VideoSearcher: Empowering Video Deep Research with Multi-Tool Agentic Reasoning via Reinforcement Learning

Zhenkun Gao[*,1,2], Yicheng Bao[*,1], Jinlong Peng[*,†,✉,2], Xueheng Li[*,✉,3], Suyuan Huang[*,4,2]
Bangwei Liu[1], Kunquan Li[5], Zhenye Gan[2], Tao Hu[3], Chengjun Xie[3], Xuanhua He[✉,6]
Zhizhong Zhang[1], Xin Tan[1], Chengjie Wang[2], Yuan Xie[✉,1]

[*] Equal Contribution, [†] Project Lead, ✉ Corresponding Author
[1]East China Normal University, [2]Tencent Youtu Lab, [3]University of Science and Technology of China
[4]Wuhan University, [5]Xiamen University, [6]The Hong Kong University of Science and Technology

Video understanding is moving beyond closed-context perception toward open-world evidence exploration, a paradigm formalized as Video Deep Research (VDR). However, existing multimodal search agents primarily target static images, and the current VDR benchmark relies on text-centric retrieval that discards crucial visual information. To address these limitations, we propose VideoSearcher, a closed-loop agentic framework that empowers Vision-Language Models with multi-tool reasoning for VDR. VideoSearcher unifies temporal localization, spatial focusing, and multimodal search within a single reasoning trajectory, enabling agents to progressively ground visual clues, retrieve relevant evidence, and synthesize answers. To optimize knowledge-intensive reasoning trajectories, we propose Bi-branch Sequence Policy Optimization (BiSPO), a reinforcement learning algorithm that decouples tool-invocation optimization from answer-accuracy optimization. This design provides stable learning signals for both evidence-grounded reasoning and purposeful tool use. Furthermore, we construct VideoSearch-QA, the first benchmark designed to evaluate open-world video information grounding and multimodal search-based reasoning. Extensive experiments demonstrate that VideoSearcher significantly outperforms prior open-source agentic baselines across various search-oriented and multimodal understanding benchmarks.

1 Introduction

Video understanding [1, 2, 3] remains a critical bottleneck for deploying Vision-Language Models (VLMs) [4, 5] in real-world applications [6, 7]. Unlike static images, videos encompass evolving temporal dynamics, requiring models to track state transitions across frames for accurate spatio-temporal modeling. Particularly in long videos [8], crucial evidence is frequently submerged within massive redundant frames and manifests only during fleeting temporal windows. Therefore, models need to localize the relevant moments in time, focus on fine-grained visual details such as text or small objects, and reason over the collected evidence to answer the query. Robust video understanding thus requires both global context modeling and fine-grained spatio-temporal perception [9].

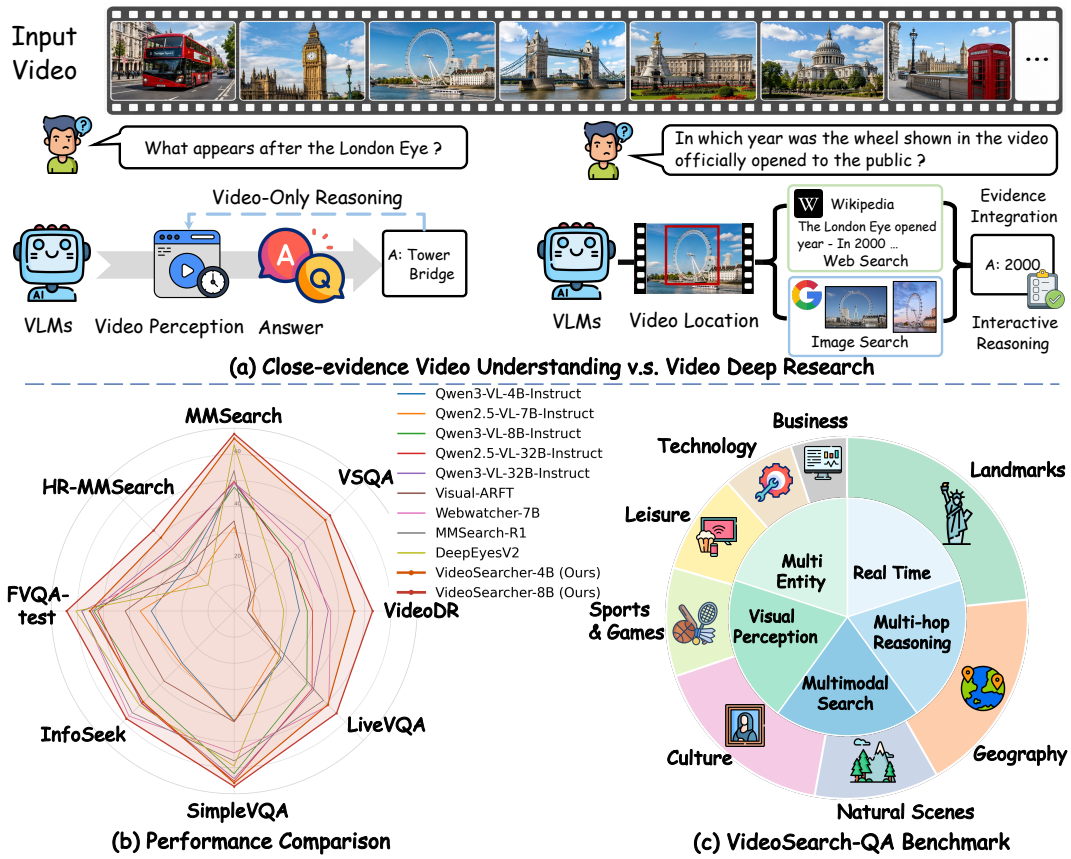


Figure 1. Overview of VideoSearcher. (a) Video Deep Research extends closed-evidence video understanding by requiring agents to ground visual cues and retrieve open-web evidence. (b) Performance comparison on search-oriented benchmarks. (c) Our constructed VideoSearch-QA benchmark covers diverse domains for evaluating video-grounded multimodal search ability.

However, most existing video understanding works [10, 11] adhere to a closed-evidence paradigm, assuming that all necessary information resides within the input video. While this setting is useful for evaluating spatio-temporal video understanding, it remains insufficient for real-world scenarios. In practical video QA, videos often provide only visual anchors (e.g., logos or people), while the final answer must be retrieved from open-web sources, including webpages, images, news, or knowledge bases, as shown in Fig. 1(a). Recently, VideoDR [12] proposed a benchmark and formalized this setting as **Video Deep Research** (VDR), where models must identify cross-frame visual anchors and combine them with open-web retrieval to produce verifiable answers. This paradigm extends video understanding from closed-context perception to open-world evidence exploration, positioning videos as entry points for agents to retrieve and verify information in expansive web environments.

In recent years, deep research [13] and multimodal search [14] agents have shown that tool-augmented closed-loop reasoning can substantially improve knowledge-intensive problem solving. Text-based agents solve open-web QA through iterative query formulation and evidence integration [15], while multimodal agents [16, 17] further introduce visual tools such as image cropping and image search to combine external knowledge with localized visual analysis. However, these methods are primarily designed for static images, making them insufficient for VDR, where agents

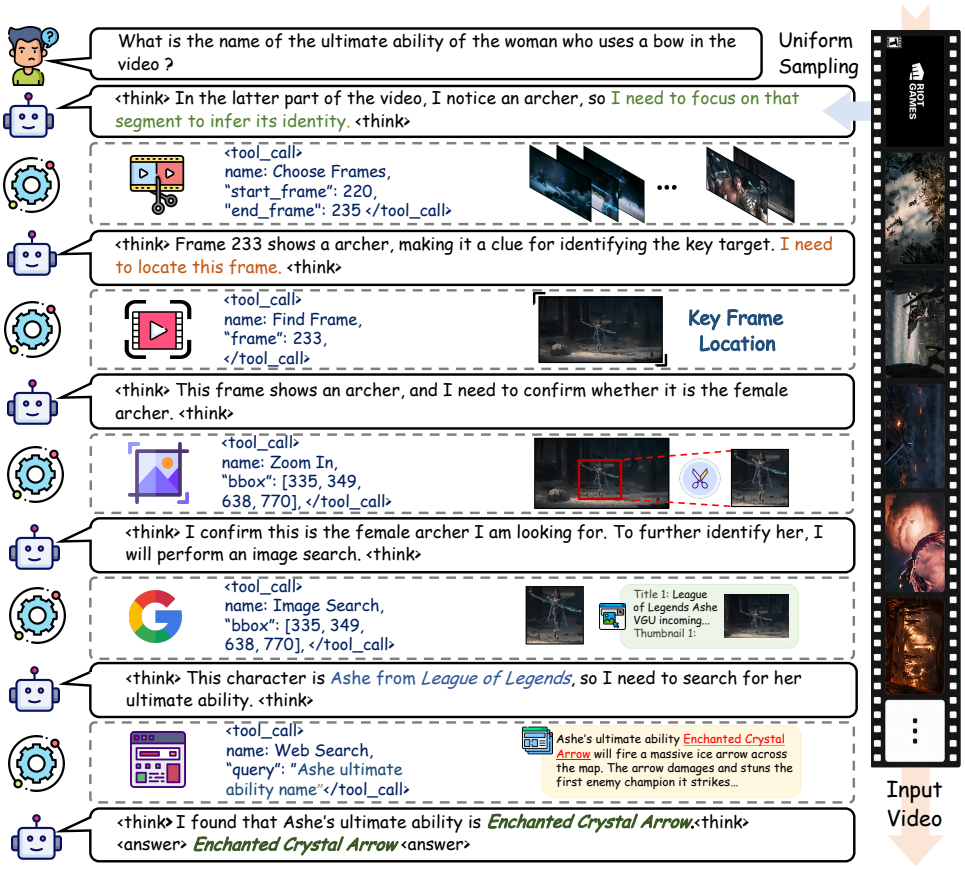


Figure 2. Agentic reasoning trajectory of VideoSearcher. VideoSearcher performs agentic reasoning by localizing key frames, inspecting regions, invoking web/image search, and integrating multimodal evidence for answer generation.

must track cross-frame cues, model spatio-temporal state changes, and jointly decide temporal localization, spatial focus, and search modality. Although VideoDR [12] represents an important step, it remains largely text-centric by converting visual cues into textual descriptions before web search. This setting loses the original visual information and cannot fully reflect real-world video retrieval. Capable VDR agents should localize visual anchors across frames, inspect fine-grained regions, construct multimodal queries, and reason over evidence gathered through long-horizon web interaction. This demands fine-grained video understanding and precise tool use, including discriminative text queries and targeted cropped regions for image search.

To address these challenges, we introduce **VideoSearcher**, a pioneering agentic reasoning and search framework for Video Deep Research. VideoSearcher equips the model with five tools: `choose_frames` for coarse temporal selection, `find_frame` for key frame localization, `zoom_in` for local region inspection, as well as `web_search` and `image_search` for textual and visual web retrieval. These tools enable a unified trajectory in which the model progressively narrows the video timeline, focuses on informative regions, retrieves multimodal evidence, and integrates the acquired evidence for reasoning. We adopt a two-stage training paradigm comprising cold-start Supervised Fine-tuning (SFT) and Reinforcement Learning (RL). To construct multi-tool reasoning trajectories, we design a VDR-specific data synthesis pipeline. We extract and cross-validate core entities from search-oriented image-text QA datasets, and use them as retrieval targets to collect

relevant videos. The retrieved videos are normalized into unified frame sequences. We then rewrite the original QA pairs to align with the video context, followed by visual-QA alignment verification to discard noisy samples with missing entities, answer leakage, or inconsistent visual anchors. We then synthesize trajectories by first localizing key frames and generating complete reasoning traces with diverse tool invocations. Through rigorous filtering and quality assurance, we derive high-quality SFT data. The remaining hard video-QA instances are reserved for RL rollouts in the VDR environment.

Furthermore, we propose Bi-branch Sequence Policy Optimization (**BiSPO**) to optimize long-horizon multi-tool reasoning trajectories in VDR. Built upon GSPO [18], BiSPO applies sequence-level importance sampling and clipping, and further separates answer accuracy optimization from tool invocation optimization. This dual-branch design provides more stable learning signals for knowledge-intensive and reasoning-heavy VDR. We also introduce a bell-shaped gated tool invocation reward that encourages sufficient video observation and external retrieval within a reasonable range, while penalizing excessive tool calls that may cause invalid searches or incorrect tool sequences. Finally, we introduce **VideoSearch-QA** (VSQA), a benchmark for evaluating whether VLM-based agents can jointly capture video cues and perform multimodal search-based reasoning in dynamic videos. Extensive experiments on VideoDR [12], VideoSearch-QA, and a suite of multimodal search and video understanding benchmarks validate the effectiveness of VideoSearcher.

In summary, our contributions are as follows:

- We propose VideoSearcher, the first closed-loop Video Deep Research agent that integrates temporal localization, spatial focusing and multimodal search for dynamic videos.
- We propose BiSPO, a dual-branch RL algorithm that separately optimizes answer accuracy and tool invocation behavior for stable long-horizon tool-intensive reasoning.
- We construct VideoSearch-QA, the first benchmark for evaluating VDR agents on video information grounding and multimodal search-based reasoning in open-world scenarios.
- Extensive experiments on VideoDR, VSQA, and multiple multimodal search and video understanding benchmarks validate the effectiveness of VideoSearcher.

2 Related Work

2.1 Video Understanding

Recent advances in Vision-Language Models (VLMs) have substantially improved video understanding, particularly in spatio-temporal perception and reasoning [4]. Advanced video-centric models such as Video-R1 [7] and VideoRFT [19] leverage Reinforcement Learning to strengthen the video reasoning capabilities of models. FrameThinker [10] and LongVT [11] further introduce agentic techniques, such as iterative tool use and adaptive frame selection, to improve long video understanding. Despite these advances, most existing works still follow a closed-evidence paradigm [12], where answers are assumed to be inferable solely from the input video, leaving open-web information acquisition and multimodal search-oriented reasoning insufficiently explored.

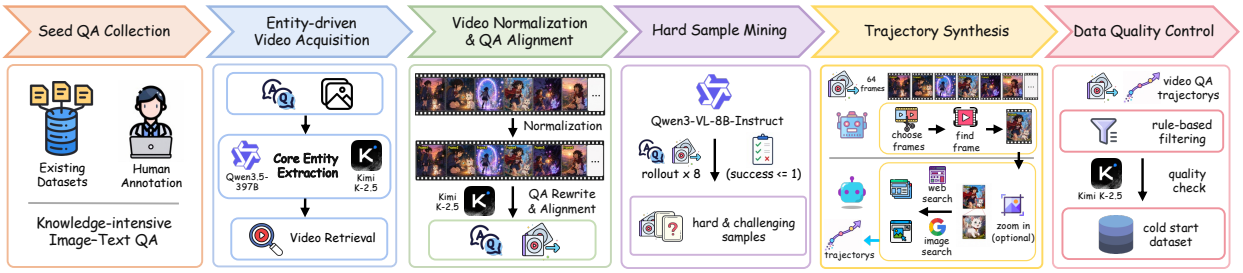


Figure 3. The video-centric training data synthesis pipeline of VideoSearcher.

2.2 Tool-Augmented Agentic VLMs

Existing studies have advanced VLMs from passive perception to tool-augmented agentic reasoning. Under the "thinking with images" paradigm, models such as OpenThinkIMG [20] and DeepEyesV2 [17] use interactive visual tools to manipulate, generate or inspect intermediate visual states for fine-grained reasoning. In addition, search agents such as MMSearch-R1 [16] and Webwatcher [21] integrate multimodal search tools, enabling models to acquire open-web information beyond their parametric knowledge. Other efforts [22, 23] further explore tool use for complex visual reasoning, including visual analysis and code execution. Despite these advances, existing models are mainly designed for text or static images. They remain insufficient for VDR, where agents must perform cross-frame localization, spatial focusing, multimodal search, and long-chain evidence reasoning over dynamic videos.

3 Methods

3.1 Problem Formulation

Task Definition. We formulate Video Deep Research (VDR) as an open-world video question answering task. Given a video $\mathcal{V}$, a question q, and an external web evidence space $\mathcal{W}$, an agent π_θ is required to produce an answer y through a multi-step reasoning trajectory $\tau = (a_1, o_1, \ldots, a_T, o_T)$. At each turn t, the agent selects an action $a_t \in \mathcal{A}$, such as temporal localization, spatial inspection, image search, or web search, and receives an observation o_t from either the video or the open web. The trajectory induces a joint evidence set $\mathcal{E}_\tau = \mathcal{E}_\tau^v \cup \mathcal{E}_\tau^w$, where $\mathcal{E}_\tau^v$ contains localized video evidence such as frames, regions, entities, and visual texts, and $\mathcal{E}_\tau^w \subseteq \mathcal{W}$ contains retrieved webpages, images, or textual snippets. Therefore, VDR requires agents to link sparse cross-frame anchors with multimodal search and reason over video and web evidence.

Observation Space. At turn t, the agent observes the interaction history $\mathcal{H}_t$, where each $o_i \in \mathcal{O}$ is the compact multimodal output returned by the corresponding tool call. The observation space is defined as $\mathcal{O} = \mathcal{O}^v \cup \mathcal{O}^w$, where $\mathcal{O}^v$ contains video-grounded observations (e.g., selected frames or localized regions) and $\mathcal{O}^w$ contains web-grounded observations (e.g., retrieved images or webpages). After executing a_t, the returned observation o_t is appended to the history to form $\mathcal{H}_{t+1} = (\mathcal{H}_t, a_t, o_t)$.

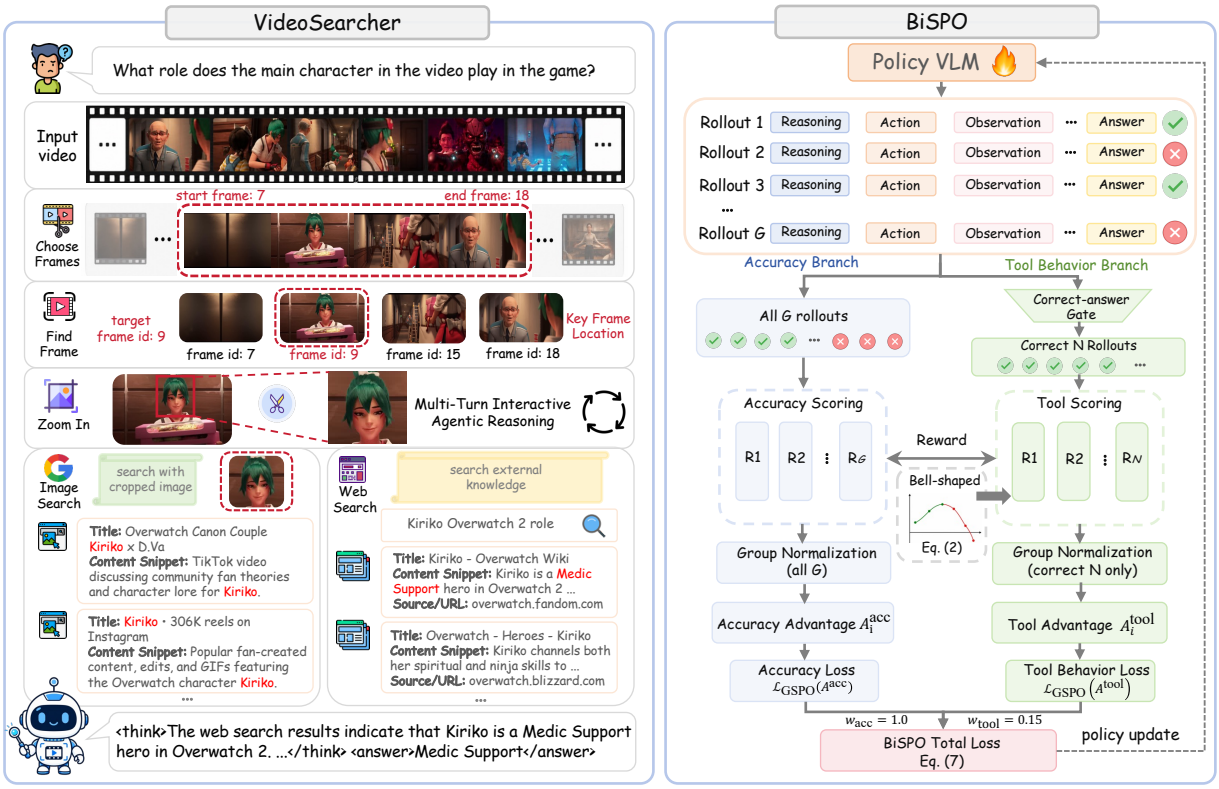


Figure 4. Overview of the VideoSearcher framework with BiSPO training. VideoSearcher performs multi-turn reasoning by selecting relevant frames, localizing key evidence, zooming into target regions, and invoking multimodal search tools in VDR. BiSPO optimizes the policy VLM with complementary accuracy and tool-behavior branches.

Action Space. At each turn, the agent first generates a reasoning state and then executes one valid action from $\mathcal{A}$:

1. `choose_frames`: Select a coarse temporal interval from the video.
2. `find_frame`: Localize a frame from the video.
3. `zoom_in`: Inspect a specified region of a frame.
4. `web_search`: Retrieve web evidence using a textual query.
5. `image_search`: Perform visual retrieval with a frame or cropped region.
6. `answer`: Produce the final prediction.

It is noted that validity constraints require spatial and image-search actions to be grounded in selected frames with meaningful bounding boxes, and each step to include both a reasoning state and a well-formed action. Non-terminal actions return observations that update the interaction history.

3.2 Data Construction

3.2.1 Video-Centric Data Pipeline

To equip VideoSearcher with video grounding and tool-augmented deep research abilities, we build an automated video-centric data pipeline, as shown in Fig. 3. The pipeline converts knowledge-intensive image-text QA samples into VDR instances, synthesizes multi-tool reasoning trajectories, and applies strict quality control before training. Detailed procedures are provided in Appendix A.1.

Entity-driven Video Acquisition. We start from knowledge-intensive image-text QA datasets, including FVQA [16], DeepEyes [24] and DeepEyesV2 [17]. For each QA pair, we extract the core visual entities required for answering (e.g., landmarks, logos and people). To improve reliability, we query Qwen3.5-397B [25] and Kimi-K2.5 [26], and retain only entities consistently identified by both models. These verified entities are then used as retrieval targets to collect relevant videos from online platforms.

Video Normalization and QA Alignment. We normalize collected videos by resampling them to 1 FPS and overlaying a "Frame N" marker as a temporal reference. We then rewrite the original image-text QA pairs to align with the video context. The visual-QA alignment verification is applied to discard noisy samples with missing key entities, answer leakage, direct OCR exposure of the answer, or inconsistent visual anchors.

Hard Sample Mining. To focus on challenging VDR instances, we evaluate each candidate with Qwen3-VL-8B-Instruct and retain samples that the model consistently fails to solve. The resulting hard samples emphasize temporal localization, fine-grained visual inspection, and external evidence acquisition. We reserve 3,285 video instances for RL online rollouts, while the remaining samples are used for SFT trajectory synthesis.

Hierarchical Trajectory Synthesis. For each retained sample, we synthesize multi-tool trajectories with two proprietary models. Gemini-3.1-Flash [27] performs coarse temporal screening via `choose_frames`, while Gemini-3.1-Pro [28] handles fine-grained localization, multimodal web retrieval, and evidence reasoning within a limited turn budget.

Data Quality Control. Strict quality control is applied before assembling the final SFT dataset. Heuristic filters remove trajectories with invalid tool invocations, malformed bounding boxes, and repetitive reasoning. Samples with hallucinated evidence or broken reasoning chains are further rejected by Kimi-K2.5. This process yields 3,811 high-quality SFT trajectories, which provide VideoSearcher with diverse tool-use patterns and basic VDR interaction capabilities.

3.2.2 VideoSearch-QA Construction

VideoDR [12] first formalizes VDR, but its evaluation remains largely text-centric, as visual cues are transformed into textual descriptions before web search. We introduce VideoSearch-QA (VSQA) as shown in Fig. 1(c) and Fig. 5, a manually annotated benchmark of 278 samples for evaluating multimodal video-grounded search, covering diverse domains such as landmarks, geography, culture, and recent 2026 events. For each video, we manually annotate fine-grained, search-oriented questions around key visual cues, enabling the evaluation of whether agents can capture visual

information in dynamic videos and use it for open-web search and evidence-based reasoning. More details can be found in the Appendix A.2.

3.3 Model Training

We train VideoSearcher in two stages: cold-start SFT initializes the VDR interaction protocol and basic tool-use behaviors, while online RL further improves the long-horizon evidence acquisition, tool coordination, and reasoning capabilities. The framework of VideoSearcher and the RL algorithm is shown in Fig. 4.

3.3.1 Cold-start SFT

Given the curated trajectory set $\mathcal{D}_{\text{SFT}} = \{(x_i, \tau_i^\star)\}_{i=1}^{M}$, where $x_i = (\mathcal{V}_i, q_i)$ denotes the question and $\tau_i^\star$ is a validated multi-turn trajectory, we perform supervised fine-tuning with $\mathcal{L}_{\text{SFT}} = -\sum_{i=1}^{M} \log \pi_\theta(\tau_i^\star \mid x_i)$. Each trajectory contains interleaved reasoning steps, tool calls, tool observations (excluded from the loss), and the final answer. This stage initializes the VDR interaction protocol and basic tool-use behaviors before RL.

3.3.2 RL with Bi-branch Sequence Policy Optimization

After cold-start SFT, we further optimize VideoSearcher with RL in the same VDR environment, and introduce Bi-branch Sequence Policy Optimization (BiSPO) on top of GSPO [18] to decouple answer optimization from tool-behavior optimization. For each prompt x, the old policy $\pi_{\theta_{\text{old}}}$ samples a group of G trajectories $\{\tau_i\}_{i=1}^{G}$, where each trajectory contains interleaved reasoning steps, tool interactions, observations, and a final answer. Let y_i denote the model-generated token sequence in τ_i, with tool observations used only as conditioning context.

Bi-branch Reward Design. Successful VDR trajectories should produce correct answers while using tools purposefully. To preserve effective tool-use signals and promote purposeful tool invocation in VDR, we define two reward branches. The accuracy branch evaluates answer correctness and protocol validity:

$$R_i^{\text{acc}} = R_i^{\text{judge}} + \lambda_f R_i^{\text{fmt}} + \lambda_d R_i^{\text{dep}}, \tag{1}$$

where R_i^{judge} is an LLM-judge score for final-answer correctness, R_i^{fmt} checks format compliance, and R_i^{dep} verifies correct tool-dependency constraints (e.g., selecting a frame before invoking region-level inspection or image search).

The tool branch is correctness-gated, so tool behavior is optimized only for trajectories that reach a correct answer:

$$R_i^{\text{tool}} = \mathbb{I}\left[R_i^{\text{judge}} > \delta\right] \cdot \text{clip}\left(b + \gamma\sqrt{T_i^{\text{vid}}} + \sum_{m\in\mathcal{M}} \sum_{k=1}^{U_{i,m}} \Delta_k,\ r_{\min},\ 1\right). \tag{2}$$

where T_i^{vid} counts valid video-grounding actions, $\mathcal{M} = \{\text{image}, \text{web}\}$ denotes search modalities, and $U_{i,m}$ is the number of unique successful searches under modality m. The marginal term Δ_k induces a bell-shaped search utility per modality: early unique useful searches increase the tool reward, whereas excessive ones receive negative marginal rewards. Failed, empty, or duplicated calls receive zero marginal reward and do not advance the schedule. This design encourages sufficient tool use in knowledge-intensive VDR while suppressing excessive or uninformative tool invocations. See Appendix B for details.

Decoupled Advantage Estimation. For the accuracy branch, we compute group-relative advantages over all rollouts:

$$A_i^{\text{acc}} = \frac{R_i^{\text{acc}} - \mu^{\text{acc}}}{\sigma^{\text{acc}} + \epsilon}, \tag{3}$$

where μ^{acc} and σ^{acc} are the mean and standard deviation of $\{R_j^{\text{acc}}\}_{j=1}^{G}$. While for the tool branch, comparison is restricted to correct trajectories. Let $\mathcal{Q} = \{j \mid R_j^{\text{judge}} > \delta\}$. The tool advantage is:

$$A_i^{\text{tool}} = \mathbf{1}\{i \in \mathcal{Q},\ |\mathcal{Q}| \geq 2\} \frac{R_i^{\text{tool}} - \mu_{\mathcal{Q}}^{\text{tool}}}{\sigma_{\mathcal{Q}}^{\text{tool}} + \epsilon}. \tag{4}$$

where $\mu_{\mathcal{Q}}^{\text{tool}}$ and $\sigma_{\mathcal{Q}}^{\text{tool}}$ are computed over $\{R_k^{\text{tool}}\}_{k \in \mathcal{Q}}$. This avoids comparing tool behaviors from incorrect rollouts with evidence-seeking behaviors that actually support correct answers. When fewer than two rollouts are correct, we skip tool-behavior optimization and update only the accuracy branch. Once multiple correct trajectories emerge, the tool branch compares them to favor more effective tool use.

BiSPO Objective. Long VDR trajectories contain many reasoning tokens and tool-conditioned contexts, making token-level clipping unstable. We therefore build BiSPO on sequence-level importance sampling. For trajectory i, the sequence importance ratio is:

$$s_i(\theta) = \exp\big(\bar{\ell}_i(\theta)\big), \quad \bar{\ell}_i(\theta) = \frac{1}{|y_i|} \sum_t \big[\log \pi_\theta(y_{i,t} \mid x_i, y_{i,<t}) - \log \pi_{\theta_{\text{old}}}(y_{i,t} \mid x_i, y_{i,<t})\big]. \tag{5}$$

Given an advantage A, the clipped sequence-level surrogate objective is:

$$\mathcal{L}_{\text{GSPO}}(A) = -\mathbb{E}_i[\min(s_i(\theta)A_i,\ \tilde{s}_i(\theta)A_i)], \quad \tilde{s}_i(\theta) = \text{clip}\big(s_i(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\big). \tag{6}$$

The final BiSPO objective combines the two branches only at the loss level:

$$\mathcal{L}_{\text{BiSPO}} = w_{\text{acc}}\mathcal{L}_{\text{GSPO}}(A^{\text{acc}}) + w_{\text{tool}}\mathcal{L}_{\text{GSPO}}\Big(A^{\text{tool}}\Big). \tag{7}$$

We set $w_{\text{acc}} = 1.0$ and $w_{\text{tool}} = 0.15$. During RL, the vision tower is frozen and the actor is optimized with sequence-level aggregation. By separating answer correctness from tool behavior, BiSPO preserves the primary task objective while providing a dedicated learning signal for efficient temporal grounding, multimodal retrieval, and evidence-aware stopping for VDR.

4 Experiment

Model	VideoDR	VSQA	MMSearch	HR-MMSearch	FVQA-test	InfoSeek	SimpleVQA	LiveVQA	Avg.
Direct Answer									
Open-source									
Qwen3-VL-4B-Instruct	8.00	12.59	13.45	3.61	25.33	25.75	46.89	23.35	19.87
Qwen2.5-VL-7B-Instruct	5.00	9.35	7.60	0.58	26.28	31.95	47.88	19.63	18.53
Qwen3-VL-8B-Instruct	8.00	16.91	11.70	12.13	24.22	23.15	42.94	23.18	20.28
Qwen2.5-VL-32B-Instruct	16.00	13.67	11.70	3.93	30.50	36.65	48.57	21.40	22.80
Qwen3-VL-32B-Instruct	16.00	20.86	16.96	19.02	32.17	28.95	45.90	31.59	26.43
Proprietary									
GPT-4o	35.00	36.33	23.39	13.11	48.00	52.90	51.73	28.18	36.08
Gemini-3-Flash	57.00	56.12	57.31	21.97	56.50	54.85	63.57	38.90	50.78
GPT-5.2	40.00	38.85	43.27	24.92	50.94	50.40	59.92	47.00	44.41
Gemini-3-Pro	61.00	54.32	62.57	26.89	59.22	56.30	64.07	40.06	53.05
Agentic Model (zero-shot)									
Open-source									
Qwen3-VL-4B-Instruct	25.00	29.14	49.12	24.92	31.72	28.10	41.95	26.15	32.01
Qwen2.5-VL-7B-Instruct	7.00	11.15	32.16	19.34	36.00	28.80	42.35	22.52	24.92
Qwen3-VL-8B-Instruct	28.00	30.94	47.37	27.87	53.61	46.15	62.29	39.37	41.95
Qwen2.5-VL-32B-Instruct	30.00	29.14	49.71	33.44	52.22	50.10	65.15	42.17	43.99
Qwen3-VL-32B-Instruct	36.00	37.77	49.12	34.43	54.28	49.85	64.17	42.87	46.06
Proprietary									
GPT-4o	48.00	49.64	49.12	30.16	66.34	59.55	63.67	40.09	50.82
Gemini-3-Flash	60.00	58.99	62.57	41.64	64.89	61.10	67.92	48.06	58.15
GPT-5.2	58.00	57.19	66.08	48.20	68.78	65.55	78.18	65.99	63.50
Gemini-3-Pro	77.00	65.11	74.27	48.52	72.61	66.45	75.91	59.69	67.45
Agentic Model									
Visual-ARFT	5.00	8.99	34.50	24.92	41.72	37.95	42.45	25.40	27.62
Webwatcher-7B	37.00	34.53	49.10	26.89	58.17	56.90	54.30	51.20	46.01
MMSearch-R1	5.00	11.87	53.80	20.33	58.40	55.10	57.40	48.40	38.79
DeepEyesV2	19.00	21.58	63.70	14.10	60.60	51.10	59.40	24.43	39.24
VideoSearcher-4B	46.00	49.28	66.08	39.67	55.17	49.25	65.84	50.72	52.75
Δ v.s. Qwen3-VL-4B-Instruct	**+21.00**	**+20.14**	**+16.96**	**+14.75**	**+23.45**	**+21.15**	**+23.89**	**+24.57**	**+20.74**
VideoSearcher-8B	**53.00**	**51.80**	**67.84**	**43.61**	**64.10**	**58.30**	**67.20**	**55.40**	**57.66**
Δ v.s. Qwen3-VL-8B-Instruct	**+25.00**	**+20.86**	**+20.47**	**+15.74**	**+10.49**	**+12.15**	**+4.91**	**+16.03**	**+15.71**

Table 1. Performance on search-oriented benchmarks under Direct Answer and Agentic Model workflows. Zero-shot denotes tool-augmented inference without task-specific training. Scores are averaged over benchmark samples.

4.1 Implementation Details

Model and Training. VideoSearcher is initialized from Qwen3-VL-8B-Instruct and trained with a two-stage pipeline, using LLaMA-Factory [29] for SFT and veRL [30] for online RL. In SFT, we freeze the vision encoder and multi-modal projector and fine-tune only the language model with a learning rate of 1×10^{-5}. The RL stage starts from the SFT checkpoint and is trained for one epoch on 16 H20 GPUs, with the vision tower kept frozen. We use a prompt batch size of 64, sample 4 rollouts per prompt, and set the actor learning rate to 1×10^{-5}. More implementation details about hyperparameter setting are provided in the Appendix B.

Benchmarks. To comprehensively evaluate our proposed VideoSearcher, we consider three categories of benchmarks. For Video Deep Research, we use VideoDR [12] and our newly constructed VideoSearch-QA (VSQA). For multimodal search, we evaluate on MMSearch [31], HR-MMSearch [32], FVQA-test [16], InfoSeek [33], SimpleVQA [34], and LiveVQA [35]. For general video understanding, we further test on MMVU [36], TempCompass [37], VideoMMMU [38], and

VideoMathQA [39]. Additional benchmark details are provided in the Appendix D.

Baselines. We compare VideoSearcher with a broad set of advanced baseline models. Proprietary baselines include GPT-4o [40], GPT-5.2 [41], and Gemini-3-Flash/Pro [42, 43], while the powerful open-source VLMs include the Qwen2.5-VL [44] and Qwen3-VL [4] series. We further compare with multimodal search-oriented agentic models, including Visual-ARFT [45], Webwatcher [21], MMSearch-R1 [16], and DeepEyesV2 [17], as well as the video reasoning models, including Video-R1 [7], VideoChat-R1.5 [46], VideoRFT [19], and VideoCoM [47]. For search-oriented benchmarks, models are evaluated under two settings: (1) *Direct Answer*, where the model answers directly without using external tools, and (2) *Agentic Model*, where the model is provided with the available tools and autonomously decides how to invoke them during rollout reasoning. For video understanding tasks, baselines use uniformly sampled frames in the Direct Answer setting, while VideoSearcher is evaluated as an Agentic Model.

4.2 Main Results

Search-oriented Benchmarks. As shown in Tab. 1, our method achieves strong performance across search-oriented evaluations. On VideoDR and VSQA, VideoSearcher substantially improves over existing agentic baselines, demonstrating its effectiveness for VDR scenarios that require joint video grounding, tool use, and open-web evidence reasoning. Notably, although trained only on video-based data, VideoSearcher also generalizes well to general multimodal search benchmarks, substantially outperforming Qwen3-VL agentic baselines.

Model	MMVU	TempCompass	VideoMMMU	VideoMathQA	Avg.
Qwen3-VL-4B-Instruct	63.36	70.95	60.78	22.62	54.43
Qwen3-VL-8B-Instruct	<u>69.28</u>	<u>73.62</u>	<u>65.67</u>	26.90	58.87
Video-R1	63.80	73.20	52.40	23.30	53.18
VideoChat-R1.5	62.08	72.30	51.40	25.70	52.87
VideoRFT	68.50	**73.70**	51.10	25.20	54.63
VideoCoM	65.40	71.30	50.20	27.80	53.68
VideoSearcher-4B	68.64	70.27	65.33	<u>27.86</u>	<u>58.03</u>
VideoSearcher-8B	**70.40**	73.14	**66.33**	**31.67**	**60.39**

Table 2. Performance on general video understanding benchmarks.

General Video Understanding. Tab. 2 further evaluates VideoSearcher on general video understanding benchmarks. Despite being designed as a search-enabled agent, where retrieved evidence may introduce distracting or irrelevant information, VideoSearcher maintains competitive general video reasoning ability. The experimental results indicate that the proposed agentic training does not merely optimize search behavior, but also preserves and improves transferable video understanding under more complex evidence conditions.

Method	Setting	VideoDR	VSQA	MMSearch	HR-MMSearch
SFT	SFT only	46.00	47.84	58.48	40.98
GRPO	R_{acc} only	45.00	48.16	65.50	41.14
GSPO	R_{acc} only	47.00	50.72	63.16	40.66
GSPO	$R_{\text{acc}} + 0.15R_{\text{tool}-\text{mono}}$	47.00	51.08	65.50	40.33
BiSPO	$R_{\text{acc}} + 0.15R_{\text{tool}-\text{mono}}$	50.00	51.44	65.50	40.33
BiSPO	**Full**	**53.00**	**51.80**	**67.84**	**43.61**

Table 3. Algorithmic ablation of RL. GRPO and GSPO are single-branch RL variants; and BiSPO is the proposed bi-branch variant, with BiSPO Full using the correctness-gated bell-shaped tool reward.

4.3 Ablation Study

RL Ablation Study. Tab. 3 ablates the RL optimization design. With only the accuracy reward R_{acc}, GSPO improves over GRPO on VideoDR and VSQA, reflecting the benefit of sequence-level optimization for long VDR trajectories. However, its mixed VDR gains suggest that answer-level supervision alone does not reliably induce tool use. We further compare a monotonic tool reward, $R_{\text{tool-mono}}$, which rewards every valid tool call. Our full reward instead follows a bell-shaped design, encouraging sufficient grounding and retrieval while penalizing unnecessary calls. By decoupling R_{acc} from the tool reward, BiSPO optimizes tool use in reasoning-intensive VDR tasks, improving VideoDR from 47.00% to 50.00% with $R_{\text{tool-mono}}$ and achieving the best overall results with the full bell reward. These results show that decoupled optimization and bell-shaped tool reward are both important for stable and effective VDR training.

Locate	Search	VideoDR	VSQA	MMSearch	HR-MMSearch
✓		12.00	19.78	12.28	3.93
	✓	43.00	47.48	66.67	40.98
✓	✓	**53.00**	**51.80**	**67.84**	**43.61**

Table 4. Tool ablation study. Locate denotes the video localization tools and Search denotes the external web retrieval tools.

Tool Ablation Study. Tab. 4 ablates the localization and search tools. Localization alone performs poorly, indicating that grounded visual evidence is insufficient for open-world questions without external knowledge. Search alone also degrades markedly, especially when retrieval requires prior grounding in video-centric benchmarks. These results show that the two tools are complementary: localization anchors the query in visual evidence, while search augments it with external information for knowledge-intensive reasoning.

5 Conclusion

In this work, we study Video Deep Research (VDR), which extends video understanding from closed-context perception to open-world information exploration. We introduce VideoSearcher, a closed-loop agentic framework for temporal grounding, spatial inspection, and multimodal

retrieval over dynamic videos. We further develop a video-centric data pipeline for synthesizing high-quality multi-tool trajectories and propose BiSPO, a dual-branch RL algorithm that decouples answer accuracy from tool-invocation optimization for knowledge-intensive VDR. Finally, we introduce VideoSearch-QA, a benchmark for evaluating video information grounding and multimodal search-based reasoning in open-world scenarios. Extensive experiments across multiple benchmarks demonstrate the effectiveness of VideoSearcher.

Limitations

Although VideoSearcher improves search-oriented video question answering, several limitations remain. It relies on an external tool environment, so answer quality depends on the latency, coverage, and stability of image/web search, making reproduction less deterministic than closed-book video QA. Online RL with multi-turn tool use is also computationally expensive, as long contexts and repeated rollouts increase both training cost and engineering complexity. In addition, our training and evaluation mainly focus on search-oriented visual/video QA, and may not fully cover tasks requiring dense temporal localization, audio understanding, embodied interaction, specialized domain knowledge, or short-event motion cues under the 1-fps frame representation. Finally, open-ended evaluation relies on an LLM judge, which can introduce noise for ambiguous or semantically equivalent answers; stronger human evaluation and calibrated automatic metrics remain useful future directions.

References

[1] Vidi Team, Celong Liu, Chia-Wen Kuo, Dawei Du, Fan Chen, Guang Chen, Jiamin Yuan, Lingxi Zhang, Lu Guo, Lusha Li, et al. Vidi: Large multimodal models for video understanding and editing. *arXiv preprint arXiv:2504.15681*, 2025.

[2] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118, 2025.

[3] Munan Ning, Bin Zhu, Yujia Xie, Bin Lin, Jiaxi Cui, Lu Yuan, Dongdong Chen, and Li Yuan. Video-bench: A comprehensive benchmark and toolkit for evaluating video-based large language models. *Computational Visual Media*, 2025.

[4] 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.

[5] 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.

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

[7] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *Advances in Neural Information Processing Systems*, 38:99114–99137, 2026.

[8] Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, et al. Scaling rl to long videos. *Advances in Neural Information Processing Systems*, 38:172842–172870, 2026.

[9] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, et al. Video understanding with large language models: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

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

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

[12] Chengwen Liu, Xiaomin Yu, Zhuoyue Chang, Zhe Huang, Shuo Zhang, Heng Lian, Kunyi Wang, 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*, 2026.

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

[14] Kartik Narayan, Yang Xu, Tian Cao, Kavya Nerella, Vishal M Patel, Navid Shiee, Peter Grasch, Chao Jia, Yinfei Yang, and Zhe Gan. Deepmmsearch-r1: Empowering multimodal llms in multimodal web search. *arXiv preprint arXiv:2510.12801*, 2025.

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

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

[17] Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

[18] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[19] Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. Videorft: Incentivizing video reasoning capability in mllms via reinforced fine-tuning. *Advances in neural information processing systems*, 38:4350–4376, 2026.

[20] Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, et al. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025.

[21] Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025.

[22] Runqi Qiao, Qiuna Tan, Minghan Yang, Guanting Dong, Peiqing Yang, Shiqiang Lang, Enhui Wan, Xiaowan Wang, Yida Xu, Lan Yang, et al. V-thinker: Interactive thinking with images. *arXiv preprint arXiv:2511.04460*, 2025.

[23] 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.

[24] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

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

[26] Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

[27] Gemini Team. Gemini-3.1-flash-lite, March 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-flash-lite/.

[28] Gemini Team. Gemini-3.1-pro, February 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/.

[29] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pages 400–410, 2024.

[30] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

[31] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, et al. Mmsearch: Benchmarking the potential of large models as multi-modal search engines. *arXiv preprint arXiv:2409.12959*, 2024.

[32] Yong Xien Chng, Tao Hu, Wenwen Tong, Xueheng Li, Jiandong Chen, Haojia Yu, Jiefan Lu, Hewei Guo, Hanming Deng, Chengjun Xie, et al. Sensenova-mars: Empowering multimodal agentic reasoning and search via reinforcement learning. *arXiv preprint arXiv:2512.24330*, 2025.

[33] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14948–14968, 2023.

[34] Xianfu Cheng, Wei Zhang, Shiwei Zhang, Jian Yang, Xiangyuan Guan, Xianjie Wu, Xiang Li, Ge Zhang, Jiaheng Liu, Yuying Mai, et al. Simplevqa: Multimodal factuality evaluation for multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4637–4646, 2025.

[35] Mingyang Fu, Yuyang Peng, Benlin Liu, Yao Wan, and Dongping Chen. Livevqa: Live visual knowledge seeking. *arXiv:2504.05288*, 2025.

[36] Yilun Zhao, Haowei Zhang, Lujing Xie, Tongyan Hu, Guo Gan, Yitao Long, Zhiyuan Hu, Weiyuan Chen, Chuhan Li, Zhijian Xu, et al. Mmvu: Measuring expert-level multi-discipline video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8475–8489, 2025.

[37] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8731–8772, 2024.

[38] Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*, 2025.

[39] Hanoona Rasheed, Abdelrahman Shaker, Anqi Tang, Muhammad Maaz, Ming-Hsuan Yang, Salman Khan, and Fahad Shahbaz Khan. Videomathqa: Benchmarking mathematical reasoning via multimodal understanding in videos. *arXiv preprint arXiv:2506.05349*, 2025.

[40] 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.

[41] 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.

[42] Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. A new era of intelligence with gemini 3. *Google. URL: https://blog. google/products-and-platforms/products/gemini/gemini*, 3, 2025.

[43] Google DeepMind. Gemini 3 pro model card, May 2026. URL https://deepmind.google/models/model-cards/gemini-3-pro.

[44] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[45] Ziyu Liu, Yuhang Zang, Yushan Zou, Zijian Liang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual agentic reinforcement fine-tuning. *arXiv preprint arXiv:2505.14246*, 2025.

[46] Ziang Yan, Yinan He, Xinhao Li, Zhengrong Yue, Xiangyu Zeng, Yali Wang, Yu Qiao, Limin Wang, and Yi Wang. Videochat-r1. 5: Visual test-time scaling to reinforce multimodal reasoning by iterative perception. *Advances in Neural Information Processing Systems*, 38:119152–119184, 2026.

[47] Hanoona Rasheed, Mohammed Zumri, Muhammad Maaz, Ming-Hsuan Yang, Fahad Shahbaz Khan, and Salman Khan. Video-com: Interactive video reasoning via chain of manipulations. *arXiv preprint arXiv:2511.23477*, 2025.

Appendix

A Data Pipeline

A.1 Training Data Construction

We provide more details on the construction of the training data used for cold-start SFT and online RL. As illustrated in Fig. 3, our goal is to convert knowledge-intensive image–text QA samples into video-centric deep-research instances, where the model must ground visual clues in a video, invoke appropriate tools, retrieve external evidence, and synthesize a final answer.

Seed QA Collection. We begin with knowledge-intensive image–text QA data from existing multimodal search and visual information-seeking datasets. These samples are suitable seeds because their answers usually cannot be obtained from visual perception alone, but require external knowledge from web pages or image search. In addition to existing datasets, we also include human-annotated seed questions to increase coverage over diverse entities and search scenarios. For each seed sample, we retain the question, answer, reference image, and the key visual entity that supports the answer.

Entity-driven Video Acquisition. Since the original seed samples are static image–text pairs, we first identify the core visual entity that should appear in a video, such as a landmark, logo, product, person, scene, or event-related object. To reduce entity ambiguity, we query two strong multimodal models, Qwen3.5-397B [25] and Kimi-K2.5 [26], and only keep entities that are consistently recognized by both models. The verified entities are then used as retrieval targets to collect relevant videos from multiple platforms, including web video sources such as YouTube and Bilibili. This produces an initial pool of video-QA candidates grounded in real video content.

Video Normalization and QA-alignment. Collected videos are normalized into a unified format before trajectory synthesis. We resample each video into frame sequences and add explicit frame indices as temporal references, enabling subsequent tools to localize and refer to specific moments. We then rewrite the original image-based questions into video-grounded questions. During rewriting, we ensure that the question naturally refers to the video context rather than the original image, and we add discriminative visual or temporal descriptions when multiple similar entities appear in the same video. We further apply visual-QA alignment verification to remove samples where the target entity is missing, the answer is directly exposed by OCR, the video contains inconsistent visual anchors, or the question can be answered without using the video evidence.

Hard Sample Mining. To avoid constructing a training set dominated by visually obvious or shallow-search examples, we perform hard sample mining with Qwen3-VL-8B-Instruct. For each candidate video-QA pair, the model is rolled out multiple times under the agentic setting. We retain samples that the model fails to solve consistently, following the criterion that the number of successful rollouts is no more than one. This step encourages the final data to focus on challenging cases that require temporal localization, fine-grained visual inspection, multimodal retrieval, and evidence-grounded reasoning. The resulting samples are then split into two branches: 3,285 video instances are reserved for online RL rollouts, while the remaining samples are used for SFT trajectory synthesis. **Hierarchical Trajectory Synthesis.** For each SFT sample, we synthesize a complete multi-tool reasoning trajectory with a hierarchical teacher pipeline. Gemini-3.1-Flash [27] is first used for coarse temporal screening through `choose_frames`, which narrows the video to a smaller set of potentially relevant frames. Gemini-3.1-Pro [28] then performs fine-grained localization and reasoning, including `find_frame`, optional `zoom_in`, `image_search`, and `web_search`. The generated trajectory records the complete interaction process, including intermediate reasoning states, tool calls, tool observations, retrieved evidence, and the final answer. To reduce visual token cost, we follow an incremental visual-context strategy: newly acquired visual observations are introduced at each turn, while the full textual reasoning history is preserved.

Quality Control. We apply strict filtering before adding synthesized trajectories to the cold-start dataset. Rule-based filters remove trajectories with malformed tool calls, invalid frame references, illegal bounding boxes, failed searches, repeated queries, or repetitive reasoning patterns. We further use Kimi-K2.5 as a quality verifier to reject trajectories with hallucinated evidence, unsupported conclusions, inconsistent tool

observations, or broken reasoning chains. After filtering, we obtain 3,811 high-quality SFT trajectories. These trajectories provide diverse demonstrations of video localization, spatial inspection, multimodal search, and evidence synthesis, thereby initializing the interaction protocol and basic tool-use behaviors of VideoSearcher before RL.

A.2 VideoSearch-QA Benchmark Construction

We construct VideoSearch-QA (VSQA) as a human-centered benchmark for evaluating video-grounded multimodal search. Unlike existing VDR evaluation that often converts visual cues into text before retrieval, VSQA is designed to test whether an agent can directly identify visual anchors from dynamic videos, use these anchors to perform web or image search, and reason over multimodal evidence. As shown in Tab. 5, the final benchmark contains 278 manually annotated samples across diverse domains, including landmarks, geography, natural scenes, culture, sports and gaming, leisure, technology, and business. Fig. 5 illustrates the VSQA construction pipeline.

Video Collection and Filtering. We collect candidate videos from public web video platforms and online sources, prioritizing videos that are informative, dynamic, multi-entity, and temporally grounded. In particular, we favor videos containing recognizable and searchable visual anchors, such as landmarks, logos, signs, products, public figures, maps, or event-specific scenes. We also include recent or real-time videos, especially from 2026 events, to reduce potential memorization from model pretraining. Human annotators then screen the collected videos and remove cases that are visually ambiguous, low-resolution, dominated by irrelevant content, or unsuitable for open-web retrieval. The retained videos must contain at least one stable visual anchor, require localized video grounding, and cannot be answered solely from the video or solely from web evidence without using video cues.

Human Annotation and Verification. For each retained video, expert annotators manually construct the question-answer pair based on key visual cues in the video. The question is designed to be grounded in concrete video evidence, such as a visible object, text region, location, person, logo, or event clue, while requiring multimodal retrieval through web search or image search to obtain the final answer. Annotators also provide the reference answer and verify it with reliable external evidence. A separate human checking stage further reviews each video-question-answer triple in terms of visual-QA alignment, difficulty, multi-hop reasoning, and multimodal retrieval necessity. Samples with ambiguous visual anchors, recognition-only answers, unsupported answers, or insufficient need for external retrieval are revised or discarded.

Domain	# Samples	Percentage
Landmarks	65	23.4%
Geography	51	18.3%
Natural Scenes	31	11.2%
Culture	47	16.9%
Sports & Games	27	9.7%
Leisure	25	9.0%
Technology	18	6.5%
Business	14	5.0%
Summary	**278**	**100.0%**

Table 5. Data distribution of the VideoSearch-QA benchmark.

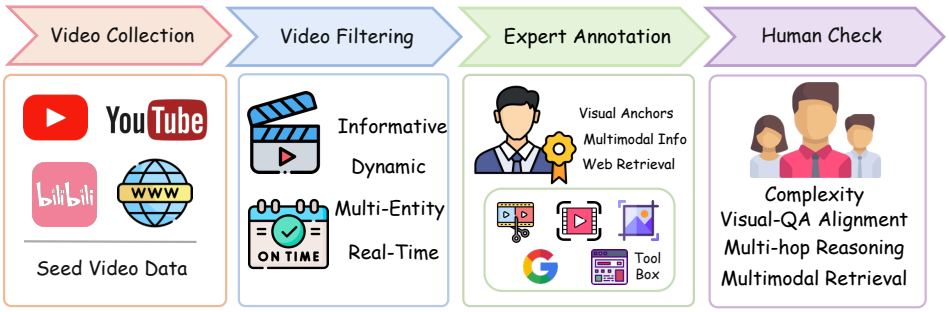


Figure 5. VideoSearch-QA Benchmark Construction Pipeline

B Implementation Details

B.1 Training Setting

VideoSearcher is trained with a two-stage pipeline. The first stage is cold-start supervised fine-tuning on curated multi-turn Video Deep Research trajectories. These trajectories follow the same interaction format used during inference, including intermediate reasoning, tool invocation, tool observation, and final answer generation. During SFT, we freeze the vision encoder and multi-modal projector and update only the language model.

The second stage is online reinforcement learning in the tool-interactive environment. The RL stage is initialized from the SFT checkpoint and trained for one epoch on two nodes with 16 H20 GPUs. The model can interact with five tools: `choose_frames`, `find_frame`, `zoom_in`, `image_search`, and `web_search`. During RL, the vision tower remains frozen to stabilize long-horizon multi-turn training.

B.2 Hyperparameter Setting

For SFT, we use bf16 precision and a learning rate of 1×10^{-5}. The vision encoder and multi-modal projector are frozen, while the language model is optimized.

For RL, we use a prompt batch size of 64 and sample 4 rollouts for each prompt. The actor learning rate is 1×10^{-5}. The maximum response length is 16,384 tokens, and the maximum context length is 49,152 tokens. The maximum number of assistant turns is 12. We use asymmetric clipping with $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$. The answer-correctness branch has weight 1.0, and the tool-behavior branch has weight 0.15. We do not use an additional KL reward or KL loss during RL.

For the reward design, we instantiate the answer-correctness branch by absorbing the coefficients in Eq. (1) into the corresponding terms:

$$R_i^{\text{acc}} = R_i^{\text{judge}} + R_i^{\text{fmt}} + R_i^{\text{dep}}.$$

where $R_i^{\text{judge}} \in \{0, 1\}$ is given by the LLM judge, $R_i^{\text{fmt}} = 0.5$ if the trajectory satisfies the required output format and has no model-caused tool error, and 0 otherwise. The dependency term penalizes invalid tool ordering as $R_i^{\text{dep}} = -\min(0.1N_i^{\text{dep}}, 0.5)$, where N_i^{dep} is the number of detail/search tool calls made without a valid locked frame. The correctness gate threshold for the tool branch is $\delta = 0$.

For the tool-behavior branch, non-search video-grounding actions use $b = 0.3$ and $\gamma = 0.35$ in the $b + \gamma\sqrt{T_i^{\text{vid}}}$ term. For search-tool shaping, image search and web search are counted separately. For each modality, the marginal reward for the k-th unique useful search call is $[0.35, 0.20, 0.10, -0.10, -0.25]$ for $k = 1, \ldots, 5$, and -0.60 for every subsequent unique useful call. Failed calls, empty image-search results, duplicated text queries, and repeated visual regions receive zero marginal reward and do not advance the count. The final tool score is clipped to $[-1.0, 1.0]$.

B.3 Evaluation Setting

Evaluation is conducted in a tool-interactive environment consistent with the capabilities required by each benchmark. For VideoDR and VSQA, which require video grounding and open-world evidence retrieval, the model is allowed to use all five tools: `choose_frames`, `find_frame`, `zoom_in`, `image_search`, and `web_search`. For multimodal search benchmarks, where the input is a static image rather than a video, we provide the image-level tools `zoom_in`, `image_search`, and `web_search`. For general video understanding benchmarks, we keep the same five-tool environment as VDR to evaluate whether the trained agent can generalize beyond search-oriented tasks. During evaluation, we use temperature 0.7, top-$p = 0.8$, and top-$k = 20$ for decoding, with a maximum of 12 assistant turns. We report average scores over all samples in each benchmark.

For open-ended benchmarks, we use Qwen3.5-27B as the automatic judge. This choice balances judgment quality and evaluation throughput: the model provides strong instruction-following ability while remaining efficient to serve as a dense 27B judge, which is important because each open-ended evaluation requires many independent judge calls. For multiple-choice benchmarks, we use exact-match accuracy.

C Baseline Models

C.1 Open-source Models

We compare our method against a broad set of open-source VLMs and multimodal agentic models:

- **Qwen-VL Series.** We evaluate Qwen2.5-VL [44] and Qwen3-VL [4] models as strong open-source generalist VLMs. Qwen2.5-VL supports dynamic-resolution visual encoding and temporal localization for image/video understanding, while Qwen3-VL further improves long-context multimodal reasoning, interleaved image-video-text understanding, and spatial-temporal modeling.
- **Visual-ARFT.** Visual-ARFT [45] improves VLMs with agentic reinforcement fine-tuning. It equips models with external tool-use abilities, such as web browsing and image manipulation, making it a relevant baseline for visual agentic reasoning.
- **WebWatcher.** WebWatcher [21] is a multimodal deep-research agent trained with synthetic trajectories and reinforcement learning. It focuses on visual-textual information seeking with dynamic tool use, serving as a strong open-web research baseline.
- **MMSearch-R1.** MMSearch-R1 [16] is an RL-based multimodal search agent that learns when and how to invoke text and image search tools. It is designed for on-demand, multi-turn search in real-world Internet environments.
- **DeepEyesV2.** DeepEyesV2 [17] studies how to build agentic multimodal models with external tools such as web search and code execution. It emphasizes interleaved multimodal reasoning and tool-integrated problem solving.
- **Video-R1.** Video-R1 [7] adapts the R1-style reinforcement learning paradigm to video reasoning. It introduces temporal-aware RL training to improve spatial, temporal, and logical reasoning over videos.
- **VideoChat-R1.5.** VideoChat-R1.5 [46] introduces visual test-time scaling with iterative perception. It progressively refines spatio-temporal attention during inference to strengthen multimodal video reasoning.
- **VideoRFT.** VideoRFT [19] extends reinforced fine-tuning to video reasoning through SFT on video CoT data followed by RL. It further uses semantic-consistency rewards to align textual reasoning with visual evidence.
- **VideoCoM.** VideoCoM [47] proposes interactive video reasoning via a chain of tool-augmented manipulations. It treats video as an active reasoning workspace, where models iteratively manipulate and inspect visual evidence.

C.2 Proprietary Models

We also include several powerful proprietary models for evaluation:

- **GPT-4o.** GPT-4o [40] is a proprietary multimodal model with strong visual understanding, instruction following, and general reasoning ability.
- **GPT-5.2.** GPT-5.2 [41] is an advanced proprietary multimodal model with strong reasoning and tool-use capabilities. We evaluate it as a competitive closed-source baseline.
- **Gemini-3-Flash/Pro.** Gemini-3-Flash and Gemini-3-Pro [42] are proprietary multimodal models from the Gemini family. Gemini-3-Flash emphasizes efficient inference, while Gemini-3-Pro provides stronger reasoning and multimodal understanding.

D Evaluation Benchmarks

D.1 Search-oriented Benchmarks

For evaluating Video Deep Research and multimodal image-text search capabilities, we primarily adopt the following benchmarks:

- **VideoDR.** VideoDR [12] is an open-domain video deep research benchmark that evaluates whether agents can combine video-grounded clues with open-web evidence. The full benchmark contains 500 samples (released on 2026.05.19) across six domains: Daily Life, Technology, Culture, History, Economics, and Geography. Each question requires extracting multi-frame visual anchors, performing interactive web search, and synthesizing video–web evidence for a verifiable answer. In our evaluation, we use the 100-sample version released on 2026.01.14 from the official repository, which additionally provides Category and Difficulty labels. Overall, VideoDR evaluates the agentic ability of multimodal models to connect video-grounded clues with external web evidence for open-domain factual reasoning.
- **VideoSearch-QA (VSQA).** VideoSearch-QA is our manually annotated benchmark for evaluating video-grounded multimodal search. It contains 278 expert-annotated video–question pairs across eight domains, including Landmarks, Geography, Natural Scenes, Culture, Sports & Gaming, Leisure, Technology, and Business. Each question is grounded in key visual cues from the video, such as objects, text regions, locations, logos, people, or event-specific clues, and requires external web or image search to obtain the final answer. VSQA evaluates whether agents can localize relevant visual evidence in dynamic videos, transform video-grounded clues into effective retrieval queries, and synthesize multimodal evidence for open-world factual reasoning.
- **MMSearch.** MMSearch [31] consists of 300 manually curated examples spanning 14 subdomains, organized into two parts: News and Knowledge. The News split is designed around events after August 2024, reducing potential contamination from model pretraining, whereas the Knowledge split emphasizes obscure or long-tail facts that remain difficult even for strong models. Following MMSearch-R1, we restrict our evaluation to the 171 image-based questions and remove text-only samples, so that the benchmark better reflects visually grounded information-seeking scenarios.
- **HR-MMSearch.** HR-MMSearch [32] contains 305 image–question pairs, built from 4K-resolution news images from 2025 to reduce potential overlap with pre-training data and provide rich visual details. The dataset spans eight diverse domains: Sports, Entertainment & Culture, Science & Technology, Business & Finance, Games, Academic Research, Geography & Travel, and Others. HR-MMSearch contains 188 Hard and 117 Easy samples. Each question is knowledge-intensive and grounded in key visual evidence, often involving small objects or text regions that occupy less than 5% of the image. Solving these questions requires the use of at least one multimodal tool, including image search, text search, or image cropping. Overall, HR-MMSearch provides a challenging and diverse testbed for evaluating fine-grained visual understanding and agentic search in VLM agents.
- **FVQA-test.** FVQA-test [16] is a multimodal evaluation set designed to cover both visual and textual knowledge domains. It contains 1,800 high-quality examples from three sources. Specifically, 600 examples are drawn from FVQA-auto-vc after accuracy verification and training-data separation, 600 examples come from the InfoSeek Human Split with manually corrected answers, and the remaining 600 examples are newly annotated by humans to further broaden the benchmark coverage.
- **InfoSeek.** InfoSeek [33] is a real-world knowledge retrieval benchmark built from Wikidata triples. Its questions are generated by converting structured triples into natural-language queries with human-

designed templates covering 300 relations, where unit and entity-type placeholders are incorporated to improve question clarity. The dataset is further refined by removing unanswerable questions and balancing samples across entities. In our setting, the evaluation subset contains 2,000 instances sampled from the test split, covering a diverse range of factual queries.

- **SimpleVQA.** SimpleVQA [34] is a factual VQA benchmark designed to evaluate objective real-world knowledge. It consists of two types of samples: image–question pairs collected from post-2023 VQA datasets, and newly constructed examples created by experts based on internet search results. All samples are filtered for difficulty and quality to ensure reliable factual evaluation. The evaluation subset contains 1,013 English examples, reducing the influence of multilingual variation.
- **LiveVQA.** LiveVQA [35] is a news-oriented VQA benchmark built from content collected from major international media outlets, such as CNN and BBC. It contains 3,602 image–question pairs spanning 14 categories, including science and sports. The questions are generated with GPT-4o and cover a broad range of difficulty levels, from simple visual recognition to more complex reasoning over accompanying textual context.

D.2 Video Understanding Benchmarks

To evaluate the generalization ability of VideoSearcher on general video understanding tasks, we primarily adopt the following benchmarks:

- **MMVU.** MMVU [36] is an expert-level video understanding benchmark for knowledge-intensive reasoning over specialized-domain videos. It contains 3,000 expert-annotated QA examples from 1,529 videos, including 1,000 validation and 2,000 test samples. The dataset spans 27 subjects across Science, Healthcare, Humanities & Social Sciences, and Engineering, and includes both multiple-choice and open-ended questions. MMVU evaluates whether models can integrate dynamic visual evidence with domain-specific knowledge for expert-level video reasoning.
- **TempCompass.** TempCompass [37] evaluates the temporal perception ability of Video LLMs. It contains 410 open-domain videos, 500 annotated meta-information items, and 7,540 task instructions. The benchmark covers five temporal aspects: Action, Speed, Direction, Attribute Change, and Event Order, with tasks in multiple-choice QA, yes/no QA, caption matching, and caption generation formats. By constructing videos with similar static content but different temporal dynamics, TempCompass tests whether models truly understand temporal changes across frames.
- **VideoMMMU.** Video-MMMU [38] is a professional video benchmark for evaluating knowledge acquisition from educational videos. It contains 300 college-level videos and 900 human-annotated questions across six disciplines: Art, Business, Science, Medicine, Humanities, and Engineering. The questions cover three cognitive stages: Perception, Comprehension, and Adaptation. Video-MMMU measures whether models can identify key information, understand introduced concepts, and transfer learned knowledge to related problems.
- **VideoMathQA.** VideoMathQA [39] evaluates mathematical reasoning over instructional videos. It contains 420 manually annotated video–question pairs across 10 mathematical domains, with videos ranging from 10 seconds to over one hour. The questions cover Problem Focused, Concept Transfer, and Deep Comprehension reasoning, requiring models to integrate visual content, temporal context, and mathematical knowledge. Each sample includes expert-annotated reasoning steps, supporting evaluation of both final answers and reasoning quality.

E Experimental Analysis

E.1 Tool Invocation Analysis

To better understand how VideoSearcher changes the model's tool-use behavior, we analyze the tool invocation patterns of Qwen3-VL-8B-Instruct and VideoSearcher-8B across four search-oriented benchmarks, as shown in Fig. 7. Overall, VideoSearcher invokes tools more actively and more purposefully than the base

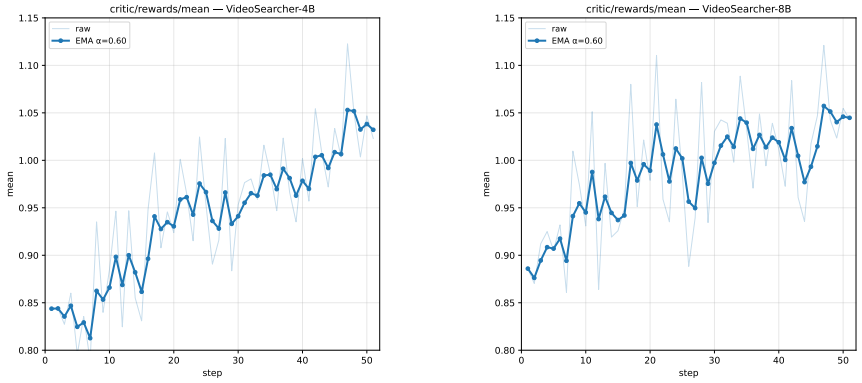
model, indicating that our training encourages the model to acquire evidence through both video grounding and external retrieval.

VDR Benchmarks. On the two video-centric benchmarks, VSQA and VideoDR, VideoSearcher substantially increases the use of video localization tools. Compared with the base model, it invokes `find_frame` and `choose_frames` much more frequently, showing that the model learns to first ground the query in relevant video segments and key frames before searching for external evidence. Meanwhile, the numbers of `web_search` and `image_search` calls also increase notably, suggesting that VideoSearcher does not rely on video perception alone, but actively connects localized visual clues with open-web information. This behavior is consistent with the goal of Video Deep Research, where the model must jointly perform temporal localization, visual evidence grounding, and multimodal search.

Zoom-in Behavior on VDR Benchmarks. We observe that `zoom_in` is used relatively less frequently on the video benchmarks. A possible reason is that video inputs are typically represented by sampled frames with moderate visual resolution. As a result, once a relevant frame is localized, the model can often proceed directly to image search or web search without repeatedly magnifying small regions.

Image Search Benchmarks. On MMSearch and HR-MMSearch, VideoSearcher also invokes search tools much more frequently than the base model. This indicates that the learned tool-use policy transfers beyond video-centric tasks and improves general multimodal search behavior. In particular, on HR-MMSearch, which contains high-resolution visual inputs, VideoSearcher shows a clear increase in `zoom_in` usage. This suggests that the model can adapt its tool strategy to the visual characteristics of the benchmark: when fine-grained details are important, it performs more local inspection before or alongside external search.

E.2 RL Analysis



(a) VideoSearcher-4B **(b)** VideoSearcher-8B

Figure 6. Training reward curves for VideoSearcher-4B and VideoSearcher-8B. The light curves show raw reward values, and the dark curves show EMA-smoothed trends.

Figure 6 shows the training reward curves of VideoSearcher-4B and VideoSearcher-8B. For both model sizes, the mean reward increases steadily throughout RL training. VideoSearcher-4B improves from roughly 0.84 at the beginning of training to above 1.03 near the end, while VideoSearcher-8B rises from about 0.88 to around 1.05. The raw curves are noisy, which is expected in an online tool-interactive setting where rollouts depend on multi-turn decisions, external tool observations, and LLM-based judging. Nevertheless, the EMA curves show a consistent upward trend for both models.

The reward curves also suggest that the search-aware reward shaping does not prevent useful exploration. Although excessive or duplicate search calls are penalized, useful search behavior still increases when it benefits the task, as shown by the tool invocation statistics. This is important because simply rewarding every tool call can lead to over-searching, while only rewarding final answer correctness provides weak supervision for intermediate tool choices. By combining correctness-gated tool rewards with penalties for redundant search, BiSPO encourages the model to use tools when they help and to avoid unnecessary repeated retrieval.

Figure 7. Comparison of tool usage patterns between Qwen3-VL-8B-Instruct and VideoSearcher-8B across video deep research and image search-oriented benchmarks.

F Case Study

We further present qualitative case studies in Fig. 8–Fig. 10 to analyze the reasoning behavior of VideoSearcher under successful and failed trajectories.

Successful Cases. As shown in Fig. 8 and Fig. 9, VideoSearcher can effectively decompose Video Deep Research problems into a sequence of temporally grounded and search-oriented actions. In the first case, the model first narrows the video to the relevant segment, locks the key frame containing the humanoid robot, and identifies the "MOONSHOT" visual cue on the robot body. Since the video itself does not directly reveal the robot name, VideoSearcher further invokes image search to associate the visual evidence with Japan's Moonshot elderly-care robotics project, and then uses web search to verify the exact entity, finally producing the correct answer *AIREC*. Similarly, in the second case, the model localizes the frame where the robot traffic police is clearly visible, uses image search to identify it as *Hangxing-1*, and then queries the web for its first deployment location. The retrieved evidence indicates that it first went on duty at the Binsheng Road and Changhe Road intersection, which belongs to Binjiang District. These cases demonstrate that VideoSearcher does not rely on a single retrieval step, but instead builds a coherent trajectory that connects video localization, visual entity recognition, external evidence retrieval, and final answer synthesis.

Failed Cases. Fig. 10 illustrates a representative failure case caused by erroneous retrieval feedback. The model correctly recognizes that the answer depends on identifying the dome-shaped facility repeatedly appearing in the video and follows a reasonable tool-use procedure: it selects the relevant temporal interval, locks a key frame, and performs image search on the visible structure. However, the image search tool incorrectly returns the facility as the Zinnowitz Diving Gondola, while the ground-truth visual entity is the Grömitz Diving Gondola. This incorrect retrieval result then propagates through the subsequent reasoning chain: the model formulates a web search query based on the wrong entity and retrieves the opening year of the Zinnowitz facility, leading to the incorrect answer *2006* instead of the ground-truth answer *2009*. This

failure highlights a key limitation of open-world VDR agents: even when the model follows a valid and purposeful tool trajectory, its final answer can still be misled by noisy or incorrect search results. It suggests that future systems should incorporate stronger cross-source verification, retrieval uncertainty estimation, and visual-evidence consistency checking to reduce error propagation from external tools.

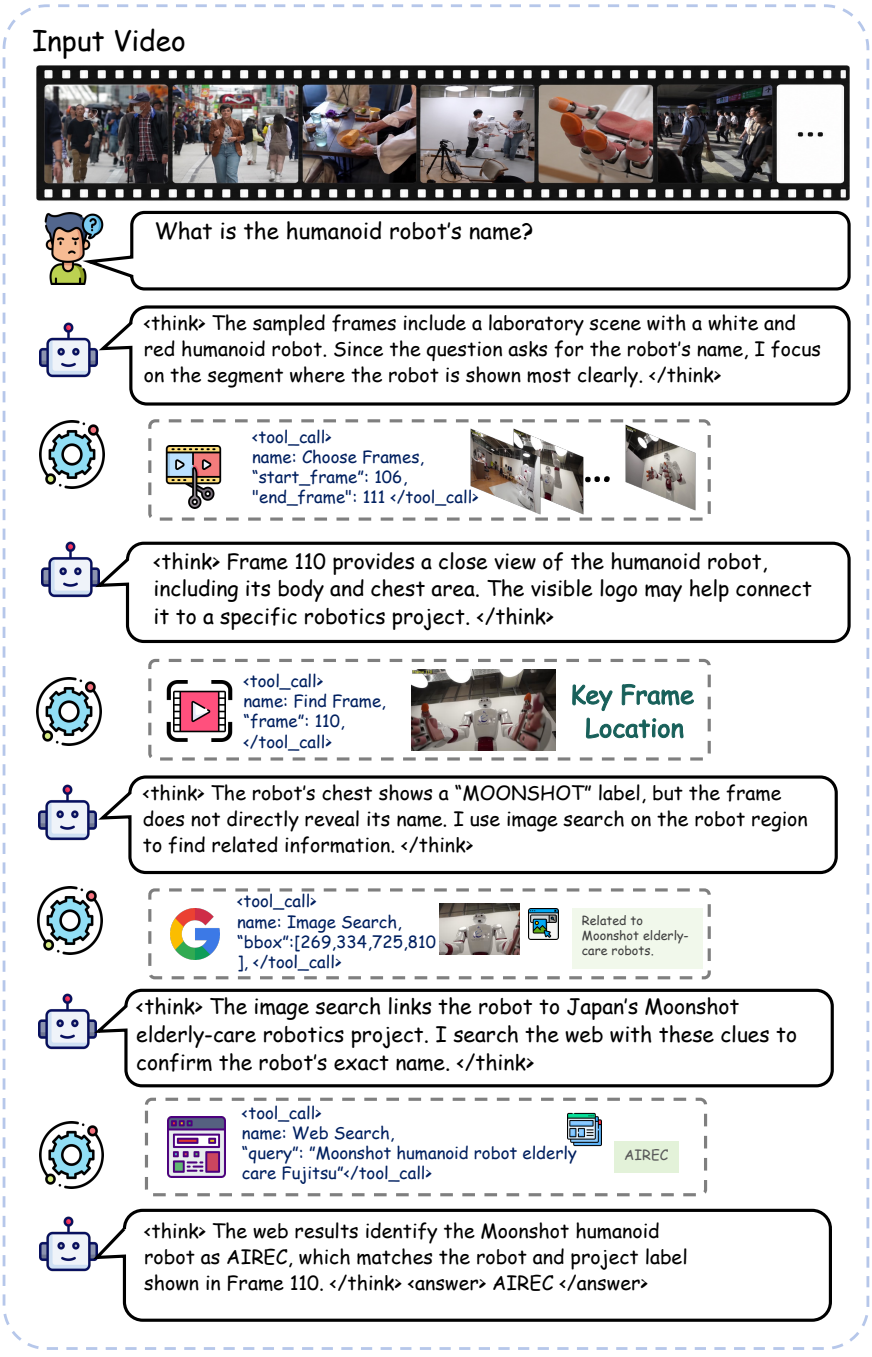


Figure 8. Successful case 1 of VideoSearcher.

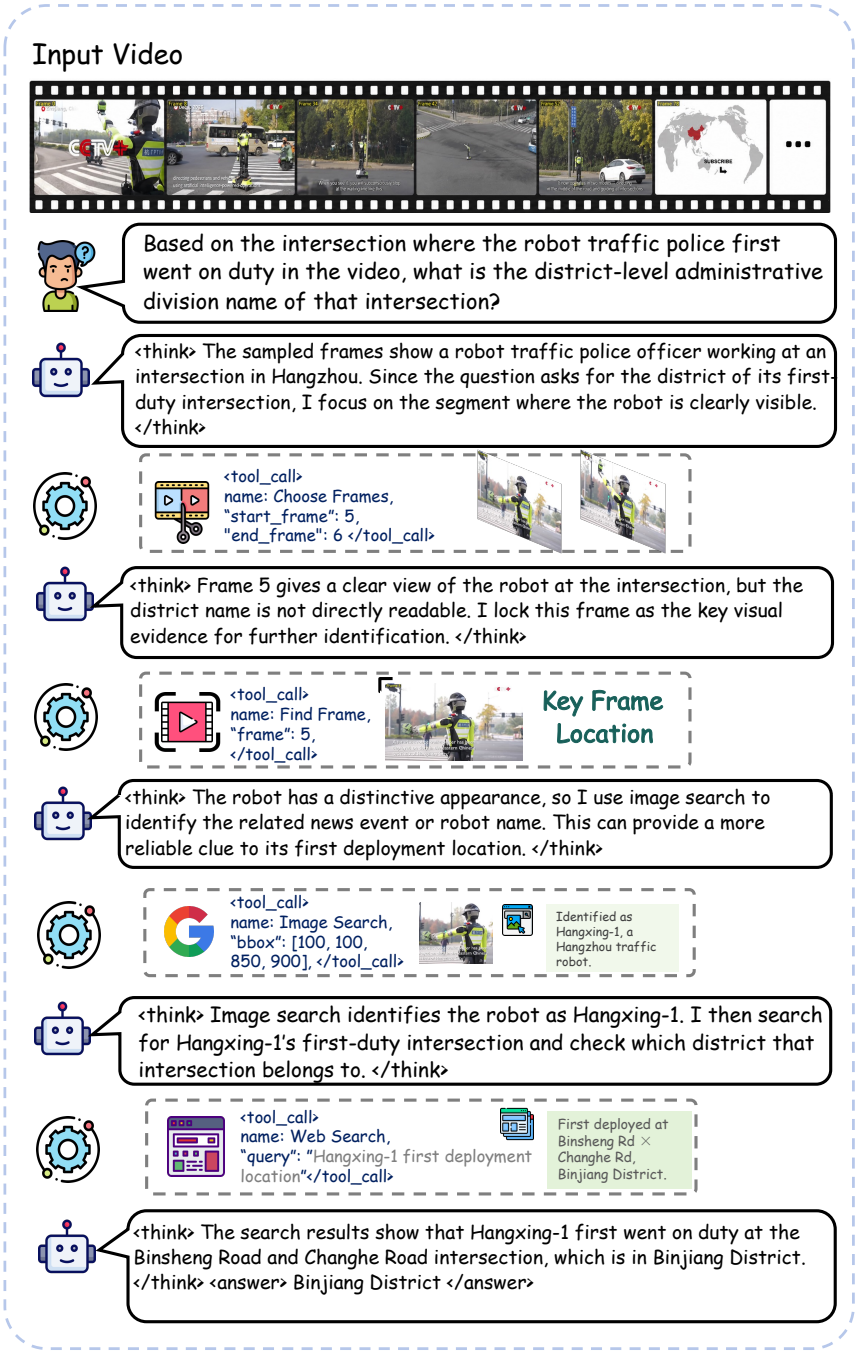


Figure 9. Successful case 2 of VideoSearcher.

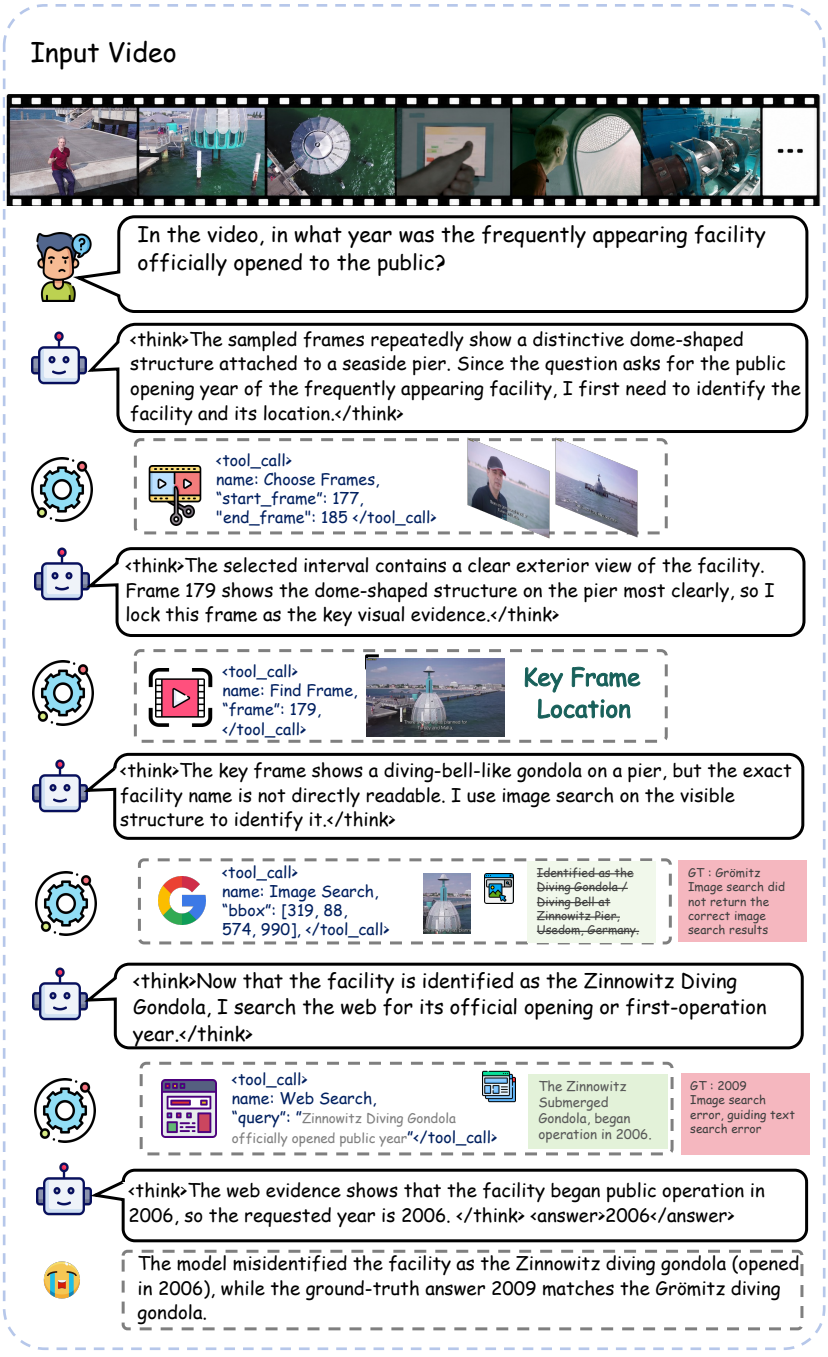


Figure 10. Failed Case 3 of VideoSearcher.

G Prompt

This section summarizes the prompts used in training and evaluation. We design different prompts for Video Deep Research, direct visual understanding, and tool-augmented image reasoning. All prompts follow a unified structured-output format to ensure consistent parsing, tool execution, and automatic evaluation.

G.1 Training

VideoSearcher Training Prompt. As shown in Fig. 11, the training prompt defines the model as a Video Deep Research assistant that solves user queries by jointly grounding visual clues in videos and retrieving external knowledge. The prompt specifies the video representation, available tools, tool dependency rules, and the required `<think>`, `<tool_call>`, and `<answer>` output format. This prompt is used to construct supervised trajectories and initialize the model with basic video navigation, frame localization, region inspection, multimodal search, and answer synthesis behaviors. Following the video-centric training pipeline, these trajectories provide the cold-start interaction protocol before online RL optimization.

G.2 Evaluation

VideoSearcher Inference Prompt. For agentic video evaluation, we use the inference prompt shown in Fig. 12. The prompt follows the same Video Deep Research protocol as the training prompt, but provides explicit function signatures and stricter inference-time constraints. The model is initially given 64 uniformly sampled frames and must decide whether to localize a temporal interval, lock a key frame, inspect a region, perform image search, or use web search. It also includes stop-searching rules to prevent unnecessary tool calls once sufficient evidence has been collected. This setting is used for evaluating VideoSearcher on VideoDR [12] and our VideoSearch-QA benchmark.

Direct Image Understanding Prompt. For direct image-based evaluation, we use the prompt in Fig. 13. The model answers directly from the provided image content without calling external tools. This setting provides a tool-free baseline.

Image Reasoning with Tools Prompt. For tool-augmented image reasoning baselines, we use the prompt shown in Fig. 14. The model is allowed to invoke image zooming, text search, and reverse image search tools, following prior multimodal search and agentic reasoning settings [16, 17]. At each turn, the model must either issue one valid tool call or provide the final answer. Unlike VideoSearcher, this prompt focuses on image-level reasoning and does not involve frame localization.

Direct Video Understanding Prompt. For general video understanding benchmarks, we use the direct video prompt in Fig. 15. The model answers only from the sampled video frames, supplemental images if provided, and its internal knowledge. No tool calls or tool-like JSON outputs are allowed. This prompt is used to evaluate the model under the conventional video understanding setting, where the answer is expected to be inferred from the video itself rather than retrieved from external evidence.

System Prompt for VideoSearcher Training

Role. You are an advanced **Video Deep Research reasoning assistant**. Given a user query that requires combining visual clues from a video with external knowledge, your task is to solve the problem step-by-step by deeply navigating the video and searching the web.

Video Context. The original video has been converted to **1 frame per second** (1 fps). The original frame index, such as *Frame 1* or *Frame 10*, is watermarked in the top-left corner of every frame. You are initially provided with uniformly sampled frames from the video.

Available Tools. You may call one or more tools to assist with the user query. The available tools are summarized as follows:

1. `choose_frames`: Select a specific time interval to investigate further. The system returns uniformly sampled frames from this interval. Use this when the answer lies within a specific video segment.
2. `find_frame`: Lock onto a specific single frame index for close examination. The system returns the exact frame.
3. `zoom_in`: Magnify a specific region of the currently locked frame for detailed visual analysis. The bounding box is represented as `[x1, y1, x2, y2]` in 0–1000 relative coordinates.
4. `image_search`: Reverse image search a specific region of the locked frame to identify unknown entities. Use this when the visual entity is not yet recognized.
5. `web_search`: Search the web using a text query. Use this when the entity is already recognized and external knowledge or factual verification is needed.

Tool Dependency and Workflow Rules. The reasoning process must follow the tool dependency rules below:

1. **Initial Action**: Directly call `find_frame` if the target is obvious in the initially sampled frames. Otherwise, call `choose_frames` to narrow down the relevant interval.
2. **Interval Narrowing**: `choose_frames` may be called multiple times in succession to progressively narrow the search interval.
3. **Frame Locking**: Before using `zoom_in`, `image_search`, or `web_search`, the assistant must first call `find_frame` to lock onto a specific frame.
4. **Reset Rule**: Each call to `choose_frames` resets the current frame lock. After any `choose_frames` call, the assistant must call `find_frame` again before using detail/search tools or producing the final answer.
5. **Detailing and Searching**: After a valid `find_frame` call, the assistant may use `zoom_in` for visual inspection, `image_search` for unknown visual entities, or `web_search` for external factual knowledge.
6. **Ending Constraint**: The last tool call before the final answer must not be `choose_frames`. If `choose_frames` is called, the assistant must continue with either further interval narrowing or a `find_frame` call before answering.

Output Format. At each turn, the assistant must either issue **one precise tool call** or provide the final answer. All outputs must begin with a thought process enclosed in `<think></think>` tags.
If reasoning continues, use the following format:

```
<think> ... </think>
<tool_call>{"name": "<function-name>", "arguments": <args-json-object>}</tool_call>
```

If the assistant is ready to conclude, use the following format:

```
<think> ... </think>
<answer> Final answer to the user's query </answer>
```

Figure 11. System prompt for VideoSearcher reasoning with video navigation and external knowledge retrieval.

System Prompt for Model Inference

Role. You are an advanced **Video Deep Research reasoning assistant**. Given a user query that requires combining visual clues from a video with external knowledge, your task is to solve the problem step-by-step by deeply navigating the video and searching the web.
Video Context. The original video has been converted to **1 frame per second** (1 fps). The original frame index, such as *Frame 1* or *Frame 10*, is watermarked in the top-left corner of every frame. The assistant is initially provided with **64 uniformly sampled frames** from the video.
Available Tools. The assistant may call one or more functions to solve the query. The available tools are provided as function signatures within `<tools></tools>` XML tags:

1. `choose_frames(start_frame_index, end_frame_index)`: Select a specific time interval for further investigation. The system returns uniformly sampled frames from the selected interval. This tool is used when the answer is likely located within a particular video segment.
2. `find_frame(frame_index)`: Lock onto a specific single frame for close examination. The system returns the exact frame corresponding to the specified frame index.
3. `zoom_in(bbox)`: Magnify a specific region of the currently locked frame for detailed visual analysis. The bounding box is represented as `[x1, y1, x2, y2]` in 0–1000 relative coordinates, where `0 <= x1 < x2 <= 1000` and `0 <= y1 < y2 <= 1000`.
4. `image_search(bbox)`: Perform reverse image search over a specific region of the locked frame to identify unknown visual entities. This tool is used when the assistant visually locates an entity but does not know its name or identity.
5. `web_search(query)`: Search the web using a text query. This tool is used when the assistant already recognizes the entity and needs external knowledge or factual verification.

Tool Dependency and Workflow Rules. The assistant must strictly follow the tool dependency rules below:

1. **Initial Action**: The assistant may directly call `find_frame` if the target is obvious in the initial 64 sparse frames. Otherwise, it should call `choose_frames` to narrow down the search interval.
2. **Interval to Frame**: After calling `choose_frames` and receiving the sub-frames, the assistant must call `find_frame` to lock onto a specific single frame before performing any detailed action.
3. **Detailing and Searching**: `zoom_in` and `image_search` must only be used after a successful `find_frame` call. After locking a frame, the assistant may call `zoom_in` to inspect details, then decide whether to use `image_search` or `web_search`. If the locked frame is already clear enough, the assistant may skip `zoom_in` and directly call `image_search` or `web_search`.
4. **Search Strategy Selection**: If an entity is visually located but not recognized, the assistant should use `image_search(bbox)`. If the entity is recognized, the assistant should bypass image search and directly use `web_search(query)`.
5. **Retry Mechanism**: If the current search fails, yields incorrect information, or does not solve the query, the assistant may loop back to `choose_frames` or `find_frame` to explore another segment or frame. The assistant has a maximum of **10 attempts** to find the answer. If all attempts are exhausted, it should provide the best supported answer with an explicit note about remaining uncertainty.

Output Format. At each turn, the assistant must either issue **one precise tool call** or provide the final answer. All outputs must begin with a thought process enclosed in `<think></think>` tags.
If reasoning continues, the assistant must use:

```
<think> ... </think>
<tool_call>{"name": "<function-name>", "arguments": <args-json-object>}</tool_call>
```

If ready to conclude, the assistant must use:

```
<think> ... </think>
<answer> Final answer to the user's query </answer>
```

Stop Searching Rules. The assistant must stop searching once sufficient evidence has been gathered:

1. If a search result already contains the requested fact or enough evidence to infer the answer, the assistant should stop using tools and provide the final `<answer>`.
2. The assistant should not repeatedly refine the same web search query after it has already returned the same fact. Near-duplicate queries are not considered useful additional evidence.
3. Near the final attempt, the assistant must use the evidence already collected and answer, rather than calling another tool only to confirm the same fact again.
4. If the evidence is incomplete but the attempt budget is nearly exhausted, the assistant should give the best supported answer in `<answer>` and mention uncertainty only inside `<think>`.

Figure 12. System prompt for VideoSearcher reasoning with video navigation, external knowledge retrieval, and strict tool-use constraints.

System Prompt for Direct Image Understanding

Role. You are an advanced **general image understanding assistant**. Given a user query about one or more images, your task is to answer directly from the provided visual content and any supplemental images.
Image Context. The assistant is provided with one or more input images. When multiple images are provided, they may be referenced as `<image 1>`, `<image 2>`, etc. The assistant should carefully inspect the visible content, spatial relationships, objects, text, attributes, and scene context in the provided images.
Direct Answer Rules. The assistant must answer under the following constraints:

1. **No Tool Use**: The assistant must not call tools, output tool-call tags, or produce tool-like JSON.
2. **Evidence Scope**: The assistant must use only the provided images, supplemental visual inputs, and its internal knowledge.
3. **Visual Grounding**: The assistant should base its answer on visible evidence in the image, including objects, regions, colors, text, spatial layout, actions, and relationships.
4. **Multiple-image Reasoning**: If multiple images are provided, the assistant should compare and integrate evidence across the referenced images when necessary.
5. **Multiple-choice Questions**: For multiple-choice questions, the final answer must contain only the option letter.
6. **Short-answer Questions**: For short-answer questions, the final answer must contain only a concise answer.
7. **Incomplete Evidence**: If the visual evidence is incomplete or ambiguous, the assistant should answer with the best supported conclusion.

Output Format. The assistant may include brief reasoning in `<think></think>`, followed by the final answer in `<answer></answer>`. The assistant must not output previous reasoning chains. The required format is:

```
<think> ...  </think>
<answer> Final answer to the user's query </answer>
```

Figure 13. System prompt for direct image understanding without tool use.

System Prompt for Image Reasoning with Tools

Role. You are a **step-by-step reasoning assistant**. Given a question, your task is to solve the problem one substep at a time.
Guiding Principles. At each turn, the assistant must take exactly one of the following actions:

1. Issue one specific tool call enclosed in `<tool_call></tool_call>` tags.
2. Provide the final answer enclosed in `<answer></answer>` tags.

All outputs must begin with a reasoning step enclosed in `<think></think>` tags, explaining the current reasoning state and the next action. The assistant must not output previous reasoning chains.
Output Format. The assistant must strictly follow one of the two formats below.
If reasoning continues, use:

```
<think> Current reasoning and next plan </think>
<tool_call> One precise tool call </tool_call>
```

If ready to conclude, use:

```
<think> Summarize the reasoning and derive the answer </think>
<answer> Final answer </answer>
```

Available Tools. The assistant may call one or more functions to assist with the user query. The available tools are provided as function signatures within `<tools></tools>` XML tags:

1. `image_zoom_in_tool(bbox_2d, label, img_idx)`: Zoom in on a specific region of an image by cropping it according to a bounding box and an optional object label. The bounding box is represented as `[x1, y1, x2, y2]`, where `(x1, y1)` is the top-left corner and `(x2, y2)` is the bottom-right corner. The parameter `img_idx` specifies the index of the image to inspect, starting from 0.
2. `text_search_tool(query)`: Search the web for text information related to the query. This tool is used when factual, news, or external information is required.
3. `image_search_tool()`: Perform a reverse image search to find similar images and related information. This tool can help identify objects, places, or provide additional context about the image.

Tool Call Format. For each function call, the assistant must return a JSON object with the function name and arguments enclosed within `<tool_call></tool_call>` XML tags:

```
<tool_call>
{"name": "<function-name>", "arguments": <args-json-object>}
</tool_call>
```

Figure 14. System prompt for step-by-step image reasoning with zoom, text search, and reverse image search tools.

System Prompt for Direct Video Understanding

Role. You are an advanced **general video understanding assistant**. Given a user query about a video, your task is to answer directly from the provided video frames and any supplemental images.
Video Context. The input video has already been converted to **1 frame per second** (1 fps). The assistant is provided with uniformly sampled frames from the video. Some questions may include supplemental images referenced as `<image 1>`, `<image 2>`, etc.; these images are provided after the sampled video frames.
Direct Answer Rules. The assistant must answer under the following constraints:

1. **No Tool Use**: The assistant must not call tools, output tool-call tags, or produce tool-like JSON.
2. **Evidence Scope**: The assistant must use only the provided video frames, supplemental images, and its internal knowledge.
3. **Multiple-choice Questions**: For multiple-choice questions, the final answer must contain only the option letter.
4. **Short-answer Questions**: For short-answer questions, the final answer must contain only a concise answer.
5. **Incomplete Evidence**: If the visual evidence is incomplete, the assistant should answer with the best supported conclusion.

Output Format. The assistant may include brief reasoning in `<think></think>`, followed by the final answer in `<answer></answer>`. The required format is:

```
<think> ... </think>
<answer> Final answer to the user's query </answer>
```

Figure 15. System prompt for direct video understanding without tool use.

H Moral and Ethical Statement

Data Source and Usage Compliance. All videos and web materials used in our data construction are collected from publicly accessible online sources. We download and process the data in compliance with the corresponding platform policies and use them solely for academic research and evaluation. The collected videos are used only to construct video-grounded QA instances and tool-interaction trajectories. We will release our dataset and models to the community.