

AdaVDR: Adaptive Tool Use and Reflection for Video Deep Research

Xintong Zhang^{1,2,*}, Xiaomeng Fan^{2,*}, Shilin Yan¹, Ekko He¹, Zicheng Liu¹,
Zijian Zou¹, Guannan Zhang¹, Yuwei Wu², Zhi Gao^{2,†}, Hongwei Xue^{1,†}

¹Accio Team, Alibaba Group

²Beijing Key Laboratory of Intelligent Information Technology,
School of Computer Science & Technology, Beijing Institute of Technology

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 15.20 and 29.00 percentage points on VDR-EE and the existing VideoDR, respectively, demonstrating its superiority.

 **Project Page:** <https://Accio-Lab.github.io/AdaVDR>

 **GitHub Repo:** <https://github.com/Accio-Lab/AdaVDR>

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

*Equal contribution. †Corresponding authors. This work was done during Xintong Zhang’s internship at the Accio Team, Alibaba Group.



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

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 intermediate 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 15.20 and 29.00 percentage points 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 improve long-video reasoning by iteratively locating question-relevant evidence within videos. Representative systems such as VideoMind [Liu et al., 2025b], 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 video exploration. Recent approaches 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 and Wang, 2025], or dynamic temporal grounding guided by intermediate sub-questions [Yuan et al., 2025]. Their evidence acquisition primarily remains within the video, whereas Video Deep Research additionally requires external information associated with grounded entities or events. Our work studies how video grounding and external retrieval can be adaptively coordinated according to task requirements and model capabilities, while allowing the agent to revise unreliable intermediate results during multi-turn reasoning.

2.2 Multimodal Deep Research

Deep-search agents iteratively reason and invoke external tools to acquire information [Yao et al., 2022, Jin et al., 2025, Li et al., 2025]. Multimodal extensions incorporate image search, visual grounding, cropping, and other visual operations, and learn tool-use policies through supervised fine-tuning or reinforcement learning [Wu et al., 2025, Hong et al., 2026, Huang et al., 2026, Chen et al., 2026, Zhang et al., 2026c]. Related work also explores visual tool coordination [Gao et al., 2024, Zhang et al., 2025, He et al., 2025], adaptive tool use and mode selection [Yan et al., 2026, Zhang et al., 2026b], and longer multimodal search trajectories with richer visual interactions [Li et al., 2026b, Chu et al., 2026]. These methods primarily operate on static input images or visual evidence retrieved from the Web. Video Deep Research introduces an additional temporal dimension: relevant evidence may concern an entity appearing at a particular moment or an event unfolding over an extended segment, requiring video grounding to be coordinated with external retrieval.

Recent studies formulate and evaluate video deep research [Liu et al., 2026a] and develop agents that combine video perception with search. VideoSearcher integrates video grounding with multimodal

search, VideoRover coordinates video cropping, search, and Web browsing, and Video-DeepResearch adopts a staged perception–exploration pipeline [Gao et al., 2026, Liu et al., 2026b, Fang et al., 2026]. Beyond combining video and search tools, an effective agent must determine which grounding and retrieval operations are needed for each question and whether the model can obtain the required information without invoking them. AdaVDR addresses these requirements through adaptive tool use and reflection. Its data construction adapts trajectories to task requirements and model capabilities, while NASA-GRPO combines segment-guided evidence credit with model-conditioned tool necessity to discourage redundant interactions and reduce 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 raw video V and a user question q , the task of *video deep research* aims to generate an accurate textual answer a by iteratively reasoning and leveraging a suite of external tools $\mathcal{T} = \{T_1, T_2, \dots, T_K\}$. The toolset encompasses three primary categories: *video grounding* (including temporal grounding, timestamp grounding, and spatial grounding), *external retrieval* (including image search, web search, and page visits), and *termination* for final answer generation. Formally, at each interaction turn t , the agent observes the interaction history $h_t = (q, v, a_1, o_1, \dots, a_{t-1}, o_{t-1})$ and generates an assistant output $a_t = \pi(h_t)$, where π is typically a VLM acting as the agent brain. At each non-terminal turn, a_t jointly contains the agent’s reasoning and a selected tool action from $\mathcal{T}$ with its corresponding arguments, while o_t denotes the observation returned by executing that action. Once sufficient evidence has been acquired, the final assistant output instead contains the agent’s reasoning and textual answer, without an additional tool observation.

Compared with *image deep research*, video deep research involves more diverse question types, video content, and tool-use requirements, leading to substantially different information needs and reasoning trajectories across tasks. Two representative question types are:

- **Temporal Event.** Event-centric questions require the agent to identify an action or event and localize the corresponding temporal interval. They often rely on temporal or timestamp grounding to determine when the relevant evidence appears, while spatial grounding is typically unnecessary. Some event-related questions can further proceed to text-based web search once the video content has been sufficiently understood.
- **Entity.** Entity-centric questions require the agent to identify or ground an object, person, product, or other entity at a particular timestamp or across multiple frames. Such questions often require locating a relevant frame, grounding the target region, and further using OCR, image search, or web search to resolve the entity’s fine-grained identity and related information.

These differences make the choice of tool-use strategy critical. *An inappropriate reasoning trajectory may fail to acquire the information required by the task or introduce irrelevant operations, ultimately leading to incorrect reasoning results.*

The outputs of video grounding and external retrieval tools are inherently uncertain. Temporal or timestamp grounding may locate an inaccurate segment or frame, spatial grounding may focus on an uninformative region, and image or web search may return irrelevant results. As a result, when the required information can already be obtained from the model’s video understanding capability or internal knowledge, executing additional tool calls unnecessarily exposes the reasoning process to these uncertainties. *Such redundant interactions not only increase inference latency, but may also introduce noisy or incorrect evidence and degrade the final reasoning result.*

To address these challenges, we propose an **adaptive reasoning** paradigm with *adaptive tool invocation* and *adaptive reflection*. For tool invocation, the agent first selects the most appropriate grounding and retrieval tools according to the question type and video content, allowing different tasks to follow different reasoning trajectories. It further adapts tool use to its own capability boundary and internal

knowledge, skipping grounding or retrieval steps when the required information can already be obtained directly. Reflection is also performed adaptively: reliable intermediate results are directly used for subsequent reasoning, whereas unreliable or insufficient results trigger the agent to identify the problematic grounding or retrieval step and backtrack for correction. In this way, the agent avoids both inappropriate and unnecessary tool interactions while reducing the propagation of unreliable information.

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

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

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

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

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

To enable the adaptive behaviors described above, we construct training data that reflects task-specific tool requirements and model-specific tool necessity. As illustrated in Fig. 2, our pipeline consists of two stages: *QA generation* and *trajectory generation*, where we construct questions requiring both video and external knowledge, and build verified trajectories with unnecessary tool interactions removed.

4.1 QA Generation

We first perform *video selection and anchor discovery* to identify retrieval-relevant events and entities from diverse videos. We then conduct *video-to-Web evidence acquisition*, progressively grounding the target content and retrieving related external knowledge. Finally, through *QA generation and filtering*, we construct questions that jointly require video information and external evidence, and remove those that can be answered without either source.

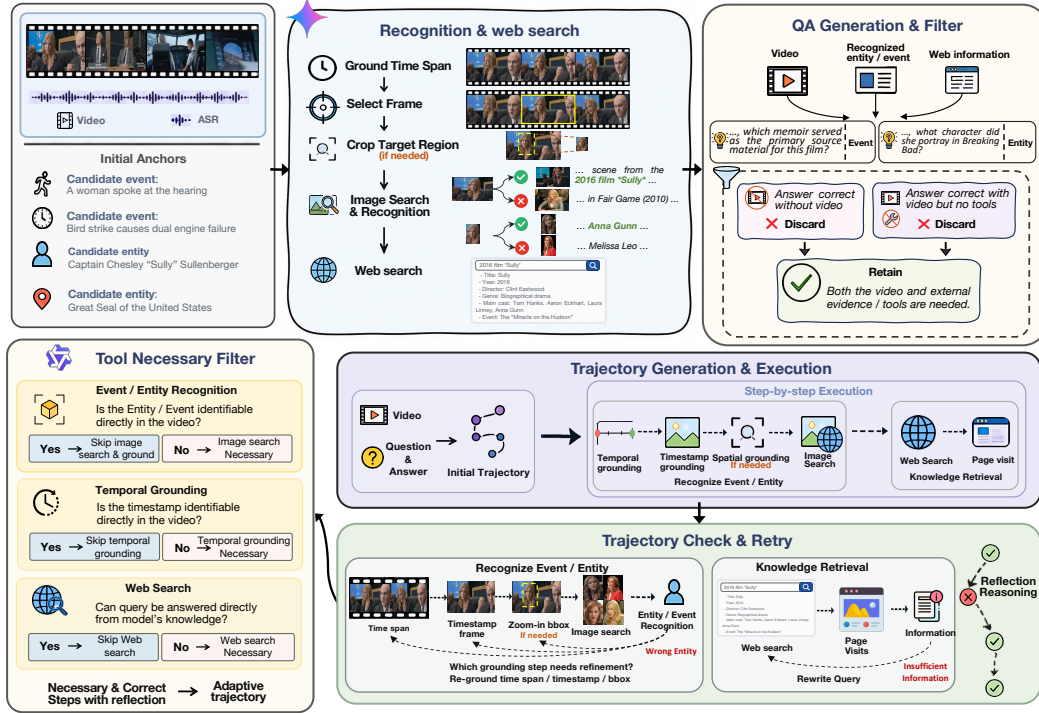


Figure 2: Overview of the data construction pipeline, including QA and trajectory generation, trajectory refinement, and tool-necessary filtering.

Video Selection and Anchor Discovery. We primarily collect videos from YouTube for the news, industry, culture, entertainment, sports, and science domains. The scene-understanding subset is instead primarily sourced from EgoExo4D [Grauman et al., 2024], EgoLife [Yang et al., 2025a], CityWalker [Liu et al., 2025a], ARKitScenes [Baruch et al., 2021], ScanNet++ [Yeshwanth et al., 2023], and Sekai [Li et al., 2026c], covering diverse indoor, outdoor, egocentric, and urban environments. The resulting collection covers seven semantic domains: news, industry, culture, entertainment, sports, science, and scene understanding.

Gemini-3.5-Flash analyzes each video with its ASR transcript to identify fine-grained and retrieval-relevant visual anchors. At this stage, the model does not need to determine the exact identity of each entity or event. Instead, it records textual descriptions of potentially useful visual content for subsequent retrieval and question construction. These descriptions include both recognizable entities or events with clear retrieval value and unfamiliar visual targets whose identities can be further resolved through search.

Video-to-Web Evidence Acquisition. To collect fine-grained external knowledge, we first extract sufficient information from the video to identify the relevant event or entity, and then use the extracted information to conduct further image retrieval or web retrieval. This process consists of two successive stages: video-based information extraction and external evidence retrieval.

For *video-based information extraction*, we progressively identify the target through multi-step grounding and retrieval. For events or actions, we perform temporal grounding to locate the relevant segment and timestamp grounding to select informative frames. The selected frames are then used for image search to retrieve visually related results, which are combined with complementary textual information from ASR. For entities, we perform spatial grounding to isolate the target region and use the resulting crop for image search to obtain possible identities. OCR and ASR further provide visible or spoken information. We combine these results until sufficiently specific information about the event or entity is obtained for subsequent retrieval.

Depending on the identified event or entity, we then perform *external evidence retrieval* by using web search or image search to retrieve relevant evidence from authoritative or task-specific sources, and organize the retrieved facts into a traceable evidence path.

QA Generation and Filtering. Based on the verified video information and source-backed external evidence, we construct question–answer pairs that require reasoning over both sources. Each question combines video-grounded information with a temporal or timestamp-related condition, such as an event occurring within a particular interval or an entity appearing at a specific moment, while the answer is derived from the corresponding external evidence.

To ensure that the resulting questions genuinely depend on both the video and external information, we further filter the generated QAs using Qwen3-VL-235B-A22B-Instruct under two restricted settings: (1) answering without access to the video, and (2) answering with the video but without using external tools. We remove a question if it can be answered correctly under either setting.

4.2 Trajectory Generation

For each retained QA pair, we first perform *trajectory generation and execution*, constructing task-specific tool-use trajectories, refining their logical dependencies and parameters, and executing them step by step. We then conduct *trajectory check and retry* to verify intermediate results, retry unreliable steps when necessary, and preserve the resulting correction process as adaptive reflection data. Finally, through *model-conditioned tool necessity filtering*, we further remove tool calls or tool chains that are unnecessary for the target model based on its own video understanding capability and internal knowledge.

Trajectory Generation and Execution. For each retained QA pair, we first organize the information acquired in the previous stage, including video grounding results, event or entity identification, and the corresponding external evidence. Different questions follow different information acquisition processes through task-specific prompts; for example, event-centric questions are explicitly instructed not to invoke spatial grounding. Based on these processes, we further organize the grounding and retrieval operations, together with their intermediate results, into corresponding tool-use trajectories that are aligned with the specific information requirements of each question.

We refine each initial trajectory according to the question type, removing tool calls that are irrelevant to its information requirements. For example, spatial grounding is removed for event-centric questions when no fine-grained spatial localization is needed. We then check the logical dependencies and tool parameters of the remaining trajectory, ensuring that each tool is supported by the outputs of its prerequisite steps. For instance, image search over a target entity should be preceded by the corresponding grounding operation, with its input taken from the resulting crop. We further verify that each tool call correctly refers to the intended intermediate results and revise invalid dependencies or parameters accordingly. We then invoke the tools to execute the refined trajectory step by step according to the specified tool calls and parameters.

Trajectory Check and Retry. We then execute the refined trajectory and verify whether each tool produces the expected information. For image search, an incorrect result often originates from low-quality grounding, such as selecting a frame where a person is shown only from the side or cropping an uninformative region. In such cases, we trace back to the corresponding grounding step, select a better frame or region, and repeat the image search until the expected target appears in the retrieved results or the maximum number of attempts is reached. For web search, when the retrieved results do not contain the expected information, we reformulate the search query and repeat the retrieval. This iterative execution and verification process continues until the trajectory can reliably recover the expected evidence or the maximum number of attempts is reached.

We retain these failed attempts, failure diagnoses, and subsequent corrections in the trajectory, producing natural reflection and recovery steps that teach the model to revise unreliable intermediate results. By introducing reflection only when unreliable intermediate results occur, while leaving reliable steps unchanged, the constructed trajectories teach the agent to trigger reflection on demand and perform targeted correction rather than reflecting at every reasoning step.

Model-conditioned Tool Necessity Filtering. We further evaluate the necessity of tool calls in each successfully executed trajectory. Given a tool or a consecutive tool chain $\mathcal{S}$, we denote the information already known to the agent before executing $\mathcal{S}$ as $x_{\mathcal{S}}$, and the target result obtained after executing $\mathcal{S}$ as $y_{\mathcal{S}}$. Based on the parameters of $\mathcal{S}$, such as the grounding description or retrieval target, we construct a verification query $q_{\mathcal{S}}$ that asks for the target result $y_{\mathcal{S}}$. We use the base model that will initialize the agent for subsequent training, denoted by $\mathcal{M}_{\phi}$, to perform necessity filtering. In our experiments, this model is Qwen3.5-9B, the base model of AdaVDR-9B. Tool necessity is

therefore evaluated against the video understanding capability and internal knowledge of the model to be trained. The base model then directly obtains the target result from x_S without executing $\mathcal{S}$:

$$\hat{y}_S = \mathcal{M}_\phi(x_S, q_S). \quad (3)$$

The necessity of $\mathcal{S}$ is defined as

$$N(\mathcal{S}) = \mathbf{1}[\hat{y}_S \neq y_S]. \quad (4)$$

If $\hat{y}_S$ matches y_S , the target result can be obtained without executing $\mathcal{S}$, indicating that the tool chain is unnecessary and can be removed. Otherwise, $\mathcal{S}$ is retained as necessary.

We apply this evaluation to the longest candidate tool chain and then progressively examine its individual tools. For example, consider the chain Temporal Grounding $\rightarrow$ Timestamp Grounding $\rightarrow$ Spatial Grounding $\rightarrow$ Image Search, which is used to identify “the person wearing blue.” If the base model can directly identify the person from the original video, the entire tool chain is removed. Then, we further evaluate the necessity of the tools within the chain. For example, to evaluate Temporal Grounding, we bypass its localized temporal interval and directly provide the original video together with a query asking the model to locate a usable frame containing “the person wearing blue.” If the base model can directly locate such a frame from the full video, Temporal Grounding is unnecessary and removed. We apply this principle to the remaining grounding and retrieval tools, progressively removing tool calls.

4.3 Training Data

Following the above pipeline, we 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 RL data, we construct QA pairs following the procedure in Sec. 4.1.

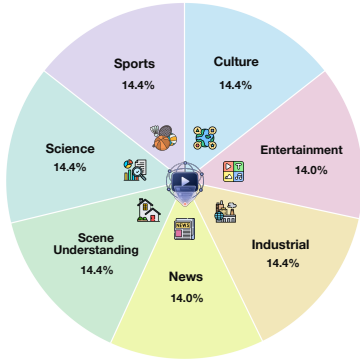


Figure 3: Distribution of the 250 VDR-EE questions across seven semantic domains.

Table 1: Statistics and video-centric annotations of VDR-EE.

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

4.4 Benchmark

By the QA generation process, we introduce **VDR-EE**, a manually verified Video Deep Research benchmark. Each QA pair 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. 3.

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 (≤ 30 s), medium (30–120 s), and long (> 120 s). All questions are also grouped by full-video duration into short (≤ 5 min), medium ($5 < t < 10$ min), and long (≥ 10 min) videos. Table 1 summarizes these annotations, including 153 entity and 97 event questions. Representative questions are shown in Fig. 4.

These annotation dimensions capture different localization demands: entity questions may require identifying multiple visual targets, while event questions may require locating an interval before external retrieval.

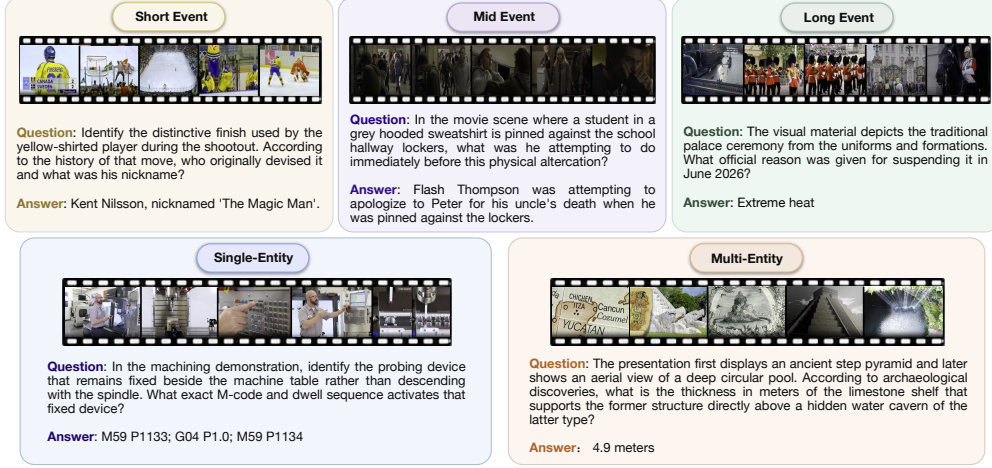


Figure 4: Examples of event-oriented and entity-oriented questions in VDR-EE.

5 Training

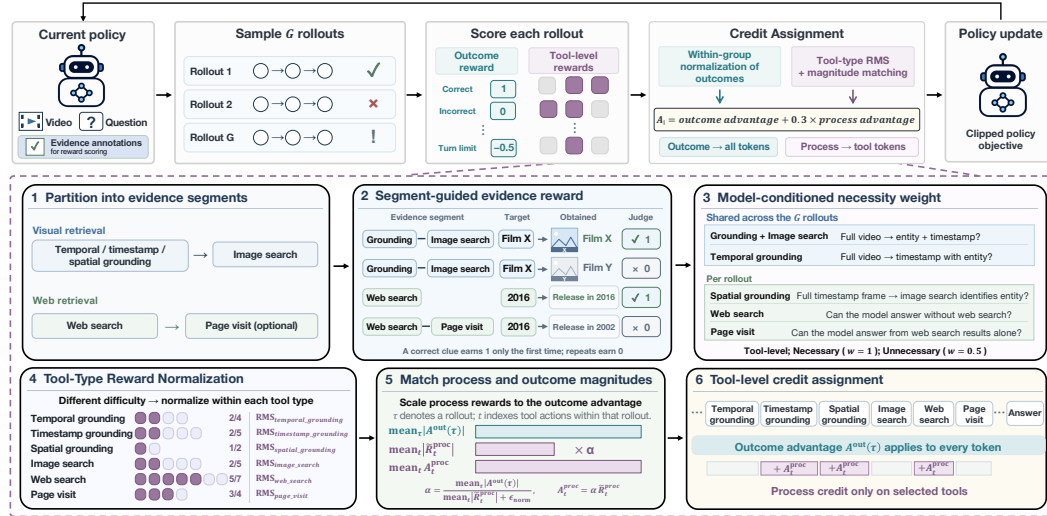


Figure 5: Overview of NASA-GRPO, a necessity-aware segment-guided credit assignment RL.

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 $y^{(n)} = \{y_1^{(n)}, \dots, y_{N_n}^{(n)}\}$ the target assistant output, we optimize:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{M} \sum_{n=1}^M \sum_{i=1}^{N_n} \log \pi_{\theta}(y_i^{(n)} | x^{(n)}, y_{<i}^{(n)}). \quad (5)$$

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. 5, 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

$$R^{\text{out}}(\tau) = \begin{cases} 1, & \text{if the final answer is correct,} \\ 0, & \text{if the final answer is incorrect,} \\ -0.5, & \text{if the rollout is invalid or reaches the turn limit.} \end{cases} \quad (6)$$

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

$$A^{\text{out}}(\tau) = \frac{R^{\text{out}}(\tau) - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}} + \varepsilon_{\text{norm}}}, \quad (7)$$

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 $\tau = (a_1, o_1, \dots, a_T, o_T)$ 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 $\hat{c}_m$ denote the clue obtained by $\mathcal{S}_m$. We assign evidence reward only when the clue is both correct and has not been obtained,

$$S_m^{\text{evid}} = \mathbb{I}[J(\hat{c}_m, c_m^*) = 1] \cdot \mathbb{I}[c_m^* \notin \mathcal{C}_{<m}^{\text{credit}}], \quad (8)$$

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. 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 current policy to recover the same target information. At each RL iteration, we use the policy snapshot $\pi_{\theta_{\text{old}}}$ that generates the rollout group. This snapshot is held fixed during reward evaluation and refreshed at the next iteration, so necessity judgments track the evolving capabilities of the policy. 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.

An action receives full reward 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

$$w_t^{\text{nec}} = \begin{cases} 1, & \text{if } \widehat{y}_{a_t} \neq y_{a_t}, \\ 0.5, & \text{if } \widehat{y}_{a_t} = y_{a_t}, \end{cases} \quad (9)$$

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

$$R_t^{\text{proc}} = S_{s(t)}^{\text{evid}} \cdot w_t^{\text{nec}}. \quad (10)$$

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

$$\widetilde{R}_t^{\text{proc}} = \frac{R_t^{\text{proc}}}{\sqrt{\frac{1}{|\mathcal{B}_{k(t)}|} \sum_{j \in \mathcal{B}_{k(t)}} (R_j^{\text{proc}})^2 + \varepsilon_{\text{norm}}}}. \quad (11)$$

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

$$\alpha = \frac{\text{mean}_{\tau} |A^{\text{out}}(\tau)|}{\text{mean}_t |\widetilde{R}_t^{\text{proc}}| + \varepsilon_{\text{norm}}}, \quad A_t^{\text{proc}} = \alpha \widetilde{R}_t^{\text{proc}}. \quad (12)$$

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

$$A_i = A^{\text{out}}(\tau) + \lambda \sum_t m_{t,i} A_t^{\text{proc}}, \quad \lambda = 0.3. \quad (13)$$

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 E. 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.

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 2. Additional results on VDR-EE and VideoDR, as well as evaluations on image deep research and long-video understanding, are provided in Appendix C. Qualitative examples are provided in Sec. 6.6.

SFT improves the Qwen3.5-9B base model on both benchmarks. As shown in Table 2, AdaVDR-9B-SFT raises accuracy from 37.00% to 52.00% on VideoDR and from 40.00% to 48.00% on VDR-EE,

Table 2: Accuracy (%) on benchmarks. 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		
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-SFT (Qwen3.5-9B)	52.00	60.53	33.77	47.06	48.57	47.06	53.57	49.48	48.00	50.00
Δ vs. Qwen3.5-9B	+15.00	+3.95	+5.20	+4.58	+14.28	+14.71	+10.71	+13.40	+8.00	+11.50
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

corresponding to gains of 15.00 and 8.00 percentage points, respectively. Subsequent NASA-GRPO training further improves accuracy to 66.00% and 55.20%, adding 14.00 and 7.20 percentage points over SFT.

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 15.20 and 29.00 percentage points, 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 2). 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 3: 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
<i>VideoDR</i>							
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
<i>VDR-EE</i>							
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 3 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 14.0 and 7.2 percentage points over the best SFT configuration on VideoDR and VDR-EE, respectively (Table 4). Detailed tool-use analyses are outlined in Appendix D.

6.4 Ablation

Table 4 evaluates the contributions of SFT data composition and RL reward components on both benchmarks.

Table 4: Ablation of SFT data and RL rewards. Accuracy is reported in %; Δ is the gain in percentage points over the base model for SFT and over Reflection + Adaptive Data for RL.

Configuration	VideoDR		VDR-EE	
	Acc.	Δ	Acc.	Δ
Base Model	37.00	–	40.00	–
<i>SFT data composition (Δ vs. Base Model)</i>				
Base Data	50.00	+13.00	47.20	+7.20
Adaptive Data	48.00	+11.00	46.00	+6.00
Reflection Data	51.00	+14.00	47.60	+7.60
Reflection + Adaptive Data	52.00	+15.00	48.00	+8.00
<i>RL reward components (Δ vs. full SFT)</i>				
Accuracy + Format	53.00	+1.00	48.80	+0.80
+ Segment-Guided Evidence Reward	57.00	+5.00	50.80	+2.80
+ Tool-Type Normalization	59.00	+7.00	52.00	+4.00
+ Model-Conditioned Necessity Weight	66.00	+14.00	55.20	+7.20

SFT data composition. Training on base tool-use trajectories improves accuracy over the untrained model, establishing the benefit of learning to acquire evidence through tools. Adaptive data alone slightly lowers accuracy relative to base data, suggesting that learning to skip interactions does not by itself ensure that useful evidence is preserved. Reflection data improves performance by teaching the agent to revisit unreliable intermediate results. Combining reflection with adaptive data yields the strongest SFT configuration, reaching 52.00% on VideoDR and 48.00% on VDR-EE, or gains of 15.00 and 8.00 percentage points over the base model. These results support combining selective tool use with recovery from tool failures.

RL reward components. Starting from the combined SFT model, accuracy and format rewards produce only a small improvement, while segment-guided evidence rewards provide more direct feedback for useful evidence acquisition and raise accuracy to 57.00% and 50.80%. Tool-type normalization further improves these results to 59.00% and 52.00% by reducing the imbalance between tool types with different positive-reward rates. Finally, model-conditioned necessity weighting distinguishes useful evidence from tool calls that the model could bypass, increasing accuracy to 66.00% and 55.20%. This corresponds to total gains of 14.00 and 7.20 percentage points over SFT, while average interaction turns decrease from 13.67 to 11.67 on VideoDR and from 15.43 to 13.26 on VDR-EE relative to the variant without necessity weighting. Further ablations are reported in Appendix F.

6.5 Training Curves

Figure 6 compares the training rollout rewards of GRPO, the variant without the necessity weight, and NASA-GRPO. All three runs exhibit substantial fluctuations in raw reward, while the smoothed curves reveal different training trends. GRPO improves after an initial decline but shows no sustained upward trend in the later steps. Adding segment-guided evidence rewards and tool-type normalization without the necessity weight produces a clearer early improvement, followed by fluctuations around a relatively stable reward level. NASA-GRPO also improves rapidly at the beginning and maintains a higher smoothed reward toward the end of the displayed training window.

These trends are consistent with the ablation results in Table 4: segment-guided credit assignment strengthens the learning signal beyond rollout-level feedback, while necessity weighting further improves the resulting policy. Evidence contribution alone can reward a successful tool call even when the model could recover the same information without it. The necessity weight reduces credit

for such bypassable actions, directing the process signal toward evidence acquisition that the model actually needs. The reward curves provide a training diagnostic; the held-out accuracy gains and tool-necessity analysis in Appendix D.2 provide complementary evidence for improved tool selection.

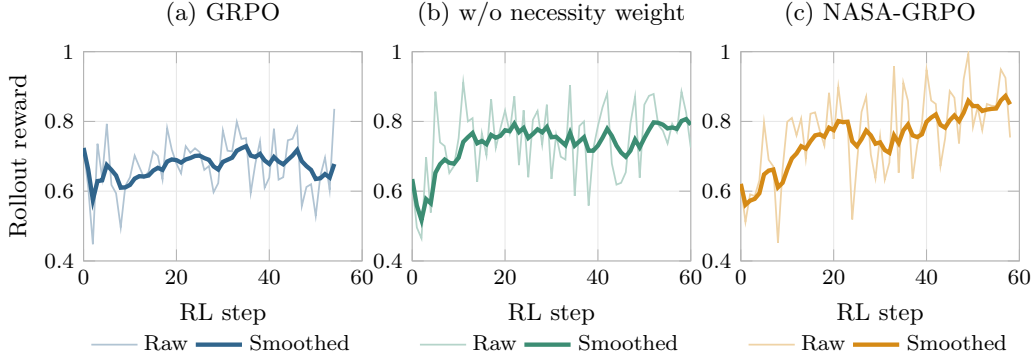


Figure 6: Training rollout rewards for (a) GRPO, (b) the variant without the necessity weight, and (c) NASA-GRPO. Pale and dark lines show raw and smoothed rewards. The curves in (b) are approximately digitized from the training-dashboard screenshot.

6.6 Qualitative Analysis

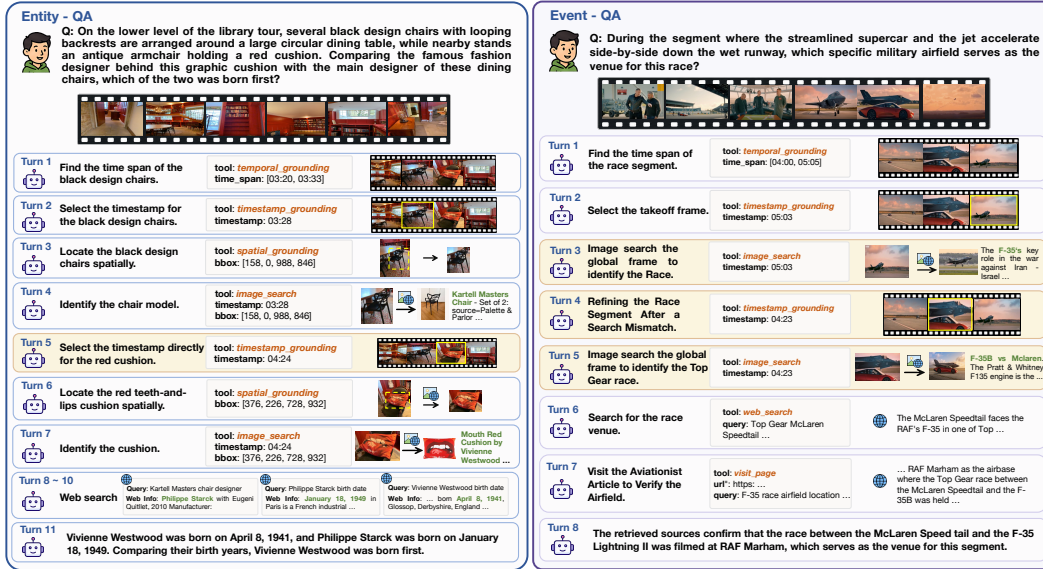


Figure 7: Qualitative examples of AdaVDR on entity-oriented and event-oriented questions.

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.

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 adaptive tool use and reflection through SFT, while NASA-GRPO further improves the policy through necessity-aware credit assignment. We also construct VDR-EE, a manually verified benchmark covering entity- and event-centric questions. The resulting AdaVDR selects necessary tools and corrects unreliable intermediate results when needed. Experiments demonstrate improvements in both accuracy and efficiency, with accuracy gains of 29.00 and 15.20 percentage points over the base model on VideoDR and VDR-EE, respectively.

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.
- Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.
- Boyuan 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)*, pages 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)*, pages 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*, pages 19383–19400, 2024.
- Xingqi He, Yujie Zhang, Shuyong Gao, Wenjie Li, Lingyi Hong, Mingxi Chen, Kaixun Jiang, Jiyan Fu, and Wenqiang Zhang. Rsgent: 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, pages 114851–114872, 2026.
- Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, et al. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*, 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, Serkan 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 video llms 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.
- Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, et al. Sekai: A video dataset towards world exploration. *Advances in Neural Information Processing Systems*, 38, 2026c.
- 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.
- Xinhao Liu, Jintong Li, Yicheng Jiang, Niranjan Sujay, Zhicheng Yang, Juexiao Zhang, John Abanes, Jing Zhang, and Chen Feng. Citywalker: Learning embodied urban navigation from web-scale videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6875–6885. IEEE, 2025a.
- Ye Liu, Kevin Qinghong Lin, Chang Wen Chen, and Mike Zheng Shou. Videomind: A chain-of-lora agent for long video reasoning. *arXiv preprint arXiv:2503.13444*, 2025b.
- 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 llms 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, Xinfu 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 multimodal 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)*, pages 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)*, pages 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.
- Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12–22. IEEE, 2023.
- 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*, pages 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	19
A.1	Training Parameters	19
B	Proof of Theorem 1	19
C	More Experimental Results	21
C.1	Results on VDR-EE	21
C.2	Results on VideoDR	22
C.3	Image Deep Research	23
C.4	Long-Video Understanding	23
D	More Tool-Use Analysis	24
D.1	Tool Effectiveness	24
D.2	Tool Necessity Analysis	25
E	Training Dynamics	25
E.1	Process Credit During Training	27
E.2	Effect of Model-Conditioned Necessity Weight on Tool Use	27
F	More Ablations	28
F.1	Necessity Weight Ablation	28
F.2	Process-Advantage Coefficient Ablation	28
F.3	Tool Ablation	29
G	Human–Credit Agreement	29
H	Prompts	30
H.1	LLM Judge Prompt	30
H.2	Visual Anchor Extraction Prompt	31
H.3	QA Generation Prompt	32
H.4	Initial Trajectory Generation Prompt	33
H.5	Trajectory Refinement Prompt	34
H.6	Model-Conditioned Tool Necessity Filtering Prompts	35
H.7	Reflection and Retry Prompts	36

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 5 summarizes the RL configuration.

Table 5: RL training configuration for NASA-GRPO.

Parameter	Value
<i>Compute and parallelism</i>	
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
<i>Sampling and batching</i>	
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
<i>Optimization</i>	
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

B Proof of Theorem 1

Definition 2. 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

$$\Gamma_M := I(Y; X | H_M). \quad (14)$$

Definition 3. We define the misleadingness δ of erroneous tool observations, which measures how strongly erroneous tool outputs distort the model prediction, formulated as

$$\delta := \mathbb{E}[\text{TV}(P(Y | H_M, Z), P(Y | H_M, Z; M)) | C = 1]. \quad (15)$$

Definition 4. 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

$$\xi_M := \mathbb{E}_{H_M} [D_{\text{KL}}(P(Y | H_M) \| P(Y | H_M; M))]. \quad (16)$$

Lemma 5. We denote the base risk R_{base} and the tool-augmented risk R_{tool} as

$$R_{\text{base}} = \mathbb{E}[-\log P(Y | H_M; M)], R_{\text{tool}} = \mathbb{E}[-\log P(Y | H_M, Z; M)]. \quad (17)$$

Let

$$\xi_{\text{tool}} = \mathbb{E}_{H_M, Z} [D_{\text{KL}}(P(Y | H_M, Z) \| P(Y | H_M, Z; M))] \quad (18)$$

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

$$R_{\text{tool}} - R_{\text{base}} = -I(Y; Z | H_M) + \xi_{\text{tool}} - \xi_M. \quad (19)$$

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

$$R_{\text{base}} = \mathbb{E}[-\log P(Y | H_M; M)] \quad (20)$$

$$= H(Y | H_M) + \mathbb{E}_{H_M} [D_{\text{KL}}(P(Y | H_M) \| P(Y | H_M; M))] \quad (21)$$

$$= H(Y | H_M) + \xi_M. \quad (22)$$

Similarly, the tool-augmented risk satisfies

$$R_{\text{tool}} = \mathbb{E}[-\log P(Y | H_M, Z; M)] \quad (23)$$

$$= H(Y | H_M, Z) + \mathbb{E}_{H_M, Z} [D_{\text{KL}}(P(Y | H_M, Z) \| P(Y | H_M, Z; M))] \quad (24)$$

$$= H(Y | H_M, Z) + \xi_{\text{tool}}. \quad (25)$$

Therefore,

$$R_{\text{tool}} - R_{\text{base}} = H(Y | H_M, Z) - H(Y | H_M) + \xi_{\text{tool}} - \xi_M \quad (26)$$

$$= -I(Y; Z | H_M) + \xi_{\text{tool}} - \xi_M, \quad (27)$$

where the last equality follows from the definition of conditional mutual information. $\square$

Theorem 6. We define the base risk R_{base} and the tool-augmented risk R_{tool} as

$$R_{\text{base}} = \mathbb{E}[-\log P(Y | H_M; M)], R_{\text{tool}} = \mathbb{E}[-\log P(Y | H_M, Z; M)]. \quad (28)$$

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

$$R_{\text{tool}} - R_{\text{base}} \geq 2\epsilon\delta^2 - \Gamma_M - \xi_M. \quad (29)$$

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

$$R_{\text{tool}} - R_{\text{base}} = -I(Y; Z | H_M) + \xi_{\text{tool}} - \xi_M, \quad (30)$$

where

$$\xi_{\text{tool}} = \mathbb{E}_{H_M, Z} [D_{\text{KL}}(P(Y | H_M, Z) \| P(Y | H_M, Z; M))]. \quad (31)$$

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

$$I(Y; Z | H_M) \leq I(Y; X | H_M) = \Gamma_M. \quad (32)$$

Therefore,

$$-I(Y; Z | H_M) \geq -\Gamma_M. \quad (33)$$

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

$$\begin{aligned} \xi_{\text{tool}} &= (1 - \epsilon)\mathbb{E}[D_{\text{KL}}(P(Y | H_M, Z) \| P(Y | H_M, Z; M)) | C = 0] \\ &\quad + \epsilon\mathbb{E}[D_{\text{KL}}(P(Y | H_M, Z) \| P(Y | H_M, Z; M)) | C = 1] \\ &\geq \epsilon\mathbb{E}[D_{\text{KL}}(P(Y | H_M, Z) \| P(Y | H_M, Z; M)) | C = 1]. \end{aligned} \quad (34)$$

By Pinsker's inequality,

$$D_{\text{KL}}(P \| Q) \geq 2 \text{TV}(P, Q)^2, \quad (35)$$

and hence

$$\begin{aligned} \xi_{\text{tool}} &\geq 2\epsilon\mathbb{E}[\text{TV}(P(Y | H_M, Z), P(Y | H_M, Z; M))^2 | C = 1] \\ &\geq 2\epsilon(\mathbb{E}[\text{TV}(P(Y | H_M, Z), P(Y | H_M, Z; M)) | C = 1])^2 \\ &= 2\epsilon\delta^2, \end{aligned} \quad (36)$$

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

Substituting Eq. 33 and Eq. 36 into Eq. 30, we obtain

$$\begin{aligned} R_{\text{tool}} - R_{\text{base}} &\geq -\Gamma_M + 2\epsilon\delta^2 - \xi_M \\ &= 2\epsilon\delta^2 - \Gamma_M - \xi_M. \end{aligned} \quad (37)$$

Therefore,

$$R_{\text{tool}} - R_{\text{base}} \geq 2\epsilon\delta^2 - \Gamma_M - \xi_M, \quad (38)$$

which completes the proof. $\square$

C 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.

C.1 Results on 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 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.

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

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

Model	Agentic			
	Low	Medium	High	Avg.
#Samples	32	36	32	100
<i>Proprietary</i>				
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
<i>Open-source</i>				
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

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

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

C.3 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.
<i>Agentic workflow</i>						
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	—
<i>Multimodal deep-research models</i>						
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

C.4 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

D More Tool-Use Analysis

D.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

Table 12 and Fig. 8, together with Table 3, 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

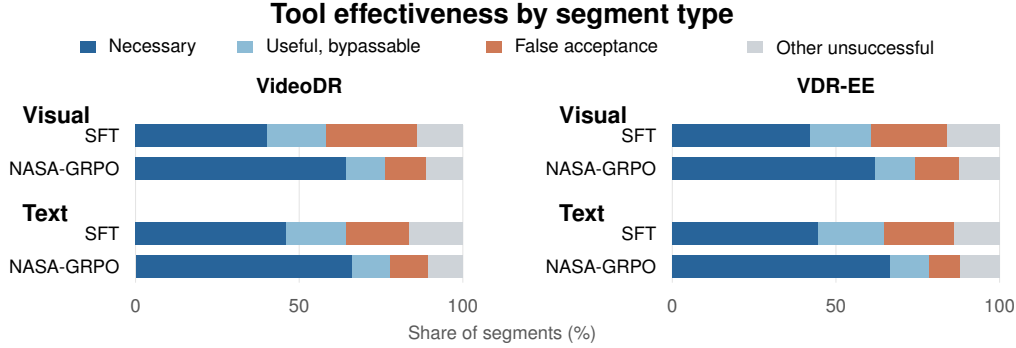


Figure 8: Tool effectiveness by segment type on VideoDR and VDR-EE. Necessary and useful but bypassable segments partition useful segments; false acceptance and other unsuccessful segments partition the remaining segments.

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.

D.2 Tool Necessity Analysis

We compare the RL variants after tool-type normalization, with and without the model-conditioned necessity weight (Table 4), on the same VideoDR and VDR-EE questions under identical decoding settings. We identify useful evidence-acquisition segments using the criteria in Sec. D.1 and apply the model-conditioned recovery criterion of Eq. 9. 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 4 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

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 4 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.

E Training Dynamics

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

Question: Approximately how many meters above sea level is the church in the video located?

Answer: 2,580 meters



Video frame: RL observation at 00:00:07

Red: incorrect content. Green: correct identification, evidence use or answer.

SFT · 9 steps

STEP 1 | REASONING + CALL

Initial belief is stated before retrieval

SFT identifies the church as Fayita and recalls 2,000–2,100 m; temporal_grounding selects 00:05–00:10.

STEP 2 | CALL

timestamp_grounding: extract the church frame

SFT labels the target Fayita Church at 00:06.

STEP 3 | CALL

image_search: query the extracted frame

The image query already includes “Fayita Gorge”.

STEP 3 | RETURN

Image search supplies a different identity

Returned title: “Follow @greatbigstory → Abuna Yemata Guh is a church ...”
Another result also names Abuna Yemata Guh.

STEP 4 | ISSUE A · REASONING

Wrong church identity: Fayita Church

The church is Abuna Yemata Guh, as identified in the paired retrievals. SFT instead asserts Fayita Church and continues querying that incorrect identity.
“but the video itself shows Fayita Church”

STEPS 4-8 | CALLS + RETURNS

Five web searches fail to establish elevation

Search 1: unrelated text / no Fayita elevation (M9).
Search 2: empty results + tool error (M11).
Search 3: unrelated text (M13).
Searches 4-5: empty results + tool errors (M15, M17).

STEP 9 | ISSUE B · REASONING

Wrong elevation: approximately 2,000 m

SFT says it relies on its knowledge, but gives 2,000 m. This value is wrong: the standard answer and matching retrieved snippets give 2,580 m.

“at an altitude of approximately 2,000 meters”

STEP 9 | FINAL ANSWER · WRONG

Approximately 2,000 meters

Incorrect against the standard answer of 2,580 meters. Red marks the wrong value, not the use of prior knowledge.

RL · 5 steps

STEP 1 | CALL

Locate the church in the video

temporal_grounding: 00:05–00:10; target is “church in the video”.

STEP 2 | CALL

timestamp_grounding: extract the church frame

RL describes the church in Tigray and selects 00:07.

STEP 3 | CALL

image_search: query the extracted frame

The query describes a cliff church in Tigray, without choosing a church name.

STEP 3 | RETURN

Image results identify Abuna Yemata Guh

Titles name Abuna Yemata Guh; another result describes its entrance in the Gheralta Mountains.

STEP 4 | CONTROL A · REASONING + CALL

Use the returned identity instead of overriding it

“The church is Abuna Yemata Guh ...”
web_search query: Abuna Yemata Guh church elevation above sea level
meters Tigray Ethiopia.

STEP 4 | CONTROL B · RETURN → ANSWER

Obtain explicit elevation evidence before answering

Several returned snippets state 2,580 meters. The Wikipedia snippet gives the altitude above sea level.

“altitude above sea level of 2,580 metres”

STEP 5 | FINAL ANSWER · CORRECT

Approximately 2,580 meters

Returned church identity → matching web query → explicit elevation evidence → answer.

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.

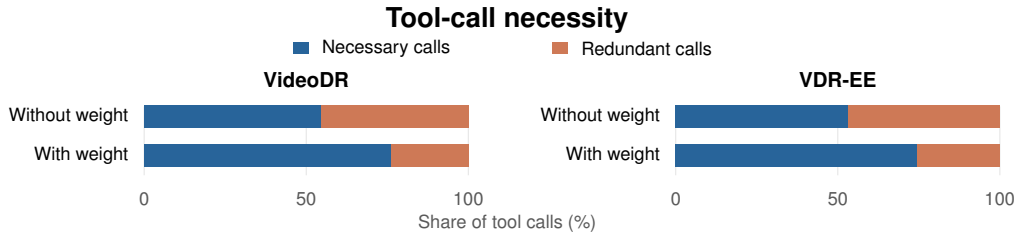


Figure 10: Necessary and redundant tool calls on VideoDR and VDR-EE, with and without the model-conditioned necessity weight. Shares are computed over all tool calls.

E.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.

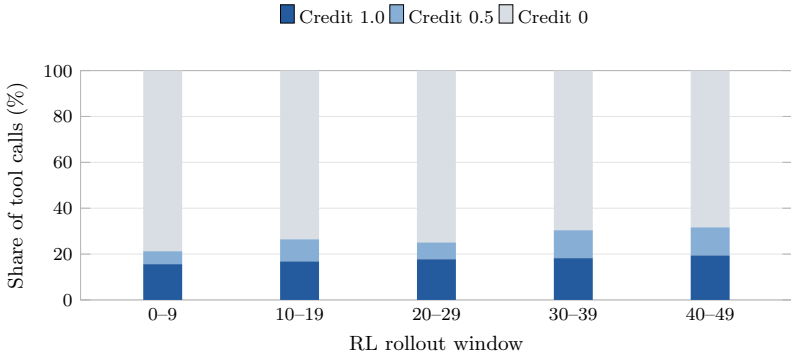


Figure 11: Process-credit composition across RL rollout windows. Each bar shows the share of calls assigned credit 0, 0.5, or 1.0, using the counts in Table 14.

E.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 4.

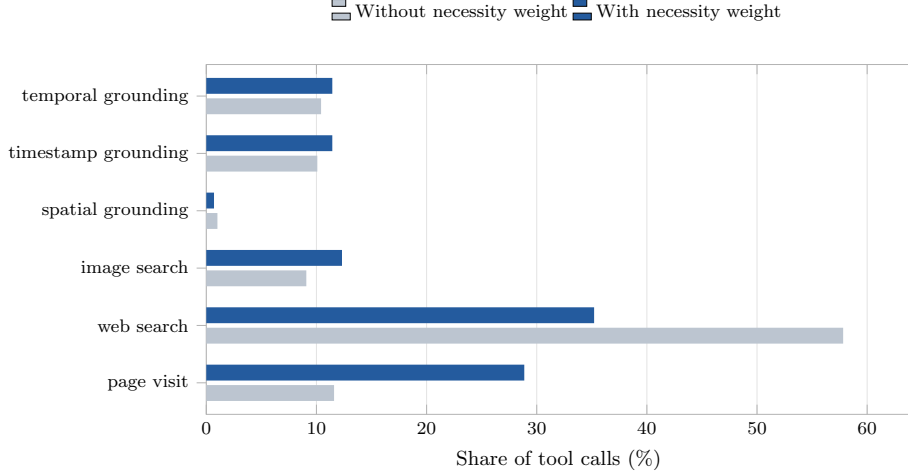


Figure 12: Tool-use distribution in a training snapshot with and without the model-conditioned necessity weight. Bars show each tool’s share of all calls.

F More Ablations

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

F.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. 9); all other training settings remain fixed.

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

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.

F.2 Process-Advantage Coefficient Ablation

We vary the coefficient λ that scales the process advantage in Eq. 13, keeping the necessity weights fixed at their default values. The default is $\lambda = 0.3$.

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.

E.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 2.*

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 3.

G 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

H Prompts

H.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."
}
```

H.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": "..."
  }]
}
```

H.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": [...]
  }]
}
```

H.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": [...]
  }]
}
```

H.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": "..."
  }]
}
```

H.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"}.

H.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"}.
```