ADAVDR: ADAPTIVE TOOL USE AND REFLECTION FOR VIDEO DEEP RESEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Video deep research aims to answer questions by jointly understanding video content and retrieving external knowledge from the open Web. However, these tasks often require diverse and multi-step tool-use trajectories, which may lead to unnecessary video tool calls, incurring additional costs and failure risk. In this paper, we propose AdaVDR, an *adaptive video deep research agent* with adaptive tool use and reflection, which selects tools based on task and model capabilities while backtracking when needed. To this end, we develop a data construction pipeline that builds adaptive tool-use trajectories according to task types and model capabilities. Specifically, we construct QA pairs that require both video and external information, and organize the information acquisition processes into task-specific tool-use trajectories. We then apply model-conditioned tool necessity filtering to remove tools that can be bypassed using video understanding and internal knowledge of the model. Based on this pipeline, we construct a training dataset with adaptive trajectories and perform SFT on them to initialize these capabilities of AdaVDR. To further improve adaptive capabilities, we propose NASA-GRPO, a necessity-aware segment-guided credit assignment method, which evaluates segment-guided evidence contribution and tool necessity to construct normalized tool-level rewards. We also construct VDR-EE, a benchmark built using the proposed pipeline. Experiments show that AdaVDR outperforms the baselines by up to 15% and 29% on VDR-EE and the existing VideoDR, respectively, demonstrating its superiority.

1 INTRODUCTION

Video deep research is an important task that aims to answer complex questions by jointly reasoning over video content and external knowledge. Solving such tasks requires not only understanding the video, but also retrieving, verifying, and integrating relevant information from the Web (Liu et al., 2026a; Gao et al., 2026; Liang et al., 2025). A video deep research agent invokes video understanding tools, such as grounding, and retrieval tools, such as image search, to progressively acquire evidence and complete the task (Zhang et al., 2026a; Li et al., 2026a; Wu et al., 2025).

A key challenge in video deep research is that unnecessary or inappropriate tool calls lead to inefficient costs and incorrect answers during multi-step tool use. First, diverse questions and videos require different tool-use trajectories, and inappropriate tool use may fail to acquire critical evidence (Yuan et al., 2025; Li et al., 2026a). Second, video grounding and external retrieval tools are inherently error-prone (Zhi et al., 2025; Zuo et al., 2026; Liu et al., 2026a), so unnecessary calls can introduce erroneous information and disrupt subsequent process (Li et al., 2026b; Chu et al., 2026). Our theoretical analysis further shows additional tool calls increase prediction risk when tools are unreliable. These observations motivate us to adaptively select necessary tools according to task requirements, model capabilities, and internal knowledge.

In this paper, we propose AdaVDR, as illustrated in Fig. 1, an *adaptive video deep research agent* with adaptive tool invocation and reflection. For tool invocation, the agent selects necessary tools according to the question type, its video understanding capability, and internal knowledge. For example, AdaVDR skips temporal grounding when the relevant frame can be directly located, and image search when the entity can be recognized. For reflection, AdaVDR backtracks when inter-

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.

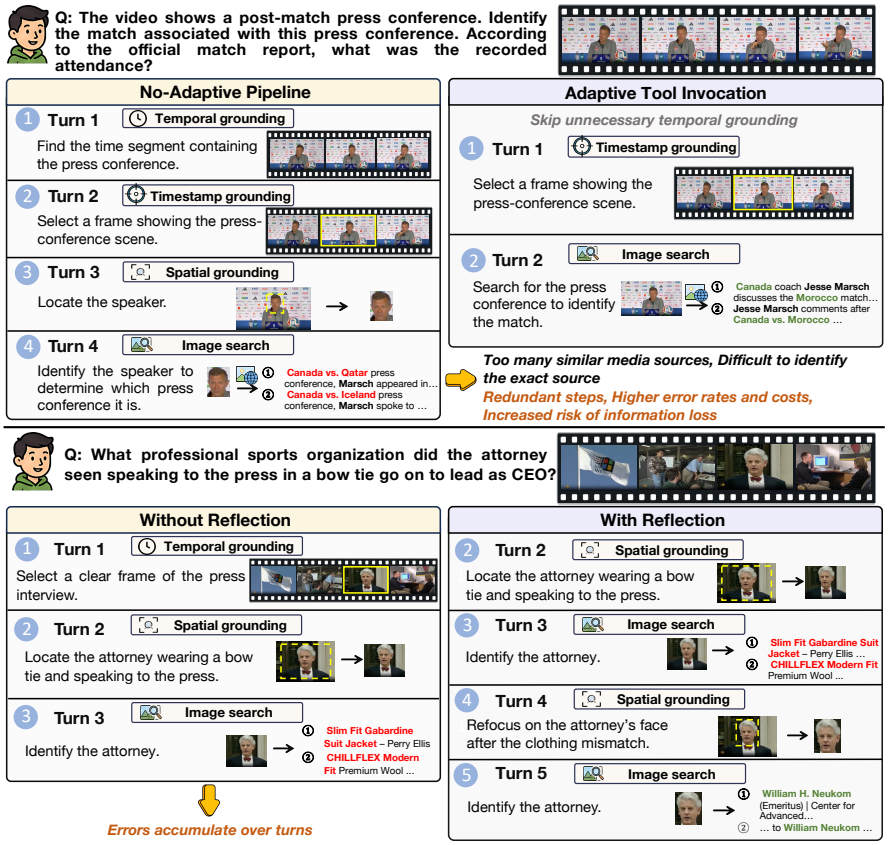

Figure 1: **Adaptive tool use and reflection in AdaVDR.** AdaVDR adaptively selects tools and reflects on unreliable evidence, reducing redundant calls and error propagation.

mediate evidence is unreliable, re-performing according to the failure. AdaVDR reduces redundant tool use and prevent unreliable evidence from affecting subsequent processes.

To equip AdaVDR with the adaptive tool-use capability, we propose a video deep research data construction pipeline consisting of two stages: *QA generation* and *trajectory generation*. For QA generation, we identify events and entities in videos that require external retrieval, and combine grounding and retrieval to acquire relevant information and construct QA pairs that rely on both video content and external knowledge. For trajectory generation, we organize and refine the grounding and retrieval processes into tool-use trajectories according to the information needs of different questions. We then perform tool necessity filtering based on video understanding capability and internal knowledge of the target model, removing tools that the model can bypass to obtain adaptive training trajectories tailored to its capabilities. Based on the pipeline, we construct a training dataset and a benchmark, VDR-EE, which includes fine-grained annotations of question types.

We perform supervised fine-tuning (SFT) on the constructed adaptive trajectories to equip AdaVDR with initial adaptive capabilities. To further improve adaptive capabilities of AdaVDR, we propose NASA-GRPO, a necessity-aware segment-guided credit assignment method, which assigns tool-level rewards to tool calls according to their importance to strengthen the adaptive ability. NASA-GRPO evaluates segment-guided evidence contribution and tool necessity, and applies tool-type reward normalization to construct tool-level rewards. We evaluate AdaVDR on both the constructed VDR-EE benchmark and the existing VideoDR benchmark. Our method outperforms the baselines by up to 15% and 29% on VDR-EE and VideoDR, respectively, demonstrating its superiority. Results also demonstrate that our agent effectively reduces redundant tool calls, leading to improved performance and lower inference latency. The contributions can be summarized as four-fold.

- We propose AdaVDR, an adaptive video deep research agent that selects tools according to task requirements and model capabilities, and selectively reflects on unreliable intermediate evidence.

- We develop a video deep research data construction pipeline that generates task-specific tool-use trajectories and filters unnecessary tools according to target model capabilities.
- We provide training data and a benchmark with fine-grained question-type annotations to support future research.
- We propose NASA-GRPO, a necessity-aware segment-guided credit assignment method that computes tool-level rewards for tool importance and improves necessary tool identification.

2 Related Work

2.1 Video Understanding Agents

Video understanding agents iteratively locate essential evidence within videos. Representative systems such as LVAgent (Chen et al., 2025), VCA (Yang et al., 2025b), and Deep Video Discovery (Zhang et al., 2026a) identify relevant segments, frames, actions, or entities through agentic exploration. Recent methods further improve this process through uncertainty-aware plan adjustment (Zhi et al., 2025), hierarchical memory backtracking (Zuo et al., 2026), parallel segment-level perception and global aggregation (Pang & Wang, 2025), or dynamic temporal grounding (Yuan et al., 2025). However, these methods acquire evidence within videos, while Video Deep Research requires external knowledge. We build a video deep research agent that jointly coordinates video understanding and external retrieval to acquire sufficient evidence.

2.2 Multimodal Deep Research

Deep-search agents use external tools (Yao et al., 2022; Jin et al., 2025; Li et al., 2025), while multimodal agents add visual search and grounding (Wu et al., 2025; Hong et al., 2026; Chen et al., 2026; Zhang et al., 2026c; Gao et al., 2024; Zhang et al., 2025; He et al., 2025). Recent work explores adaptive tool use and longer visual search (Yan et al., 2026; Zhang et al., 2026b; Li et al., 2026b; Chu et al., 2026). VideoSearcher, VideoRover, and Video-DeepResearch extend deep research to videos (Gao et al., 2026; Liu et al., 2026b; Fang et al., 2026). Our work instead adapts grounding and retrieval to the task and model, while reflecting only on unreliable evidence to reduce redundant calls and error propagation.

3 Adaptive Video Deep Research

We first formulate video deep research as a multi-turn reasoning and evidence-acquisition task over videos and external tools, and characterize its unique challenges. We present a theoretical analysis of the risks induced by unnecessary tool use. We then introduce an adaptive agent mechanism that dynamically determines both which tools to invoke and when reflection is necessary.

3.1 Problem Definition

Given a video V and a question q, video deep research aims to generate an answer a by iteratively reasoning and using tools $\mathcal{T} = \{T_1, \ldots, T_K\}$. The toolset includes three categories: grounding (temporal grounding, timestamp grounding, and spatial grounding), retrieval (image search, web search, and page visits), and termination for final answer. At each interaction turn t, the agent observes the history $h_t = (q, v, a_1, o_1, \ldots, a_{t-1}, o_{t-1})$ and generates an output $a_t = \pi(h_t)$, where π is a VLM acting as the agent brain. At each non-terminal turn, a_t contains reasoning and a selected tool with arguments, while o_t denotes the observation returned by executing a_t. Once sufficient evidence been acquired, the final output only contains the reasoning and answer.

A key challenge in video deep research is that unnecessary or inappropriate tool calls lead to incorrect answers during multi-step tool use, which is further proved in Section 3.2. The challenge arises from two aspects. Video deep research involves more diverse question types and video styles, leading to substantially different information needs and reasoning trajectories across tasks. *Event-centric questions* focus on identifying events, spatial grounding is often unnecessary. *Entity-centric questions* focus on identifying fine-grained entities, and thus spatial grounding and OCR are necessary to acquire fine-grained information. **An inappropriate tool-use trajectory fails to acquire the essential information, leading to incorrect results.** Moreover, grounding and retrieval are

inherently uncertain. When the required information is available from video understanding or internal knowledge, **redundant tool calls increase latency and may introduce noisy evidence that degrades subsequent reasoning.**

3.2 Theoretical Analysis

Let $X = \{V, q\}$ denote the input and Y the target output. For a model M, let $H_M = h_M(X)$ denote the information extracted by the model from X. Let Z denote the observation from an external tool. Let $C \in \{0, 1\}$ indicate whether the observation is erroneous, and let $\epsilon = \Pr(C = 1)$ denote the tool error probability.

Theorem 1. *We define the* base risk R_{base} *and the* tool-augmented risk R_{tool} *as*

$$R_{\text{base}} = \mathbb{E}\left[-\log P(Y \mid H_M; M)\right], R_{\text{tool}} = \mathbb{E}\left[-\log P(Y \mid H_M, Z; M)\right]. \tag{1}$$

The difference between the base risk R_{base} and the tool-augmented risk R_{tool} satisfies

$$R_{\text{tool}} - R_{\text{base}} \geq 2\epsilon\delta^2 - \Gamma_M - \xi_M. \tag{2}$$

Γ_M measures the necessary evidence in X that is not captured by H_M. δ measures how strongly erroneous tool outputs distort the model prediction. ξ_M measures the mismatch between the ideal prediction and the actual prediction of model M given H_M.

Specific definition and proof are presented in appendix. Theorem 1 shows that using external tools introduce additional prediction risk, when $2\epsilon\delta^2 > \Gamma_M + \xi_M$. Video deep search is characterized by a high tool error probability ϵ, resulting in a large $2\epsilon\delta^2$. When the model captures sufficient information (small Γ_M), such that $2\epsilon\delta^2 > \Gamma_M + \xi_M$, it tends to $R_{\text{tool}} > R_{\text{base}}$. In contrast, when information remains uncaptured (large Γ_M), it tends to $R_{\text{tool}} < R_{\text{base}}$. This motivates our adaptive tool use, which selects tools according to the requirements of each task and the capability of the target model, thereby minimizing the prediction risk.

3.3 Adaptive Agent

We develop AdaVDR, an adaptive agent with adaptive tool invocation and reflection. For tool invocation, AdaVDR selects the most appropriate tool-use trajectory according to the question type. AdaVDR further adapts tool use to its own capability and internal knowledge, skipping grounding or retrieval steps when the information can be obtained directly. For reflection, only unreliable results trigger AdaVDR to identify the problematic tool calls and backtrack for correction. Thus, AdaVDR avoids inappropriate or unnecessary tool calls and limits the spread of unreliable information.

To equip AdaVDR with such adaptive capabilities, we construct adaptive tool-use trajectories according to task types and target model capabilities (Section 4) and perform SFT on them to initialize these capabilities. We introduce a novel reinforcement learning (RL) method that builds tool-level rewards based on tool contribution and necessity, further strengthening adaptive tool use (Section 5).

4 Data Collection

4.1 QA Generation

We discover retrieval-relevant events and entities from videos, then ground these visual anchors and retrieve associated external knowledge. Based on the evidence, we construct and filter questions that require both video information and external knowledge. Details are provided in Appendix C.1.

Video Selection and Anchor Discovery. We collect videos from YouTube and scene-understanding datasets, such as EgoExo4D (Grauman et al., 2024) and EgoLife (Yang et al., 2025a). Gemini-3.5-Flash analyzes each video together with its ASR transcript to identify events and entities for subsequent processes. These anchors include both recognizable targets and unfamiliar visual content whose exact identity must be resolved through search.

Video-to-Web Evidence Acquisition. We first acquire sufficient video information to identify each target. For events and actions, we use temporal grounding to locate the relevant segment, timestamp grounding to select informative frames, and image search together with ASR to identify the event. For entities, we additionally apply spatial grounding and use the resulting crop for image search,

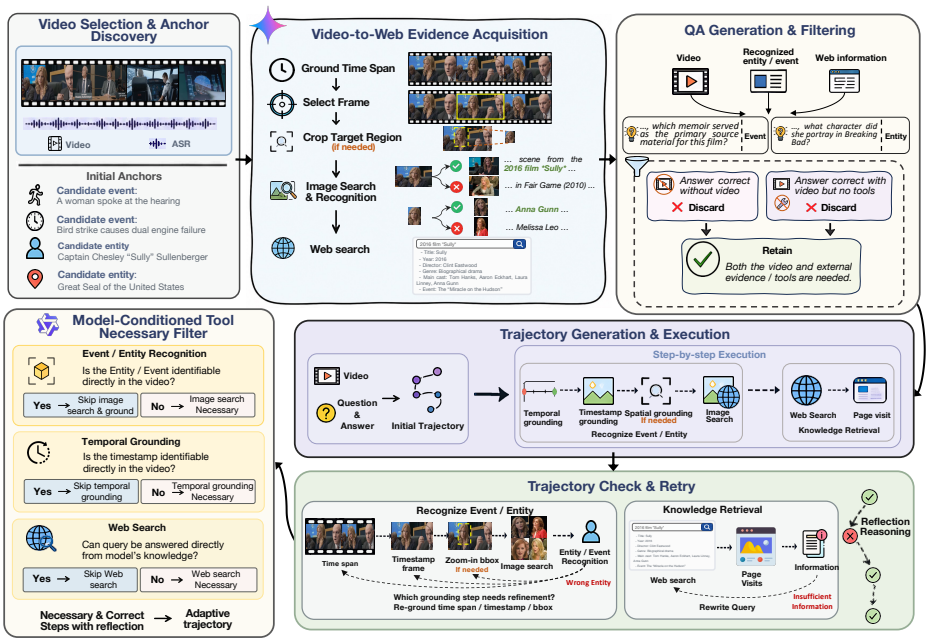

Figure 2: Overview of the data construction pipeline.

while OCR and ASR provide complementary visual and spoken information. We then use web or image search to retrieve related external knowledge. We prioritize recent and time-sensitive facts, fine-grained details, and multi-hop relations involving multiple entities or events, and organize the retrieved information into a traceable evidence path for subsequent processes.

QA Generation and Filtering. We construct QA pairs using grounded video information and external evidence, with each question requiring both sources for answering. To ensure that both information sources are necessary, we filter the generated questions using Qwen3-VL-235B under two restricted settings: without access to the video, and with the video but without external tools. Questions answered correctly under either setting are removed.

4.2 Trajectory Generation

For each retained QA pair, we construct and verify a task-specific tool-use trajectory, retaining retry and correction steps as adaptive reflection data. We then apply model-conditioned tool necessity filtering to remove tool calls or chains that the target model can bypass using its own video understanding or internal knowledge. Further details are provided in Appendix C.2.

Trajectory Generation and Execution. Using the collected grounding results and external evidence, we construct executable trajectories tailored to the information requirements of each question. We verify the logical dependencies and parameters of tool calls to ensure that each action uses the correct prerequisite output, and then execute the refined trajectory step by step.

Trajectory Check and Retry. After execution, we verify whether each tool produces the expected information. When retrieval fails, we revise the relevant input and retry. For image search, we select a better frame or crop; for web search, we reformulate the query. This process continues until the required evidence is recovered or the maximum number of attempts is reached. We retain failed attempts, failure diagnoses, and subsequent corrections as adaptive reflection data, teaching the model to reflect only when intermediate results are unreliable.

Model-conditioned Tool Necessity Filtering. For each successfully executed trajectory, we evaluate whether a tool or tool chain $\mathcal{S}$ is necessary for the target model that will subsequently be fine-tuned on the resulting trajectories. Let $\mathcal{M}_{\text{tgt}}$ denote this target model, $x_{\mathcal{S}}$ the context available before executing $\mathcal{S}$, $y_{\mathcal{S}}$ its target information, and $q_{\mathcal{S}}$ a verification query asking for that information. We bypass $\mathcal{S}$ and compute

$$\hat{y}_{\mathcal{S}} = \mathcal{M}_{\text{tgt}}(x_{\mathcal{S}}, q_{\mathcal{S}}), \qquad N(\mathcal{S}) = \mathbf{1}[\hat{y}_{\mathcal{S}} \neq y_{\mathcal{S}}]\,. \tag{3}$$

If $N(\mathcal{S}) = 0$, the target information can be recovered without $\mathcal{S}$, so the tool or chain is removed; otherwise, it is retained. We first evaluate the longest candidate chain and then examine its individual tools, producing model-specific trajectories that retain only actions that cannot be bypassed using the model's own video understanding or internal knowledge.

4.3 Training Data

Following the above pipeline, we can automatically construct SFT data with tool-use trajectories. Different question types correspond to different tool-use trajectories, while model capability filtering retains only necessary tool calls, enabling the agent to acquire adaptive capabilities through SFT. For data of RL, we only construct QA pairs following the procedure in Section 4.1.

4.4 Benchmark

By the QA generation process, we introduce **VDR-EE**, a manually verified Video Deep Research benchmark. Each QA pairs is manually verified to ensure clear identification of the related entity or event in the video, dependence on both video and external evidence, question clarity, and answer support from retrieved evidence. We retain 250 manually verified questions.

Semantic domain. Each retained question is manually assigned to one of seven domains: culture, entertainment, industry, news, scene understanding, science, and sports. The resulting distribution across these seven domains is shown in Fig. 5 in Appendix D.

Video-centric categories. VDR-EE contains entity-centric and event-centric questions. We further divide entity questions into single-entity questions and multi-entity questions. Event questions are grouped by the duration of the grounded target event: short ($\leq$ 30 s), medium (30–120 s), and long ($>$ 120 s). All questions are also grouped by full-video duration into short ($\leq$ 5 min), medium (5–10 min), and long ($\geq$ 10 min) videos. Table 5 in Appendix D summarizes these annotations, including 153 entity and 97 event questions. Representative questions are shown in Fig. 6 in Appendix D.

5 Training

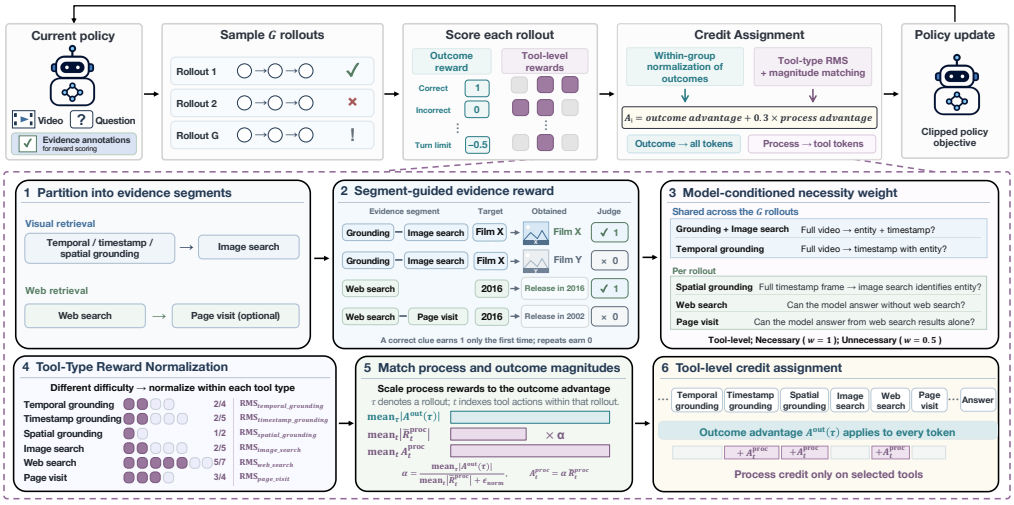

Figure 3: Overview of NASA-GRPO, a necessity-aware segment-guided credit assignment RL.

5.1 Cold-Start Supervised Fine-Tuning

We perform supervised fine-tuning (SFT) on 2.9K verified trajectories from Sec. 4.2 to initialize adaptive tool use and reflection. Given M trajectories, where $x^{(n)}$ denotes the interaction context and $y^{(n)} = \{y_1^{(n)}, \ldots, y_{N_n}^{(n)}\}$ the target assistant output, we optimize:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{M} \sum_{n=1}^{M} \sum_{i=1}^{N_n} \log \pi_\theta \left(y_i^{(n)} \mid x^{(n)}, y_{<i}^{(n)} \right). \tag{4}$$

The loss is applied only to assistant-generated tokens, while video inputs, user messages, and tool observations serve as conditioning context. The resulting checkpoint serves as the initial RL policy.

5.2 NASA-GRPO

In video deep research, a rollout typically involves multiple tool calls, some of which may fail or be redundant. Relying solely on the outcome reward provides sparse supervision, making it difficult to learn tool-use policies. Moreover, evidence often requires a sequence of consecutive tool calls to obtain. We propose NASA-GRPO, a necessity-aware segment-guided credit assignment method for GRPO, which assigns tool-level rewards according to their importance within each evidence-acquisition segment. As shown in Fig. 3, NASA-GRPO partitions rollouts into evidence-acquisition segments, scores each segment's evidence contribution and each tool's necessity, normalizes rewards by tool type, and combines them with the outcome reward.

Outcome Reward. Given a training sample (V, q, a^*), the outcome reward of a rollout τ is

$$R^{\text{out}}(\tau) = \begin{cases} 1, & \text{if the final answer is correct,} \\ 0, & \text{if the final answer is incorrect,} \\ -0.5, & \text{if the rollout is invalid or reaches the turn limit.} \end{cases} \tag{5}$$

GRPO normalizes the outcome reward within a sampled rollout group $\mathcal{G}$, which is computed as

$$A^{\text{out}}(\tau) = \frac{R^{\text{out}}(\tau) - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}} + \varepsilon_{\text{norm}}}, \tag{6}$$

where $\mu_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ denote the group mean and standard deviation.

Segment-Guided Evidence Reward. Given (V, q, a^*), our data construction provides target clues $\mathcal{C}^*$, including video entities and external facts required to answer the question. We partition rollout $\tau = (a_1, o_1, \ldots, a_T, o_T)$ into segments $\mathcal{S} * m * m = 1^M$, each of which corresponds to one external information-acquisition process, including both the tool actions that prepare the retrieval input and the retrieval actions that obtain the target information. Each segment targets one clue in $\mathcal{C}^*$. For visual retrieval, a segment includes grounding and image-search tools. For Web retrieval, a segment includes a Web-search tool and may include page-visit tool. Let $\widehat{c}_m$ denote the clue obtained by $\mathcal{S}_m$. We assign evidence reward only when the clue is both correct and has not been obtained,

$$S_m^{\text{evid}} = \mathbb{I}\left[J(\widehat{c}_m, c_m^*) = 1\right] \cdot \mathbb{I}\left[c_m^* \notin \mathcal{C}_{<m}^{\text{credit}}\right], \tag{7}$$

where $\mathcal{C}_{<m}^{\text{credit}}$ contains the target clues credited in earlier segments. The first term measures clue correctness, while the second measures whether the clue has been previously obtained.

Model-Conditioned Necessity Weight. A segment that obtains valid evidence may still contain unnecessary tool actions. We measure tool necessity by bypassing each action and testing whether the same target information can still be recovered without it. An action receives full rewards when the model cannot recover its target information without the action. Otherwise, the action retains half reward. Mathematically, for an action a_t in segment $s(t)$, the necessity weight is computed as

$$w_t^{\text{nec}} = \begin{cases} 1, & \text{if } \widehat{y}_{a_t}^{-} \neq y_{a_t}, \\ 0.5, & \text{if } \widehat{y}_{a_t}^{-} = y_{a_t}, \end{cases} \tag{8}$$

where y_{a_t} denotes its target information, h_t^- denotes the input used after bypassing the action, and $\widehat{y}_{a_t}^{-} = \mathcal{M}_\phi(h_t^-, q_{a_t}^{\text{verify}})$ denotes the output of the model without a_t. Combining the segment-guided evidence credit, the credit of action a_t is computed as

$$R_t^{\text{proc}} = S_{s(t)}^{\text{evid}} \cdot w_t^{\text{nec}}. \tag{9}$$

Tool-Type Reward Normalization. Different tool types receive positive rewards at different rates due to differences in task difficulty. We therefore normalize process rewards within each tool type to prevent easier tools from dominating the learning signal. Let $k(t)$ denote the type of action a_t, and let $\mathcal{B}_k$ contain all actions of type k in the rollout batch. The process reward is normalized as

$$\widetilde{R}_t^{\text{proc}} = \frac{R_t^{\text{proc}}}{\sqrt{\frac{1}{|\mathcal{B}_{k(t)}|}\sum_{j \in \mathcal{B}_{k(t)}} \left(R_j^{\text{proc}}\right)^2} + \varepsilon_{\text{norm}}}. \tag{10}$$

We further match the average magnitude of the process signal to that of the outcome advantage:

$$\alpha = \frac{\mathrm{mean}_\tau\,|A^{\mathrm{out}}(\tau)|}{\mathrm{mean}_t\,|\widetilde{R}_t^{\mathrm{proc}}| + \varepsilon_{\mathrm{norm}}}, \qquad A_t^{\mathrm{proc}} = \alpha \widetilde{R}_t^{\mathrm{proc}}. \tag{11}$$

Tool-Level Credit Assignment. The outcome advantage is applied to all tokens. The tool-level advantage is applied only to the tokens of the credited tool action. Let $m_{t,i} = 1$ if token i belongs to action a_t, and $m_{t,i} = 0$ otherwise. The final token reward is

$$A_i = A^{\mathrm{out}}(\tau) + \lambda \sum_t m_{t,i} A_t^{\mathrm{proc}}, \qquad \lambda = 0.3. \tag{12}$$

6 Experiment

6.1 Implementation Details

We evaluate AdaVDR-9B, initialized from Qwen3.5-9B (Team, 2026), on VDR-EE and VideoDR (Liu et al., 2026a). Training details for AdaVDR-9B are provided in Appendix A and Appendix G. Baselines include Gemini-3.1-Pro and Gemini-3-Flash (Team et al., 2023), GPT-5.4 (Singh et al., 2025), Qwen3.5-35B-A3B and Qwen3.5-9B (Team, 2026), and Qwen3-VL-32B-Instruct and Qwen3-VL-8B-Instruct (Bai et al., 2025). We also include VideoSearcher-8B (Gao et al., 2026), VideoRover-8B (Liu et al., 2026b), and Video-DeepResearch-30B-A3B (Fang et al., 2026). Direct answers from the video alone; Workflow separates video-cue extraction from Web retrieval; Agentic lets a multimodal agent coordinate video understanding, tool use, and evidence integration. We report AdaVDR-9B under Agentic and use GPT-5.4 to judge answer accuracy.

Table 1: Accuracy (%) on benchmkarks. VDR-EE uses the **Direct** and **Agentic** settings, while VideoDR uses the **Workflow** and **Agentic** settings. AdaVDR results are averaged over three runs.

Model	**VideoDR**	**VDR-EE**								**Avg.**
	Avg.	**Entity**			**Event**				**Avg.**	
		Single	**Multiple**	**Overall**	**Short**	**Medium**	**Long**	**Overall**		
	Workflow	*Direct*								
Proprietary										
Gemini-3.1-Pro (Team et al., 2023)	–	56.58	28.57	42.48	54.29	55.88	50.00	53.61	46.80	–
Gemini-3-Flash (Team et al., 2023)	59.00	42.11	22.08	32.03	51.43	61.76	46.43	53.61	40.40	49.70
GPT-5.4 (Singh et al., 2025)	51.00	35.53	23.38	29.41	28.57	41.18	39.29	36.08	32.00	41.50
Open-source										
Qwen3-VL-8B-Instruct (Bai et al., 2025)	28.00	21.05	5.19	13.07	8.57	14.71	10.71	11.34	12.40	20.20
Qwen3-VL-32B-Instruct (Bai et al., 2025)	36.00	30.26	9.09	19.61	8.57	14.71	14.29	12.37	16.80	26.40
Qwen3.5-9B (Team, 2026)	35.00	23.68	11.69	17.65	5.71	2.94	7.14	5.15	12.80	23.90
Qwen3.5-35B-A3B (Team, 2026)	39.00	32.89	9.09	20.92	11.43	5.88	7.14	8.25	16.00	27.50
	Agentic	*Agentic*								
Proprietary										
Gemini-3.1-Pro (Team et al., 2023)	–	73.68	50.65	62.09	71.43	64.71	67.86	68.04	64.40	–
Gemini-3-Flash (Team et al., 2023)	–	64.47	40.26	52.29	68.57	82.35	60.71	71.13	59.60	–
GPT-5.4 (Singh et al., 2025)	–	59.21	40.26	49.67	54.29	61.76	57.14	57.73	52.80	–
Open-source										
Qwen3-VL-8B-Instruct (Bai et al., 2025)	30.00	42.11	22.08	32.03	22.86	26.47	17.86	22.68	28.40	29.20
Qwen3-VL-32B-Instruct (Bai et al., 2025)	38.00	60.53	32.47	46.41	22.86	32.35	17.86	24.74	38.00	38.00
Qwen3.5-9B (Team, 2026)	37.00	56.58	28.57	42.48	34.29	32.35	42.86	36.08	40.00	38.50
Qwen3.5-35B-A3B (Team, 2026)	42.00	56.58	38.96	47.71	42.86	52.94	35.71	44.33	46.40	44.20
VideoSearcher-8B (Gao et al., 2026)	53.00	53.95	32.47	43.14	54.29	55.88	53.57	54.64	47.60	50.30
VideoRover-8B (Liu et al., 2026b)	56.00	–	–	–	–	–	–	–	–	–
Video-DeepResearch-30B-A3B (Fang et al., 2026)	62.00	61.84	36.36	49.02	62.86	64.71	60.71	62.89	54.40	58.20
AdaVDR-9B (Qwen3.5-9B)	**66.00**	**63.16**	**37.66**	**50.33**	**62.86**	**64.71**	**60.71**	**62.89**	**55.20**	**60.60**
Δ vs. Qwen3.5-9B	+29.00	+6.58	+9.09	+7.85	+28.57	+32.36	+17.85	+26.81	+15.20	+22.10

6.2 Main Results

We evaluate the models on VDR-EE under the Direct and Agentic settings and on VideoDR under the Workflow and Agentic settings. Results are shown in Table 1. Additional results on VDR-EE and VideoDR, as well as evaluations on image deep research and long-video understanding, are provided in Appendix E. Additional qualitative examples are provided in Appendix E.3.

AdaVDR consistently improves performance on both benchmarks. AdaVDR outperforms the base model Qwen3.5-9B on both VDR-EE and VideoDR, improving average accuracy by up to 15% and 29%, respectively. The improvements are consistent across entity- and event-centric questions, all seven domains on VDR-EE, and different domains and difficulty levels on VideoDR (Tables 6, 8, and 9). These results demonstrate the superiority of the proposed method. Agentic interaction

consistently improves performance. For all baselines with paired results, the Agentic setting outperforms the Direct setting on VDR-EE and the Workflow setting on VideoDR (Table 1). These results show the benefit of allowing the model to autonomously coordinate video grounding and external retrieval rather than relying on direct answering or a predefined workflow.

6.3 Tool-Use Analysis

Table 2: Average tool calls and turns per question across training stages on VideoDR and VDR-EE.

Model	Video Grounding			External Retrieval			Trajectory
	Temporal	Timestamp	Spatial	Image Search	Web Search	Visit Page	Avg. Turns
VideoDR							
Qwen3.5-9B	0.46	0.24	0.07	0.12	2.56	0.67	5.12
AdaVDR-9B-SFT	1.55	1.73	0.44	0.93	4.07	0.46	10.18
AdaVDR-9B-RL	1.09	1.14	0.02	1.09	4.81	2.52	11.67
VDR-EE							
Qwen3.5-9B	0.99	0.79	0.12	0.40	2.12	0.28	5.70
AdaVDR-9B-SFT	1.28	1.16	0.78	0.99	5.05	0.55	10.81
AdaVDR-9B-RL	1.04	1.11	0.06	1.09	5.54	3.42	13.26

Table 2 shows that SFT makes more tool calls than the base model on both benchmarks. Compared with SFT, NASA-GRPO makes fewer grounding calls but more image searches, web searches, and page visits. On the one hand, SFT may hallucinate citations when retrieval is inconclusive, whereas NASA-GRPO rewards obtaining supporting evidence. On the other hand, correct grounding does not guarantee a useful image-search result, so NASA-GRPO may favor more informative frames and crops. Accuracy improves by 13.0% and 7.2% over the best SFT configuration on VideoDR and VDR-EE, respectively (Table 3). Detailed tool-use analyses are outlined in Appendix F.

6.4 Ablation

Table 3 shows how the training components affect performance. SFT on tool-use trajectories teaches the model to use tools and improves accuracy. Adding adaptive data reduces the number of tool calls, but accuracy declines slightly. This suggests that SFT learns to call tools less often without reliably identifying which calls are unnecessary. Adding reflection data to the adaptive data increases tool use again and improves accuracy by helping the model revisit unreliable results.

Table 3: Ablation of SFT data composition and RL reward components on VideoDR and VDR-EE (accuracy).

Configuration	VideoDR	Δ	VDR-EE	Δ
Base Model	37.00	–	40.00	–
SFT Data Composition				
Base Data	50.00	+13.00	47.20	+7.20
Adaptive Data	48.00	−2.00	46.00	−1.20
Reflection Data	51.00	+1.00	47.60	+0.40
Reflection + Adaptive Data	52.00	+2.00	48.00	+0.80
Necessity-aware segment-guided credit assignment				
Accuracy + Format	53.00	+1.00	48.80	+0.80
+ Segment-Guided Evidence Reward	57.00	+4.00	50.80	+2.00
+ Tool-Type Normalization	59.00	+2.00	52.00	+1.20
+ Model-Conditioned Necessity Weight	**66.00**	**+7.00**	**55.20**	**+3.20**

In RL, segment-guided credit assignment provides explicit rewards for actions that obtain necessary evidence, producing a much larger gain. We observe that web search is called more often, which is easier to obtain useful evidence than image search. As a result, rewards can bias the model toward web search even when visual retrieval is needed. Tool-type normalization reduces this bias and further improves accuracy. A clue can be new to the trajectory yet obtainable without a tool; rewarding such calls encourages redundant interactions. Moreover, the model-conditioned necessity weight guides the agent to focus on tool calls that the model actually needs. It yields the largest accuracy gain and reduces average interaction turns from 13.67 to 11.67 on VideoDR and from 15.43 to 13.26 on VDR-EE. More ablations are in Appendix H.

7 Conclusion

In this work, we study Video Deep Research and propose AdaVDR, an adaptive video deep research agent designed to handle diverse tool-use requirements. The developed data construction pipeline generates adaptive tool-use trajectories according to task requirements and model capabilities. The constructed training dataset initializes AdaVDR with adaptive tool-use capabilities through SFT, while NASA-GRPO further improves its tool-use policy. The resulting AdaVDR selects tools according to task requirements and model capabilities and corrects unreliable intermediate results only when necessary. Experiments demonstrate that AdaVDR improves both accuracy and efficiency.

AI USE STATEMENT

Generative AI tools were used to assist with drafting and editing the manuscript, organizing and interpreting experimental analyses, and preparing LaTeX tables and figures. They also helped improve the clarity and structure of the presentation. Numerical results and scientific claims must be verified against the underlying experiments and cited sources, not accepted solely from AI-generated suggestions. The authors reviewed the AI-assisted material and take responsibility for the final text, analyses, and artifacts.

ETHICS STATEMENT

This work uses videos from YouTube and existing datasets, which may contain identifiable people or copyrighted material. Public availability does not itself grant permission to redistribute these videos; any use or release of source material must follow the relevant terms and privacy requirements. Retrieved web evidence may be incomplete, outdated, or misleading. We check video-grounded clues and external evidence during data construction and manually review benchmark questions, but errors may remain. The benchmark is intended for research on video reasoning and tool use and should not be used for surveillance or profiling.

REPRODUCIBILITY STATEMENT

To support reproducibility, Appendix A reports the SFT and RL hyperparameters, rollout configuration, and reward details. Appendix C describes video selection, QA filtering, trajectory construction, and tool-necessity checks, while Appendix D documents the VDR-EE composition and annotations. We specify the evaluation settings in the main text and provide the judge and data-construction prompts in Appendix J. Because external search results can change over time, exact tool observations may vary across runs.

REFERENCES

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

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20237–20246. IEEE, 2025.

Shuang Chen, Kaituo Feng, Hangting Chen, Wenxuan Huang, Dasen Dai, Quanxin Shou, Yunlong Lin, Xiangyu Yue, Shenghua Gao, and Tianyu Pang. Opensearch-vl: An open recipe for frontier multimodal search agents. *arXiv preprint arXiv:2605.05185*, 2026.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, Shengchao Hu, et al. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*, 2026.

Zhen Fang, Yu Zeng, Wenxuan Huang, Yiming Zhao, Shiting Huang, Tianfei Ren, Qi Lu, Qingnan Ren, Qisheng Su, Lionel Z Wang, et al. Video-deepresearch: Towards the next-generation multimodal deepresearch agent. *arXiv preprint arXiv:2608.03979*, 2026.

Zhenkun Gao, Yicheng Bao, Jinlong Peng, Xueheng Li, Theo Huang, Bangwei Liu, Kunquan Li, Zhenye Gan, Tao Hu, Chengjun Xie, et al. Videosearcher: Empowering video deep research with multi-tool agentic reasoning via reinforcement learning. *arXiv preprint arXiv:2607.02927*, 2026.

Zhi Gao, Yuntao Du, Xintong Zhang, Xiaojian Ma, Wenjuan Han, Song-Chun Zhu, and Qing Li. Clova: A closed-loop visual assistant with tool usage and update. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13258–13268. IEEE, 2024.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d:

Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19383–19400, 2024.

Xingqi He, Yujie Zhang, Shuyong Gao, Wenjie Li, Lingyi Hong, Mingxi Chen, Kaixun Jiang, Jiyuan Fu, and Wenqiang Zhang. Rsagent: Learning to reason and act for text-guided segmentation via multi-turn tool invocations. *arXiv preprint arXiv:2512.24023*, 2025.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, and Guohai Xu. Deepeyesv2: Toward agentic multimodal model. In *International Conference on Learning Representations*, volume 2026, pp. 114851–114872, 2026.

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

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

Chenglin Li, Qianglong Chen, Feng Han, Yikun Wang, Xingxi Yin, Yan Gong, Ruilin Li, Yin Zhang, and Jiaqi Wang. Videothinker: Building agentic videollms with llm-guided tool reasoning. *arXiv preprint arXiv:2601.15724*, 2026a.

Guankai Li, Jiabin Chen, Yi Xu, Xichen Zhang, and Yuan Lu. Hypereyes: Dual-grained efficiency-aware reinforcement learning for parallel multimodal search agents. *arXiv preprint arXiv:2605.07177*, 2026b.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025.

Zhengyang Liang, Yan Shu, Xiangrui Liu, Minghao Qin, Kaixin Liang, Nicu Sebe, Zheng Liu, and Lizi Liao. Video-browser: Towards agentic open-web video browsing. *arXiv preprint arXiv:2512.23044*, 2025.

Chengwen Liu, Xiaomin Yu, Zhuoyue Chang, Zhe Huang, Shuo Zhang, Heng Lian, Jisheng Dang, Rui Xu, Sen Hu, Jianheng Hou, et al. Watching, reasoning, and searching: A video deep research benchmark on open web for agentic video reasoning. *arXiv preprint arXiv:2601.06943*, 2026a.

Wenqi Liu, Shijie Ma, Yunxiao Wang, Meng Liu, Qile Su, Han Liu, Bohan Hou, Zeyu Wang, Xuanyu Zheng, Changyi Liu, et al. Thinking beyond videos: Unifying video reasoning and deep research for open-world video agents. *arXiv preprint arXiv:2608.23329*, 2026b.

Ziqi Pang and Yu-Xiong Wang. Mr. video:" mapreduce" is the principle for long video understanding. *arXiv preprint arXiv:2504.16082*, 2025.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. Mmsearch-r1: Incentivizing lmms to search. *arXiv preprint arXiv:2506.20670*, 2(4), 2025.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

Shilin Yan, Jintao Tong, Hongwei Xue, Xiaojun Tang, Yangyang Wang, Kunyu Shi, Guannan Zhang, Ruixuan Li, and Yixiong Zou. Act wisely: Cultivating meta-cognitive tool use in agentic multi-modal models. *arXiv preprint arXiv:2604.08545*, 2026.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, et al. Egolife: Towards egocentric life assistant. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28885–28900. IEEE, 2025a.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. Vca: Video curious agent for long video understanding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20168–20179. IEEE, 2025b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Ranchi Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11704–11715, 2026.

Huaying Yuan, Zheng Liu, Junjie Zhou, Hongjin Qian, Yan Shu, Nicu Sebe, Ji-Rong Wen, and Zhicheng Dou. Videoexplorer: Think with videos for agentic long-video understanding. *arXiv preprint arXiv:2506.10821*, 2025.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. *Advances in Neural Information Processing Systems*, 38:89863–89895, 2026a.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Adaptive chain-of-focus reasoning via dynamic visual search and zooming for efficient vlms. *arXiv preprint arXiv:2505.15436*, 2025.

Xintong Zhang, Xiaowen Zhang, Jingrong Wu, Zhi Gao, Shilin Yan, Zhenxin Diao, Kunpeng Gao, Xuanyan Chen, Yuwei Wu, Yunde Jia, et al. Adaptmmbench: Benchmarking adaptive multimodal reasoning for mode selection and reasoning process. *arXiv preprint arXiv:2602.02676*, 2026b.

Zhengbo Zhang, Changtao Miao, Jinbo Su, Zhaowen Zhou, Chunxia Zhang, Xukai Wang, Ruiqi Liu, Kaiyuan Zheng, Jiansheng Cai, Bo Zhang, et al. Visual-seeker: Towards visual-native multimodal agentic search via active visual reasoning. *arXiv preprint arXiv:2606.15231*, 2026c.

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

Jialong Zuo, Yongtai Deng, Lingdong Kong, Jingkang Yang, Rui Jin, Yiwei Zhang, Nong Sang, Liang Pan, Ziwei Liu, and Changxin Gao. Videolucy: Deep memory backtracking for long video understanding. *Advances in Neural Information Processing Systems*, 38:25660–25691, 2026.

APPENDIX CONTENTS

A Training Details . . . 14
A.1 Training Parameters . . . 14
A.2 Reinforcement Learning Details . . . 14
A.3 NASA-GRPO Reward Curve . . . 15
B Proof of Theorem 1 . . . 15
C Data Collection Details . . . 17
C.1 QA Generation . . . 17
C.2 Trajectory Generation . . . 17
D Benchmark Details . . . 18
E More Experimental Results . . . 18
E.1 Results on VDR-EE . . . 18
E.2 Results on VideoDR . . . 20
E.3 Qualitative Analysis . . . 21
E.4 Image Deep Research . . . 22
E.5 Long-Video Understanding . . . 22
F More Tool-Use Analysis . . . 22
F.1 Tool Effectiveness . . . 22
F.2 Tool Necessity Analysis . . . 23
G Training Dynamics . . . 25
G.1 Process Credit During Training . . . 25
G.2 Effect of Model-Conditioned Necessity Weight on Tool Use . . . 26
H More Ablations . . . 26
H.1 Necessity Weight Ablation . . . 26
H.2 Process-Advantage Coefficient Ablation . . . 26
H.3 Tool Ablation . . . 28
I Human–Credit Agreement . . . 28
J Prompts . . . 29
J.1 LLM Judge Prompt . . . 29
J.2 Visual Anchor Extraction Prompt . . . 30
J.3 QA Generation Prompt . . . 31
J.4 Initial Trajectory Generation Prompt . . . 32
J.5 Trajectory Refinement Prompt . . . 33
J.6 Model-Conditioned Tool Necessity Filtering Prompts . . . 34
J.7 Reflection and Retry Prompts . . . 35

A Training Details

A.1 Training Parameters

We train the model in two stages: supervised fine-tuning followed by reinforcement learning.

For SFT, we use five epochs, a learning rate of 2×10^{-6}, an effective batch size of 16, a maximum sequence length of 131,072, a LoRA rank of 8, sequence parallelism of 8, and gradient accumulation of 16.

The resulting SFT checkpoint initializes the policy for GRPO. Table 4 summarizes the RL configuration.

Table 4: RL training configuration for NASA-GRPO.

Parameter	**Value**
Compute and parallelism	
Backend	Megatron
GPUs	2 nodes $\times$ 8 GPUs (16 total)
Training tensor parallelism	4
Rollout tensor parallelism	1
Pipeline parallelism	1
Context parallelism	1
Data parallelism	4
Sequence parallelism	Enabled
Dynamic batching	Enabled
Sampling and batching	
Prompts per iteration	8
Rollouts per prompt	8
Trajectories per iteration	64
Global batch size	64
Micro batch size	1
Rollout temperature	1.0
Maximum tool-interaction turns	20
Optimization	
Optimizer	Adam
Learning-rate schedule	Constant
Learning rate	1×10^{-6}
Weight decay	0.1
KL reward coefficient (`kl_coef`)	0
KL loss coefficient (`kl_loss_coef`)	0

A.2 Reinforcement Learning Details

Model-Conditioned Necessity Weight. Here, we provide a detailed explanation of how the model-conditioned necessity weight is computed for different tool actions. Evidence contribution does not imply tool necessity. A tool may return a correct clue even when the target model can obtain the same information without using it. Such a tool call is not necessary. We assign it less credit so that the policy is not encouraged to make unnecessary tool calls.

We evaluate necessity by bypassing the tool and asking the target model to recover the same target information. Some evaluations use information shared by all rollouts in the same GRPO group. We first provide the full video and ask the model to identify the target entity and its timestamp. If the model identifies the entity correctly, `image_search` and its preceding grounding actions are unnecessary. Otherwise, the visual retrieval chain is retained. We then ask the model to locate a timestamp that contains the target entity directly from the full video. If the returned timestamp contains the correct entity, the preceding `temporal_grounding` action is unnecessary; otherwise, temporal grounding is necessary. These evaluations are performed once for the rollout group because its rollouts share the same video, question, and target entities.

Other evaluations depend on the intermediate result or information target of a specific rollout. We therefore evaluate them separately for each rollout. For `spatial_grounding`, we use the full

frame selected by that rollout and perform image search without the predicted crop. If the full frame is sufficient to identify the target entity, spatial grounding is not necessary. Otherwise, the crop is necessary. We perform this evaluation only for rollouts that invoke spatial grounding. For `web_search`, we ask the model to answer the knowledge question targeted by the current rollout without using external retrieval.

A.3 NASA-GRPO Reward Curve

Figure 4 shows that the smoothed NASA-GRPO rollout reward rises over training despite fluctuations in the raw reward.

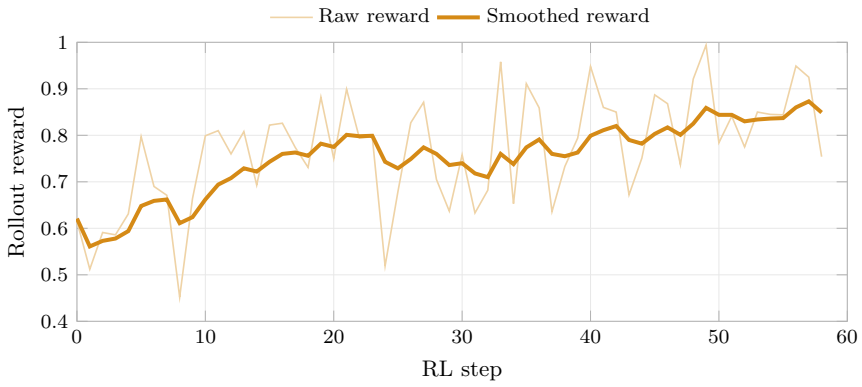

Figure 4: Rollout reward during NASA-GRPO training. Pale and dark lines show raw and smoothed rewards.

B Proof of Theorem 1

Definition 1. *We define the* residual task-relevant information Γ_M *to measure the task-relevant information in* X *that is not captured by* H_M*, formulated as*

$$\Gamma_M := I(Y; X \mid H_M). \tag{13}$$

Definition 2. *We define the* misleadingness δ *of erroneous tool observations, which measures how strongly erroneous tool outputs distort the model prediction, formulated as*

$$\delta := \mathbb{E}\left[\mathrm{TV}\left(P(Y \mid H_M, Z), P(Y \mid H_M, Z; M)\right) \mid C = 1\right]. \tag{14}$$

Definition 3. *We define the* base approximation error ξ_M*, which measures the mismatch between the ideal prediction and the actual prediction of model* M *given* H_M*, formulated as*

$$\xi_M := \mathbb{E}_{H_M}\left[D_{\mathrm{KL}}\left(P(Y \mid H_M) \,\|\, P(Y \mid H_M; M)\right)\right]. \tag{15}$$

Lemma 1. *We denote the* base risk R_{base} *and the* tool-augmented risk R_{tool} *as*

$$R_{\mathrm{base}} = \mathbb{E}\left[-\log P(Y \mid H_M; M)\right], R_{\mathrm{tool}} = \mathbb{E}\left[-\log P(Y \mid H_M, Z; M)\right]. \tag{16}$$

Let

$$\xi_{\mathrm{tool}} = \mathbb{E}_{H_M, Z}\left[D_{\mathrm{KL}}\left(P(Y \mid H_M, Z) \,\|\, P(Y \mid H_M, Z; M)\right)\right] \tag{17}$$

denote the approximation error of the model with tool augmentation. Then,

$$R_{\mathrm{tool}} - R_{\mathrm{base}} = -I(Y; Z \mid H_M) + \xi_{\mathrm{tool}} - \xi_M. \tag{18}$$

Proof. By the cross-entropy decomposition, the base risk satisfies

$$\begin{aligned} R_{\mathrm{base}} &= \mathbb{E}\left[-\log P(Y \mid H_M; M)\right] && (19)\\ &= H(Y \mid H_M) + \mathbb{E}_{H_M}\left[D_{\mathrm{KL}}\left(P(Y \mid H_M) \,\|\, P(Y \mid H_M; M)\right)\right] && (20)\\ &= H(Y \mid H_M) + \xi_M. && (21) \end{aligned}$$

Similarly, the tool-augmented risk satisfies

$$R_{\text{tool}} = \mathbb{E}\left[-\log P(Y \mid H_M, Z; M)\right] \tag{22}$$
$$= H(Y \mid H_M, Z) + \mathbb{E}_{H_M,Z}\left[D_{\text{KL}}\left(P(Y \mid H_M, Z) \,\|\, P(Y \mid H_M, Z; M)\right)\right] \tag{23}$$
$$= H(Y \mid H_M, Z) + \xi_{\text{tool}}. \tag{24}$$

Therefore,

$$R_{\text{tool}} - R_{\text{base}} = H(Y \mid H_M, Z) - H(Y \mid H_M) + \xi_{\text{tool}} - \xi_M \tag{25}$$
$$= -I(Y; Z \mid H_M) + \xi_{\text{tool}} - \xi_M, \tag{26}$$

where the last equality follows from the definition of conditional mutual information. □

Theorem 2. *We define the* base risk R_{base} *and the* tool-augmented risk R_{tool} *as*

$$R_{\text{base}} = \mathbb{E}\left[-\log P(Y \mid H_M; M)\right], R_{\text{tool}} = \mathbb{E}\left[-\log P(Y \mid H_M, Z; M)\right]. \tag{27}$$

The difference between the base risk R_{base} *and the tool-augmented risk* R_{tool} *satisfies*

$$R_{\text{tool}} - R_{\text{base}} \geq 2\epsilon\delta^2 - \Gamma_M - \xi_M. \tag{28}$$

Γ_M *measures the necessary evidence in* X *that is not captured by* H_M. δ *measures how strongly erroneous tool outputs distort the model prediction.* ξ_M *measures the mismatch between the ideal prediction and the actual prediction of model* M *given* H_M.

Proof. According to the lemma, we have

$$R_{\text{tool}} - R_{\text{base}} = -I(Y; Z \mid H_M) + \xi_{\text{tool}} - \xi_M, \tag{29}$$

where

$$\xi_{\text{tool}} = \mathbb{E}_{H_M,Z}\left[D_{\text{KL}}\left(P(Y \mid H_M, Z) \,\|\, P(Y \mid H_M, Z; M)\right)\right]. \tag{30}$$

Since the tool observation Z is generated from the original input X, we assume that $Y \to X \to Z$ forms a Markov chain. As $H_M = h_M(X)$ is a deterministic representation of X, the conditional data processing inequality gives

$$I(Y; Z \mid H_M) \leq I(Y; X \mid H_M) = \Gamma_M. \tag{31}$$

Therefore,

$$-I(Y; Z \mid H_M) \geq -\Gamma_M. \tag{32}$$

We next derive a lower bound for ξ_{tool}. Let $C \in \{0, 1\}$ indicate whether the tool output is erroneous, with $\Pr(C = 1) = \epsilon$. By conditioning on C, we obtain

$$\begin{aligned}\xi_{\text{tool}} &= (1-\epsilon)\mathbb{E}\left[D_{\text{KL}}\left(P(Y \mid H_M, Z) \,\|\, P(Y \mid H_M, Z; M)\right)|C = 0\right] \\ &\quad + \epsilon\mathbb{E}\left[D_{\text{KL}}\left(P(Y \mid H_M, Z) \,\|\, P(Y \mid H_M, Z; M)\right)|C = 1\right] \\ &\geq \epsilon\mathbb{E}\left[D_{\text{KL}}\left(P(Y \mid H_M, Z) \,\|\, P(Y \mid H_M, Z; M)\right)|C = 1\right].\end{aligned} \tag{33}$$

By Pinsker's inequality,

$$D_{\text{KL}}(P\|Q) \geq 2\ \text{TV}(P, Q)^2, \tag{34}$$

and hence

$$\begin{aligned}\xi_{\text{tool}} &\geq 2\epsilon\mathbb{E}\left[\text{TV}\left(P(Y \mid H_M, Z), P(Y \mid H_M, Z; M)\right)^2 \middle| C = 1\right] \\ &\geq 2\epsilon\left(\mathbb{E}\left[\text{TV}\left(P(Y \mid H_M, Z), P(Y \mid H_M, Z; M)\right)|C = 1\right]\right)^2 \\ &= 2\epsilon\delta^2,\end{aligned} \tag{35}$$

where the second inequality follows from Jensen's inequality and the last equality follows from the definition of δ.

Substituting Eq. 32 and Eq. 35 into Eq. 29, we obtain

$$\begin{aligned}R_{\text{tool}} - R_{\text{base}} &\geq -\Gamma_M + 2\epsilon\delta^2 - \xi_M \\ &= 2\epsilon\delta^2 - \Gamma_M - \xi_M.\end{aligned} \tag{36}$$

Therefore,

$$R_{\text{tool}} - R_{\text{base}} \geq 2\epsilon\delta^2 - \Gamma_M - \xi_M, \tag{37}$$

which completes the proof. □

C Data Collection Details

This appendix supplements the data collection pipeline described in Sec. 4.1 and Sec. 4.2. For QA generation, we provide additional details on video sources, anchor discovery, evidence acquisition, and filtering statistics. For trajectory generation, we describe task-specific trajectory construction, dependency verification, retry procedures, and tool-specific necessity evaluation.

C.1 QA Generation

This subsection provides additional implementation details for video selection, anchor discovery, video-to-Web evidence acquisition, and QA filtering.

Video Selection and Anchor Discovery. The scene-understanding subset covers diverse indoor, outdoor, egocentric, and urban environments. Gemini-3.5-Flash analyzes each video together with its ASR transcript and records textual descriptions of fine-grained, retrieval-relevant visual anchors. At this stage, an exact identity is not required. The recorded anchors include both recognizable entities or events with clear retrieval value and unfamiliar visual targets whose identities can be resolved through subsequent search.

Video-to-Web Evidence Acquisition. For events and actions, we verify the temporal boundaries, participants, and event semantics over the localized segment rather than relying on a single frame. ASR provides complementary context, such as event names or participants, that may not be visually apparent. For entities, we seek fine-grained identities, such as a product's series, model, version, or packaging style, rather than only a coarse category. We crop the target region and use OCR to extract visible text, logos, model numbers, product names, or titles. Image-search candidates are then compared with the video crop to verify visual consistency.

After the target is identified, we retrieve information from sources appropriate to the query. Public entities, entity pairs, and named events are searched for recent information, relations among multiple entities or events, and evidence chains that require multiple retrieval steps. Products, models, books, documents, and devices are matched with fine-grained sources such as official websites, manuals, specification pages, and product pages. We extract traceable facts from these sources and organize them into a source-backed evidence path.

QA Generation and Filtering. The temporal condition in a generated question may refer to an event occurring within a particular interval or an entity appearing at a specific timestamp. Under the two restricted evaluation settings described in Sec. 4.1, only 2 of 1,000 candidate questions can be answered without the video, while 6 can be answered from the video alone without external tool use; these questions are removed.

C.2 Trajectory Generation

This subsection provides additional details on task-specific trajectory construction, execution verification, adaptive retry, and model-conditioned tool necessity filtering.

Trajectory Generation and Execution. For each retained QA pair, we organize the video grounding results, event or entity identification, external evidence, and their intermediate outputs into a tool-use trajectory. Task-specific prompts reflect different information requirements; for example, event-centric questions are instructed not to invoke spatial grounding when fine-grained spatial localization is unnecessary. We then verify that every tool is supported by its prerequisite outputs. In particular, image search over an entity must follow the corresponding grounding operation and use the resulting crop. We also check that tool parameters refer to the intended intermediate results, revise invalid dependencies or parameters, and execute the refined trajectory step by step.

Trajectory Check and Retry. An incorrect image-search result often originates from low-quality grounding, such as a frame showing only a side view of a person or an uninformative crop. We trace such failures back to the responsible temporal, timestamp, or spatial grounding step, select a better frame or region, and repeat image search. When web retrieval does not contain the expected information, we reformulate the query and search again. This process continues until the required evidence is recovered or the maximum number of attempts is reached.

We retain failed attempts, failure diagnoses, and subsequent corrections in the trajectory as natural reflection and recovery steps. Reliable steps remain unchanged, so reflection is introduced only when an intermediate result is unreliable and requires targeted correction.

Model-conditioned Tool Necessity Filtering. Following the criterion defined in Sec. 4.2, we first evaluate the longest candidate tool chain and then progressively examine its individual tools. Consider the chain `Temporal Grounding` → `Timestamp Grounding` → `Spatial Grounding` → `Image Search` for identifying "the person wearing blue." If Qwen3.5-9B can directly identify the person from the original video, the entire chain is removed. Otherwise, we inspect its components. To test temporal grounding, we bypass the localized interval and ask the model to find a usable frame directly from the full video. To test spatial grounding, we perform the downstream image search using the full selected frame instead of the predicted crop. To test web search, we ask the model to answer the targeted knowledge question without external retrieval. We apply the same bypass principle to the remaining grounding and retrieval tools and retain only those needed to recover the expected downstream information.

D Benchmark Details

VDR-EE contains 250 questions spanning seven semantic domains (Fig. 5). We annotate question type, entity multiplicity, grounded-event duration, and full-video duration (Table 5). These dimensions capture different localization demands: entity questions may require identifying multiple visual targets, while event questions may require locating an interval before external retrieval. Fig. 6 shows examples of both types.

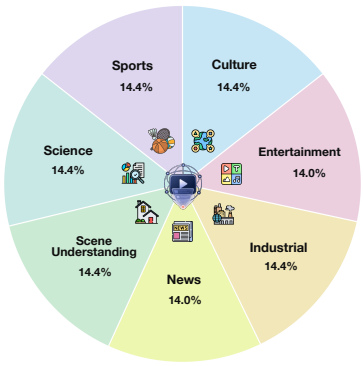

Figure 5: Distribution of the 250 VDR-EE questions across seven semantic domains.

Table 5: Statistics and video-centric annotations of VDR-EE.

Dimension	Category	Definition	Count	Proportion
Question Type (250 instances)				
Question Type	Entity	Entity-oriented questions	153	61.20%
	Event	Event-oriented questions	97	38.80%
Entity Multiplicity (153 entity instances)				
Entity Type	Single	One entity	76	49.67%
	Multi	Multiple entities	77	50.33%
Grounded-Event Duration (97 event instances)				
Event Duration	Short	≤ 30 s	35	36.08%
	Medium	30–120 s	34	35.05%
	Long	> 120 s	28	28.87%
Full-Video Duration (250 instances)				
Video Duration	Short	≤ 5 min	77	30.80%
	Medium	$5 < t < 10$ min	80	32.00%
	Long	≥ 10 min	93	37.20%

E More Experimental Results

We provide additional breakdowns of VDR-EE and VideoDR results by domain, video duration, and question difficulty, followed by qualitative examples of adaptive tool use and reflection.

E.1 Results on VDR-EE

Table 6 reports domain-wise results on VDR-EE. AdaVDR-9B improves the overall accuracy of its Qwen3.5-9B base model from 40.00% to 55.20% and achieves gains across all seven domains. The largest improvements are observed in Entertainment and Sports, with gains of 22.85 and 22.22 percentage points, respectively.

Table 7 groups VDR-EE questions by full-video duration. AdaVDR-9B outperforms Qwen3.5-9B in all three groups, with gains of 33.76, 2.50, and 10.75 percentage points on short, medium, and long videos, respectively. The smaller gain on medium-length videos indicates that the benefit varies with video duration.

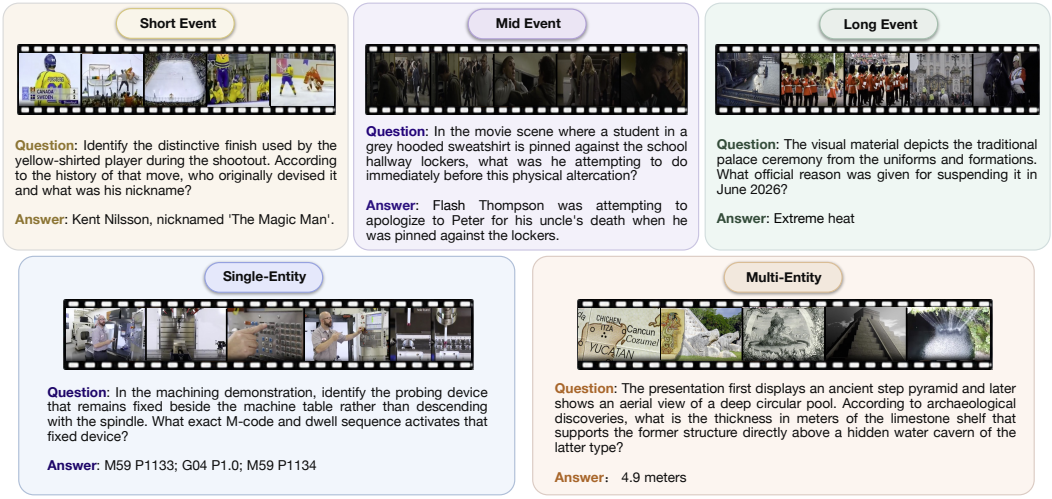

Figure 6: Examples of event-oriented and entity-oriented questions in VDR-EE.

Table 6: Domain-wise accuracy (%) on VDR-EE under the **Direct** and **Agentic** settings.

Model	Domains							Avg.
	Culture	Entertainment	Industrial	News	Scene understanding	Sports	Science	
Direct								
Proprietary								
Gemini-3.1-Pro (Team et al., 2023)	38.89	54.29	27.78	37.14	63.89	50.00	55.56	46.80
Gemini-3-Flash (Team et al., 2023)	44.44	40.00	16.67	22.86	66.67	44.44	47.22	40.40
GPT-5.4 (Singh et al., 2025)	41.67	14.29	16.67	22.86	50.00	36.11	41.67	32.00
Open-source								
Qwen3-VL-8B-Instruct (Bai et al., 2025)	2.78	14.29	11.11	8.57	25.00	8.33	16.67	12.40
Qwen3-VL-32B-Instruct (Bai et al., 2025)	11.11	14.29	13.89	5.71	36.11	11.11	25.00	16.80
Qwen3.5-9B (Team, 2026)	8.33	5.71	16.67	8.57	25.00	8.33	16.67	12.80
Qwen3.5-35B-A3B (Team, 2026)	13.89	5.71	5.56	11.43	38.89	8.33	27.78	16.00
Agentic								
Proprietary								
Gemini-3.1-Pro (Team et al., 2023)	58.33	74.29	50.00	51.43	75.00	66.67	75.00	64.40
Gemini-3-Flash (Team et al., 2023)	72.22	51.43	41.67	48.57	66.67	69.44	66.67	59.60
GPT-5.4 (Singh et al., 2025)	47.22	51.43	38.89	62.86	66.67	44.44	58.33	52.80
Open-source								
Qwen3-VL-8B-Instruct (Bai et al., 2025)	33.33	28.57	25.00	28.57	38.89	8.33	36.11	28.40
Qwen3-VL-32B-Instruct (Bai et al., 2025)	50.00	37.14	27.78	25.71	52.78	11.11	61.11	38.00
Qwen3.5-9B (Team, 2026)	50.00	34.29	30.56	40.00	50.00	30.56	44.44	40.00
Qwen3.5-35B-A3B (Team, 2026)	55.56	34.29	36.11	40.00	55.56	36.11	66.67	46.40
AdaVDR-9B (Qwen3.5-9B)	**63.89**	**57.14**	**36.11**	**42.86**	**69.44**	**52.78**	**63.89**	**55.20**
Δ vs. Qwen3.5-9B	+13.89	+22.85	+5.55	+2.86	+19.44	+22.22	+19.45	+15.20

Table 7: Accuracy (%) across full-video duration groups on VDR-EE under the **Direct** and **Agentic** settings.

Model	Video Duration			Avg.
	Short ≤ 5 min	**Medium** $5 < t < 10$ min	**Long** ≥ 10 min	
Direct				
Proprietary				
Gemini-3.1-Pro (Team et al., 2023)	50.65	52.50	38.71	46.80
Gemini-3-Flash (Team et al., 2023)	41.56	48.75	32.26	40.40
GPT-5.4 (Singh et al., 2025)	31.17	36.25	29.03	32.00
Open-source				
Qwen3-VL-8B-Instruct (Bai et al., 2025)	6.49	15.00	15.05	12.40
Qwen3-VL-32B-Instruct (Bai et al., 2025)	10.39	22.50	17.20	16.80
Qwen3.5-9B (Team, 2026)	11.69	18.75	8.60	12.80
Qwen3.5-35B-A3B (Team, 2026)	11.69	22.50	13.98	16.00
Agentic				
Proprietary				
Gemini-3.1-Pro (Team et al., 2023)	68.83	62.50	62.37	64.40
Gemini-3-Flash (Team et al., 2023)	71.43	63.75	46.24	59.60
GPT-5.4 (Singh et al., 2025)	59.74	50.00	49.46	52.80
Open-source				
Qwen3-VL-8B-Instruct (Bai et al., 2025)	31.17	32.50	22.58	28.40
Qwen3-VL-32B-Instruct (Bai et al., 2025)	35.06	41.25	37.63	38.00
Qwen3.5-9B (Team, 2026)	32.47	51.25	36.56	40.00
Qwen3.5-35B-A3B (Team, 2026)	50.65	46.25	43.01	46.40
AdaVDR-9B (Qwen3.5-9B)	**66.23**	**53.75**	**47.31**	**55.20**
Δ vs. Qwen3.5-9B	+33.76	+2.50	+10.75	+15.20

E.2 Results on VideoDR

Table 8: Domain-wise accuracy (%) on VideoDR under the **Workflow** and **Agentic** settings.

Model	VideoDR Domains						
	History	**Geography**	**Culture**	**Economy**	**Technology**	**Daily Life**	**Avg.**
Workflow							
Proprietary							
Gemini-3-Pro-Preview (Team et al., 2023)	72.73	70.00	80.00	62.50	64.29	69.70	69.00
GPT-4o (Hurst et al., 2024)	63.64	40.00	33.33	43.75	42.86	39.39	42.00
GPT-5.2 (Singh et al., 2025)	72.73	70.00	80.00	56.25	64.29	72.73	69.00
Open-source							
Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025)	36.36	30.00	26.67	43.75	50.00	36.36	37.00
InternVL3.5-14B (Wang et al., 2025)	9.09	50.00	20.00	25.00	21.43	30.30	27.00
MiniCPM-V 4.5 (Yu et al., 2026)	27.27	10.00	46.67	25.00	14.29	24.24	25.00
Qwen3-VL-8B-Instruct (Bai et al., 2025)	36.36	30.00	33.33	31.25	33.33	18.18	28.00
Qwen3-VL-32B-Instruct (Bai et al., 2025)	45.45	40.00	46.67	50.00	26.67	24.24	36.00
Qwen3.5-9B (Team, 2026)	54.55	20.00	26.67	50.00	33.33	30.30	35.00
Qwen3.5-35B-A3B (Team, 2026)	63.64	20.00	26.67	56.25	40.00	33.33	39.00
Agentic							
Proprietary							
Gemini-3-Pro-Preview (Team et al., 2023)	81.82	50.00	86.67	68.75	85.71	78.79	76.00
GPT-4o (Hurst et al., 2024)	63.64	20.00	53.33	50.00	35.71	39.39	43.00
GPT-5.2 (Singh et al., 2025)	90.91	70.00	73.33	56.25	71.43	66.67	69.00
Open-source							
Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025)	54.55	40.00	26.67	43.75	35.71	33.33	37.00
InternVL3.5-14B (Wang et al., 2025)	36.36	40.00	26.67	31.25	28.57	24.24	30.00
MiniCPM-V 4.5 (Yu et al., 2026)	9.09	10.00	26.67	12.50	14.29	18.18	16.00
Qwen3-VL-8B-Instruct (Bai et al., 2025)	36.36	40.00	33.33	31.25	33.33	21.21	30.00
Qwen3-VL-32B-Instruct (Bai et al., 2025)	63.64	40.00	46.67	37.50	40.00	24.24	38.00
Qwen3.5-9B (Team, 2026)	54.55	20.00	40.00	43.75	33.33	33.33	37.00
Qwen3.5-35B-A3B (Team, 2026)	63.64	20.00	33.33	62.50	40.00	36.36	42.00
Qwen3.5-397B-A17B (Team, 2026)	54.55	70.00	46.67	81.25	53.33	48.48	57.00
AdaVDR-9B (Qwen3.5-9B)	**63.64**	**60.00**	**80.00**	**62.50**	**66.67**	**63.64**	**66.00**
Δ vs. Qwen3.5-9B	+9.09	+40.00	+40.00	+18.75	+33.34	+30.31	+29.00

Table 8 breaks down VideoDR accuracy by domain. AdaVDR-9B improves on the agentic Qwen3.5-9B base model in all six domains, raising overall accuracy from 37.00% to 66.00%. The largest gains occur in Geography and Culture (both +40.00 percentage points), while History shows the smallest gain (+9.09 points). Thus, the overall improvement is not driven by a single domain, although its magnitude varies across domains.

Table 9 reports VideoDR results across difficulty levels. AdaVDR-9B reaches 66.00% overall accuracy, outperforming its Qwen3.5-9B base model by 29.00 percentage points. The improvements are

Table 9: Accuracy (%) across VideoDR difficulty levels under the **Agentic** setting.

Model	Agentic			
	Low	Medium	High	Avg.
#Samples	32	36	32	100
Proprietary				
Gemini-3-Pro-Preview (Team et al., 2023)	93.75	69.44	65.62	76.00
GPT-4o (Hurst et al., 2024)	62.50	38.89	28.12	43.00
GPT-5.2 (Singh et al., 2025)	84.38	58.33	65.62	69.00
Open-source				
Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025)	65.62	27.78	18.75	37.00
InternVL3.5-14B (Wang et al., 2025)	46.88	22.22	21.88	30.00
MiniCPM-V 4.5 (Yu et al., 2026)	18.75	19.44	9.38	16.00
Qwen3-VL-8B-Instruct (Bai et al., 2025)	46.88	27.78	15.62	30.00
Qwen3-VL-32B-Instruct (Bai et al., 2025)	46.88	38.89	28.12	38.00
Qwen3.5-9B (Team, 2026)	53.13	33.33	25.00	37.00
Qwen3.5-35B-A3B (Team, 2026)	68.75	41.67	15.62	42.00
Qwen3.5-397B-A17B (Team, 2026)	90.62	58.33	21.88	57.00
AdaVDR-9B (Qwen3.5-9B)	**84.38**	**61.11**	**53.13**	**66.00**
Δ vs. Qwen3.5-9B	+31.25	+27.78	+28.13	+29.00

consistent across the Low-, Medium-, and High-difficulty subsets, with gains of 31.25, 27.78, and 28.13 points, respectively.

E.3 Qualitative Analysis

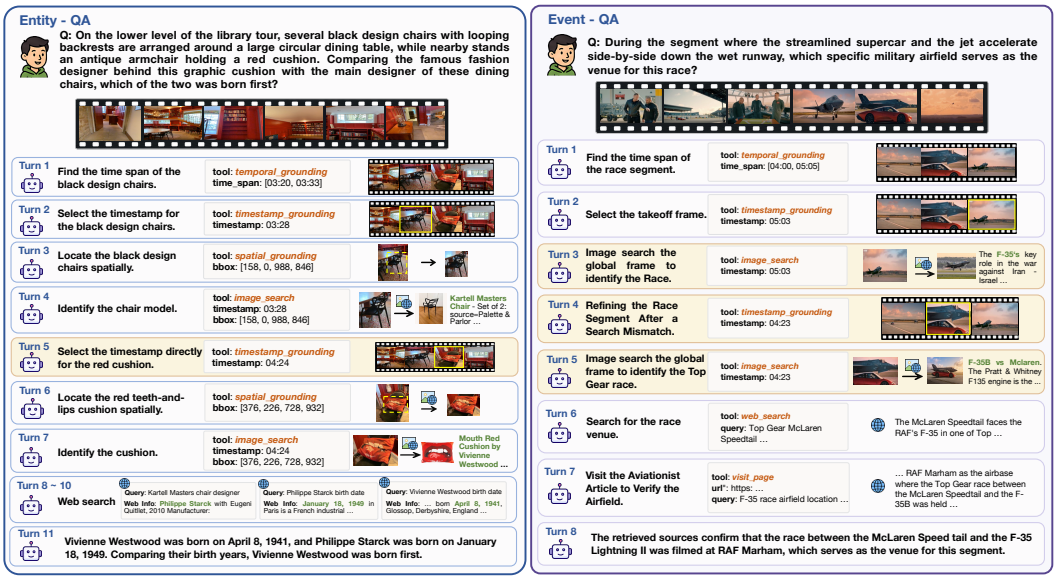

Figure 7: Qualitative examples of AdaVDR on entity-oriented and event-oriented questions.

Fig. 7 illustrates how AdaVDR adapts its reasoning trajectory to the question and the evidence acquired during interaction. In the entity-oriented example, the model uses temporal grounding to locate the chairs. For the red cushion, the relevant timestamp can be directly identified from the current video context, so the model skips temporal grounding and proceeds to spatial grounding. It then identifies both objects before retrieving information about their designers and comparing their birth dates. This trajectory shows that AdaVDR can omit temporal grounding when the relevant timestamp can be identified directly. In the event-oriented example, the initial image search based on a takeoff frame returns mismatched evidence. The model therefore selects a more informative timestamp, identifies the race, and verifies the airfield through web search and page inspection. Together, the two examples demonstrate adaptive tool invocation and reflection-based recovery from unreliable intermediate retrieval results.

We also examine whether AdaVDR generalizes beyond video deep research in two related settings.

E.4 IMAGE DEEP RESEARCH

We evaluate AdaVDR on image-based research tasks with video-specific grounding tools disabled. Table 10 reports results on five image deep-research benchmarks.

Table 10: Accuracy (%) on five image deep-research benchmarks.

Model	SimpleVQA	MMSearch	LiveVQA	BC-VL	FVQA	Avg.
Agentic workflow						
GPT-5	67.3	62.7	–	57.6	62.0	–
Gemini-2.5-Pro	74.3	55.7	–	42.3	65.0	–
Gemini-3.1-Pro	–	86.1	76.6	64.1	84.0	–
Claude-4-Sonnet	–	67.2	–	48.6	69.0	–
Claude-4.6-Opus	–	76.2	67.4	48.3	74.5	–
Kimi-K2.5	–	76.6	76.6	50.3	76.5	–
Multimodal deep-research models						
Visual-ARFT-7B	42.4	34.5	25.4	16.5	41.7	32.1
MMSearch-R1-7B	57.4	53.8	48.4	20.9	58.4	47.8
DeepEyes-V2-7B	59.4	63.7	–	–	60.6	–
WebWatcher-7B	54.3	49.1	51.2	21.2	–	–
Qwen3-VL-8B	52.0	37.4	50.6	27.9	58.7	45.3
SenseSearch-8B	61.7	67.4	56.2	35.1	67.1	57.5
OpenSearch-VL-8B	71.6	64.5	59.6	37.6	71.5	61.0
Qwen3-VL-30B	55.1	44.2	62.0	34.1	63.0	51.7
WebWatcher-32B	59.0	55.3	58.7	27.0	–	–
Qwen3.5-9B (Agentic)	56.8	41.6	55.4	31.7	63.2	49.7
Qwen3.5-27B (Agentic)	64.3	59.1	50.6	39.4	67.3	56.1
AdaVDR-9B	**72.0**	**70.7**	**64.5**	**39.5**	**74.0**	**64.1**

E.5 LONG-VIDEO UNDERSTANDING

We evaluate AdaVDR-9B on three long-video understanding benchmarks, as shown in Table 11.

Table 11: Video question-answering accuracy (%).

Model	VideoMME	MV-Bench	LongVideoBench	Avg.
InternVL3.5-8B	66.0	72.1	62.1	66.7
Keye-VL-1.5-8B	73.0	56.9	66.0	65.3
Eagle2.5-8B	72.4	74.8	66.4	71.2
MiniCPM-V-4.5-8B	67.9	60.5	63.9	64.1
LLaVA-OneVision-2-8B	71.9	66.2	66.9	68.3
Qwen3.5-9B	71.1	68.2	64.6	68.0
Video-RTS	63.0	–	56.6	–
Video-MTR	59.0	–	56.4	–
LOVE-R1	64.3*	–	58.0*	–
Conan	60.5	–	56.6	–
LongVT	64.3	–	–	–
Video-Zoomer	64.6*	–	55.9*	–
Video-o3	66.5	–	60.5	–
AdaVDR-9B	72.8	74.9	68.2	72.0

F MORE TOOL-USE ANALYSIS

F.1 TOOL EFFECTIVENESS

We compare SFT and NASA-GRPO on the same evaluation questions using the visual and text segments defined in Sec. 5. A visual segment is useful when its grounding, with image search if used, provides the target visual information; a timestamp-only segment can therefore be useful. A

text segment is useful when its Web search and associated page visits, or a standalone page visit, recover the target fact. We then inspect whether the model treats the target as established despite an unsuccessful segment, either by answering from prior knowledge without retrieved support or by relying on an incorrect tool result.

We manually inspect trajectories for 100 VideoDR questions. For VDR-EE, a VLM judge applies the same two checks to each segment. For useful segments, we also bypass the segment and ask the evaluated model to recover its target from the preceding context; we count it as necessary if the model cannot do so. Table 12 reports percentages computed from pooled segment counts. Necessary segments are reported as a percentage of useful segments. False acceptance is the share of all segments that fail to obtain useful information but whose target the model nevertheless treats as established.

Table 12: Tool effectiveness and necessity for visual and text segments on 100 manually reviewed VideoDR questions and VLM-judged VDR-EE questions. Useful segments and false acceptance use all segments as the denominator; necessary segments use useful segments as the denominator.

Benchmark	Model	Segment	Useful segments (%)	Necessary among useful (%)	False acceptance (%)
VideoDR	SFT	Visual	58.4	68.9	27.6
		Text	64.4	71.4	19.4
	NASA-GRPO	Visual	76.1	84.6	12.8
		Text	77.8	85.2	11.7
VDR-EE	SFT	Visual	60.7	69.7	23.4
		Text	64.6	68.8	21.6
	NASA-GRPO	Visual	74.1	83.9	13.6
		Text	78.4	85.0	9.4

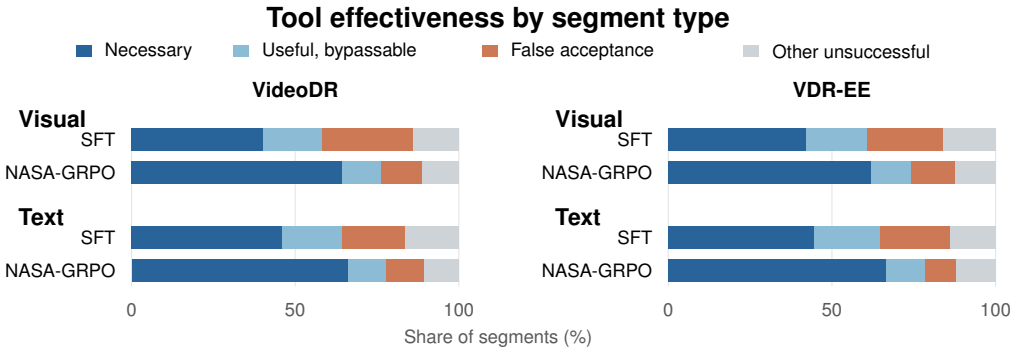

Figure 8: Tool effectiveness by segment type on VideoDR and VDR-EE. Necessary and useful but bypassable segments partition useful segments; false acceptance and other unsuccessful segments partition the remaining segments.

Table 12 and Fig. 8, together with Table 2, support the view that NASA-GRPO improves evidence acquisition rather than merely increasing tool calls, because it has higher useful- and necessary-segment rates and lower false-acceptance rates than SFT for both visual and text segments. Its higher visual usefulness despite fewer grounding calls is also consistent with selecting inputs that better support image search.

Fig. 9 illustrates a false-acceptance failure in one SFT trajectory. Although image search returns the correct church identity, SFT retains an earlier, incorrect identification and gives an unsupported elevation after subsequent searches fail to establish the target fact. The RL trajectory instead follows the retrieved identity and obtains explicit elevation evidence before answering.

F.2 Tool Necessity Analysis

We compare the RL variants after tool-type normalization, with and without the model-conditioned necessity weight (Table 3), on the same VideoDR and VDR-EE questions under identical decoding settings. We identify useful evidence-acquisition segments using the criteria in Sec. F.1 and apply



Figure 9: A VideoDR case illustrating unsupported answering after failed evidence acquisition. SFT retains an incorrect church identity despite the image-search result and answers approximately 2,000 m without supporting elevation evidence. The RL trajectory uses the retrieved church identity and elevation evidence to answer approximately 2,580 m.

the model-conditioned recovery criterion of Eq. 8. A segment is necessary when it is useful and its target information cannot be recovered after bypassing it. We assess tool calls likewise: a call is necessary when it retrieves useful target information that the evaluated model cannot recover after bypassing the call; otherwise, it is redundant.

Table 13 reports necessary segments as a percentage of all segments and necessary and redundant calls as percentages of all calls, with counts pooled across questions. This analysis complements the answer-accuracy ablation in Table 3 by measuring whether the necessity weight shifts tool use toward evidence and calls that provide otherwise unavailable information.

Table 13: Necessity of evidence-acquisition segments and tool calls on VideoDR and VDR-EE. Segment percentages use all segments; call percentages use all tool calls.

Benchmark	Necessity weight	Necessary segments (%)	Necessary calls (%)	Redundant calls (%)
VideoDR	Without	54.9	54.6	45.4
	With	65.9	76.2	23.8
VDR-EE	Without	50.8	53.2	46.8
	With	65.9	74.5	25.5



Figure 10: Necessary and redundant tool calls on VideoDR and VDR-EE, with and without the model-conditioned necessity weight. Shares are computed over all tool calls.

Table 13 shows that necessity-aware training raises the share of necessary segments from 54.9% to 65.9% on VideoDR and from 50.8% to 65.9% on VDR-EE. The necessary tool-call share rises from 54.6% to 76.2% and from 53.2% to 74.5%, respectively, as illustrated in Fig. 10. Despite the corresponding reduction in redundant calls, Table 3 shows that accuracy rises from 59.00% to 66.00% on VideoDR and from 52.00% to 55.20% on VDR-EE. This pattern indicates that the weight improves tool selection rather than simply reducing tool use.

G Training Dynamics

We examine the evolution of process credits and the effect of the model-conditioned necessity weight on tool use.

G.1 Process Credit During Training

Table 14: Tool-level process credits during RL training with the model-conditioned necessity weight. Positive-credit calls receive 0.5 or 1.0; no-credit calls receive 0.

Rollouts	All tool calls	Credit 1.0 calls	Credit 0.5 calls	Positive-credit calls	Positive-credit share (%)	No-credit share (%)
0–9	3775	589	213	802	21.2	78.8
10–19	3878	652	374	1026	26.5	73.5
20–29	4947	878	362	1240	25.1	74.9
30–39	5059	922	616	1538	30.4	69.6
40–49	4753	923	582	1505	31.7	68.3

Table 14 and Fig. 11 show that the share of calls receiving positive process credit rises overall from 21.2% to 31.7%, while the share receiving no credit falls from 78.8% to 68.3%, despite a small

reversal in the middle of training. Both full- and half-credit calls increase in share, with full-credit calls remaining more numerous throughout. Together, these trends suggest that the model becomes more effective at acquiring useful external evidence as training progresses.

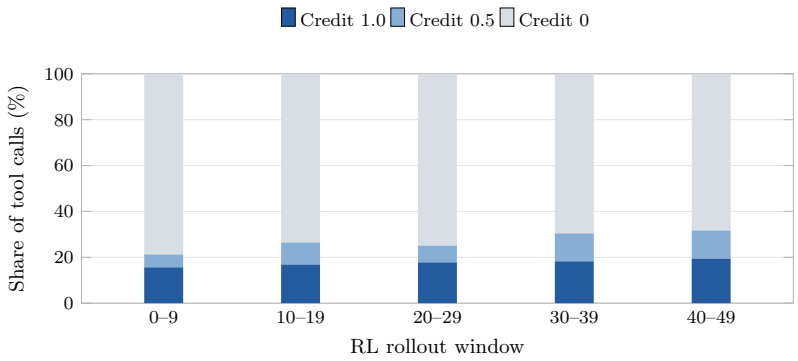

Figure 11: Process-credit composition across RL rollout windows. Each bar shows the share of calls assigned credit 0, 0.5, or 1.0, using the counts in Table 14.

G.2 Effect of Model-Conditioned Necessity Weight on Tool Use

Table 15: Tool-call distribution in a training snapshot with and without the model-conditioned necessity weight.

Tool type	**Without necessity weight**		**With necessity weight**		**Δ Calls**
	Calls	**Share (%)**	**Calls**	**Share (%)**	
temporal grounding	62	10.42	65	11.44	+3
timestamp grounding	60	10.08	65	11.44	+5
spatial grounding	6	1.01	4	0.70	-2
image search	54	9.08	70	12.32	+16
web search	344	57.82	200	35.21	-144
page visit	69	11.60	164	28.87	+95
Total	**595**	**100**	**568**	**100**	**-27**

In this training snapshot, the model without the model-conditioned necessity weight makes more tool calls than the weighted model (595 vs. 568; Table 15). Fig. 12 highlights the shift in tool use: web search falls from 57.82% to 35.21% of calls, while page visits rise from 11.60% to 28.87% and image search rises from 9.08% to 12.32%. This shift toward visual retrieval and page inspection complements the accuracy gains in Table 3.

H More Ablations

We examine the two reward coefficients separately and assess the contribution of each available tool.

H.1 Necessity Weight Ablation

For bypassable actions, we vary the necessity weight while keeping the weight of necessary actions at 1. The default bypassable-action weight is 0.5 (Eq. 8); all other training settings remain fixed. The comparisons suggest a trade-off: smaller weights weaken the process signal even for bypassable calls that return correct evidence, whereas larger weights narrow the reward gap between necessary and bypassable calls and may encourage redundant tool use.

H.2 Process-Advantage Coefficient Ablation

We vary the coefficient λ that scales the process advantage in Eq. 12, keeping the necessity weights fixed at their default values. The default is $\lambda = 0.3$.

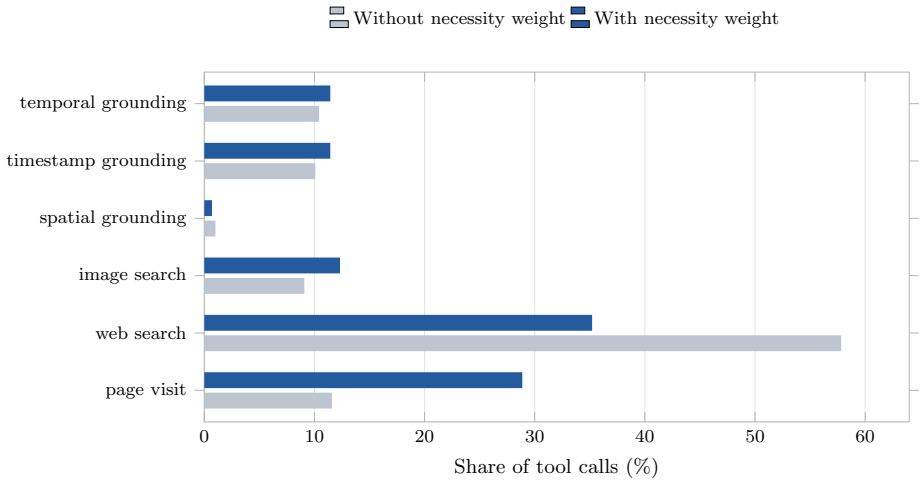

Figure 12: Tool-use distribution in a training snapshot with and without the model-conditioned necessity weight. Bars show each tool's share of all calls.

Table 16: Ablation of the necessity weight assigned to bypassable tool calls. The default is 0.5.

Bypassable-call weight	**VideoDR**		**VDR-EE**	
	Acc. (%)	**Avg. calls**	**Acc. (%)**	**Avg. calls**
0.2	61.00	9.10	50.80	10.50
0.3	63.00	9.55	52.40	11.05
0.4	64.00	10.10	54.00	11.65
0.5 (default)	66.00	10.67	55.20	12.26
0.6	62.00	11.75	52.00	13.40

Table 17: Ablation of the process-advantage coefficient λ. The default is 0.3.

λ	**VideoDR**		**VDR-EE**	
	Acc. (%)	**Avg. calls**	**Acc. (%)**	**Avg. calls**
0.2	62.00	8.40	51.60	10.00
0.3 (default)	66.00	10.67	55.20	12.26
0.4	64.00	12.00	53.60	13.60
0.5	61.00	13.40	51.20	15.00

The values peak at $\lambda = 0.3$. A smaller coefficient may underweight evidence-acquisition actions, reducing useful retrieval. A larger coefficient may overemphasize process rewards, leading to more tool calls without better answers.

H.3 Tool Ablation

We keep the trained AdaVDR model fixed and disable one tool at a time during inference, leaving the other tools available. Table 18 compares each setting with the full tool set; Δ denotes the accuracy change in percentage points.

Table 18: Inference-time tool ablation. Each row independently removes only the named tool from the full model; all other tools remain available. Full-model accuracies match Table 1.

Configuration	**VideoDR**		**VDR-EE**	
	Acc. (%)	Δ	**Acc. (%)**	Δ
Full model (all tools)	66.00	–	55.20	–
− temporal grounding	62.00	−4.00	52.80	−2.40
− timestamp grounding	61.00	−5.00	52.00	−3.20
− spatial grounding	65.00	−1.00	54.80	−0.40
− image search	58.00	−8.00	48.80	−6.40
− web search	53.00	−13.00	44.00	−11.20
− page visit	59.00	−7.00	49.60	−5.60

Removing web search or image search produces the largest declines, consistent with their roles in acquiring external facts and identifying visual entities. Removing spatial grounding has a smaller effect, consistent with its low invocation frequency in Table 2.

I Human–Credit Agreement

Three annotators independently inspect 50 training tool calls of each type, including the question, relevant video and trajectory context, tool output, target clue, and bypass response. Each annotator assigns every call a credit of 0, 0.5, or 1, and the majority label is used as the human consensus. Table 19 reports the agreement between this consensus and the assigned pre-normalization credits.

Table 19: Agreement between human consensus and training-time tool credits by tool type.

Tool type	**Agreement (%)**
temporal grounding	96.0
timestamp grounding	94.0
spatial grounding	92.0
image search	98.0
web search	98.0
page visit	96.0

J PROMPTS

J.1 LLM JUDGE PROMPT

We use the following prompt to determine whether a model prediction is semantically consistent with the reference answer. The judge is instructed to focus on factual correctness rather than exact lexical matching and to return a binary decision.

Prompt for LLM-Based Answer Evaluation

You are an answer evaluator. Based on the question and the standard answer, determine whether the model answer is correct.

Allow semantically equivalent wording, abbreviations, formatting differences, and reasonable differences in numerical precision. Mark the answer as incorrect if it contains factual errors, omits essential information, or fails to answer the question.

Question: {question}
Standard Answer: {standard_answer}
Model Answer: {model_answer}

Output only the following JSON:

```
{
  "is_correct":  true/false,
  "reasoning":  "A brief explanation of the judgment"
}
```

Evaluation guidelines:

- Focus on semantic correctness rather than exact wording.
- Accept synonyms, abbreviations, alternative names, and reasonable formatting differences.
- For numerical answers, allow equivalent units, representations, and reasonable precision differences.
- The model answer must contain the essential information required by the question.
- Additional information is acceptable unless it contradicts the correct answer.
- Mark the answer as incorrect if it contains a factual error, refers to a different entity or concept, omits critical information, or does not answer the question.
- If the standard answer itself contains an explanation, compare the core conclusion and required facts rather than requiring identical detail.

Return only valid JSON in the following format:

```
{
  "is_correct":  true/false,
  "reasoning":  "A concise explanation of why the answer is correct
or incorrect."
}
```

J.2 Visual Anchor Extraction Prompt

The following prompt is used to extract video-grounded entities and events that can serve as visual anchors for subsequent question construction and external retrieval. The ASR transcript is used only to name or disambiguate targets that are visibly grounded in the video.

Prompt for Visual Anchor Extraction

Watch the video carefully and use the ASR transcript only to help name or disambiguate visible targets.

ASR transcript: {asr_text}

Extract video-grounded entities and events. Ignore loose ASR-only mentions and ASR/VSR timestamps.

Rules:

- Every item must include at least one concrete video locator based on visual attributes, readable text, spatial relations, or temporal evidence.
- Extract high-value visible targets that can support downstream multi-hop QA with external source facts and can be located again for OCR, image search, Web retrieval, temporal grounding, spatial reasoning, or entity-oriented QA construction.
- Scan the entire video for useful anchors, including products, packages, books, posters, signs, screen items, devices, furniture, clothing designs, artworks, places, and named public entities.
- Prefer targets with readable text, a logo, model or title information, distinctive design, public-facing identity, clear spatial relations, temporal interactions, or state changes.
- For people, retain only named or public-facing individuals, people identified by readable on-screen information, or participants required to localize a useful event.
- Set `identity_status` to `known` only when the exact identity is supported by ASR or unambiguous on-screen text. Use `partially_resolved` when only a brand, line, coarse name, or title fragment is available, and `unknown` when only the generic visual category is known.
- Do not infer an exact subtype, model, variant, title, or identity solely from visual familiarity. Leave such identification to subsequent OCR or image search.
- Drop targets that cannot be localized, cannot be cropped meaningfully, are too weakly visible, or constitute irrelevant background clutter. Avoid near-duplicates and retain at most six unknown or partially resolved entities.
- Extract representative visible events, including actions, procedures, state changes, interactions, maneuvers, outcomes, and causal or trigger-like moments. Event descriptions must be based on visible evidence rather than dialogue or external plot knowledge.
- If the exact name of an action or event is uncertain, leave the name empty and mark it as requiring expert identification.

Return valid JSON only:

```
{
  "visually_searchable_entities": [{
    "entity": "...", "category": "...",
    "identity_status": "known | partially_resolved | unknown",
    "exact_name": "...", "identity_need": "...",
    "visual_locator": "...", "temporal_locator": "...",
    "spatial_locator": "...", "asr_evidence": "..."
  }],
  "video_events": [{
    "event": "...", "event_status": "recognized |
needs_expert_name",
    "visual_locator": "...", "observed_event": "...",
    "temporal_locator": "...", "fine_grained_actions": [...],
    "asr_evidence": "..."
  }]
}
```

J.3 QA Generation Prompt

Given the extracted visual anchors and source-backed paths, the following prompt first constructs questions and reference answers that require both video evidence and external knowledge. We omit the in-prompt examples for brevity while retaining the task definition, principal constraints, and output schema.

Prompt for QA Generation

Read the source-backed video path payload carefully and create exactly {num_qa} natural, high-quality multi-hop QA records whose answers require combining visual evidence from the video with the provided external source facts.

Input payload: {source_payload}

Use `source_facts_for_answer` as the external information supporting the final answer. The final answer must come from an explicit fact item or be computed directly from explicit facts. Prefer rare facts over common biographical or brand facts.

Question requirements:

- Each question must be a single fluent natural-language question and must require both video evidence and external source facts.
- Use indirect visual or temporal descriptions instead of explicitly naming the target entities or events. Do not reveal the final answer or copy the decisive source fact into the question.
- Do not mention tools or operations such as timestamps, bounding boxes, grounding, frames, or Web search.
- Do not rely on dialogue, speech, captions, subtitles, transcripts, commentary, music, or lyrics.
- Event questions must depend on an action, state change, event order, interaction, procedure, outcome, or causal relation unfolding over time.
- Entity questions must require identifying one or more visual targets before retrieving external information related to them.

Return valid JSON only:

```
{
  "qa_records": [{
    "qa_kind": "event | entity",
    "source_path_ids": ["..."],
    "question": "...", "final_answer": "...",
    "answer_sources": [...]
  }]
}
```

J.4 Initial Trajectory Generation Prompt

The generated QA records are then paired with initial executable research trajectories. The following prompt determines the grounding, retrieval, and reasoning operations needed to obtain the video and external evidence supporting each answer.

Prompt for Initial Trajectory Generation

Generate an executable reasoning trajectory for each QA record using the source-backed video path payload.

Input QA records: {qa_records}
Input payload: {source_payload}

Trajectory requirements:

- Use only `temporal_grounding`, `spatio_grounding`, `image_search`, `web_search`, and `reasoning`.
- A visible entity identity must be obtained through `spatio_grounding` followed by `image_search`. OCR is not represented as a standalone tool action; when a readable clue is required, first localize the relevant region with `spatio_grounding` and record the recognized text in the subsequent `reasoning` step.
- Entity records must include the visual grounding chains required to identify their target entities.
- Each Web query may use only names or terms already present in the question or produced by earlier steps. It must not contain the answer that the same step is intended to discover.
- Use `reasoning` to combine, compare, select, count, or calculate from previously obtained evidence, and to record OCR evidence from a preceding spatially grounded region. Do not use it to introduce other new video observations or external facts.
- Every step must depend on the question or earlier trajectory outputs. Avoid redundant tools and repeated grounding operations.

Return valid JSON only:

```
{
  "qa_records": [{
    "qa_kind": "event | entity",
    "source_path_ids": ["..."],
    "question": "...", "final_answer": "...",
    "reasoning_trajectory": [{
      "step": 1, "tool": "temporal_grounding", "temporal_target":
"..."
    }],
    "answer_sources": [...]
  }]
}
```

J.5 TRAJECTORY REFINEMENT PROMPT

Before executing an initial trajectory, we refine its tool dependencies and evidence flow using the following prompt. This stage preserves the question and answer while ensuring that every operation is supported by information available at that point in the trajectory.

Prompt for Initial Trajectory Refinement

You are a meticulous grounded QA dataset editor. Watch the video frames and refine the existing QA records before downstream temporal and spatial grounding.

Input QA records: {qa_records}

Rules:

- Preserve `qa_kind`, `evidence_intent`, question, and final answer unless the record is internally inconsistent. Rewrite only the reasoning trajectory so that it is complete, ordered, and traceable.
- Every concrete fact used in a query or reasoning step must come from the question or an earlier trajectory step. Do not introduce hidden names, years, places, events, products, or people in Web queries.
- Use only `temporal_grounding`, `spatio_grounding`, `image_search`, `web_search`, and `reasoning`.
- If a later step uses a visible entity identity, first apply `spatio_grounding` followed by `image_search`. OCR is not retained as a standalone tool action; readable text from a spatially grounded region must instead be incorporated into the subsequent `reasoning` step.
- When different visual targets appear at different moments, temporally localize each moment before spatial grounding. The visual query for image search must match the preceding spatial target.
- Each Web query may use only names or terms already present in the question or earlier outputs. If additional disambiguating context is required, introduce an earlier retrieval step that explicitly obtains it.
- Use reasoning to combine, compare, select, or calculate from evidence produced by earlier steps, and to record OCR evidence from a preceding spatially grounded region. Add Web search before any new external factual claim.
- Omit unnecessary tools, keep search queries concise, and renumber all steps from 1 in logical order.

Return valid JSON only:

```
{
  "qa_records": [{
    "qa_kind": "event | entity",
    "question": "...", "evidence_intent": "...",
    "reasoning_trajectory": [...],
    "final_answer": "...", "why_long_horizon": "..."
  }]
}
```

J.6 Model-Conditioned Tool Necessity Filtering Prompts

Tool necessity is evaluated against the capabilities of the base model. For a tool or consecutive tool chain, we withhold its output and ask the model to recover the downstream target directly from the information available before execution. The tool is removed only when the direct prediction is semantically consistent with the expected result.

Prompts for Model-Conditioned Tool Necessity Filtering

Temporal or timestamp grounding.
Watch the full video and find one clear timestamp where the following visual target is visible: {visual_target}. Return only JSON containing a timestamp in `MM:SS.ff` format and a brief description of the visible evidence. If the target is not visible, return an empty timestamp.

Spatial grounding.
Look at the provided frame and locate the following visual target: {visual_target}. Return only JSON containing a tight bounding box `[x_min, y_min, x_max, y_max]` and a brief description of the visible evidence. If the target is not visible, return an empty bounding box.

Image search.
Look at the provided image crop and answer what visible entity it shows. Return only JSON in the form {`"answer": "concise entity name"`}.

Web search.
Answer the following query using the model's internal knowledge without invoking Web search: {query}. Return only JSON in the form {`"answer": "concise answer"`}.

Consistency decision.
Given `model_answer` and `expected_answer`, determine whether they are semantically consistent. Return only JSON:
{`"consistent": true/false, "reason": "short reason"`}.

J.7 Reflection and Retry Prompts

When execution produces insufficient or unreliable evidence, the pipeline invokes a targeted retry according to the failed operation. The following prompts show the two principal recovery behaviors: revising video localization and rewriting an external search query.

Prompt for Grounding Retry

The previous grounding attempt did not provide a crop that clearly contains the required visual target.

Visual target: {visual_target}
Previous attempt: {previous_attempt}
Failure reason: {issue}

Re-examine the available video evidence and determine how the target should be localized again. If the target appears at a different moment, propose a revised temporal description for locating a clearer timestamp. If the timestamp is appropriate but the crop is inaccurate, propose a tighter and more precise spatial description. Do not introduce an entity identity that has not been established by visible evidence.

Return only valid JSON describing the retry action, the revised grounding target, and a brief reason.

Prompt for Web Query Rewrite

Rewrite the Web search query so the next search is more likely to return the expected information.

Original query: {original_query}
Expected information: {expected_answer}
Available context: {available_context}
Previous attempts: {previous_attempts}

Rules:

- Keep the rewritten query concise.
- Include the key entity or topic and the missing attribute to verify.
- Do not include the expected answer itself in the rewritten query.
- Use only information from the available context, original query, or previous attempts. Do not add facts that this search step is intended to discover.

Return only valid JSON:
`{"query": "concise rewritten query", "reason": "brief reason for the rewrite"}`.