ONESEARCH-VL: UNIFIED MULTIMODAL DEEP RESEARCH AGENT FOR IMAGE AND VIDEO

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-image, multi-image, and video deep research require different visual operations but share a workflow of visual grounding, external retrieval, and fact composition. A key challenge is to preserve the dependencies linking localized visual anchors, entity relations, source-supported facts, and answer-producing operations. We introduce **OneSearch-VL**, a unified agent centered on the Visually Grounded Evidence Graph (VGEG), which encodes these dependencies as a shared task-level reference for data construction, process supervision, and operation-level evaluation. Our VGEG-based data engine constructs and verifies multi-image and video questions and filters expert trajectories. Using these data, We assemble *OneSearch-VL-SFT-110K* and *OneSearch-VL-RL-10K* for SFT and RL, respectively. We further derive the Evidence-aware Visual-Grounded Rubric reward (EVGR) from VGEG annotations to supervise evidence traceability and visual grounding during RL. For finegrained evaluation, we construct *OneSearch-MI-Bench* and *OneSearch-Video-Bench*, organizing questions by the research operations encoded in their VGEGs. Experiments show that OneSearch-VL-8B improves over Qwen3-VL-8B with tool access by 20.2 and 17.6 percentage points on the two new benchmarks, respectively, while also achieving substantial gains across 7 image benchmarks and VideoDR.

1 INTRODUCTION

Multimodal deep research enables vision-language models to move beyond direct answering by actively locating visual clues, retrieving external knowledge, and integrating evidence through multi-turn tool use (Wu et al., 2025; Geng et al., 2025a; Huang et al., 2026; Chen et al., 2026). Recent work has extended this capability from images to videos by combining temporal localization, visual inspection, and open-web retrieval (Liu et al., 2026; Gao et al., 2026; Fang et al., 2026). Meanwhile, unified visual reasoning models have shown that joint image-video training can facilitate transfer across visual tasks (Feng et al., 2025b). These advances raise a natural question: can one policy jointly learn deep research over both images and videos? Although these settings require different visual operations, they share a workflow from visual grounding to external search and fact composition, motivating unified training across multiple visual input types.

Moving beyond prior work that primarily considers single-image or video inputs, we additionally study multi-image deep research and develop a unified framework spanning all three visual input types. Training data for this unified setting must link visual entities distributed across regions, images, or frames with relevant facts scattered across webpages. Prior work represents evidence through perception–knowledge chains, factual rubrics, and evidence graphs for task construction, supervision, and evaluation (Jiao et al., 2026; Zhang et al., 2026; Xu et al., 2026; Sun et al., 2026). However, these structures do not jointly preserve the dependencies from localized visual anchors, through entity relations and source-supported facts, to answer composition. Video-centric pipelines similarly use extracted entities or key-frame crops mainly as retrieval or question-generation seeds (Gao et al., 2026; Fang et al., 2026), without explicitly retaining this end-to-end provenance.

To address this gap, we introduce the **Visually Grounded Evidence Graph (VGEG)**, a unified task-level reference structure for single-image, multi-image, and video research. VGEG links localized visual anchors to real-world entities, records multi-hop relational paths and source-supported facts, and specifies the operations that compose these facts into an answer. Based on VGEG, we

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: **Tool suite of OneSearch-VL.**

Tool Set	Tool	Description	Arguments	Visual Input Type
$\mathcal{T}_{\text{vis}}$	CROP	Crop an image region.	Image + Coordinates	$I, \mathcal{I}, V$
	OCR	Extract visible text and layout.	Image	
	PERSPECTIVECORRECT	Correct perspective distortion.	Image	
	SUPERRESOLUTION	Upscale low-resolution images.	Image + Scale	
	SHARPEN	Reduce blur and enhance details.	Image + Amount	
$\mathcal{T}_{\text{ret}}$	IMAGESEARCH	Identify entities through image search.	Image	$I, \mathcal{I}, V$
	TEXTSEARCH	Search, read, and summarize webpages.	Query + Language + TopK	
$\mathcal{T}_{\text{temp}}$	SELECTTIMESPAN	Select a contiguous video interval.	Video + Start + End	V
	SELECTFRAME	Extract a frame by index.	Video + Frame Index	

develop a data engine for single-image, multi-image and video research that extracts visual entities, retrieves and verifies web facts, generates compositional questions, and synthesizes expert tool-use trajectories while retaining visual locations, source provenance, and evidence dependencies. These retained dependencies support question-answer construction and verification, expert-trajectory filtering, fine-grained trajectory rewards for RL, and operation-level evaluation. With this data engine, we construct *OneSearch-VL-SFT-110K* and *OneSearch-VL-RL-10K* for SFT and RL, respectively.

Using these data, we develop **OneSearch-VL**, a unified agent trained on data spanning single-image, multi-image, and video inputs through SFT and RL. Existing multimodal agents supervise search through outcome rewards, query-quality signals, tool-use objectives, intermediate entity anchors, or evidence rubrics. We further introduce the **Evidence-aware Visual-Grounded Rubric reward (EVGR)**, constructed from our VGEG annotations to evaluate complete trajectories along two complementary dimensions: *evidence traceability*, which verifies whether the required facts are supported by tool observations, and *visual grounding*, which verifies whether the relevant objects, regions, or frames are correctly identified and used in the solution.

For fine-grained evaluation, we also introduce **OneSearch-MI-Bench** and **OneSearch-Video-Bench**. Existing video deep-research benchmarks primarily evaluate answer accuracy, with analyses commonly organized around video content or source categories (Liu et al., 2026; Gao et al., 2026). Our benchmarks instead organize questions according to the research operations represented in their corresponding VGEGs, which capture the entity-relation paths and fact-composition dependencies required to derive the answer. This organization emphasizes how an agent retrieves and composes evidence rather than what the visual input contains.

We evaluate OneSearch-VL on established and two new benchmarks. OneSearch-VL-8B outperforms Qwen3-VL-8B by 20.2, 17.6, and 27.0 points on OneSearch-MI-Bench, OneSearch-Video-Bench, and VideoDR, respectively. Joint SFT on all three visual input types achieves a six-benchmark average of 55.8, versus 51.7-55.1 for single-type training. EVGR further raises the average from 57.3 to 61.1 over answer and query rewards.

Our contributions are summarized as follows:

- **Visually grounded data construction.** We propose VGEG, a unified structure linking visual anchors, source-supported facts, and answer composition, together with a data engine for constructing and verifying multi-image and video questions and expert trajectories.
- **Unified agent and evidence supervision.** We develop OneSearch-VL for joint single-image, multi-image, and video deep research, and use EVGR to incorporate evidence traceability and visual grounding into RL.
- **Operation-oriented benchmark.** We construct OneSearch-MI-Bench and OneSearch-Video-Bench benchmarks spanning research operations for fine-grained evaluation of fact retrieval and composition.

2 UNIFIED MULTIMODAL DEEP RESEARCH

2.1 PROBLEM FORMULATION

We study multimodal deep research over three types of visual input. Given a visual input X and a question q, an agent actively inspects the visual content, retrieves external information, and synthe-

sizes a supported answer. The input is either a single image, an image collection, or a video:

$$X \in \left\{ I,\ \mathcal{I} = \{I_j\}_{j=1}^{K},\ V \right\}. \tag{1}$$

We learn a unified policy π_θ that selects visual inspection, temporal localization, and web retrieval operations according to the input and question.

At step t, the policy conditions on the interaction history $h_t = (X, q, a_1, o_1, \ldots, a_{t-1}, o_{t-1})$ and generates an action $a_t = (z_t, c_t)$, where z_t denotes the reasoning trace and c_t is either a tool call or the final response. A tool call produces an observation o_t, such as a cropped region, a video frame, recognized text, or retrieved information. The complete trajectory is $\tau = (X, q, a_1, o_1, \ldots, a_{T-1}, o_{T-1}, a_T)$ where a_T contains the final answer. Its likelihood is defined as:

$$\pi_\theta(\tau \mid X, q) = \prod_{t=1}^{T} \pi_\theta(a_t \mid h_t). \tag{2}$$

All three types of visual input share the same action–observation protocol while retaining input-specific behaviors: single-image tasks inspect local regions, multi-image tasks aggregate information across indexed images, and video tasks localize relevant temporal segments and frames.

2.2 Tool Environment

OneSearch-VL exposes tools according to the visual input type, as detailed in Table 1:

$$\mathcal{T}(X) = \begin{cases} \mathcal{T}_{\text{vis}} \cup \mathcal{T}_{\text{ret}}, & X \in \{I, \mathcal{I}\}, \\ \mathcal{T}_{\text{vis}} \cup \mathcal{T}_{\text{ret}} \cup \mathcal{T}_{\text{temp}}, & X = V. \end{cases} \tag{3}$$

All tools share a common dispatcher and observation format. This interface allows the policy to compose operations specific to each visual input type while using the same trajectory representation for expert synthesis, reinforcement learning, and inference. Detailed execution semantics are provided in Appendix D.

3 VGEG-Centered Multimodal Data Engine

Figure 1 presents the complete data engine. Starting from raw visual sources, the pipeline identifies localized visual anchors, connects them to source-supported web facts, constructs task-level VGEGs, and synthesizes verified question–answer pairs and expert trajectories. See Appendix B for details.

3.1 Visual Source Curation

As shown in Figure 1(a), we first curate source videos that contain identifiable entities, concrete event cues, and sufficient potential for external knowledge retrieval. The pipeline combines metadata screening, task-suitability assessment, category balancing, and duration balancing, reducing the initial pool of 2.5M web videos to 70,781 candidates. See Appendix B.1 for details.

3.2 Dense Visual Anchor Discovery

Figure 1(b) converts each retained video into a hierarchical visual representation. We sample and group frames into local clips, generate clip- and frame-level descriptions, remove redundant key frames, aggregate temporally related clips into events, and localize searchable objects in representative frames. The resulting dense caption tree links each event to its temporal range, key frames, and localized object instances, which serve as visual anchors for subsequent retrieval. See Appendix B.2 for details.

3.3 Web Evidence Graph Construction

As illustrated in Figure 1(c), we connect visual anchors to external knowledge through image search, OCR, and text retrieval. Candidate identities are verified against the visual context and retrieved sources, after which source-supported relations and attributes are extracted and iteratively expanded. These results form an input-level evidence graph $\mathcal{G}_X$ that preserves the correspondence among visual anchors, real-world entities, external facts, and supporting sources. See Appendix B.3 for details.

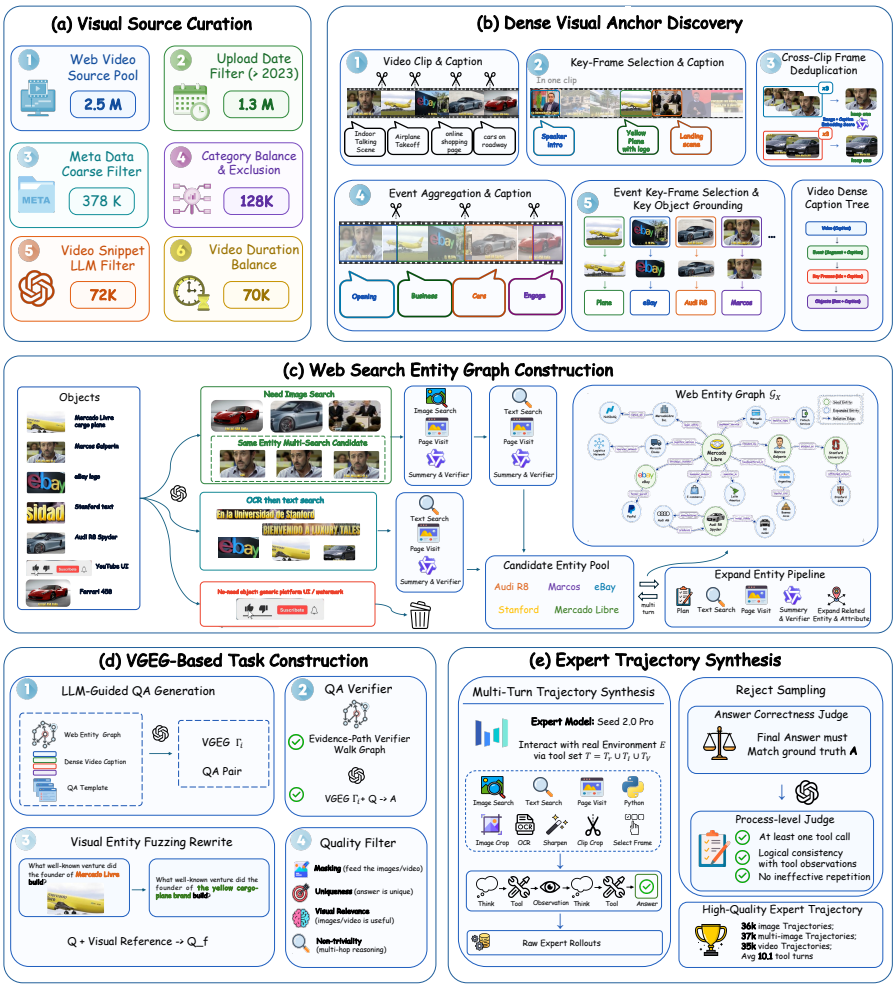


Figure 1: **Overview of the VGEG-centered multimodal data engine.** (a) Visual source curation retains videos suitable for external-knowledge research. (b) Dense visual anchor discovery organizes each video into events, key frames, and localized objects. (c) Web entity graph construction links these anchors to real-world entities, source-supported facts, and webpages, forming the evidence graph $\mathcal{G}_X$. (d) VGEG-based task construction produces VGEG Γ_i for question generation, evidence verification, and visual-reference rewriting. (e) Expert trajectories in the tool environment are filtered by answer correctness and process quality to yield multi-turn training trajectories.

3.4 VGEG-Based Task Construction

Figure 1(d) constructs candidate research tasks and their task-level references from the input-level graph $\mathcal{G}_X$. For each candidate task i, its Visually Grounded Evidence Graph (VGEG) is derived through a task-conditioned projection Φ_i that selects the required visual anchors and facts from $\mathcal{G}_X$ and augments them with explicit answer-producing operations:

$$\Gamma_i = \Phi_i(\mathcal{G}_X) = (\mathcal{A}_i, \mathcal{F}_i, \mathcal{O}_i, \mathcal{R}_i)\,. \tag{4}$$

Here, $\mathcal{A}_i$ contains localized visual anchors and their associated entities, $\mathcal{F}_i$ contains source-supported external facts, $\mathcal{O}_i$ specifies how the selected facts produce the answer, and $\mathcal{R}_i$ combines grounding and support relations inherited from $\mathcal{G}_X$ with task-specific dependencies among anchors, facts, and operations.

The generator jointly outputs (q_i, y_i, Γ_i) from $\mathcal{G}_X$. We verify the question and answer against the selected evidence and rewrite explicit entity mentions into visually grounded references. We then instantiate each task for its target visual input type. For a multi-image task, the deduplicated key



Figure 2: **Operation-oriented design of the OneSearch benchmarks.** Representative questions and reference VGEGs are shown for six principal research operations. Numbered markers bind visual evidence to graph nodes; green nodes denote retrieved entity/fact and purple nodes denote answer-producing operations. The center summarizes the operation distribution and the sizes of the multi-image and video benchmarks.

frames containing the required anchors form an unordered image collection, and frame-level references are remapped to image indices. For a video task, the original video is retained together with its event, timestamp, frame, and region bindings. Additional filters remove answer leakage, ambiguous references, visually irrelevant questions, and instances that can be confidently answered without the visual input. See Appendix B.4 for details. For single-image task, we reuse existing data (Chen et al., 2026) and convert its Wikipedia entity-sampling paths and associated visual anchors into VGEGs.

3.5 Expert Trajectory Synthesis

Finally, as shown in Figure 1(e), we synthesize expert trajectories for the verified tasks. An expert model receives only the visual input, question, and available tools and interacts with the environment until producing a final answer or reaching the interaction budget. We retain trajectories that pass both answer-correctness and process-quality evaluation and combine them with single-image trajectories in a unified multi-turn format to construct **OneSearch-VL-SFT-110K**. See Appendix B.5 for details. For RL, we construct **OneSearch-VL-RL-10K** from task pools spanning the same three visual input types, with construction details provided in Appendix E.

4 OneSearch-Bench

Existing multimodal search and video deep-research benchmarks primarily report aggregate answer accuracy, often with additional breakdowns by visual content or data source (Liu et al., 2026; Gao et al., 2026). These summaries measure overall performance but do not isolate failures in entity lookup, relation tracing, conditional filtering, or fact composition. We therefore construct **OneSearch-MI-Bench** and **OneSearch-Video-Bench** and organize their examples by the *principal research operation* required to answer each question. Both benchmarks use the same operation taxonomy, enabling a consistent analysis across multi-image and video inputs.

Figure 2 presents representative examples and their reference VGEGs, while Appendix Table 6 defines the six principal research operations and reports their distributions across the two benchmarks. Together, they support operation-level analysis of evidence retrieval and composition.

For OneSearch-MI-Bench, deduplicated evidence frames from the same visual source form an unordered image set $X = \{I_1, \ldots, I_K\}$. Each example depends on at least two images and retains image indices that bind visual anchors to their sources. We remove references to video playback, frame numbers, and unnecessary temporal order so that each question is self-contained over the image set, while preserving the relations needed for fact composition.

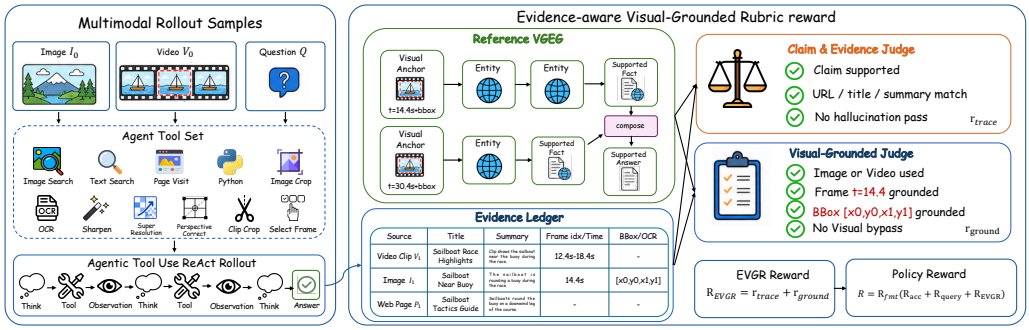


Figure 3: **Overview of the Evidence-aware Visual-Grounded Rubric reward (EVGR).** A task-level VGEG and evidence ledger align each multimodal rollout with the required visual anchors and source-supported facts. Claim-and-evidence and visual-grounding judges produce r_{trace} and r_{ground}, whose combination forms R_{EVGR} for policy optimization.

For OneSearch-Video-Bench, we preserve temporal structure and link visual anchors to their events, key frames, and web facts. Questions locate clues through visible objects, events, or temporal descriptions, requiring the agent to find relevant video content before retrieving and composing external knowledge. The two branches share candidate sources, VGEG representations, the operation taxonomy, and quality criteria, while producing separate multi-image and video sets.

Candidates pass checks for information masking, referential uniqueness, visual relevance, and non-triviality. A text-only probe further removes questions that can be answered confidently without the visual input. We then jointly inspect the question, reference answer, and supporting facts for clarity, answer uniqueness, factual correctness, and visual grounding, followed by a strict review of the visual input.

5 OneSearch-VL Training

We train OneSearch-VL in two stages. Supervised fine-tuning learns a unified multimodal tool-interaction policy from expert trajectories, and reinforcement learning further optimizes answer quality, search behavior, and evidence use through online interaction with tool environment.

5.1 Agentic Supervised Fine-Tuning

We first perform SFT on *OneSearch-VL-SFT-110K*. Each trajectory interleaves reasoning, tool calls, tool observations, and a final answer. All three visual input types share the same action–observation schema, allowing one policy to select input-appropriate tools within a unified action space.

Tool observations are exogenous environment outputs and serve only as conditioning context. Let $y_{t,k}$ denote the k-th policy-generated token at turn t, and let $M^{\text{pol}}_{t,k}$ be one for reasoning, tool-command, and final-response tokens and zero for serialized tool observations. The SFT objective is

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{\tau}\sum_{t}\sum_{k} M^{\text{pol}}_{t,k} \log \pi_\theta(y_{t,k} \mid h_t, y_{t,<k}). \tag{5}$$

This stage yields the initial policy $\pi_{\theta_{\text{SFT}}}$ for reinforcement learning.

5.2 Reinforcement Learning

Starting from $\pi_{\theta_{\text{SFT}}}$, we perform online rollouts on *OneSearch-VL-RL-10K* with the same dispatcher and tool environment used for expert trajectory synthesis and inference. Given visual input X, question Q, and reference answer A, the behavior policy samples a group of trajectories

$$\tau_i \sim \pi_{\theta_{\text{old}}}(\cdot \mid X, Q; \mathcal{E}), \qquad i = 1, \ldots, G. \tag{6}$$

Dual-dimensional EVGR. Terminal correctness does not directly measure whether a rollout obtains and uses the evidence required by the question. We therefore introduce the Evidence-aware Visual-Grounded Rubric reward (EVGR). As illustrated in Figure 3, EVGR aligns each executed trajectory with an item-specific structured rubric derived from the data engine. The rubric records the answer core, required claim facts, supporting sources and snippets, and applicable frame or region annotations. A single trajectory-level judge evaluates two complementary dimensions. **Evidence Traceability** (r_{trace}) measures whether tool observations establish the answer-core entities and required fact hops, and whether the reasoning follows those observations without unsupported substitutions. **Visual Grounding** (r_{ground}) measures whether the correct visible entities, regions, or video frames are identified and used to drive subsequent evidence retrieval. For single- and multi-image inputs, the source images are already visible to the model and no additional visual tool call is required; for video, the selected frames should cover the relevant moments and provide the visual identity used by later searches. The two dimensions are combined into the trajectory-level process reward R_{EVGR}. Their scoring and combination coefficients are specified in the experimental setup.

Composite reward and policy optimization. Following OpenSearch-VL (Chen et al., 2026), we adopt its answer-correctness reward R_{acc} and query-quality reward R_{query}, and augment them with our EVGR process reward. The resulting reward is gated by format validity:

$$R(\tau) = R_{\text{fmt}}(\tau)\Big(\lambda_{\text{acc}} R_{\text{acc}}(\tau) + \lambda_{\text{query}} R_{\text{query}}(\tau) + \lambda_{\text{EVGR}} R_{\text{EVGR}}(\tau)\Big). \tag{7}$$

Here, R_{acc} evaluates the final answer, R_{query} evaluates the relevance and progression of search queries, and R_{fmt} enforces a valid reasoning, tool-call, and response structure. The reward coefficients and implementation details are provided in Appendix E.

We optimize the policy with Group Relative Policy Optimization (GRPO), normalizing rewards across trajectories sampled for the same question and applying the clipped objective only to policy-generated tokens. Tool observations remain conditioning context and are excluded from the policy loss. For trajectories terminated by tool-execution failures, we follow the fatal-aware mechanism of OpenSearch-VL (Chen et al., 2026) and retain only the valid pre-failure prefix for optimization.

6 Experiments

6.1 Experimental Setup

Model and training data. OneSearch-VL is initialized from Qwen3-VL-8B (Bai et al., 2025) and trained in two stages. Supervised fine-tuning uses *OneSearch-VL-SFT-110K*. Reinforcement learning starts from the joint SFT model and uses *OneSearch-VL-RL-10K*. Both datasets cover all three visual input types. All online rollouts use the same tool environment.

Benchmarks. For single-image evaluation, we follow OpenSearch-VL (Chen et al., 2026) and use seven knowledge-intensive benchmarks: SimpleVQA (Cheng et al., 2025), VDR (Zeng et al., 2026), MMSearch (Jiang et al., 2025), LiveVQA (Fu et al., 2025), BrowseComp-VL (Geng et al., 2025b), FVQA (Wang et al., 2017), and InfoSeek (Chen et al., 2023). For multi-image and video evaluation, we use our OneSearch-MI-Bench and OneSearch-Video-Bench, together with VideoDR (Liu et al., 2026). On the two OneSearch benchmarks, we additionally report performance by the six principal research operations defined in Section 4. Following OpenSearch-VL, we use GPT-4o (OpenAI Team, 2024) to compare each final response with the reference answer and return a binary correctness decision.

Implementation details. Expert trajectory synthesis, online RL rollouts, and inference share the same dispatcher and tool interfaces. All training variants are compared under the same tool environment and evaluation protocol. Full optimization, rollout, reward, and inference configurations are provided in the appendix.

6.2 Main Results

Single-image multimodal deep research. Table 2 summarizes the results on seven single-image deep-research benchmarks. OneSearch-VL-8B obtains an average score of 58.3, outperforming the

Table 2: Results on single-image deep-research benchmarks.

Model	SimpleVQA	VDR	MMSearch	LiveVQA	BrowseComp-VL	FVQA	InfoSeek	Avg.
Direct Reasoning								
GPT-4o (OpenAI Team, 2024)	51.7	1.7	18.7	28.1	5.5	48.0	52.9	29.5
GPT-5 (OpenAI, 2025b)	61.6	**9.8**	35.1	44.4	**48.6**	54.4	**61.7**	45.1
Gemini-2.5-Flash (Comanici et al., 2025)	57.9	6.2	30.4	51.0	37.1	47.7	44.1	39.2
Gemini-2.5-Pro (Comanici et al., 2025)	**63.0**	8.0	**39.8**	**60.3**	43.1	**60.7**	46.9	**46.0**
Claude-4-Sonnet (Anthropic Team, 2025b)	50.9	2.0	18.7	38.5	29.3	35.3	57.3	33.1
Claude-3.7-Sonnet (Anthropic Team, 2025a)	42.7	4.6	21.1	38.0	32.3	36.7	54.8	32.9
Qwen3-VL-8B (Bai et al., 2025)	47.1	2.8	11.7	23.1	24.1	24.2	23.1	22.3
Qwen3-VL-30B-A3B (Bai et al., 2025)	53.2	3.8	18.7	42.7	29.6	34.7	26.4	29.9
Qwen3-VL-32B (Bai et al., 2025)	58.0	4.1	19.8	45.5	30.8	34.1	28.8	31.6
RAG Workflow								
GPT-4o (OpenAI Team, 2024)	**63.6**	4.5	49.1	40.1	13.4	**66.3**	59.5	42.4
GPT-5 (OpenAI, 2025b)	55.9	**22.3**	**52.6**	**56.0**	**54.9**	62.6	**70.6**	**53.6**
Claude-3.7-Sonnet (Anthropic Team, 2025a)	59.3	11.3	32.7	30.3	10.0	59.1	60.2	37.6
Qwen3-VL-8B (Bai et al., 2025)	62.3	7.3	47.3	39.3	29.3	53.6	46.1	40.7
Agentic Workflow								
DeepMMSearch-R1-7B (Narayan et al., 2025)	55.8	–	–	–	–	–	47.5	–
Visual-ARFT-7B (Liu et al., 2025)	42.4	3.3	34.5	25.4	16.5	41.7	37.9	28.8
MMSearch-R1-7B (Wu et al., 2025)	57.4	2.9	53.8	48.4	20.9	58.4	55.1	42.4
DeepEyes-v2-7B (Hong et al., 2025)	59.4	7.8	63.7	–	–	60.6	51.1	–
WebWatcher-7B (Geng et al., 2025a)	54.3	10.3	49.1	51.2	21.2	–	–	–
Qwen3-VL-8B (Bai et al., 2025)	52.0	17.0	37.4	50.6	27.9	58.7	50.3	42.0
SenseNova-MARS-8B (Chng et al., 2025)	61.7	19.4	**67.4**	56.2	35.1	67.1	61.7	52.7
OpenSearch-VL-8B (Chen et al., 2026)	71.6	20.8	64.5	59.6	37.6	71.5	70.2	56.6
OneSearch-VL-8B (Ours)	**72.4**	**23.6**	66.5	**61.2**	**39.0**	**73.4**	**72.3**	**58.3**

Table 3: Results on multi-image and video deep-research benchmarks.

	VideoDR	OneSearch-MultiImage-Bench							OneSearch-Video-Bench						
Model	All	SA	MH	Count	Join	Arith.	Comp.	All	SA	MH	Count	Join	Arith.	Comp.	All
Direct Reasoning															
GPT-4o (OpenAI Team, 2024)	42.0	35.7	31.3	56.1	**46.6**	23.0	**44.9**	39.9	36.0	25.5	29.5	**30.8**	13.3	29.9	27.4
Gemini-2.5-Flash (Comanici et al., 2025)	39.0	32.1	33.3	61.4	**46.6**	21.3	42.9	40.2	**44.0**	27.7	32.8	23.1	**20.0**	33.8	29.6
Gemini-2.5-Pro (Comanici et al., 2025)	**50.0**	**42.9**	**39.6**	61.4	**46.6**	**34.4**	**44.9**	**45.2**	**44.0**	**34.0**	**37.7**	23.1	17.8	**37.7**	**32.3**
Qwen2.5-VL-3B (Bai et al., 2025)	9.0	10.7	12.5	35.1	13.8	1.6	14.3	15.0	12.0	6.4	18.0	5.8	4.4	3.9	8.1
Qwen2.5-VL-7B (Bai et al., 2025)	5.0	17.9	12.5	22.8	17.2	4.9	14.3	14.6	4.0	4.3	16.4	5.8	2.2	6.6*	7.2*
Qwen2.5-VL-72B (Bai et al., 2025)	16.0	14.3	14.6	43.9	24.1	6.6	22.5	21.6	28.0	6.4	32.8	13.5	2.2	13.0	15.6
Qwen3-VL-8B (Bai et al., 2025)	11.0	14.3	10.4	26.3	22.4	3.3	18.4	16.0	12.0	2.1	23.0	11.5	2.2	13.0	11.4
Qwen3-VL-30B-A3B (Bai et al., 2025)	14.0	21.4	18.8	**68.4**	25.9	11.5	26.5	29.6	16.0	8.5	29.5	13.5	4.4	14.3	15.0
Qwen3-VL-32B (Bai et al., 2025)	17.0	25.0	27.1	42.1	37.9	18.0	30.6	30.6	20.0	6.4	**37.7**	9.6	2.2	20.8	17.3
Agentic Workflow															
Qwen3-VL-8B (Bai et al., 2025)	30.0	39.3	27.1	45.6	44.8	18.0	40.8	35.6	40.0	12.8	22.9	15.4	2.2	20.8	17.9
OpenSearch-VL-8B (Chen et al., 2026)	47.0	**57.1**	35.4	63.2	55.2	24.6	36.7	44.5	**44.0**	34.0	23.0	15.4	11.1	28.6	24.7
OneSearch-VL-8B (Ours)	**57.0**	46.4	**43.8**	**75.4**	**56.9**	**44.3**	**63.3**	**55.8**	**44.0**	**38.3**	**36.1**	**34.6**	**22.2**	**39.0**	**35.5**

same-scale Qwen3-VL-8B Agent by 16.3 points and OpenSearch-VL-8B by 1.7 points. It improves over OpenSearch-VL on all seven benchmarks, with gains of 2.8, 2.1, and 2.0 points on VDR, InfoSeek, and MMSearch, respectively. It also achieves the best result among the listed agentic methods on six of the seven benchmarks.

The gains span benchmarks emphasizing visual entity identification, open-web retrieval, and multi-step knowledge reasoning. Together with the data-mixture ablation below, these results show that joint training with multi-image and video trajectories does not trade off image research performance.

Multi-image and video deep research. As shown in Table 3, OneSearch-VL-8B improves over Qwen3-VL-8B by 20.2 points on OneSearch-MI-Bench, 17.6 points on OneSearch-Video-Bench, and 27.0 points on VideoDR. It improves all six research operations on both OneSearch benchmarks, showing that the gains extend beyond simple fact lookup to questions that combine evidence across images or video frames.

The operation-level breakdown further identifies where the gains arise. On multi-image tasks, knowledge-conditioned counting, multi-anchor arithmetic, and multi-anchor comparison improve by 29.8, 26.3, and 22.5 points, respectively. On video tasks, multi-hop retrieval, multi-anchor arithmetic, and multi-anchor joining improve by 25.5, 20.0, and 19.2 points. These categories require the model to filter, connect, or compute over facts associated with multiple visual anchors, matching the compositional operations covered by our data engine.

Table 4: Ablation studies on the SFT data mixture and RL reward.

(a) SFT data mixture

Data Mixture	Image	Multi-Image	Video	SimpleVQA	InfoSeek	FVQA	OneSearch-MI	OneSearch-Video	VideoDR	Avg.
Qwen3-VL-8B	×	×	×	52.0	50.3	58.7	35.5	17.9	30.0	40.7
I	✓	×	×	66.1	62.4	65.3	44.5	24.8	47.0	51.7
M	×	✓	×	67.0	62.0	67.1	48.5	25.0	49.0	53.1
V	×	×	✓	68.6	62.5	68.5	51.2	28.0	52.0	55.1
I + M	✓	✓	×	68.4	64.5	67.1	50.8	28.6	50.0	54.9
I + V	✓	×	✓	68.1	63.8	68.2	49.1	29.3	51.0	54.9
I + M + V	✓	✓	✓	68.7	64.4	67.8	51.5	31.2	51.0	55.8

(b) RL reward

Method	R_{acc}	R_{query}	r_{trace}	r_{ground}	SimpleVQA	InfoSeek	FVQA	OneSearch-MI	OneSearch-Video	VideoDR	Avg.
OneSearch-VL-8B-SFT	–	–	–	–	68.7	64.4	67.8	51.5	31.2	51.0	55.8
RL (Acc.)	✓	×	×	×	69.2	64.5	69.3	50.8	32.8	51.0	56.3
RL (Acc. + Query)	✓	✓	×	×	70.0	65.8	70.2	52.5	33.2	52.0	57.3
RL (Acc. + Query + Trace)	✓	✓	✓	×	71.0	68.0	71.5	54.0	35.5	54.0	59.0
RL (Acc. + Query + Ground)	✓	✓	×	✓	71.2	68.2	71.8	54.2	35.2	55.0	59.3
RL (Full reward)	✓	✓	✓	✓	72.4	72.3	73.4	55.8	35.5	57.0	61.1

6.3 Ablation Studies

SFT data mixture. Table 4(a) compares SFT mixtures composed of single-image (I), multi-image (M), and video (V) trajectories using the same number of training iterations. Training on any one trajectory type raises the six-benchmark average from 40.7 to 51.7–55.1. The video-only model attains the strongest single-type average of 55.1 while also improving all three single-image benchmarks, indicating that research trajectories collected for one visual input type can provide useful supervision for others.

Combining trajectory types further improves overall performance. Adding either multi-image or video trajectories to the single-image data raises the average score. Joint training on all three types reaches 55.8 and gives the best result in this ablation on SimpleVQA, OneSearch-MI-Bench, and OneSearch-Video-Bench. This result supports the complementarity of the three trajectory sources under our training setup.

RL reward. Table 4(b) analyzes answer correctness (R_{acc}), query quality (R_{query}), and the Evidence Traceability (r_{trace}) and Visual Grounding (r_{ground}) dimensions of EVGR. Starting from the joint SFT model, accuracy-only RL improves the six-benchmark average from 55.8 to 56.3, and adding query quality raises it to 57.3. Adding Trace or Ground separately further improves the average to 59.0 and 59.3, showing that each process dimension supplies useful supervision beyond answer and query rewards.

Using both Trace and Ground yields the best average of 61.1, improving by 3.8 points over answer-plus-query rewards and by 2.1 and 1.8 points over the two single-dimension variants. The full reward performs best on five benchmarks and ties for best on the remaining one. This result indicates that evidence traceability and visual grounding provide complementary feedback on factual support and the use of visual cues during research.

7 Conclusion

We introduced OneSearch-VL, a unified agent for deep research over single-image, multi-image, and videos. VGEG links visual anchors, source-supported facts, and answer-producing operations, enabling verified task construction, expert-trajectory filtering, and process-level reward design. Building on this structure, EVGR evaluates both evidence traceability and visual grounding. Across seven established single-image benchmarks and two operation-oriented benchmarks, OneSearch-VL consistently improves over the evaluated open-source baselines; the ablations further demonstrate complementary supervision across visual input types and the benefit of EVGR. The current system remains dependent on dynamic web sources, automated judges, and costly multi-turn interaction, motivating future work on judge calibration, adaptive tool budgeting, and broader real-world video domains.

AI USE STATEMENT

We used generative AI models for data construction and evaluation, as described in the paper. We also used AI tools to assist with research discussions, result interpretation, figure preparation, and manuscript writing and translation. All content was reviewed and verified by the authors, who take responsibility for the final work.

ETHICS STATEMENT

Our human review and annotation involved five participants who participated voluntarily and provided informed consent. No sensitive or personally identifiable information was collected, and all annotations were used solely for research purposes. The study did not involve any interventions.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B describe the data construction pipeline, including source curation, visual anchor discovery, web evidence acquisition, task verification, and expert trajectory filtering. Section 4 and Appendix C detail benchmark construction, operation annotations, and human review. Appendix D specifies the tool interfaces and trajectory representation, while Section 5 and Appendix E provide the training objectives, RL data selection procedure, reward design, hyperparameters, and computational resources. Section 6 describes the evaluation protocol. Exact replication of tool observations may vary with changes in external search services, webpages, and hosted model versions. Upon acceptance, we will publicly release our code and data to facilitate reproducibility and future research.

REFERENCES

Anthropic Team. Claude 3.7 Sonnet system card. https://www-cdn.anthropic.com/9ff93dfa8f445c932415d335c88852ef47f1201e.pdf, 2025a.

Anthropic Team. System card: Claude Opus 4 and Claude Sonnet 4. https://www-cdn.anthropic.com/6d8a8055020700718b0c49369f60816ba2a7c285.pdf, 2025b.

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.

ByteDance Seed. Seed2.0 model card: Towards intelligence frontier for real-world complexity. *arXiv preprint arXiv:2607.00248*, 2026.

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

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? *arXiv preprint arXiv:2302.11713*, 2023.

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*, pp. 4637–4646, 2025.

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.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Zhen Fang, Yu Zeng, Wenxuan Huang, Yiming Zhao, Shiting Huang, Tianfei Ren, Qi Lu, Qingnan Ren, Qisheng Su, Lionel Z. Wang, Qingyu Yin, Shuang Chen, Zehui Chen, Lin Chen, Zhenfei Yin, Yao Hu, Shaohui Lin, Wanli Ouyang, Shaosheng Cao, and Feng Zhao. Video-DeepResearch: Towards the next-generation multimodal deepresearch agent. *arXiv preprint arXiv:2608.03979*, 2026.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-R1: Reinforcing video reasoning in multimodal large language models. *arXiv preprint arXiv:2503.21776*, 2025a.

Kaituo Feng, Manyuan Zhang, Hongyu Li, Kaixuan Fan, Shuang Chen, Yilei Jiang, Dian Zheng, Peiwen Sun, Yiyuan Zhang, Haoze Sun, Yan Feng, Peng Pei, Xunliang Cai, and Xiangyu Yue. OneThinker: All-in-one reasoning model for image and video. *arXiv preprint arXiv:2512.03043*, 2025b.

Mingyang Fu, Yuyang Peng, Benlin Liu, Yao Wan, and Dongping Chen. LiveVQA: Live visual knowledge seeking. *arXiv preprint arXiv:2504.05288*, 2025.

Zhenkun Gao, Yicheng Bao, Jinlong Peng, Xueheng Li, Theo Huang, Bangwei Liu, Kunquan Li, Zhenye Gan, Tao Hu, Chengjun Xie, Mingqian Yang, Xuanhua He, Zhizhong Zhang, Xin Tan, Chengjie Wang, and Yuan Xie. VideoSearcher: Empowering video deep research with multi-tool agentic reasoning via reinforcement learning. *arXiv preprint arXiv:2607.02927*, 2026.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. WebWatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025a.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. WebWatcher: Breaking new frontiers of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025b.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. DeepEyesV2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-R1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.

Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, Zehui Chen, Xu Tang, Yao Hu, Shaohui Lin, Philip Torr, Feng Zhao, and Wanli Ouyang. Vision-DeepResearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*, 2026.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, Peng Gao, Yu Liu, Chunyuan Li, and Hongsheng Li. MMSearch: Benchmarking the potential of large models as multi-modal search engines. In *International Conference on Learning Representations*, 2025.

Zhengbo Jiao, Yiming Cheng, Yilei Jiang, Kaituo Feng, Rui Huang, Tianyi Jiang, Juanxi Tian, Jiapeng Li, Qunzhong Wang, Tailai Chen, Qianshan Wei, Chuan Xiao, Shanyu Rong, Yangfu Li, Yanhan Zhou, Yunpu Ma, Yifan Zhang, and Xiangyu Yue. SearchEyes: Towards frontier multimodal deep search intelligence via search world simulation. *arXiv preprint arXiv:2607.05943*, 2026.

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.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. VideoChat-R1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

Chengwen Liu, Xiaomin Yu, Zhuoyue Chang, Zhe Huang, Shuo Zhang, Heng Lian, Kunyi Wang, Rui Xu, Sen Hu, Jianheng Hou, Hao Peng, Chengwei Qin, Xiaobin Hu, Hong Peng, Ronghao Chen, and Huacan Wang. Watching, reasoning, and searching: A video deep research benchmark on open web for agentic video reasoning. *arXiv preprint arXiv:2601.06943*, 2026.

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.

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.

OpenAI. Introducing GPT-4.1 in the API. https://openai.com/index/gpt-4-1/, 2025a.

OpenAI. Introducing GPT-5. https://openai.com/index/introducing-gpt-5/, 2025b.

OpenAI Team. GPT-4o system card, 2024.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. VLM-R1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

Yubo Sun, Chunyi Peng, Yukun Yan, Zhenghao Liu, Sen Mei, Bangrui Xu, Xuanhe Zhou, Chi Chen, and Maosong Sun. HiEviDR-Bench: A benchmark for hierarchical evidence aggregation in deep research. *arXiv preprint arXiv:2607.25151*, 2026.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. FVQA: Fact-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40 (10):2413–2427, 2017.

Shengqin Wang, Wentao Yan, Huichi Zhou, Yihang Chen, Kun Shao, Zhizhong Zhang, and Yuan Xie. DR-MMSearchAgent: Deepening reasoning in multimodal search agents. *arXiv preprint arXiv:2604.19264*, 2026.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-R1: Post-training large vision language model for temporal video grounding. *arXiv preprint arXiv:2503.13377*, 2025.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. MMSearch-R1: Incentivizing LMMs to search. *arXiv preprint arXiv:2506.20670*, 2025.

Ke Xu, Han Xu, Xinran Chen, Yuqian Wang, Zhixuan Li, Xiaojian Liu, Changwo Wu, Jianqiang Xia, and Yuchen Li. STAMP: Provenance-guided credit assignment for deep search agents. *arXiv preprint arXiv:2607.11172*, 2026.

Wentao Yan, Shengqin Wang, Huichi Zhou, Yihang Chen, Kun Shao, Yuan Xie, and Zhizhong Zhang. ProMMSearchAgent: A generalizable multimodal search agent trained with process-oriented rewards. *arXiv preprint arXiv:2604.20486*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Huanjin Yao, Qixiang Yin, Min Yang, Ziwang Zhao, Yibo Wang, Haotian Luo, Jingyi Zhang, and Jiaxing Huang. MM-DeepResearch: A simple and effective multimodal agentic search baseline. *arXiv preprint arXiv:2603.01050*, 2026.

Zuyao You and Zuxuan Wu. Seg-R1: Segmentation can be surprisingly simple with reinforcement learning. *arXiv preprint arXiv:2506.22624*, 2025.

Yu Zeng, Wenxuan Huang, Zhen Fang, Shuang Chen, Yufan Shen, Yishuo Cai, Xiaoman Wang, Zhenfei Yin, Lin Chen, Zehui Chen, Shiting Huang, Yiming Zhao, Yao Hu, Philip Torr, Wanli Ouyang, and Shaosheng Cao. Vision-DeepResearch Benchmark: Rethinking visual and textual search for multimodal large language models. *arXiv preprint arXiv:2602.02185*, 2026.

Jiajie Zhang, Xin Lv, Ling Feng, Lei Hou, and Juanzi Li. Chaining the evidence: Robust reinforcement learning for deep search agents with citation-aware rubric rewards. *arXiv preprint arXiv:2601.06021*, 2026.

Appendix Contents

A Related Work **14**

B Details of the VGEG-Centered Multimodal Data Engine **15**

B.1 Stage 1: Source Video Filter 15

B.2 Stage 2: Dense Video Caption Pipeline 16

B.3 Stage 3: Web Search Entity Graph Construction 17

B.4 Stage 4: VQA Construction 17

B.5 Stage 5: Expert Trajectory Synthesis 18

C Benchmark Construction and Evaluation **19**

C.1 Visual Input Construction 19

C.2 Research Operations and Reference Structures 19

C.3 Difficulty and Visual-Domain Distributions 19

C.4 Human Review and Annotation 20

D Tool Interface and Trajectory Representation **21**

D.1 Resource Registry and Tool Availability 21

D.2 Tool Execution Semantics 21

D.3 Observation and Trajectory Serialization 22

E Training and Evidence Reward Details **22**

E.1 Supervised Trajectory Learning 22

E.2 Reinforcement Learning Data Construction 22

E.3 From VGEG Annotations to EVGR Rubrics 24

E.4 Reward Composition and Policy Optimization 24

F Qualitative Case Studies **26**

F.1 Multi-image Evidence Composition 26

F.2 Cross-frame Evidence Composition 28

A Related Work

Multimodal search and deep research. Active search enables language models to supplement parametric knowledge with external evidence acquired through multi-turn interaction (Jin et al., 2025). Multimodal research extends this paradigm to visual inputs by combining image and text retrieval with visual inspection, and training search policies through synthetic tool-use trajectories and reinforcement learning (Wu et al., 2025; Geng et al., 2025a; Huang et al., 2026; Chen et al., 2026; Yao et al., 2026; Narayan et al., 2025). Related work explores active visual perception, generalizable tool use, and joint optimization of reasoning and search, advancing research agents that integrate visual understanding with external evidence acquisition (Liu et al., 2025; Hong et al., 2025; Chng et al., 2025).

Video deep research. Video deep research additionally requires agents to locate relevant moments, identify objects across frames, and connect them to external knowledge. VideoDR formalizes this setting through an open-web benchmark (Liu et al., 2026). Recent methods develop video-oriented research pipelines by jointly training temporal localization, spatial inspection, and multimodal retrieval (Gao et al., 2026), or by staging tool use to guide visual grounding before web exploration (Fang et al., 2026). These studies highlight temporal and spatial grounding as essential links between video content and external evidence.

Unified visual reasoning. Reinforcement learning has improved image question answering and detection (Huang et al., 2025; Shen et al., 2025), image segmentation (You & Wu, 2025), video question answering (Feng et al., 2025a), and temporal or spatio-temporal video understanding (Wang et al., 2025; Li et al., 2025), with many methods designed for specific tasks. OneThinker jointly learns question answering, captioning, grounding, tracking, and segmentation across images and videos, demonstrating the potential of training across tasks and visual inputs (Feng et al., 2025b). Our work focuses on unification at the level of deep research: a single policy combines input-specific visual operations with external retrieval and fact composition across single images, image collections, and videos.

Structural references and process supervision. Beyond answer correctness, research agents receive supervision through query-quality and tool-use rewards (Chen et al., 2026; Gao et al., 2026). Other approaches use intermediate entity anchors and citation-supported factual requirements as references for credit assignment (Jiao et al., 2026; Zhang et al., 2026), while process-oriented objectives evaluate search decisions or complete rollouts (Yan et al., 2026; Wang et al., 2026). These studies establish the value of structured evidence for agent supervision. We use VGEG to preserve visual provenance and answer-composition dependencies in a shared task-level reference, connecting task construction and verification with trajectory supervision. EVGR derives evidence-traceability and visual-grounding criteria from this reference to evaluate complete research trajectories.

B Details of the VGEG-Centered Multimodal Data Engine

This appendix expands the five stages in Figure 1. The input-level graph $\mathcal{G}_X$ collects candidate visual and web evidence for a visual source. A task-level VGEG Γ_i is derived from $\mathcal{G}_X$ through a task-conditioned projection: it selects the anchors and facts needed by a particular question and augments them with explicit answer-producing operations and dependencies. We report the recorded sampling and execution settings for each stage; prompt templates are omitted.

B.1 Stage 1: Source Video Filter

Figure 1(a) contains six blocks: Web Video Source Pool, Upload Date Filter, Metadata Coarse Filter, Category Balance & Exclusion, Video Snippet LLM Filter, and Video Duration Balance. Table 5 gives the recorded size after each block.

Table 5: Source-video curation. Counts refer to candidate video records.

Stage	Retained records
Web video source pool	2,500,000
Upload date filter	1,295,220
Metadata coarse filter	377,955
Category balance & exclusion	127,562
Video snippet LLM filter	72,403
Video duration balance	70,781

Step 1: Web Video Source Pool. The pipeline starts from 2,500,000 candidate YouTube video records. This pool provides the common input to all subsequent source-level filters.

Step 2: Upload Date Filter. We retain records with available metadata and an upload date on or after January 1, 2023. This stage reduces the pool to 1,295,220 records.

Step 3: Metadata Coarse Filter. A rule-based score combines entity and event cues in the title, description, and tags with duration and metadata richness. We retain the higher-scoring records, leaving 377,955 candidates.

Step 4: Category Balance & Exclusion. We retain Film & Animation, Autos & Vehicles, Pets & Animals, Sports, Travel & Events, Gaming, People & Blogs, Comedy, Entertainment, News & Politics, Howto & Style, and Science & Technology. Candidates are ranked within each category, with equal category quotas used to maintain content coverage and per-channel limits used to prevent concentration on a small number of sources. We exclude Music, Nonprofits & Activism, Education, and Unknown because these categories more often exhibit limited visual dynamics or contain fewer identifiable, searchable entities for evidence-grounded task construction. This stage retains 127,562 records.

Step 5: Video Snippet LLM Filter. We use GPT-4.1 (OpenAI, 2025a) to assess whether each video potentially contains multiple scenes, searchable entities, specific events, visual information, and available web evidence. We combine its assessment with the rule-based metadata score and retain 72,403 candidates. This step operates on textual metadata; visual inspection occurs in the subsequent anchor-discovery stage.

Step 6: Video Duration Balance. We group the remaining candidates by video duration and sample across duration ranges to reduce the dominance of any single interval. This step yields 70,781 records with a more balanced duration distribution.

B.2 Stage 2: Dense Video Caption Pipeline

Figure 1(b) contains five blocks: Video Clip & Caption, Key-Frame Selection & Caption, Cross-Clip Frame Deduplication, Event Aggregation & Caption, and Event Key-Frame Selection & Key Object Grounding. Their outputs form the Video Dense Caption Tree.

Step 1: Video Clip & Caption. We uniformly sample each video at a target rate of 2 FPS, and divide the frames into temporally ordered local clips of ten frames. A final group with fewer than five frames is merged into its predecessor. We use Seed 2.0 Pro (ByteDance Seed, 2026) to describe the scenes, actions, and identifiable entities in each clip.

Step 2: Key-Frame Selection & Caption. For each clip, we use Seed 2.0 Pro (ByteDance Seed, 2026) to select representative key frames and generate key-moment descriptions and candidate entity cues, while retaining the frame indices and timestamps.

Step 3: Cross-Clip Frame Deduplication. We use Qwen3-VL-Embedding-8B (Li et al., 2026) to encode each candidate frame and its description as normalized image and text embeddings. Candidates are compared in temporal order against the most recently retained frame. A candidate is removed when image similarity is at least 0.9 or text similarity is at least 0.8. The similarity thresholds are stored with the deduplication output.

Step 4: Event Aggregation & Caption. We use Seed 2.0 Pro (ByteDance Seed, 2026) to merge temporally adjacent, semantically related clips into coherent events based on the clip descriptions and deduplicated key frames. Each event records a title, temporal range, description, covered clip range, and searchable entity cues, forming the temporal hierarchy from the video to its events.

Step 5: Event Key-Frame Selection & Key Object Grounding. For each event, we select representative key frames and use Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025) to localize visible, searchable objects. Each object records a name or description, a type, a normalized bounding box, and its relation to the event. Bounding boxes use coordinates in $[0, 1000]^4$.

Video Dense Caption Tree. The resulting hierarchy preserves temporal and spatial references:

$$\text{video} \rightarrow \text{event} \rightarrow \text{key frame} \rightarrow \text{localized object}. \tag{8}$$

These records supply the visual anchors used by later retrieval and question construction.

B.3 Stage 3: Web Search Entity Graph Construction

Figure 1(c) routes localized Objects through image search or OCR followed by text search, assembles a Candidate Entity Pool, expands related entities and attributes through the Expand Entity Pipeline, and produces a Web Entity Graph.

Objects and Retrieval Routing. Object grouping and representative-view selection use deterministic matching. Objects with matching types and similar entity or name cues are grouped across frames; the largest available bounding box is selected as the representative view, with alternate views retained from other frames. Crops with a short side below 256 pixels are enlarged to 256 pixels. We use GPT-4.1 (OpenAI, 2025a) to route each object to reverse image search, OCR followed by text search, or omission based on its crop, name, type, candidate entity cues, and event relation. For the OCR route, we also use the model to read visible text and propose a candidate entity; the image-search route uses an external reverse-image-search service to retrieve candidate entities and webpages.

Candidate Entity Pool. Because reverse image search is sensitive to viewpoint and cropping, we construct a retrieval pool for each object grouped across frames, containing its representative crop and alternate views from other sampled frames. If one crop yields no usable candidates, we retry the search with another view from the same object pool. We first use Seed 2.0 Pro (ByteDance Seed, 2026) to select visually grounded anchors with research potential from the event context and initial retrieval results. We then use the model to combine the object crop, local visual description, and image-search or OCR evidence to propose an entity identity, assess whether the retrieved entity matches the visual object, and formulate entity-specific text queries. We use Qwen3-32B (Yang et al., 2025) to read the retrieved pages, produce query-focused summaries, and verify consistency between the page evidence and the candidate identity; search snippets are used when the page body is unavailable. Duplicate pages and explicitly mismatched identities are removed. The remaining identities, source objects, and webpage evidence form the candidate entity pool.

Expand Entity Pipeline. We use Seed 2.0 Pro (ByteDance Seed, 2026) to extract source-supported relations from the verified page summaries. Each fact records its head entity, relation, tail entity or attribute value, supporting URL, and evidence quotation:

$$f = (u, r, z, p, \xi). \tag{9}$$

Here, p is the source URL and ξ is the retained supporting text. A deterministic parser checks required fields and matches normalized quotations against the supplied source material; a matched quotation must contain at least six normalized characters. Numeric facts can additionally retain units and value types. For entity-valued tails, we use Seed 2.0 Pro (ByteDance Seed, 2026) to decide whether to continue based on entity specificity, relation confidence, task relevance, and evidence-expansion potential and to propose a follow-up query; we then use Qwen3-32B (Yang et al., 2025) to summarize and verify the retrieved expansion pages. For each expandable entity, the pipeline retains one query, retrieves five results, deduplicates pages by host, and reads the selected pages. Each parent selects high-confidence successors with scores of at least 0.5, and expansion stops when no suitable successor remains.

Web Entity Graph. Graph assembly uses no additional model and deterministically merges the preceding outputs. The graph contains events, frames, localized objects, entities, attributes, and source pages. Structural edges associate events with frames and frames with objects; grounding edges bind objects to candidate entities; factual edges connect entities to other entities or attributes while retaining source evidence. Entity and relation normalization merges compatible records, removes duplicate edges, and removes self-loops. The resulting input-level graph $\mathcal{G}_X$ supplies a common evidence context for multiple candidate questions.

B.4 Stage 4: VQA Construction

Figure 1(d) contains four blocks: LLM-Guided QA Generation, QA Verifier, Visual Entity Fuzzing Rewrite, and Quality Filter. VGEG is retained as the task-level reference throughout these blocks.

Step 1: LLM-Guided QA Generation. We organize clip and event descriptions from the dense video caption tree, visual-anchor records, source-supported facts from the input-level graph $\mathcal{G}_X$, and candidate relation chains into a generation context. We use Seed 2.0 Pro (ByteDance Seed, 2026) to propose candidate questions spanning the six research operations. The model uses visual descriptions and anchor mappings rather than raw video frames and outputs a question, answer, supporting facts, operation type, visual references, and corresponding frame indices. Multi-anchor questions retain a separate visual description and frame mapping for each referenced object. Relation chains can occur within each branch before the branch results are joined, counted, compared, or used in arithmetic.

The same step assembles the task-level VGEG using the main-text definition $\Gamma_i = \Phi_i(\mathcal{G}_X) = (\mathcal{A}_i, \mathcal{F}_i, \mathcal{O}_i, \mathcal{R}_i)$. The projection selects $\mathcal{A}_i$ from the visual anchors and grounded entities in $\mathcal{G}_X$, retaining their image, frame, or region locations. It selects $\mathcal{F}_i$ from the graph relations and attributes that provide source support for the task. The operation set $\mathcal{O}_i$ describes the answer-producing retrieval or composition, while $\mathcal{R}_i$ retains the selected grounding and support edges and adds task-specific dependencies connecting anchors, entities, facts, and operations. These components are assembled deterministically from the visual-reference fields, selected `used_facts`, task-structure labels, source records, and their grounding and dependency links, without an additional model call.

Step 2: QA Verifier. The verifier first checks the evidence path and task structure against the candidate graph using deterministic rules without an additional model call. Relation-chain questions are checked for continuity between successive facts. Multi-anchor questions require distinct referenced objects and corresponding supporting facts. Comparisons use a common semantic attribute; arithmetic uses numeric operands with compatible meanings; knowledge-conditioned counting associates counted entities with facts supporting the condition. Frame references are checked against the available entity-to-frame mappings, and the facts required to derive the answer must be supported by the candidate graph.

For a candidate (q_i, y_i, Γ_i), the QA Verifier then gathers graph relations, source quotations, page summaries, and visual descriptions associated with its supporting entities. We use Seed 2.0 Pro (ByteDance Seed, 2026) to re-answer the question using only this evidence and to check semantic agreement between the regenerated and proposed answers, allowing aliases and equivalent units or date formats. To prevent answer leakage, explicit occurrences of the candidate answer in the fact summary are masked when possible. This stage checks that the supplied evidence supports recovery of the answer.

Step 3: Visual Entity Fuzzing Rewrite. We use Seed 2.0 Pro (ByteDance Seed, 2026) to inspect the referenced key frames and rewrite explicit entity names into descriptions that locate the intended objects. Multi-hop questions also conceal intermediate entity names so that the final question starts from visible targets without exposing entities along the retrieval path.

The same step instantiates the task for its target visual input type. For a multi-image task, we extract the deduplicated key frames referenced by its VGEG, organize them as an unordered image collection, and remap frame- and region-level references to image indices. References to video playback, timestamps, and unnecessary temporal order are removed so that the question is self-contained over the image collection. For a video task, we retain the original video and its event, timestamp, key-frame, and region bindings, preserving the need for temporal localization.

Step 4: Quality Filter. We use Seed 2.0 Pro (ByteDance Seed, 2026) to assess information masking, referential uniqueness, visual relevance, and whether external retrieval is required. A question-only probe rejects a candidate when the model can reliably recover the reference answer without the visual input. The retained record contains the visual input, final question, answer, VGEG annotations, operation label, and visual-reference mappings.

B.5 Stage 5: Expert Trajectory Synthesis

Figure 1(e) contains two steps: multi-turn trajectory synthesis and rejection sampling.

Multi-turn trajectory synthesis. We use Seed 2.0 Pro (ByteDance Seed, 2026) as the expert model to solve each verified task in the real tool environment. The expert receives the visual input,

question, and tool definitions, while the reference answer is withheld for subsequent evaluation. At each turn, the model produces reasoning followed by a tool call or a final response. The environment executes the tool and appends its observation, including returned visual resources, to the interaction history, producing an interleaved reasoning–action–observation trajectory.

Rejection sampling: answer-correctness judge. We use GPT-4o (OpenAI Team, 2024) to compare the final response of each raw expert rollout with the reference answer while allowing semantically equivalent formulations. Rollouts with an incorrect answer or without a valid terminal response are rejected.

Rejection sampling: process-level judge. The remaining rollouts undergo process-level evaluation, which checks that each trajectory contains at least one effective tool call, remains logically consistent with tool observations, and avoids ineffective repetition. We retain only trajectories that pass both judges as high-quality expert demonstrations for SFT.

C Benchmark Construction and Evaluation

This appendix details the construction protocols for OneSearch-MI-Bench and OneSearch-Video-Bench, including visual-input construction, VGEG-based operation assignment, difficulty and visual-domain annotation, and human review.

C.1 Visual Input Construction

For OneSearch-MI-Bench, deduplicated evidence frames from the same source video form an unordered image collection. Questions are rewritten to refer to the images and visible objects, while image indices preserve the bindings to their visual anchors. References to video playback, frame numbers, and unnecessary temporal order are removed. Each retained item requires evidence from at least two images, preventing reduction to a single-image question.

For OneSearch-Video-Bench, we preserve the temporal structure of the source video and bind each visual anchor to its corresponding event and key frames. Questions refer to relevant content through visible objects, events, or temporal descriptions, requiring the model to localize visual evidence before retrieving and composing external facts.

C.2 Research Operations and Reference Structures

Each benchmark item stores $(X_i, q_i, y_i, \Gamma_i, s_i)$, where s_i is determined by the final answer-producing operation in its task-level VGEG rather than by visual content or source category. Figure 2(a)–(f) shows representative examples of the six operations, and Table 6 provides their definitions and distributions.

When evidence branches contain intermediate relation chains, the category follows the operation that combines the branch results into the answer. For example, two anchors may each initiate multi-hop retrieval, but a question is labeled as multi-anchor comparison when its final answer compares the retrieved attributes. This rule assigns one principal operation to each item while preserving all intermediate dependencies in its VGEG.

The two benchmarks contain 608 questions. The five compositional categories beyond single-anchor lookup account for 555 questions, or 91.3% of the combined set. Multi-image questions contain 2–8 images, with a mean of 3.05 images per question. These statistics show that the benchmarks primarily test relation tracing, conditional filtering, and cross-anchor fact composition.

C.3 Difficulty and Visual-Domain Distributions

We derive a structural difficulty score from the principal research operation, reasoning depth, and number of required facts. Let h_i denote the number of reasoning hops, $|\mathcal{F}_i|$ the number of used facts in the VGEG, and $w(s_i)$ the operation weight, set to 0 for single-anchor lookup, 1 for multi-hop retrieval and knowledge-conditioned counting, and 2 for the three multi-anchor operations:

$$d_i = 10w(s_i) + h_i + 0.1|\mathcal{F}_i|. \tag{10}$$

Table 6: Definitions and distribution of the six principal research operations.

Operation	Definition	Multi-image	Video	Total
Single-anchor lookup	Retrieve one external attribute for one anchor.	28	25	53
Multi-hop retrieval	Follow an external relation chain from one anchor.	48	47	95
Knowledge-conditioned count	Test a retrieved condition per anchor, then count.	57	61	118
Multi-anchor join	Combine retrieved facts across anchors.	58	52	110
Multi-anchor arithmetic	Compute over retrieved numeric facts.	61	45	106
Multi-anchor comparison	Compare the same retrieved attribute across anchors.	49	77	126
All questions		**301**	**307**	**608**

Within each benchmark, examples are ranked by this score and partitioned into easy, medium, and hard groups. Both branches yield empirical boundary scores of 14.4 and 22.2; ties at a boundary are deterministically assigned across adjacent groups to keep their sizes balanced. The three groups contain 100, 100, and 101 multi-image questions and 102, 102, and 103 video questions, respectively. Figure 4(a) shows the full score distributions and their partition boundaries. These labels represent relative evidence complexity within each visual input type rather than independently calibrated human difficulty.

We divide the source videos into six broad visual domains: Daily Life, Entertainment, Geography, Technology, Sports, and News. As shown in Figure 4(b), every visual domain spans all six principal research operations, indicating that the operation taxonomy is not tied to a particular source domain. Differences in their proportions reflect the benchmark's overall emphasis on compositional research questions.

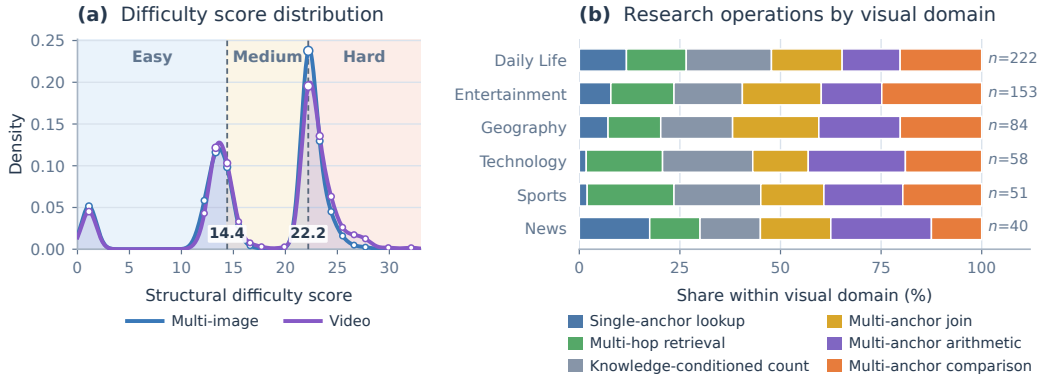


Figure 4: **Benchmark distributions.** (a) Smoothed structural-difficulty score distributions, with circles marking observed values. Dashed lines mark the empirical boundaries at 14.4 and 22.2; the easy, medium, and hard groups contain 100/100/101 multi-image and 102/102/103 video questions. (b) Operation composition within six visual domains. Bars are normalized within each domain, and n denotes the number of questions. Every domain covers all six principal research operations.

C.4 Human Review and Annotation

Candidates first pass the automatic information-masking, referential-uniqueness, visual-relevance, and non-triviality checks described in Appendix B.4. These checks prevent answer or intermediate-entity leakage, ensure that each visual reference identifies its intended object, remove questions that do not require the visual input, and reject questions that can be answered reliably from text alone.

After automatic filtering, human annotators jointly inspect the visual input, question, reference answer, task-level VGEG, and supporting webpages. They verify that each visual reference identifies the intended object, each required fact is source-supported, the reference answer follows from the VGEG dependencies, and the principal operation matches the final answer-producing step. The reviewed record retains the confirmed visual-anchor bindings, supporting facts, reference answer, and operation annotation.

A final quality review further checks question clarity, the presence and distinguishability of the referenced visual objects, and consistency across the annotations. Candidates with ambiguous visual references, unsupported required facts, indeterminate answers, or unclear operation labels are rejected from the final benchmarks.

D TOOL INTERFACE AND TRAJECTORY REPRESENTATION

This appendix specifies the execution semantics of the tools in Table 1, including visual-resource registration, tool inputs and outputs, and the common interaction records used for expert synthesis, reinforcement learning, and inference.

D.1 RESOURCE REGISTRY AND TOOL AVAILABILITY

At the beginning of an episode, the environment registers the visual input in an episode-level resource table. A single image and each member of an image collection receive separate image handles, while a video handle refers to the sampled frame grid visible to the policy. Crops, enhanced images, extracted frames, and selected clips are registered as new image or clip handles and can be reused by later calls. This enables compositions such as selecting a video frame, cropping a local region, and then applying OCR or image search.

Single-image and multi-image inputs expose $\mathcal{T}_{\text{vis}} \cup \mathcal{T}_{\text{ret}}$, while video inputs additionally expose $\mathcal{T}_{\text{temp}}$. All three visual input types use the same call protocol: one tool call is executed per interaction turn, and its observation is returned before the next action.

D.2 TOOL EXECUTION SEMANTICS

Crop. We implement spatial cropping with PIL. Given an image handle and a normalized box $(x_1, y_1, x_2, y_2) \in [0, 1000]^4$, the executor maps the box to pixel coordinates, extracts the region, and registers it as a new image resource. The tool isolates local visual cues before OCR, enhancement, or image search.

OCR. We implement OCR with PaddleOCR. The backend detects text regions, recognizes their contents, and returns the text spans together with bounding boxes and confidence scores as a textual observation. The tool is typically applied to cropped or enhanced signs, documents, logos, and captions.

PerspectiveCorrect. We implement perspective correction with OpenCV. The executor applies grayscale conversion, Gaussian smoothing, and Canny edge detection, locates a quadrilateral among the external contours, orders its corners, and estimates a four-point perspective transform. The rectified fronto-parallel view is registered as a new image resource.

SuperResolution. We implement super-resolution with EDSR through the OpenCV `dnn_superres` interface. The network upsamples the input at the scale supplied by the call and registers the result as a new image resource for subsequent OCR, cropping, or image search.

Sharpen. We implement sharpening with OpenCV unsharp masking. Given amount α, the tool combines the input with its Gaussian-blurred version as $I_{\text{out}} = (1 + \alpha)I - \alpha(G_\sigma * I)$, enhancing edges and fine details before registering the result as a new image resource.

ImageSearch. The executor materializes a registered image as a reference accessible to an external image-to-image search service. It condenses the returned candidate identities, visual matches, source pages, and titles into a textual observation. This tool provides the bridge from a visual anchor to real-world entities and related webpages.

TextSearch. Text search calls an external search service that performs web retrieval, page reading, and query-focused summarization within one request. Each returned passage retains its title, URL, and summary, enabling identity verification and source-grounded fact acquisition.

SelectTimespan. This tool operates on the sampled frame grid of a video or an existing clip. For a resource with N frames, it selects the half-open interval $[s, e)$, where $0 \leq s < e \leq N$, slices the corresponding frame list, and uses FFmpeg to materialize the associated video subclip. The selected frames are re-indexed from zero and registered with the subclip as a new clip resource.

SelectFrame. Given an index $0 \leq i < N$, this tool resolves the corresponding frame from the sampled grid and registers it as a new image resource. The returned handle can be passed directly to cropping, OCR, enhancement, and image search, connecting temporal localization to the common image-tool pipeline. Both temporal tools use sampled-grid indices rather than timestamps in seconds.

D.3 Observation and Trajectory Serialization

Each tool execution produces a textual observation o_t and may additionally create a set of visual resources $\Delta\mathcal{R}_t$. We serialize the corresponding turn as $r_t = (a_t, \hat{c}_t, o_t, \Delta\mathcal{R}_t)$, where a_t is the original policy output and $\hat{c}_t$ is the parsed tool name and arguments. Returned media and text are appended to the next interaction context, allowing subsequent reasoning to reference the executed result. A final response closes the trajectory without another tool execution.

Malformed calls, invalid resource handles, and out-of-range indices are returned as tool-error observations, allowing the policy to correct them in a later turn. Execution status is retained so that data filtering and reward computation can distinguish successful calls, recoverable errors, and incomplete trajectories.

E Training and Evidence Reward Details

This appendix complements Section 5 with implementation details on supervised trajectory learning, the conversion from task-level VGEGs to EVGR rubrics, and the use of process rewards in GRPO.

E.1 Supervised Trajectory Learning

SFT uses **OneSearch-VL-SFT-110K**, comprising approximately 110k expert trajectories: 37k single-image, 37k multi-image, and 36k video trajectories. All three visual input types use the unified multi-turn format defined in Appendix D, where each turn records model reasoning, a tool call, the resulting environment observation, and the subsequent response. Images, video clips, and textual observations returned by tools remain available through resource handles, preserving executable dependencies across tool calls.

Visual inputs, user questions, tool definitions, and tool observations serve as conditioning context, while the supervised targets comprise assistant reasoning, tool commands, and final responses. The loss mask is therefore one for policy-generated tokens and zero for environment observations and other conditioning tokens. This prevents the model from being trained to reproduce tool outputs while preserving its ability to condition subsequent actions on executed observations.

Table 7 reports the complete SFT configuration for OneSearch-VL-8B. We perform full-parameter finetuning while freezing the vision encoder and multimodal projector. Videos are sampled at 2 fps with at most 100,352 pixels per frame and 128 frames. SFT uses 64 H800 GPUs across eight nodes and runs for approximately four days.

E.2 Reinforcement Learning Data Construction

We construct **OneSearch-VL-RL-10K**, containing approximately 10k tasks after filtering and deduplication: 3.7k single-image, 2.8k multi-image, and 3.6k video tasks. Candidates are drawn from the corresponding pools for the three visual input types, exposing online training to spatial grounding, cross-image evidence composition, and temporal localization.

We use Qwen3-VL-8B (Bai et al., 2025) to perform eight independent rollouts for each candidate in the unified tool environment and retain tasks satisfying $0 < n_{\text{correct}} < 8$, where n_{correct} denotes the number of rollouts producing a correct final answer. We deduplicate the retained tasks to form the final RL dataset for subsequent GRPO optimization.

Category	Hyperparameter	Value / Setting
Model	Base Model	Qwen3-VL-8B-Instruct
	Image Max Pixels	262,144 ($\approx 512 \times 512$)
	Video Max Pixels	100,352 per frame
	Video Sampling Rate	2 fps
	Maximum Video Frames	128
	Trust Remote Code	True
Method	Finetuning Type	Full
	Vision Tower Frozen	True
	MM Projector Frozen	True
	DeepSpeed Stage	ZeRO-2
	Mixed Precision	bfloat16
Dataset	Total Samples	$\sim$ 110,000
	Visual Input Types	Single-image, multi-image, video
	Template	`qwen3_vl`
	Cutoff Length	32,768 tokens
	Preprocessing Workers	1
	Preprocessing Batch Size	16
	Dataloader Workers	4
Training	Batch Size per Device	1
	Gradient Accumulation Steps	2
	Effective Batch Size	128 ($= 1 \times 2 \times 64$ GPUs)
	Gradient Checkpointing	True
	Learning Rate	2.0×10^{-5}
	Epochs	8
	LR Scheduler	cosine
	Warmup Ratio	0.1
Infrastructure	Total GPUs	64 NVIDIA H800 GPUs (8 nodes $\times$ 8)
	Training Time	4 days
	Distributed Backend	Torchrun + DeepSpeed ZeRO-2
Logging / I O	Logging Steps	5
	Checkpoint Save Steps	500
	Maximum Saved Checkpoints	5
	Plot Loss	True
	Report Backend	TensorBoard

Table 7: Agentic SFT configuration for OneSearch-VL-8B.

E.3 From VGEG Annotations to EVGR Rubrics

As shown in Figure 3, each training item is paired with a structured rubric derived from its task-level VGEG. The rubric retains answer-core entities, required fact hops, supporting sources and snippets, and applicable image, region, or frame locations. The evaluator receives this rubric together with the question, reference answer, visual input type, and serialized executed trajectory, directly aligning the agent's observations with the evidence required by the task.

The executed trajectory is serialized in interaction order as reasoning and actions, parsed tool names and arguments, and returned observations. Visual resources are represented by their handles and location records rather than re-embedding the raw media in the judge input. We use Seed 2.0 Pro (ByteDance Seed, 2026) as the trajectory-level judge, which returns evidence-traceability and visual-grounding scores together with a brief evidence-based reason for each in one call.

Evidence traceability. The judge checks whether tool observations establish the answer-core entities and fact hops required by the rubric and whether subsequent reasoning follows those observations. Missing evidence, unsupported entity substitutions, broken fact chains, and claims that contradict the retrieved sources reduce the score. This dimension evaluates the evidence process independently of an incidental match with the reference answer.

Visual grounding. The judge checks whether the trajectory identifies the visible entities, regions, or video moments required by the question and uses them to drive subsequent retrieval. For single-image and multi-image inputs, the source images are directly visible to the model, so an additional visual-tool call is not required; the criterion instead checks whether the correct images and objects are used. For videos, the selected frames are compared with the reference moments, and the visual identities obtained from those frames must guide the subsequent evidence search.

Table 8: Anchored EVGR scoring criteria used by the trajectory judge.

Score	Evidence traceability	Visual grounding
1.00	All required fact hops are directly supported.	All required visual anchors are correctly grounded and used.
0.75	Most facts are supported, with one minor gap.	Grounding is mostly correct, with one minor localization gap.
0.50	Useful evidence is present, but a major hop is indirect or missing.	Only part of the required visual evidence is grounded.
0.25	Evidence is mostly noisy, speculative, or contradicted.	Identification is weak or incorrect, with limited recovery.
0.00	No supporting evidence is established.	No grounding is established, or the visual identity remains wrong.

The returned judgment must contain two valid anchored scores and their explanations. The rubric explicitly avoids rewarding verbosity, fluent but unsupported reasoning, ineffective tool calls, or final-answer matching alone. The resulting r_{trace} and r_{ground} can be used separately for ablations or combined into the full process reward R_{EVGR}.

E.4 Reward Composition and Policy Optimization

Using the reward terms defined in the main text, the weights for R_{acc}, R_{query}, and R_{EVGR} are 0.6, 0.2, and 0.2, respectively, and R_{fmt} gates the resulting weighted reward.

Following the main-text setup, we use Group Relative Policy Optimization (GRPO). Given a group of trajectories sampled for the same question, GRPO derives relative advantages from their within-group rewards and updates the policy with a clipped objective. The policy loss is applied only to model-generated reasoning, tool-command, and final-response tokens, while tool observations remain conditioning context. For trajectories truncated by a terminal tool error, only the valid interaction prefix preceding the failure participates in optimization (Chen et al., 2026).

Table 9 summarizes the RL configuration for OneSearch-VL-8B. Training uses asynchronous SGLang rollouts and follows the OpenSearch-VL 8B recipe for the Megatron-LM actor and learning rate. RL uses 32 H800 GPUs across four nodes and runs for approximately four days.

Category	Hyperparameter	Value / Setting
Model	Initial Policy	OneSearch-VL-8B-SFT
	Training Dtype	bfloat16
Data	Train Batch Size (prompts)	256
	Val Batch Size	64
	Max Prompt Length	40,960 tokens
	Max Response Length	16,384 tokens
	Data Seed	3407
	Image Max Pixels	262,144
	Video Max Pixels	100,352 per frame
	Video Sampling Rate	2 fps
	Maximum Video Frames	128
Rollout	Engine	SGLang (async mode)
	Rollout Tensor Parallel	1
	# Samples per Prompt (n)	8
	GPU Mem. Utilization	0.60
	Train / Val Temperature	0.7 / 0.7
	Train / Val Top-p	1.0 / 0.95
	Top-k	-1 (disabled)
Policy (Actor)	Strategy	Megatron-LM
	Tensor Parallel (TP)	4
	Pipeline Parallel (PP)	2
	Context Parallel (CP)	4
	PPO Mini-batch Size	64
	PPO Max Token Len / GPU	74,576
	Micro-batch Size / GPU	1
	Dynamic Batch Size	True
	Param / Optim / Grad Offload	CPU
	Gradient Checkpointing	Full recompute, uniform (1 layer)
Optim. / Loss	Actor LR	1×10^{-6}
	PPO Clip Ratio (high)	0.28
	Entropy Coefficient	0.0
	Use KL Loss	False
	KL Loss / Controller Coef	1×10^{-3} / 1×10^{-3}
	Loss Aggregation	`seq-mean-token-sum`
Algorithm	Advantage Estimator	RLOO (within GRPO objective)
	KL Type	low-variance KL
	Fatal-aware Masking	True (unknown + error)
Tool-Agent	# Parallel Tasks	512
	# Parallel Tool Calls	1,024
	Stepwise Advantage	False
Trainer	Cluster	32 NVIDIA H800 GPUs (4 nodes $\times$ 8)
	Training Time	4 days
	Save / Test Freq (steps)	10 / disabled
	Total Training Steps	200
	Total Epochs	100
	Critic Warmup	0 (critic-free)

Table 9: GRPO configuration for OneSearch-VL-8B.

F QUALITATIVE CASE STUDIES

This appendix presents example outputs of OneSearch-VL-8B, illustrating how the agent localizes visual anchors, links them to external facts, and produces answers. Figures 5–8 show four trajectories covering multi-image and cross-frame evidence composition. Each figure shows the question, visual material, model reasoning, tool observations, and final answer. Model responses are excerpted and tool observations are condensed, with all tool calls and arguments preserved.

Labels on input images, videos, selected frames, and crops match the handles used in the tool calls.

F.1 MULTI-IMAGE EVIDENCE COMPOSITION

Multi-anchor arithmetic. Figure 5 asks for the difference between two manufacturers' founding years. The agent crops the rear of a blue sports car and identifies a Ford Mustang through IMAGESEARCH, then crops a brake-caliper logo in another image to identify Brembo. TEXTSEARCH returns 1903 for Ford and 1961 for Brembo, and the final response computes $1961 - 1903 = 58$. The two visual anchors lead to distinct entities and numerical facts, which a subtraction operation combines into the answer, illustrating the dependencies captured by VGEG.





User Question: In one image, a blue sports car is shown. In another image, a dark grey brake caliper is shown. How many years later was the manufacturer of the brake caliper founded than the manufacturer of the sports car?

Model Responses: **<think>** The user wants the difference in founding years between the sports car manufacturer and the brake caliper manufacturer. First, I need to identify the car manufacturer from img_1 (the blue sports car). I will crop the rear section of img_1 [...] **</think>**
<tool_call> {"name": "crop", "arguments": {"image": "img_1", "bbox_1000": [150, 400, 780, 600]}} **</tool_call>**
Tool Output: **<observation>** Crop → img_3. **</observation>**

Model Responses: **<think>** Now I will run a reverse image search on the cropped car rear (img_3) to identify the manufacturer of the blue sports car. **</think>**
<tool_call> {"name": "image_search", "arguments": {"url": "img_3"}} **</tool_call>**
Tool Output: **<observation>** Results include Mustang S650 and **Ford** Mustang taillights. **</observation>**



Model Responses: **<think>** Okay, the sports car is identified as **Ford** Mustang, manufacturer **Ford** Motor Company. Now I need to confirm the brake caliper manufacturer from img_2. Let me crop the region with the brand logo on the caliper to run image search. **</think>**
<tool_call> {"name": "crop", "arguments": {"image": "img_2", "bbox_1000": [330, 280, 440, 580]}} **</tool_call>**
Tool Output: **<observation>** Crop → img_4. **</observation>**

Model Responses: **<think>** Now I will run reverse image search on the cropped caliper (img_4) to confirm the brake manufacturer. **</think>**
<tool_call> {"name": "image_search", "arguments": {"url": "img_4"}} **</tool_call>**
Tool Output: **<observation>** Result 2 identifies **Brembo** Big Brakes; other matches are irrelevant. **</observation>**



Model Responses: **<think>** Now I have confirmed the two manufacturers: the car is **Ford** Motor Company, the brake caliper is by **Brembo**. I need to search for the founding years of both companies to calculate the difference. **</think>**
<tool_call> {"name": "text_search", "arguments": {"q": "**Ford** Motor Company founding year **Brembo** founding year", "top_k": 3}} **</tool_call>**
Tool Output: **<observation>** Passages 1–3 report **Ford**: **1903**; **Brembo**: **1961**. **</observation>**

Final Model Response: **<response>** The sports car is a **Ford** Mustang, manufactured by **Ford** Motor Company, founded in **1903**. The brake caliper is made by **Brembo**, founded in **1961**. The difference in their founding years is **1961** - **1903** = **58** years. boxed{**58**} **</response>**

Figure 5: **Multi-image arithmetic.** Separate crops and image searches link the car and brake caliper to Ford and Brembo. Subtracting their retrieved founding years yields 58 years.

Knowledge-conditioned counting. Figure 6 asks how many car models in three separate images are produced by a company owned by Volkswagen. The agent first crops a logo, identifies Bentley through image search, and retrieves its ownership relation with Volkswagen. It then crops and image-searches each car to identify the Continental-family models and variants, followed by a query about their distinctions. Under the sample's model/variant counting convention, the final response counts the three pictured cars and returns 3. This case links three visual anchors to an external ownership condition before aggregating the qualifying targets, with all ten tool calls retained.



User Question: In one image, a light gray luxury convertible is displayed, in another a black luxury convertible is shown, and in a third a black luxury coupe with open red doors is visible. How many of these distinct luxury car models are produced by a company owned by **Volkswagen** AG?

Model Responses: **<think>** The user wants to know how many of the three distinct luxury car models shown are produced by a company owned by **Volkswagen** AG. First, I need to confirm [...] **</think>**
<tool_call> {"name": "crop", "arguments": {"image": "img_3", "bbox_1000": [370, 570, 500, 720]}} **</tool_call>**
Tool Output: **<observation>** Crop → img_4. **</observation>**

Model Responses: **<think>** Now that I have the cropped logo region from img_3, I need to run image search to confirm this is the **Bentley** logo, so that img_3 is a **Bentley** vehicle. [...] **</think>**
<tool_call> {"name": "image_search", "arguments": {"url": "img_4"}} **</tool_call>**
Tool Output: **<observation>** Matches include **Bentley** logo / **Bentley** horsepower specifications. **</observation>**



Model Responses: **<think>** Okay, so the logo is confirmed as **Bentley**, so all three cars are **Bentley** models. Now I need to verify how many **Bentley** car models are produced by a [...] **</think>**
<tool_call> {"name": "text_search", "arguments": {"q": "How many distinct car models does **Bentley** produce, and is **Bentley** owned by **Volkswagen** AG", "top_k": 5}} **</tool_call>**
Tool Output: **<observation>** Saved passages agree: **Bentley** is owned by **Volkswagen**; acquired in 1998. Model-line counts differ (4 / 3 named / 5). **</observation>**

Model Responses: **<think>** Wait, I need to confirm if the three cars shown are distinct models from each other, so I should first identify each of the three cars: the light gray [...] **</think>**
<tool_call> {"name": "crop", "arguments": {"image": "img_1", "bbox_1000": [75, 90, 800, 910]}} **</tool_call>**
Tool Output: **<observation>** Crop → img_5. **</observation>**

Model Responses: **<think>** Okay, now I have the cropped img_5 which is the light gray convertible **Bentley**. Let me run image search on this to identify which model it is, so I can [...] **</think>**
<tool_call> {"name": "image_search", "arguments": {"url": "img_5"}} **</tool_call>**
Tool Output: **<observation>** Results identify **Bentley** Continental GTC / GT Convertible. **</observation>**



Model Responses: **<think>** Okay, img_1 is a **Bentley** Continental GT convertible (GTC/GTC Speed). Now I need to identify the black convertible in img_2 to see if it's a different [...] **</think>**
<tool_call> {"name": "crop", "arguments": {"image": "img_2", "bbox_1000": [250, 290, 840, 740]}} **</tool_call>**
Tool Output: **<observation>** Crop → img_6. **</observation>**

Model Responses: **<think>** Now I have the cropped img_6 which is the black convertible **Bentley**. Let me run image search on this to identify its model, to see if it's a different [...] **</think>**
<tool_call> {"name": "image_search", "arguments": {"url": "img_6"}} **</tool_call>**
Tool Output: **<observation>** Results identify **Bentley** Continental GT / GT Speed Convertible. **</observation>**



Model Responses: **<think>** Now I need to identify the black coupe in img_3 to confirm its model, to see if it's a third distinct model. Let me crop img_3 and run image search. **</think>**
<tool_call> {"name": "crop", "arguments": {"image": "img_3", "bbox_1000": [80, 90, 870, 990]}} **</tool_call>**
Tool Output: **<observation>** Crop → img_7. **</observation>**

Model Responses: **<think>** Okay, now I have the cropped img_7 which is the black coupe with red interior doors, it looks like a **Bentley** Continental GT (the fastback coupe version, [...] **</think>**
<tool_call> {"name": "image_search", "arguments": {"url": "img_7"}} **</tool_call>**
Tool Output: **<observation>** Result 2: **Bentley** Continental GT V8 Coupe. **</observation>**



Model Responses: **<think>** Now I have identified all three models: img_1 is a **Bentley** Continental GT (convertible), img_2 is a **Bentley** Continental GTC (Speed Convertible), img_3 is [...] **</think>**
<tool_call> {"name": "text_search", "arguments": {"q": "Are **Bentley** Continental GT Coupe, GTC Convertible, and GT Speed considered distinct car models", "top_k": 5}} **</tool_call>**
Tool Output: **<observation>** Passages treat GT Coupe, GTC Convertible and GT Speed as distinct variants. **</observation>**

Final Model Response: **<response>** All three distinct luxury car models shown are **Bentley** models (Continental GT Coupe, Continental GTC Convertible, and Continental GT Speed variants), and **Bentley** is fully owned by **Volkswagen** AG, having been acquired by the group in 1998. boxed{3} **</response>**

Figure 6: **Knowledge-conditioned counting.** Three input images and four crops support brand and model identification, followed by counting under a retrieved ownership condition. Image handles match the tool calls.

F.2 Cross-frame Evidence Composition

The following cases retrieve evidence for distinct objects in two and three different frames of the same video, then compare their attributes. Frame strips provide neighboring visual context; blue borders and USED labels mark frames selected by the agent, and ellipses indicate omitted intervals. F denotes a selectable frame index in the benchmark input, not a timestamp in seconds.

Comparing two objects across frames. Figure 7 shows a video-only SFT checkpoint comparing two toy blasters. The agent selects F9, crops and identifies the Commander RD-6 through image search, and retrieves its release year. It then selects F116 and repeats the crop, image-search, and text-search sequence for the yellow Recon CS-6. Each branch is thus tied to a distinct frame, object, and external attribute. The final response compares the retrieved years 2020 and 2008 and returns 2020, matching the reference answer. The conflicting year summaries in the original trajectory remain visible in the figure.

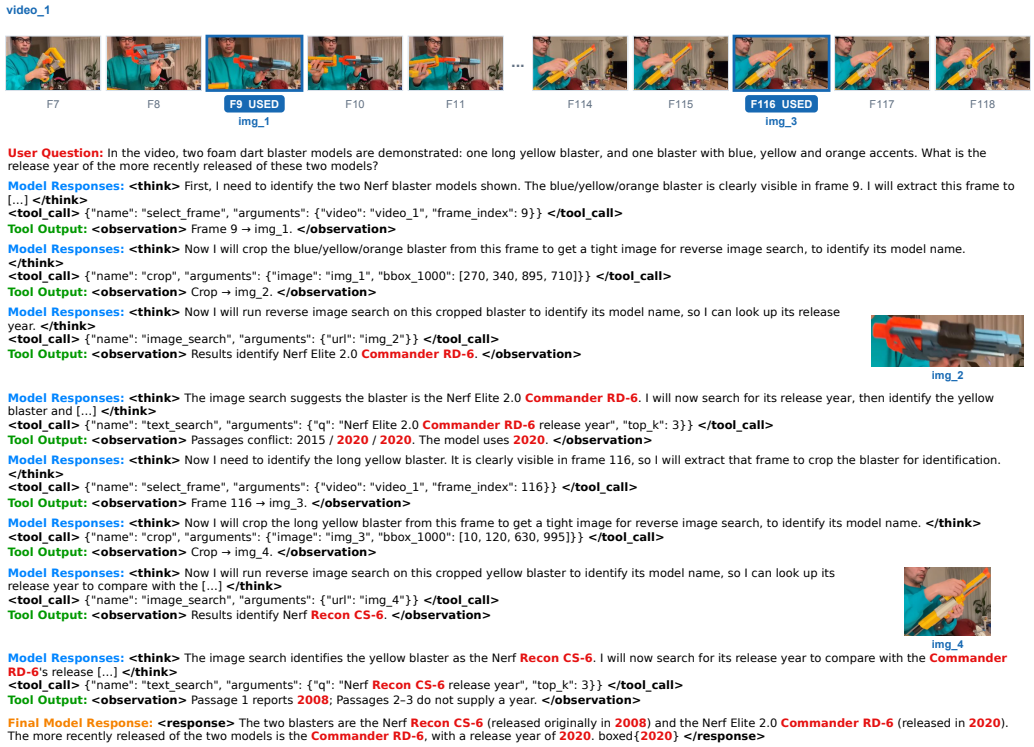


Figure 7: **Cross-frame attribute comparison.** Blue borders mark selected frames F9 and F116; adjacent frames provide context. Each object undergoes cropping, image retrieval, and text retrieval before release-year comparison. All eight calls and arguments are retained.

Comparing landmarks in three frames. Figure 8 extends cross-frame retrieval to three anchors. The agent selects F0, F12, and F15, independently crops and image-searches the buildings, and identifies the Chicago Board of Trade, Chicago Temple, and John Hancock Center. For the first building, it enlarges an initially small clock crop before searching. It then retrieves building heights and issues follow-up queries to clarify conflicting values, ultimately returning 1,128 feet, matching the reference. The saved observations retain inconsistent height definitions, and the final response adds an inaccurate antenna height. The trajectory makes the distinction between visual grounding and evidence traceability concrete: the three buildings are independently grounded, while their numerical evidence remains inconsistent.



Figure 8: **Three-anchor cross-frame retrieval.** Blue borders mark selected frames F0, F12, and F15 amid neighboring views. The three buildings are independently image-searched before height comparison. All fourteen calls, crop refinement, follow-up searches, and conflicting observations are retained.