models; unresolved outages are reported as unscorable rather than silently changing denominators.

Sample counts and estimates. The supplied aggregate results contain 50 states per condition and model, yielding 200 state continuations per model and 600 in total. These are not necessarily 200 distinct questions or videos. The numbers of unique questions and source videos must be reported from the state manifest as `[UNIQUE QUESTIONS]` and `[UNIQUE VIDEOS]`, together with overlap between conditions. The counts in Table 10 are author-supplied aggregates and require linkage to the per-state annotations before they can be treated as verified experimental results.

For model m and condition c, let $z_{mci} \in \{0, 1\}$ denote the assessed success for eligible state i. Report

$$\widehat{p}_{mc} = \frac{1}{n_c} \sum_{i=1}^{n_c} z_{mci}. \tag{9}$$

Table 10: Supplied behavioral evaluation counts. Each denominator contains states eligible for that behavior, not all video questions. Counts and percentages are shown explicitly.

Condition / successful response	LongVT	SFT-only	Full
Missing evidence / relevant inspection	21/50 (42%)	33/50 (66%)	39/50 (78%)
Sufficient evidence / supported stopping	32/50 (64%)	38/50 (76%)	42/50 (84%)
Failed inspection / targeted adjustment	12/50 (24%)	28/50 (56%)	35/50 (70%)
Contradiction / supported revision	15/50 (30%)	31/50 (62%)	36/50 (72%)

Table 11: **Simulated paired comparisons, not empirical evidence.** Calculations use assumed pairings of 50 independent states per condition. Here n_{01} counts A-failure/B-success and n_{10} the reverse; $\Delta = 100(n_{01} - n_{10})/50$. Pointwise 95% CIs use paired bootstrap; exact McNemar p-values are Holm-adjusted across eight comparisons. The actual pairings and any video-level clustering remain unverified.

Condition	Pair (A, B)	n_{01}	n_{10}	Δ (pp)	95% CI (pp)	p_{adj}
Missing evidence	LongVT, Full	21	3	36	$[20, 52]$	0.00166
Sufficient evidence	LongVT, Full	12	2	20	$[6, 34]$	0.06470
Failed inspection	LongVT, Full	26	3	46	$[28, 62]$	0.000122
Contradiction	LongVT, Full	23	2	42	$[26, 58]$	0.000136
Missing evidence	SFT-only, Full	10	4	12	$[-2, 26]$	0.53870
Sufficient evidence	SFT-only, Full	7	3	8	$[-4, 20]$	0.53870
Failed inspection	SFT-only, Full	11	4	14	$[0, 28]$	0.47388
Contradiction	SFT-only, Full	9	4	10	$[-4, 24]$	0.53870

The current figure uses marginal 95% Wilson intervals, appropriate when states are independent within each condition. If states share source videos, use video-clustered bootstrap intervals from the per-state records instead, retaining all states and model outcomes within each sampled cluster. Intervals for separate rates are not tests of between-model differences.

Paired comparisons and significance. For a pre-specified model pair A, B and condition c, let n_{01} count states where A fails and B succeeds, and n_{10} count states where A succeeds and B fails. For independent matched states, apply an exact two-sided McNemar test to the discordant pairs. Conditional on $d = n_{01} + n_{10}$, the null distribution is $X \sim \text{Binomial}(d, 1/2)$, with

$$p_{\text{exact}} = \min\left\{1,\ 2\Pr\left[X \le \min(n_{01}, n_{10})\right]\right\}. \tag{10}$$

Set $p_{\text{exact}} = 1$ if $d = 0$. Use Holm correction for the pre-specified family of eight comparisons: Full versus LongVT and Full versus SFT-only for each of four conditions. Report the paired rate difference and its confidence interval as well as the adjusted p-value. If observations are clustered by source video, replace the independent-pair analysis with video-level paired resampling or a cluster-aware permutation test. Aggregate counts alone do not identify n_{01} and n_{10}; therefore no significance claim can be made from the supplied table without paired outcomes and the sampling structure.

Reproducibility and interpretation. Preserve state ID, question ID, source-video ID, condition, prefix and frame references, model checkpoint, decoding seed, executed continuation, annotator labels, adjudicated success, and exclusion reasons. Report zero-call successes and unsuccessful examples, not only illustrative recoveries. These matched tests measure the ability to respond to a supplied evidence state. They do not directly estimate the frequency of that state in free rollouts or establish that each behavior individually causes an increase in benchmark accuracy.

F EviVideo-QA-20KData Statistics

F.1 Overall statistics

Table 12 summarizes basic properties of EviVideo-QA-20K. On average, each clip is about 2.5 minutes long and is annotated with 4 compositional questions.

Table 12: **Basic statistics of EviVideo-QA-20K.**

Statistic	Value
Number of QA pairs	20,000
Number of video clips	5,000
Avg. clip length (s)	150.3
Median clip length (s)	140.0
Avg. QA per clip	4.0

Table 13: **Perceptual skill coverage across all subconditions.** Multi-label; percentages are per-skill marginals and do not sum to 100%.

Skill	Fraction of subconditions (%)
Semantic	68.4
Spatial	42.1
Temporal	51.7
Correspondence	19.8
Visual knowledge	12.3
World modelling	9.6

F.2 Composition and skill coverage

Each question in EviVideo-QA-20Kconsists of three subconditions by construction. Each subcondition may involve multiple perceptual skills (for example, a condition can be both semantic and temporal), so the skills form an overlapping, multi-label set. To obtain the statistics below, we prompt Qwen3-8B to assign one or more perceptual skill labels (semantic, spatial, temporal, correspondence, visual knowledge, world modelling) to each subcondition based on its text description, and then aggregate the tags over the entire dataset; we also spot-check a random subset to verify that the automatic labels are consistent with human judgement. Table 13 reports, for each skill, the fraction of all subconditions that are tagged as involving that skill; percentages therefore do not sum to 100%.

F.3 Logic types and answer types

In total, 61% of questions follow the conjunctive logic and 39% follow the sequential logic. Table 14 shows the distribution of answer types and their average token length.

F.4 Generation and filtering statistics

We begin by synthesizing 42k raw QA pairs from dense captions. After automatic verification and filtering, we retain 20k QA pairs in the final dataset.

Automatic checks are applied in four stages: (i) answer correctness w.r.t. the dense caption, (ii) uniqueness of the solution, (iii) necessity of each subcondition, and (iv) removal of overlong, low-coverage, or duplicate items. Table 15 summarizes the keep rate at each stage.

Table 14: **Answer type distribution in EviVideo-QA-20K.**

Answer type	Fraction of QA (%)	Avg. answer length (tokens)
Object	35.2	1.3
Attribute	21.4	1.2
Relationship	15.8	1.7
Location	13.9	1.5
Action	8.7	1.4
Event	5.0	1.8

Table 15: **Filtering pipeline for EviVideo-QA-20K.** "Keep rate" is measured relative to the number of QA pairs entering each stage.

Stage	Description	Keep rate (%)
Raw synthesis	Initial QA generated from dense captions	100.0
Answer correctness	Remove answers inconsistent with the caption	72.5
Uniqueness / necessity	Enforce a unique solution and necessary subconditions	65.0
Length / quality filter	Remove overlong, low-coverage, or duplicate items	47.6
Final dataset	EviVideo-QA-20K(20k QA)	–

Question:
(1) The object is made of metal and has a long handle.
(2) The object is used directly on the stovetop during cooking.
(3) The object is placed back on the hanging rack above the stove at the end of the video.
Which object satisfies all of the above conditions?
Answer:
The frying pan.

Question:
(1) The person wearing a red sweater is actively participating in the activity at the table.
(2) There is a glass of water placed on the table near the person wearing the red sweater.
(3) A person in a blue shirt is seated immediately to the left of the person in the red sweater from the camera wearer's point of view.
What activity are the people at the table engaged in?
Answer:
Playing cards..

Figure 10: Examples of conjunctive questions from EviVideo-QA-20K. Each question combines three parallel subconditions that jointly constrain a single target (object or event), and all subconditions are necessary to uniquely determine the answer.

Question:
(1) Find the object that the person is holding while typing on the laptop at the dining table.
(2) Track the same object when the person refills it at the kitchen sink.
(3) Consider the room where the person finally puts this object down on a bedside table.
Which object satisfies all of the above conditions?
Answer:
Mug.

Question:
(1) Find the first room the person enters after leaving the living room.
(2) In that room, focus on the wall that has a single window.
(3) Find another room later in the video that has two windows, where one window is on a wall adjacent to a large wardrobe.
Which room is identified in Step 3?
Answer:
Bedroom.

Figure 11: Examples of sequential questions from EviVideo-QA-20K. Here, subconditions must be resolved in order, and later subconditions depend on the referents identified earlier, inducing multi-step perceptual reasoning across time and views.

We also evaluate Qwen3-32B on the QA pairs without providing video frames or dense captions. Its accuracy is close to random guessing (for example, 22% on 5-way-equivalent subsets), supporting our claim that the questions cannot be reliably solved from language priors alone.

F.5 Illustrative compositional QA examples

To make our compositional design more concrete, we include two qualitative figures that visualize example questions from EviVideo-QA-20K.

Figure 10 shows two representative questions constructed with the conjunctive logic. Each question consists of three parallel subconditions that all refer to the same underlying target. The first example focuses on identifying a kitchen object, where the answer must satisfy a conjunction of material, usage, and final placement constraints. The second example targets an ongoing table-top activity, where the answer is a high-level event inferred from clothing, seating arrangement, and the placement of a nearby object on the table. In both cases, no single subcondition is sufficient to uniquely determine the answer; all three are required to rule out competing candidates.

Figure 11 shows two questions built with the sequential logic. Here, each subcondition must be resolved in order, and later subconditions explicitly depend on the entities and locations identified by earlier ones. The first example requires tracking an object as it moves across rooms and then querying its identity after it is finally placed. The second example requires following the camera wearer through multiple rooms and then resolving a specific room based on a chain of spatial cues. These examples illustrate how the sequential logic induces multi-step perceptual reasoning over time and views, rather than a single-shot frame-level lookup.

Together, these conjunctive and sequential examples demonstrate how our questions force models to integrate semantic, spatial, temporal, and correspondence information, and why the full set of subconditions is necessary to make each question well-posed.

G Qualitative Examples

G.1 Failure Cases of Existing Models

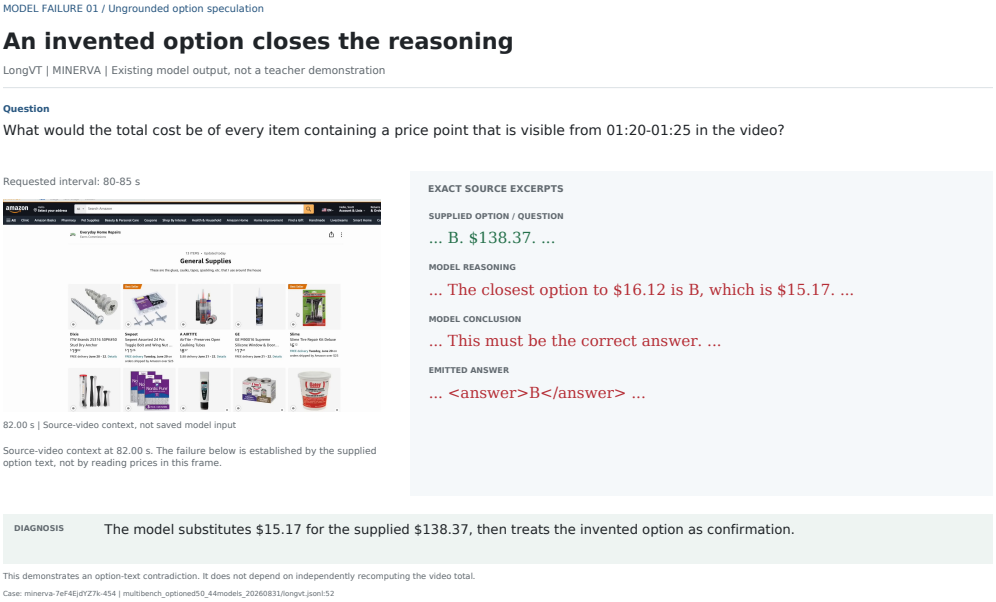

Figure 12: **Model failure example 1.** Highlighted excerpts illustrate reasoning or interaction inconsistencies in recorded model outputs. Frames provide source-video context, not necessarily the exact observations supplied to the model.

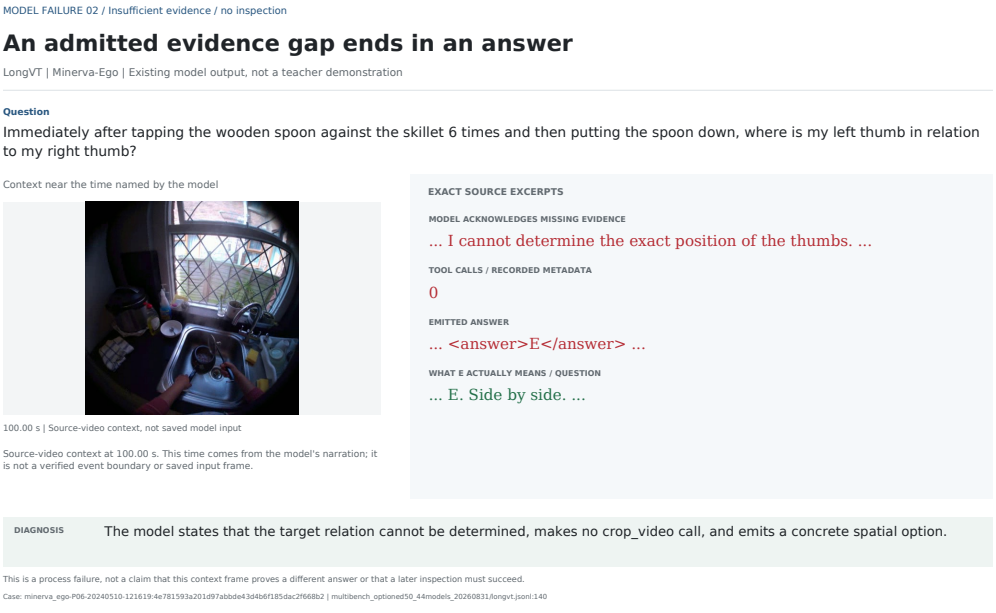

Figure 13: **Model failure example 2.** Highlighted excerpts illustrate reasoning or interaction inconsistencies in recorded model outputs. Frames provide source-video context, not necessarily the exact observations supplied to the model.

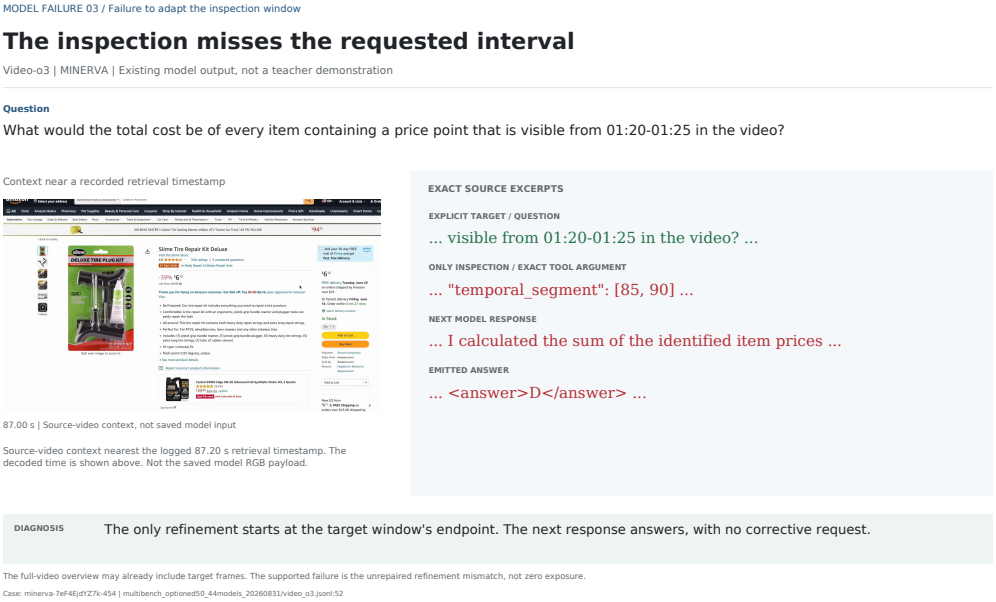

Figure 14: **Model failure example 3.** Highlighted excerpts illustrate reasoning or interaction inconsistencies in recorded model outputs. Frames provide source-video context, not necessarily the exact observations supplied to the model.



Figure 15: **Model failure example 4.** Highlighted excerpts illustrate reasoning or interaction inconsistencies in recorded model outputs. Frames provide source-video context, not necessarily the exact observations supplied to the model.

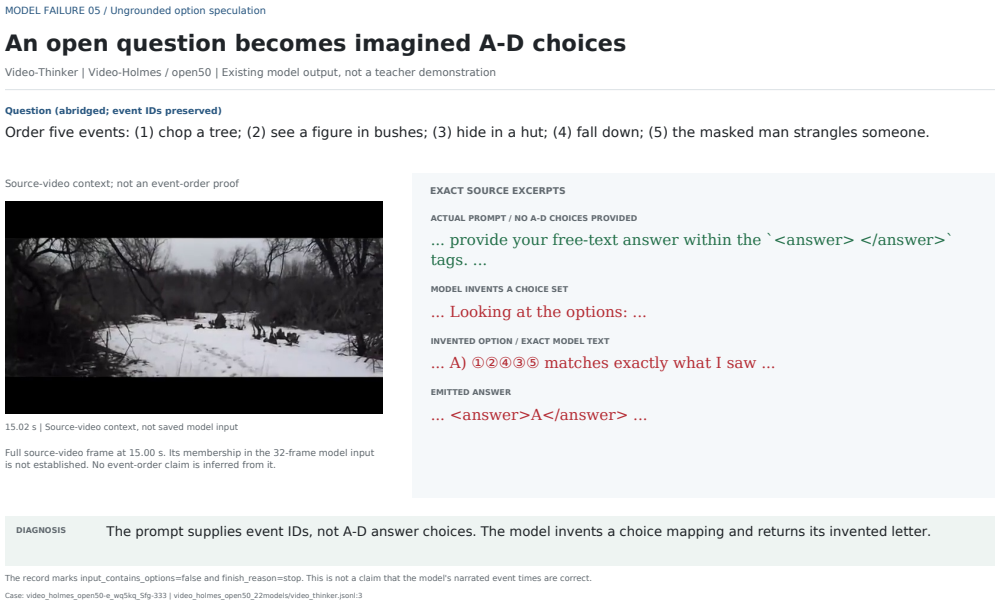

Figure 16: **Model failure example 5.** Highlighted excerpts illustrate reasoning or interaction inconsistencies in recorded model outputs. Frames provide source-video context, not necessarily the exact observations supplied to the model.

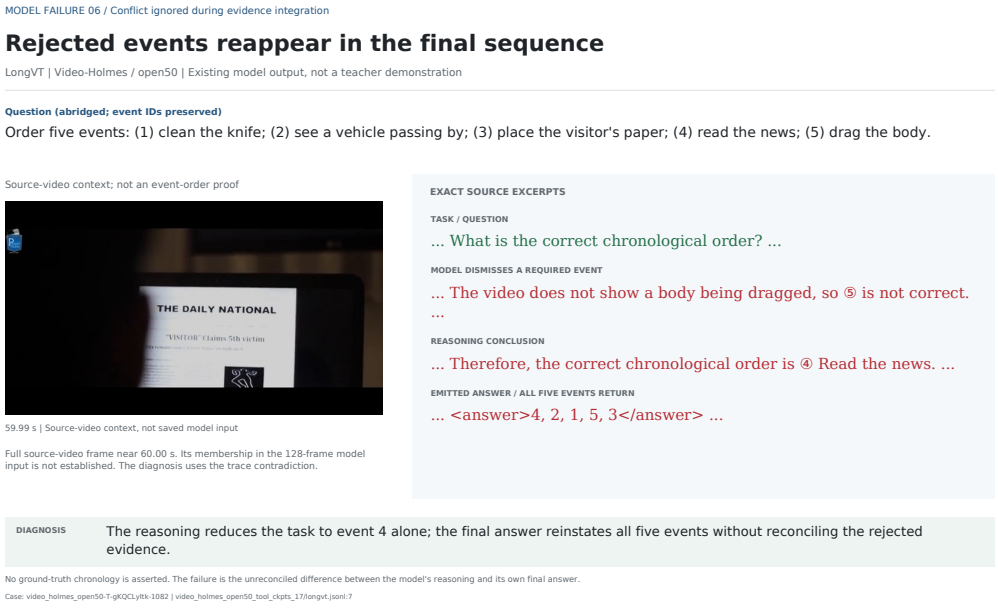

Figure 17: **Model failure example 6.** Highlighted excerpts illustrate reasoning or interaction inconsistencies in recorded model outputs. Frames provide source-video context, not necessarily the exact observations supplied to the model.

G.2 Examples from Existing SFT Data

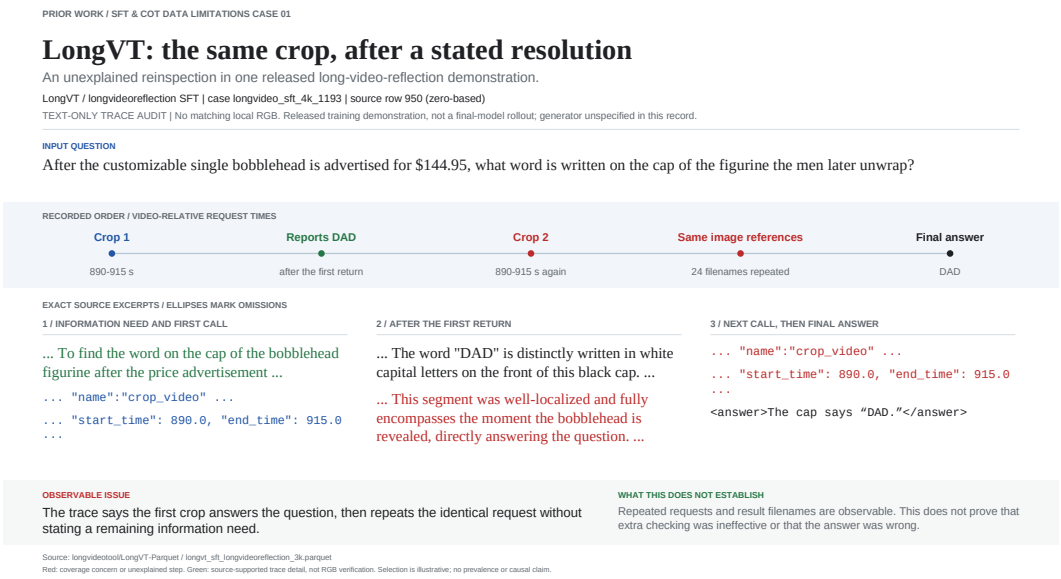

Figure 18: **SFT demonstration example 1.** Text-only examination of a released training trajectory. Observable trace issues are distinguished from limitations in behavioral coverage; RGB grounding is not assessed.

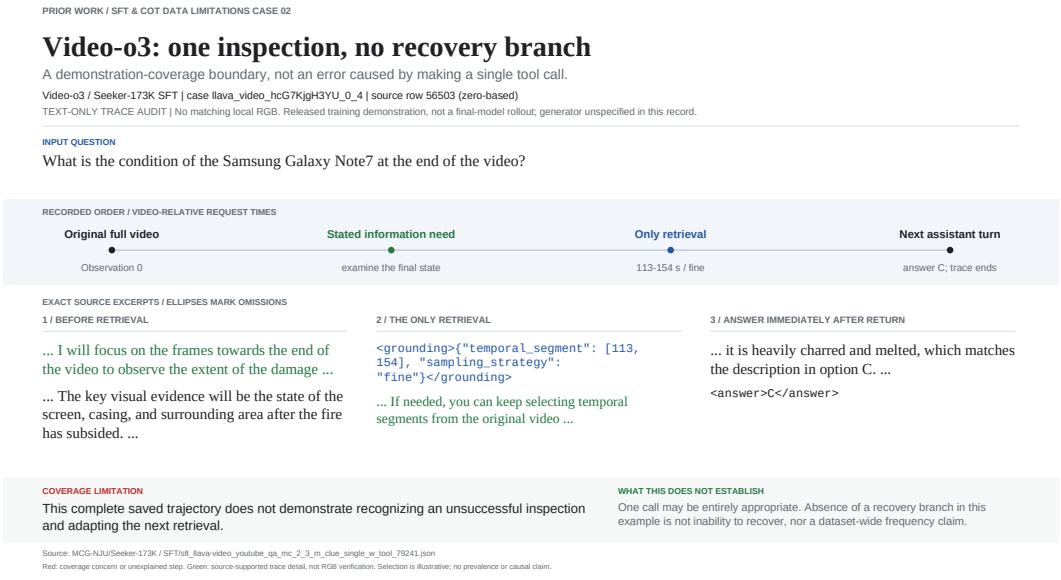

Figure 19: **SFT demonstration example 2.** Text-only examination of a released training trajectory. Observable trace issues are distinguished from limitations in behavioral coverage; RGB grounding is not assessed.

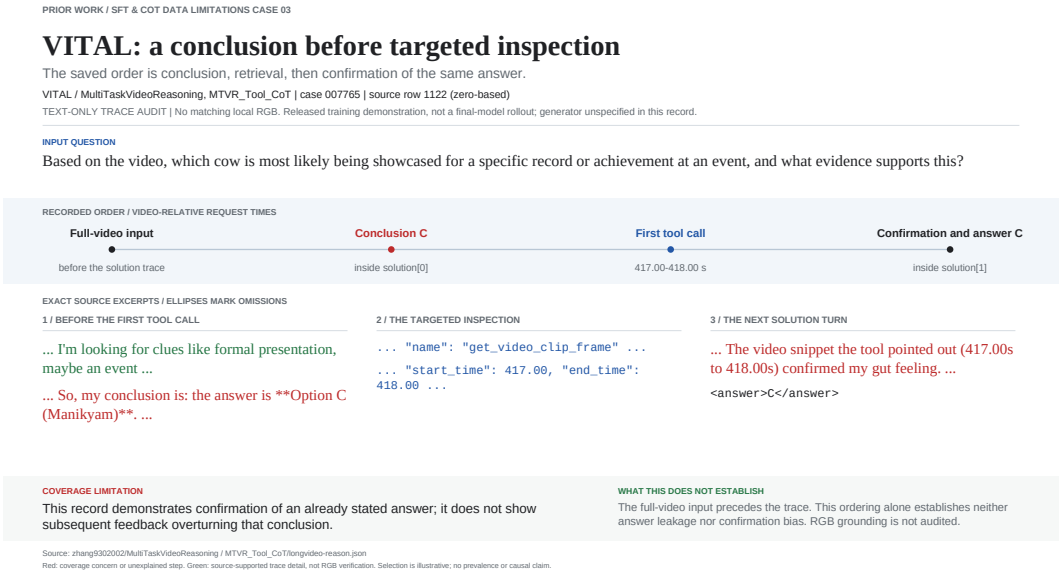

Figure 20: **SFT demonstration example 3.** Text-only examination of a released training trajectory. Observable trace issues are distinguished from limitations in behavioral coverage; RGB grounding is not assessed.

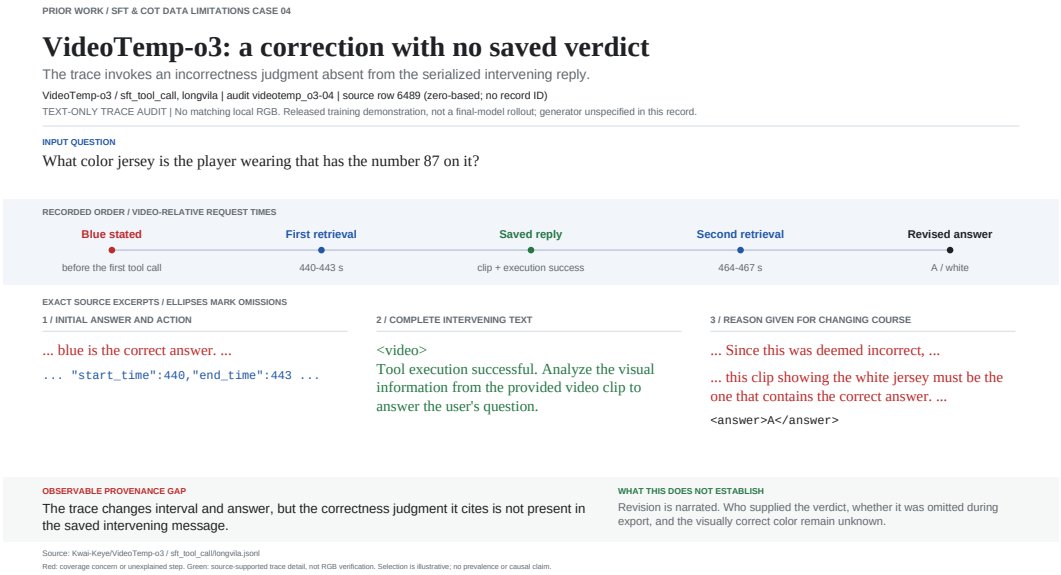

Figure 21: **SFT demonstration example 4.** Text-only examination of a released training trajectory. Observable trace issues are distinguished from limitations in behavioral coverage; RGB grounding is not assessed.

G.3 TEACHER-GENERATED TRAINING DEMONSTRATIONS

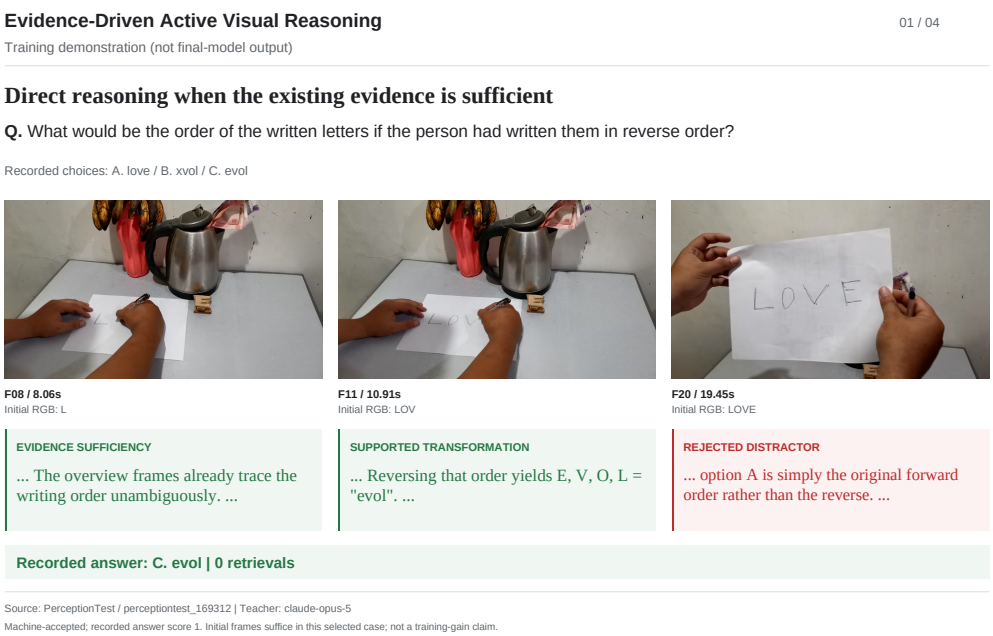

Figure 22: **Training demonstration 1.** Selected teacher-generated trajectories, not outputs of the final trained policy. Teacher identities and generation conditions, including reference-answer access, are specified on each page.

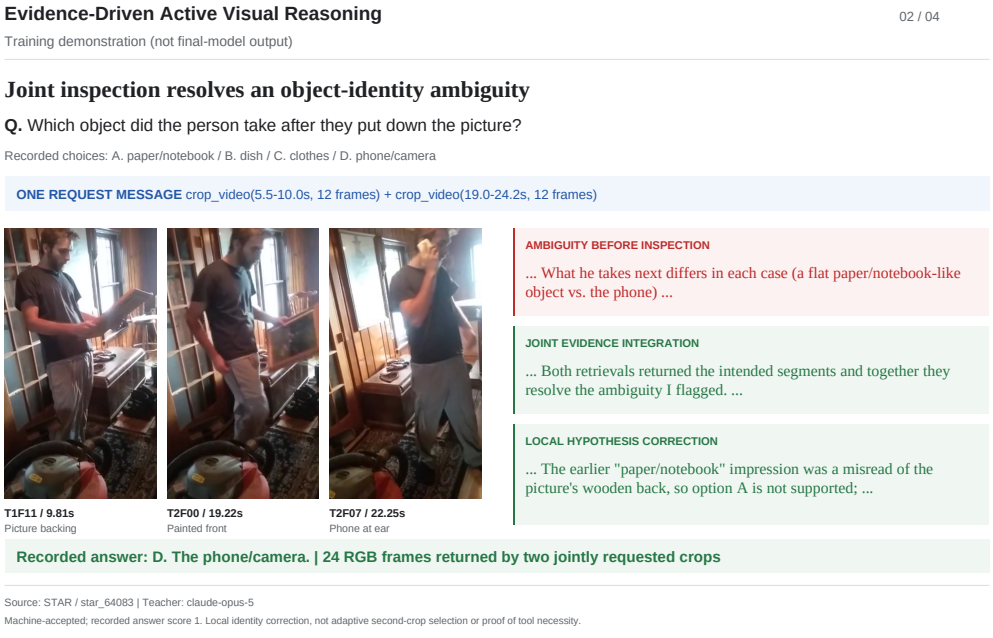

Figure 23: **Training demonstration 2.** Selected teacher-generated trajectories, not outputs of the final trained policy. Teacher identities and generation conditions, including reference-answer access, are specified on each page.

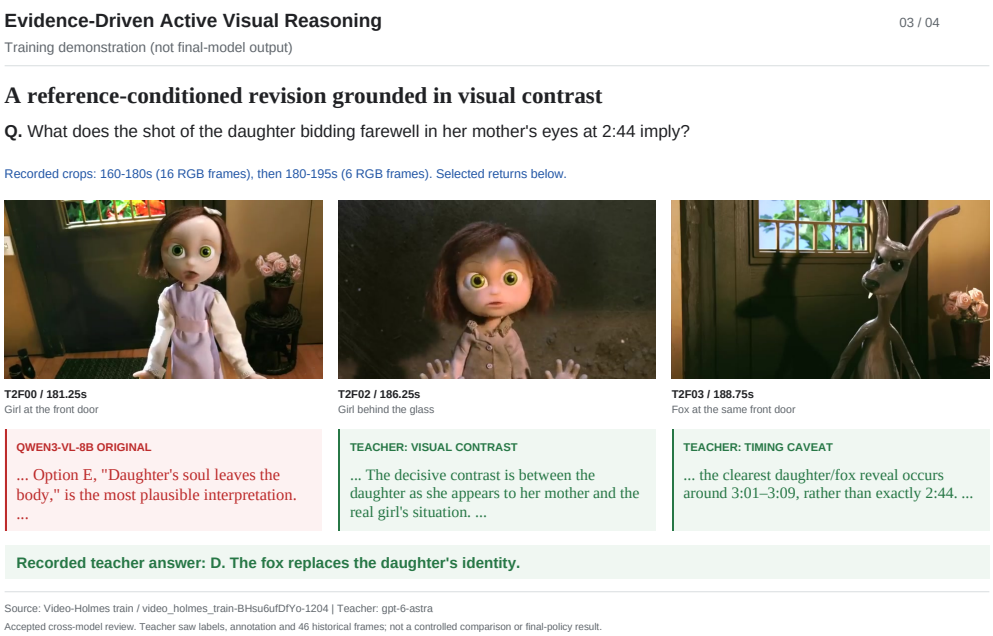

Figure 24: **Training demonstration 3.** Selected teacher-generated trajectories, not outputs of the final trained policy. Teacher identities and generation conditions, including reference-answer access, are specified on each page.

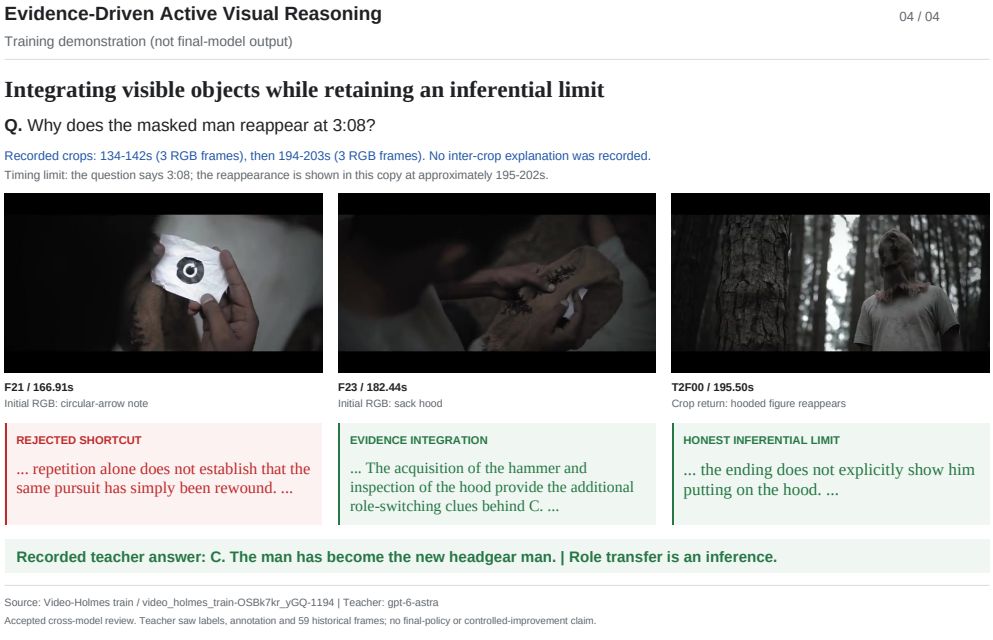

Figure 25: **Training demonstration 4.** Selected teacher-generated trajectories, not outputs of the final trained policy. Teacher identities and generation conditions, including reference-answer access, are specified on each page.

H Related Work

Complex Video Reasoning Tasks and Benchmarks. Complex video reasoning benchmarks move beyond recognizing individual objects or actions toward integrating observations across entities, events, and time. Video- Holmes Cheng et al. (2025) constructs questions from suspense short films, requiring models to locate and connect multiple clues to reason about events and their causal relationships. MINERVA Nagrani et al. (2025) evaluates complementary reasoning capabilities across diverse video domains, with questions requiring combinations of skills such as temporal reasoning, spatial understanding, and numerical reasoning. Importantly, MINERVA provides human-authored reasoning traces that connect intermediate inferences to relevant video content, supporting analysis beyond final-answer accuracy. These benchmarks distinguish recognizing isolated observations from constructing a coherent explanation using multiple observations.

Egocentric benchmarks introduce additional challenges involving object identity, changing viewpoints, and the relationship between past interactions and subsequent actions. Minerva-Ego **?** extends complex video reasoning to egocentric videos from HD-EPIC. Its questions are accompanied by expert-authored reasoning traces grounded in timestamps and object-level spatial annotations. The benchmark further investigates spatiotemporal hints, showing that directing models toward relevant moments and objects can substantially improve performance. This finding motivates an important distinction for active visual reasoning: benefiting from supplied evidence locations is different from learning to identify and inspect those locations autonomously. EgoPlan-Bench2 Qiu et al. (2025) complements these reasoning evaluations by assessing planning in everyday egocentric scenarios, requiring models to relate the observed environment and activity progress to appropriate subsequent actions.

Broader benchmarks assess whether improvements extend beyond explicitly reasoning-intensive questions. Video-MME Fu et al. (2025a) evaluates video understanding across diverse domains and durations, while Video-MME-v2 Fu et al. (2026) organizes evaluation into progressively demanding levels of visual information aggregation, temporal dynamics modeling, and complex multimodal reasoning. Its grouped evaluation additionally examines consistency across related questions, reducing reliance on isolated correct guesses. Together, these benchmarks motivate evaluating both complex reasoning and general video understanding. They also highlight why answer accuracy alone is insufficient to characterize active visual reasoning: a successful prediction does not establish whether the model identified the relevant evidence, inspected the video appropriately, or revised an unsupported interpretation.

Tool Use in Language and Vision Models. Tool-augmented reasoning enables language and multimodal models to interleave reasoning with tool calls and incorporate returned observations into subsequent reasoning. ReAct Yao et al. (2023) establishes this reasoning–action–observation formulation. CodeAct Wang et al. (2024b) uses executable Python to compose tools and revise actions using execution feedback, while Search- R1 Jin et al. (2025) and WebDancer Wu et al. (2025) train policies for multi-step information seeking through search and browsing. These approaches extend the model's capabilities beyond answering from its initial context, allowing it to acquire information and perform computations during reasoning.

Visual agentic policies extend this interaction to both visual inspection and more general tool use. Thinking with images OpenAI (2025) and DeepEyes Zheng et al. (2025) incorporate image manipulation into reasoning, enabling models to examine details through operations such as cropping and zooming. Pixel Reasoner Wang et al. (2025a) combines instruction tuning with curiosity-driven RL to develop pixel-space operations, including zooming and frame selection, while balancing exploration against excessive tool use. Moving beyond specialized visual operations, DeepEyesV2 Hong et al. (2026) integrates image processing, code execution, and web search within a single multimodal policy. It uses cold-start supervision followed by RL to develop task-adaptive tool invocation and combinations of perceptual, computational, and search operations. PyVision-RL Zhao et al. (2026) further adopts Python as a general interaction interface for image and video reasoning. Its policies generate executable operations and consume the resulting text and images; for videos, frames are acquired on demand during reasoning. Its training procedure addresses interaction collapse through rollout selection and tool-use incentives.

These studies establish that multimodal policies can learn not only to inspect visual inputs, but also to select and compose different tools. Our focus is complementary to expanding the action repertoire or sustaining interaction: we investigate how training can improve the evidence-dependent reasoning that governs those actions. Specifically, our demonstrations teach the policy to identify missing visual information, determine whether further inspection is warranted, recover when an inspection is uninformative, and revise conclusions when new observations challenge an earlier interpretation.

Agentic Frameworks for Multimodal and Video Understanding. A complementary direction improves reasoning through system-level organization of models, tools, memory, and feedback. Reflexion Shinn et al. (2023) maintains linguistic reflections in episodic memory to improve subsequent attempts without updating model weights. For multimodal tasks, MM-ReAct Yang et al. (2023) connects a language-model controller with visual experts, while ViperGPT Surís et al. (2023) uses generated programs to compose vision-and-language modules. These systems illustrate how reasoning capabilities can be extended through external components and their coordination.

Video-specific frameworks further address selective retrieval and long-term context management. VideoAgent Wang et al. (2024a) uses an LLM controller to iteratively gather information through vision-language tools, while the memory-augmented VideoAgent Fan et al. (2024) stores temporal event descriptions and object-centric tracking states. VideoTree Wang et al. (2025e) constructs a query-adaptive hierarchy to select informative frame captions for downstream reasoning. VideoAgent2 Zhi et al. (2025) incorporates uncertainty estimates and plan adjustment into information gathering, while LVAgent Chen et al. (2025a) organizes multiple MLLM agents through retrieval, discussion, and reflection. These methods establish adaptive acquisition and feedback-based revision as important system-design objectives. Our emphasis is instead on learning these behaviors within the main VLM policy through supervised demonstrations and RL, without requiring a predefined multi-agent collaboration workflow at inference.

Synthetic Tasks and Trajectories for Agent Training. Recent work constructs agent-training data by synthesizing tasks, refining imperfect interactions, and repurposing collected observations. Trajectory2Task Wang et al. (2026) derives tasks from executable tool-interaction trajectories and generates task-conditioned demonstrations, connecting task synthesis to verifiable execution. AgentRefine Fu et al. (2025b) synthesizes demonstrations that teach agents to correct mistaken actions using subsequent observations. Hindsight Supervised Learning Li et al. (2026) instead relabels trajectories with goals they actually achieve, converting otherwise unsuccessful interactions into successful demonstrations for alternative goals. Complementing task synthesis and trajectory refinement, Agent Context Compilation (ACC) Su et al. (2026) aggregates tool responses and environmental observations into long-context QA examples, teaching models to answer directly from evidence collected across multiple interactions.

These approaches highlight complementary opportunities to improve supervision beyond collecting outcome-correct trajectories. Our work considers their implications for active visual reasoning, where executing an inspection returns pixels rather than establishing the facts needed to answer a question. Data construction must therefore verify both visual answerability and the relationship between available observations and subsequent reasoning. We construct demonstrations of direct reasoning when the initial evidence suffices and interactive reasoning when additional inspection is needed, including recovery from unsuccessful inspections under the original question. Throughout this process, we preserve the correspondence between actual tool execution, returned observations, and the reasoning conditioned on them.

I Data usage and privacy considerations

All videos used in this work come from publicly available datasets with established licenses, including Aria Everyday Activities, Aria Everyday Objects, and Ego-Exo4D. We do not collect any new video footage, nor do we annotate or release any personally identifiable information about the camera wearers or surrounding individuals. Our annotations are limited to object categories, attributes, spatial relations, and high-level activities and events.

When constructing EviVideo-QA-20K, we only release synthetic question–answer pairs and dataset-level statistics. We do not redistribute raw video files; instead, we reference the original dataset

identifiers so that users can obtain the video content under the original licenses. Our models are trained and evaluated solely on these public datasets and generated annotations, and we do not attempt to infer biometric identity or other sensitive attributes from the videos.

J Discussion and Future Directions

This work takes a step toward perception-centric video thinking by coupling verifiable, open-ended RLVR data with an adaptive high-FPS rewatch mechanism. We briefly outline several directions that we believe are important for future research.

First, on the evaluation side, current video benchmarks are still heavily dominated by multiple-choice formats and relatively shallow perceptual demands. Our results suggest that high-quality, open-ended video reasoning benchmarks that are both complex and perception-centric are crucial for stress-testing multimodal test-time thinking. Designing such benchmarks at scale, with reliable verification and rich long-form reasoning traces, remains an open challenge.

Second, while our adaptive rewatch mechanism already shows clear benefits, it still represents a narrow class of agentic operations: the policy can only decide when and where to re-sample video segments at a higher frame rate, using a fixed function interface that we predefine. In the longer term, we expect more general forms of agentic perception and interaction to be necessary, where models can synthesize new tools or tool-calling programs (for example, via code generation) rather than only filling in a few arguments to a fixed set of operations.

Third, our improvements currently rely on a specific RLVR training pipeline and a particular family of synthetic environments. A broader ecosystem of multimodal RLVR data and environments will likely be needed to fully unlock test-time scaling for video reasoning. This includes more diverse video domains, richer forms of supervision (such as hierarchical or multi-turn tasks), and environments where perception, reasoning, and decision-making are all jointly optimized. Exploring such settings is a promising avenue for future work. Further, we leave intermediate-step supervision beyond outcome-based RLVR to future work.