BrowseComp-V^3: A Visual, Vertical, and Verifiable Benchmark for Multimodal Browsing Agents

Anonymous authors
Paper under double-blind review

Abstract

Multimodal large language models (MLLMs) are evolving into browsing agents capable of multimodal deep search, supported by advances in autonomous planning and tool use. However, existing benchmarks remain limited in task complexity, information searchability, and evaluation dimensions, hindering comprehensive assessment in open-world settings. To address these limitations, we introduce BrowseComp-V^3, a benchmark comprising 300 challenging, manually constructed questions across diverse domains. The tasks emphasize multi-hop, cross-modal reasoning at multiple levels of complexity and are designed to require evidence gathering through web browsing. We verify the public searchability of key evidence and provide expert-validated subgoals for process evaluation, enabling fine-grained analysis of search behavior and model capabilities. We also provide OmniSeeker, a general multimodal browsing agent framework, and evaluate a broad range of MLLMs and browsing systems. Even OpenAI Deep Research, the best-performing proprietary system in our evaluation, achieves a success rate of only 56% on BrowseComp-V^3. Further analysis reveals bottlenecks in multimodal information integration and fine-grained perception, suggesting that effectively using visual evidence in multi-step search and reasoning remains challenging.

1 Introduction

Multimodal large language models (OpenAI, 2025a; Pichai et al., 2025; Bai et al., 2025; Meta, 2025; Li et al., 2024a; Shi et al., 2025b) have demonstrated substantial progress across complex tasks. By integrating linguistic comprehension, visual perception, and tool-use capabilities, these models are increasingly capable of acting as autonomous agents that explore external environments and make decisions. Recent studies have therefore investigated how MLLMs can leverage external search and browsing tools to address multimodal deep-search tasks in open-world environments (OpenAI, 2025; Google, 2024; Wei et al., 2025; Wu et al., 2025b; Geng et al., 2025; Huang et al., 2026).

Despite rapid advances in model capabilities, existing benchmarks (Jiang et al., 2024; Geng et al., 2025; Li et al., 2025b; Tao et al., 2025a) remain limited in task complexity, information searchability, and evaluation dimensions, hindering comprehensive and reproducible assessment of multimodal browsing agents. Specifically, they exhibit three limitations: **i) Insufficient Task Complexity.** Some benchmarks (Cheng et al., 2025; Geng et al., 2025) focus on relatively short retrieval chains, with visual information concentrated in the initial stages. Consequently, they provide limited coverage of deep, cross-modal search scenarios. **ii) Inaccessibility of Key Information.** In some benchmarks (Fu et al., 2025b; Li et al., 2025b; Tao et al., 2025a), critical evidence is embedded in videos, documents, or other media that standard browsing tools cannot reliably retrieve or parse, complicating reproducible evaluation. **iii) Narrow Evaluation Dimensions.** Several studies (Jiang et al., 2024; Li et al., 2025b; Zeng et al., 2026) have introduced process-level evaluation beyond final-answer accuracy, but these approaches typically focus on predefined stages or selected types of in-

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: Comparison of our benchmark against representative deep search benchmarks along eight dimensions. § indicates that the benchmark only partially satisfies the corresponding criterion.

Benchmarks	Multimodal Context inputs	Multi-round Interaction(>2)	Thinking with Images	Multi-image Reasoning	Public-search Answerable	Hop-based Difficulty Analysis	Human-validated Trajectories	Fine-grained Progress Metrics
InfoSeek (Chen et al., 2023)	✔	✗	✗	✗	✔	✗	✗	✗
Enc-VQA (Mensink et al., 2023)	✔	✗	✗	✗	✔	✗	✗	✗
MMSearch (Jiang et al., 2024)	✔	✗	✗	✗	✔	✗	✗	✗§
DynVQA (Li et al., 2024b)	✔	✗§	✗	✗	✔	✔	✗	✗
SimpleVQA (Cheng et al., 2025)	✔	✗	✗	✗	✔	✗	✗	✗
LiveVQA (Fu et al., 2025b)	✔	✗§	✗	✗	✗§	✔	✗	✗
BrowseComp (Wei et al., 2025)	✗	✔	✗	✗	✔	✗	✗	✗
FactualVQA (Wu et al., 2025b)	✔	✗	✗	✗	✔	✗	✗	✗
BrowseComp-VL (Geng et al., 2025)	✔	✔	✗	✗	✔	✗	✗	✗
MM-BrowseComp (Li et al., 2025b)	✔	✔	✔	✗	✗§	✗	✗	✗§
MMSearch-Plus (Tao et al., 2025a)	✔	✔	✔	✔	✗§	✗	✗	✗
VDR-Bench (Zeng et al., 2026)	✔	✔	✔	✗	✔	✗	✗	✗§
InterLV-Search (Hou et al., 2026)	✗	✔	✔	✔	✗§	✗	✗§	✗§
BrowseComp-V^3 (Ours)	✔	✔	✔	✔	✔	✔	✔	✔

termediate information, providing limited coverage and traceability for open-ended search trajectories. This limits their ability to systematically identify where and why an agent fails.

To address these gaps, we present BrowseComp-V^3, a benchmark comprising 300 expert-crafted questions across 24 distinct sub-domains for evaluating multimodal deep browsing and search in open-world settings. The benchmark emphasizes long-horizon, multi-level cross-modal search and reasoning, with critical evidence distributed across text and images both within and across web pages. This design reduces reliance on text-only shortcuts and models' internal parametric knowledge. Furthermore, we verify that all annotated critical evidence is retrievable through standard public search engines and provide manually annotated reference search trajectories to support fair and reproducible evaluation. Finally, each task is annotated with expert-validated intermediate sub-goals, enabling fine-grained analysis of intermediate search progress and failure modes. Our primary contributions are summarized as follows:

- We present BrowseComp-V^3, which unifies long-horizon, multi-level cross-modal search, publicly searchable key evidence, and process-oriented evaluation within a multimodal deep search benchmark.
- We systematically define and categorize multimodal deep search scenarios. Through process-oriented evaluation, we provide a more comprehensive characterization of multimodal browsing agents' capabilities and limitations.
- We develop OmniSeeker, a unified multimodal browsing agent framework. By integrating diverse web search and visual perception tools, OmniSeeker approaches the performance of leading closed-source tool-augmented MLLMs and substantially improves open-source models on multimodal deep search tasks.

2 Related Work

2.1 Multimodal Large Language Models

Multimodal large language models (OpenAI, 2025a; Pichai et al., 2025; ByteDance Seed, 2025; Bai et al., 2025) perform strongly across VQA (Fu et al., 2025a; Cheng et al., 2025; Zhang et al., 2025; Wu et al., 2025b), grounding (Kazemzadeh et al., 2014), OCR (Masry et al., 2022; Shi et al., 2025a; Mathew et al., 2021; Shi et al., 2025c), and multimodal reasoning (Lu et al., 2023; Wang et al., 2025; 2024). However, their limited ability to acquire up-to-date information hinders performance on knowledge-intensive and information-retrieval queries, motivating tool-augmented agents that dynamically retrieve and integrate external knowledge.

2.2 Tool-Enhanced Browsing Agents

Advances in LLM/MLLM tool use (Guo et al., 2025; Yang et al., 2025; OpenAI, 2025a) have driven the development of browsing agents. Recent work uses supervised fine-tuning and

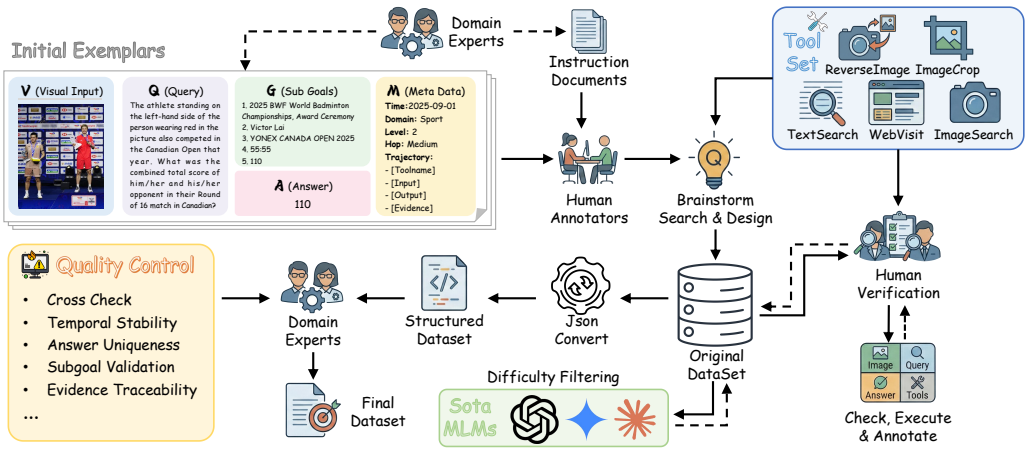


Figure 1: An overview of the data construction process of BrowseComp-V^3.

reinforcement learning to improve retrieval, reasoning, and decision-making in dynamic web environments (Jin et al., 2025; Li et al., 2025a; Wu et al., 2025a; Tao et al., 2025b; Song et al., 2025; Zheng et al., 2025). Initially developed for textual agents, these methods now extend to multimodal agents (Wu et al., 2025b; Mei et al., 2025; Hong et al., 2025; Huang et al., 2026), increasing their search depth and adaptability on complex tasks.

2.3 Multimodal Browsing Benchmarks

Early multimodal browsing benchmarks decouple visual understanding from text retrieval and focus on simple two-hop tasks (Chen et al., 2023; Mensink et al., 2023; Cheng et al., 2025; Geng et al., 2025). As performance on these tasks saturates, recent benchmarks (Li et al., 2025b; Tao et al., 2025a) introduce multi-hop designs, open-world settings, and fine-grained visual reasoning, but often place critical information in videos or non-searchable documents that standard web tools cannot retrieve. InterLV-Search (Hou et al., 2026) uses actively retrieved images as pivots for recurrent or multi-branch search from text-only information needs, but omits user-provided visual contexts and constructs its first two levels in a controlled offline environment. Process evaluation also remains limited: MMSearch (Jiang et al., 2024) evaluates stages within a fixed pipeline but does not generalize to open-ended trajectories; MM-BrowseComp (Li et al., 2025b) uses checklist signals without complete, auditable evidence chains; and VDR-Bench (Zeng et al., 2026) measures recall against gold entity sequences while missing non-entity information such as temporal constraints. BrowseComp-V^3 addresses these limitations with publicly accessible, tool-retrievable evidence and a subgoal-based Process Score that provides stronger generalizability, broader coverage, and full traceability.

3 The BrowseComp-V^3 Dataset

BrowseComp-V^3 is developed by a dedicated team of over 20 researchers, including Master's and Ph.D. candidates in artificial intelligence and related fields. The workflow follows predefined design principles and a multi-stage quality control pipeline, as described below.

3.1 Design Principles

BrowseComp-V^3 follows 3 core design principles that address key limitations of existing benchmarks in task complexity, information searchability, and evaluation dimensions.

Multi-dimensional Cross-modal Coverage. To reflect real-world search complexity, we vary tasks along two dimensions: search depth through multi-hop trajectories and cross-

modal interaction complexity across 3 levels—intra-region alignment, inter-region integration, and inter-image reasoning.

Process-oriented Granular Evaluation. We use expert-validated sub-goals to track intermediate reasoning and evidence acquisition, enabling fine-grained diagnosis of failure modes and model capability boundaries.

High Reliability and Reproducibility. For reliable and reproducible evaluation, we apply 3 filtering criteria: **(1) Evidence Traceability.** All evidence must be publicly accessible through search tools and linked to complete, manually annotated trajectories. **(2) Temporal Stability.** We prioritize objective, relatively stable knowledge to reduce sensitivity to web content drift. **(3) Answer Objectivity.** Answers must be concise and verifiable to support standardized automated evaluation.

3.2 Data Construction Pipeline

As illustrated in Figure 1, BrowseComp-V^3's construction follows a closed-loop quality assurance framework comprising 5 stages:

Stage 1: Initialization and Guideline Formulation The expert team first defines the core evaluation dimensions, including domain distribution, task hierarchy, and hop counts, and creates initial exemplars containing visual inputs, queries, sub-goals, ground-truth answers, and metadata. Together with formal instruction documents, these exemplars establish the standard for subsequent large-scale annotation.

Stage 2: Tool-Augmented Exploratory Annotation After a primary annotator drafts each question, *two* annotators with matching sub-domain expertise independently solve and label it using TextSearch, WebVisit, ImageSearch, ImageCrop, and ReverseImageSearch. Without access to each other's results, they record complete interaction trajectories, decompose the task into pivotal sub-goals, and label the capabilities required to acquire each piece of critical evidence.

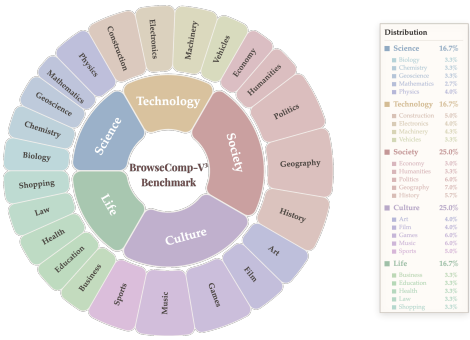
Stage 3: Multi-Stage Verification and Adversarial Filtering The resulting samples undergo three sequential screening phases. **First**, human verifiers replay the annotated trajectories and assess logical coherence, evidentiary support, and answer accuracy; failed samples are returned for revision. **Second**, the two label sets from Stage 2 are checked for consistency, and disagreements enter a consensus-based reconciliation procedure detailed in Appendix B.4. **Third**, state-of-the-art MLLMs (e.g., GPT-5.2 and Gemini-3-Pro) are used to remove questions they can solve trivially, retaining samples that require long-tail knowledge or complex reasoning.

Stage 4: Structured Data Formatting Verified samples are converted into a unified JSON schema with standardized input/output fields, sub-goals, and interaction trajectories, supporting both human inspection and automated evaluation.

Stage 5: Expert Quality Control Before release, two senior domain experts independently audit every sample for (i) content safety, (ii) privacy compliance, (iii) copyright and fair-use compliance, and (iv) domain-specific factual accuracy. Personally identifiable information is redacted from images and interaction trajectories, and only samples that pass all four audits enter the released benchmark.

3.3 Dataset Statistics

Figure 2 summarizes BrowseComp-V^3, which contains 300 questions and 383 images across 5 broad domains—Science, Technology, Society, Culture, and Life—and 24 sub-domains. The left panel shows the domain distribution, while the right reports basic statistics, task levels, and difficulty distributions.



Type	Statistic	Number
Basic Statistics	Total questions	300
	Total images	383
	Maximum question length	134
	Maximum answer length	23
	Average question length	58.58
	Average answer length	2.47
Category Statistics	Primary	5
	Secondary	24
Task Level	Level 1	89
	Level 2	140
	Level 3	71
Difficulty Distribution	Easy	45
	Medium	139
	Hard	86
	Expert	30

Figure 2: Statistics of BrowseComp-V^3. (Left) Category distribution across primary domains. (Right) Summary of statistics.

3.4 Long-term Maintenance and Reproducibility

Because BrowseComp-V^3 relies on public web evidence, we mitigate link rot and content drift through (i) Wayback-Machine and mirror-store snapshots taken at submission time, (ii) quarterly link-health audits with public `health_status` flags, (iii) versioned re-annotation of unreachable or substantially changed sources, and (iv) a public issue-reporting channel. The benchmark will be maintained on an ongoing basis through these mechanisms, with public protocols and tooling supporting continuity beyond the original maintainers. Detailed protocols, schemas, and CI configurations are provided in Appendix E.

4 OmniSeeker Framework

To support controlled evaluation across diverse MLLMs, we introduce **OmniSeeker**, a lightweight, model-agnostic multimodal browsing framework. OmniSeeker follows a ReAct-style (Yao et al., 2023) reasoning–acting loop with at most $T = 20$ turns, in which the evaluated MLLM serves as the decision-making policy and either invokes a tool or returns a final answer. It provides five browsing tools—TextSearch, WebVisit, ImageSearch, ReverseImageSearch, and CropImage—through a unified Model Context Protocol (MCP) (Anthropic, 2024) interface. All OmniSeeker instantiations share the same tool specifications, prompt template, and interaction protocol, reducing variation introduced by tool interfaces and orchestration.

OmniSeeker further connects outcome-level and process-level evaluation within the same rollout. The model incrementally reports intermediate findings using `<SUBGOAL>` tags and returns its final answer using `<FINAL_ANSWER>`. The framework collects these findings throughout the trajectory and matches them against expert-annotated reference subgoals to compute the Process Score, while the final answer is used to evaluate task success. This structured protocol captures how far an agent progresses even when it fails to produce the correct final answer, thereby supporting more fine-grained diagnosis of incomplete search trajectories. Implementation details, including the interaction loop, multimodal message format, tool back-ends, and fault-tolerance mechanisms, are provided in Appendix C.1.

5 Experiments

5.1 Experimental Setup

Evaluated Models We evaluate BrowseComp-V^{3}under six complementary settings: **(1) Human**—participants with PhD-level expertise solving each problem on the open web; **(2) Deep Research Systems**—proprietary, product-level systems evaluated through their dedicated deep-research modes, including OpenAI Deep Research (OpenAI, 2026) and Gemini Deep Research (Google, 2026); **(3) Tool-Augmented MLLMs**—proprietary models with web search and agentic reasoning capabilities, accessed through their official plat-

forms, namely GPT-5.2-Thinking (OpenAI, 2026), Gemini-3-Pro-Preview (Google, 2026), and Claude-Sonnet-4.5-Thinking (Anthropic, 2026); **(4) Tool-Free MLLMs**—leading proprietary and open-source MLLMs evaluated without external tools, including GPT-5.2 (OpenAI, 2025a), o4-mini (OpenAI, 2025b), GPT-4o (Hurst et al., 2024), Gemini-3-Flash-Preview (Google, 2025), Claude-Sonnet-4.5 (Anthropic, 2025), Doubao-Seed-1.8 (ByteDance Seed, 2025), Kimi-K2.5 (Team et al., 2026b), Qwen3.6-Plus (Qwen Team, 2026), MiMo-V2-Flash (Team et al., 2026a), Qwen3-VL-235B-A22B-Instruct (Bai et al., 2025), and Qwen3-VL-8B-Instruct (Bai et al., 2025); **(5) Open-Source Agents**—open-source multimodal browsing agents evaluated using their official implementations, namely MMSearch-R1-7B (Wu et al., 2025b), WebWatcher-7B/32B (Geng et al., 2025), and Vision-DeepResearch-8B (Huang et al., 2026); **(6) OmniSeeker**—our unified multimodal browsing-agent framework instantiated on top of the Tool-Free MLLMs above as the underlying decision policies.

Implementation Details Human participants are given 30 minutes per question. Deep Research Systems and Tool-Augmented MLLMs are evaluated through their official interfaces using the strongest publicly available research or reasoning mode, whereas Tool-Free MLLMs receive only the question text and original images. OmniSeeker and Open-Source Agents are both capped at 20 reasoning–acting turns, aligning their maximum interaction length. Detailed tool back-ends, sampling parameters, and implementation links for each setting are provided in Appendix C.3.

Evaluation Metrics We employ both outcome-level and process-level metrics. At the outcome level, **Success Rate** measures the proportion of questions for which the predicted final answer matches the ground-truth answer. At the process level, **Process Score** measures the coverage of expert-annotated critical sub-goals in the intermediate findings reported by a model or human. This metric is formally defined as:

$$\text{ProcessScore}(q) = \frac{|\hat{\mathcal{G}}_q|}{|\mathcal{G}_q|}, \tag{1}$$

where $\mathcal{G}_q$ denotes the set of reference sub-goals for question q, and $\hat{\mathcal{G}}_q \subseteq \mathcal{G}_q$ denotes the subset covered by the reported intermediate findings. Coverage is determined through semantic matching by GPT-4o and is independent of sub-goal order. Scores are computed for each question and then macro-averaged across questions (see Appendix C.4 for details).

5.2 Main Results

Based on the experimental results in Table 2, we summarize our key findings as follows:

(1) Substantial Headroom across Evaluation Settings. BrowseComp-V^{3}remains challenging across all evaluated settings. Under the 30-minute protocol, human participants achieve an SR of 68.03% and a PS of 82.93%. OpenAI Deep Research is the strongest evaluated automated system, reaching 56.00% SR, while the best Tool-Augmented MLLM reaches 39.13%. Existing open-source browsing agents (Wu et al., 2025b; Huang et al., 2026) achieve at most 11.67% SR. These results reveal substantial headroom between current automated systems and human performance.

(2) Effectiveness of Tool Augmentation and OmniSeeker. Tool-Free MLLMs achieve only 1.00%–12.00% SR, whereas OmniSeeker improves every evaluated backbone. For example, GPT-5.2 (OpenAI, 2025a) improves from 6.00% to 36.00% SR, while Doubao-Seed-1.8 (ByteDance Seed, 2025) improves from 9.00% to 33.67%. OmniSeeker with GPT-5.2 also approaches GPT-5.2-Thinking on its official platform (36.00% vs. 39.13%), demonstrating the effectiveness of the unified browsing framework.

(3) Value of Process-Level Evaluation. PS consistently exceeds SR across evaluation settings. For example, GPT-5.2 with OmniSeeker achieves 36.00% SR but 57.70% PS, while OpenAI Deep Research achieves 56.00% SR and 75.65% PS. This gap shows that systems often recover intermediate evidence without completing the task, allowing Process Score to capture progress beyond final-answer accuracy.

Table 2: **Performance on BrowseComp-V^3.** Results are reported in terms of **Success Rate** and **Process Score** under the Pass@1 setting. **Avg.** denotes the average performance, while **Sci.**, **Tech.**, **Soc.**, **Cul.**, and **Lif.** correspond to the *Science*, *Technology*, *Society*, *Culture*, and *Life* categories, respectively. Bold numbers indicate the best-performing model within each group.

Model	Success Rate (SR, %)						Process Score (PS, %)					
	Avg.	Sci.	Tech.	Soc.	Cul.	Lif.	Avg.	Sci.	Tech.	Soc.	Cul.	Lif.
Human												
Browser	**68.03**	72.00	70.00	73.33	68.00	54.00	**82.93**	87.54	85.25	84.19	82.73	74.32
Deep Research Systems												
OpenAI Deep Research	**56.00**	54.00	62.00	49.33	53.33	66.00	**75.65**	75.57	78.73	71.85	72.61	80.75
Gemini Deep Research	25.33	30.00	28.00	20.00	26.67	24.00	58.19	62.21	59.33	54.84	59.84	55.59
Tool-Augmented MLLMs												
GPT-5.2-Thinking	**39.13**	26.00	48.00	38.67	37.33	46.00	**66.05**	61.11	79.74	54.87	71.41	64.73
Gemini-3-Pro-Preview	22.90	18.00	16.00	21.33	24.00	34.00	62.43	62.02	73.78	48.70	66.33	62.50
Claude-Sonnet-4.5-Thinking	18.33	22.00	16.00	17.33	21.33	14.00	47.73	58.61	58.41	34.79	54.33	35.68
Tool-Free MLLMs												
GPT-5.2	6.00	0.00	14.00	4.00	5.33	8.00	25.02	17.50	44.50	19.22	25.25	21.43
o4-mini	7.33	0.00	16.00	2.67	6.67	14.00	29.08	29.48	46.12	17.91	29.04	28.44
GPT-4o	2.67	0.00	10.00	2.67	1.33	0.00	11.26	6.78	26.52	9.45	7.60	8.66
Gemini-3-Flash-Preview	**12.00**	8.00	18.00	12.00	10.67	12.00	**40.76**	38.98	61.94	31.89	39.72	36.60
Claude-Sonnet-4.5	4.00	4.00	6.00	2.67	5.33	2.00	25.74	33.06	47.56	15.97	24.53	13.06
Doubao-Seed-1.8	9.00	8.00	16.00	1.33	13.33	8.00	34.74	36.26	51.00	22.28	39.87	27.96
MiMo-V2-Flash	3.00	2.00	4.00	2.67	4.00	2.00	8.12	4.52	17.28	5.43	9.40	4.70
Kimi-2.5	11.33	12.00	14.00	9.33	8.00	16.00	31.55	35.79	48.85	24.51	27.52	26.59
Qwen3.6-Plus	8.00	6.00	12.00	1.33	6.67	18.00	23.03	29.59	38.08	13.09	21.63	18.44
Qwen3-VL-235B-A22B-Instruct	3.33	4.00	4.00	4.00	2.67	2.00	20.52	26.34	35.38	11.69	19.64	14.39
Qwen3-VL-8B-Instruct	1.00	0.00	2.00	1.33	0.00	2.00	6.64	3.28	15.36	6.39	4.09	5.48
Open-Source Agents												
MMSearch-R1-7B	4.00	2.00	6.00	1.34	6.67	4.00	20.61	15.17	28.07	17.07	28.08	12.68
WebWatcher-7B	4.67	2.00	8.00	4.00	5.33	4.00	17.69	22.77	26.46	13.41	19.37	7.75
WebWatcher-32B	8.67	10.00	12.00	6.67	8.00	8.00	19.40	26.57	30.84	13.04	17.98	12.45
Vision-DeepResearch-8B	11.67	8.00	12.00	13.33	16.00	6.00	15.05	9.00	19.00	15.33	17.68	12.80
OmniSeeker (Ours)												
GPT-5.2	**36.00**	50.00	28.00	33.33	33.33	38.00	57.70	67.23	55.40	49.34	60.49	58.81
o4-mini	26.00	22.00	24.00	25.33	34.67	20.00	44.66	43.70	52.11	40.73	48.72	37.94
GPT-4o	11.41	14.00	14.00	10.67	14.67	2.00	24.15	22.32	43.36	17.35	25.93	13.04
Gemini-3-Flash-Preview	23.67	32.00	24.00	18.67	25.33	20.00	47.37	50.84	68.35	40.15	43.68	39.27
Claude-Sonnet-4.5	22.67	32.00	20.00	21.33	24.00	16.00	54.17	60.94	64.25	45.29	56.73	46.78
Doubao-Seed-1.8	33.67	42.00	28.00	37.33	38.67	18.00	**58.44**	52.94	69.70	57.46	66.27	42.42
MiMo-V2-Flash	16.67	18.00	12.00	10.67	29.33	10.00	31.33	34.84	35.04	24.45	42.43	17.76
Kimi-2.5	28.00	24.00	18.00	32.00	29.33	34.00	57.34	54.64	62.20	54.47	60.33	55.08
Qwen3.6-Plus	21.00	32.00	20.00	28.00	29.33	24.00	32.43	40.90	48.19	31.01	43.78	30.46
Qwen3-VL-235B-A22B-Instruct	14.33	16.00	14.00	14.67	17.33	8.00	26.68	28.56	35.93	22.00	28.36	20.05
Qwen3-VL-8B-Instruct	5.33	2.00	2.00	9.33	8.00	2.00	13.40	8.02	19.30	15.57	15.55	6.38

6 Further Analysis

6.1 Fine-grained Analysis

Task Level As shown in Table 3, a significant performance gap exists between Level 1 and the higher-tier tasks (Levels 2 and 3). This reveals that while current models can effectively perform unitary visual search, they still face significant challenges in inter-region integration and inter-image relational reasoning across complex visual contexts during multi-hop web browsing.

Table 3: Process Score across different models and levels.

Model	L1	L2	L3
GPT-5.2	0.618	0.553	0.579
Claude-Sonnet-4.5	0.571	0.535	0.519
Doubao-Seed-1.8	0.619	0.565	0.584
MiMo-V2-Flash	0.378	0.264	0.342
Qwen3-VL-235B	0.326	0.231	0.272

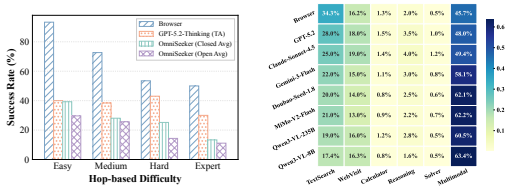


Figure 3: Difficulty and Ability Analysis

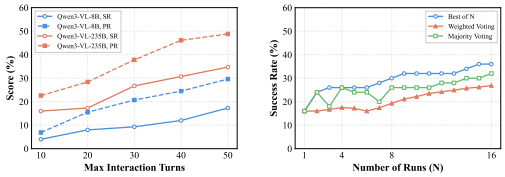


Figure 4: Test-Time Scaling

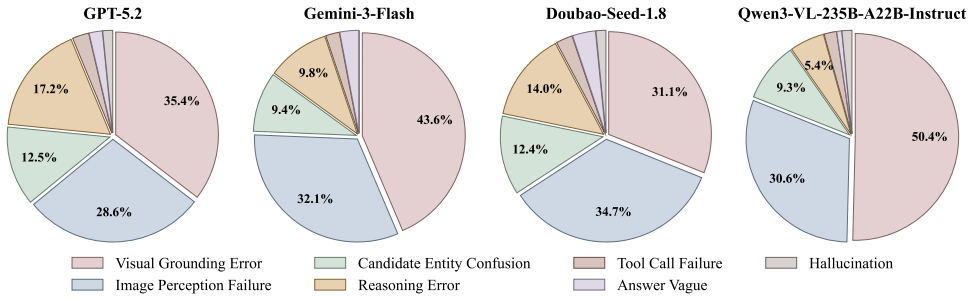


Figure 5: Failure Mode Analysis

Search Depth As illustrated in Figure 3 (Left), SR for both humans and models decline with increasing search depth, yet exhibit distinct patterns. Human performance drops sharply with longer search paths, whereas model performance declines more gradually. This discrepancy implies that models leverage internalized parametric knowledge as a compensatory mechanism to mitigate the impact of search complexity.

Ability Boundaries Figure 3 (Right) reveals distinct bottlenecks for humans and models. Human performance limitations are primarily in TextSearch, due to constraints in attention span and cognitive load when processing voluminous text. In contrast, multimodal integration remains the primary bottleneck for all models.

6.2 Test-Time Scaling

We examine how scaling test-time compute, in interaction turns and independent samples, affects performance on BrowseComp-V^3.

- **Scaling Interaction Steps.** Increasing the maximum number of interaction turns substantially improves SR (Figure 4, Left), and Qwen3-VL-235B benefits more than its 8B counterpart, suggesting that larger models make better use of additional steps for iterative refinement.
- **Scaling Sampling Consistency.** As the number of independent samples N grows (Figure 4, Right), Best-of-N scales most effectively among the three strategies for Qwen3-VL-235B, improving steadily with N.

6.3 Failure Mode Analysis

Using GPT-5.2 as an LLM-as-a-judge, we classify the trajectories of four representative models into seven error categories (Figure 5; taxonomy in Appendix C.6).

- **Multimodal Grounding and Perception.** Visual Grounding and Perception Failure dominate across all models, indicating that MLLMs still struggle to extract visual information in noisy web environments.
- **Multimodal Progress, Planning Constraint.** Closed-source models make fewer perception and grounding errors than open-source models, making long-horizon planning a more prominent bottleneck for stronger models.

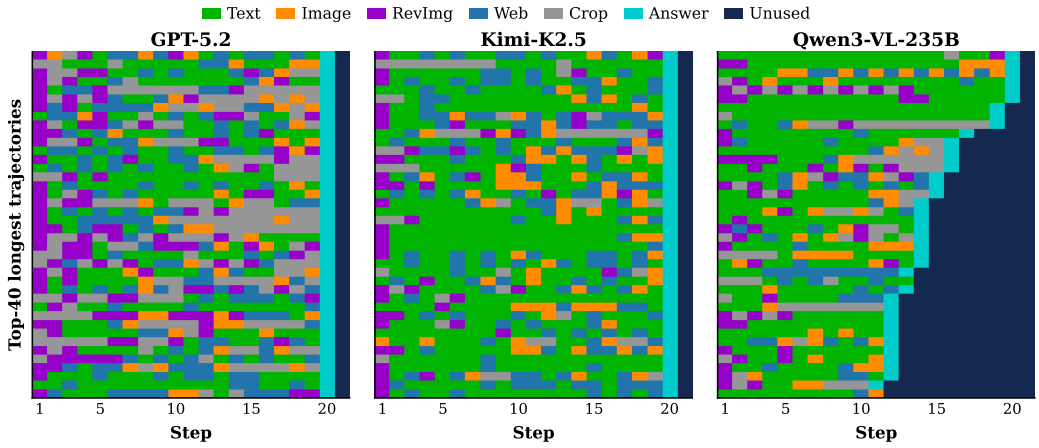


Figure 6: Episode Raster Plots for Three Representative Models.

6.4 Behavioral Trajectory Patterns

Figure 6 compares the 40 longest trajectories for each model under the same 20-turn budget. Each row shows a rollout, with colors indicating tool calls, final answers, and unused turns. GPT-5.2 uses a relatively balanced mix of tools and typically terminates near the budget limit. Kimi-K2.5 relies mainly on TextSearch and WebVisit, with sparse visual-tool use, while Qwen3-VL-235B often stops early, around steps 6–12, leaving much of the interaction budget unused. These patterns align with their SR ordering (36.0% > 28.0% > 14.3%), suggesting that fuller budget utilization and more balanced multimodal tool allocation are associated with higher task success.

6.5 Tool Ablation and Oracle Analysis

Cross-Benchmark Tool Ablation. With the same GPT-5.2 + OmniSeeker agent, full-toolset accuracy is similar on BrowseComp-V^3 and three related benchmarks (28.80%–37.00%), but only BrowseComp-V^3 drops by over 15% when any single visual tool is removed, so it jointly exercises all three visual-search capabilities as designed (Appendix A.1).

Staged Oracle Analysis. Supplying gold evidence stage by stage to GPT-5.2 + OmniSeeker, even the full evidence chain yields 50.33% SR when the target image is not identified, rising to 68.00% once it is; selecting the decisive image and extracting its visual facts thus remain major difficulties beyond retrieval (Appendix A.2).

7 Conclusion

We introduced BrowseComp-V^3, a benchmark for evaluating multimodal browsing agents on deep search. Its 300 expert-constructed questions emphasize multi-hop cross-modal search and publicly searchable key evidence, with sub-goal-based evaluation supporting fine-grained capability analysis and failure diagnosis. Current systems still have substantial room for improvement: the strongest evaluated deep-research system achieves a success rate of only 56%. Cross-benchmark tool ablations show that BrowseComp-V^3 jointly assesses models' ability to use multiple visual tools to acquire and process key evidence. Further analysis reveals that selecting key images, extracting visual facts, and integrating evidence across sources remain challenging even when complete reference evidence is provided. By evaluating both final answers and intermediate search progress, BrowseComp-V^3 provides a reusable foundation for assessing multimodal deep search, diagnosing bottlenecks in evidence acquisition and use, and guiding future improvements.

AI USE STATEMENT

We used generative AI tools to assist with grammar correction and language polishing of the manuscript, and to assist with the visual presentation of experimental results in figures. All reported results come from our own experiments. The authors reviewed all AI-assisted text and figures, checked that the figures faithfully reflect the experimental results, and take full responsibility for the content of this paper.

REFERENCES

Anthropic. Model context protocol specification. https://github.com/jlowin/fastmcp, 2024.

Anthropic. Introducing Claude Sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5, September 2025. Accessed: 2026-01-29.

Anthropic. Claude: Think fast, build faster, 2026. URL https://claude.ai/. Accessed: 2026-01-29.

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

ByteDance Seed. Seed1.8: A generalized agentic model that can efficiently and accurately accomplish complex tasks in real-world scenarios. https://seed.bytedance.com/en/seed1_8, December 2025. Accessed: 2026-01-29.

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.

Cloudflare, Inc. Cloudflare R2: S3-compatible object storage with zero egress fees. https://developers.cloudflare.com/r2/, 2022.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24108–24118, 2025a.

Mingyang Fu, Yuyang Peng, Dongping Chen, Zetong Zhou, Benlin Liu, Yao Wan, Zhou Zhao, Philip S Yu, and Ranjay Krishna. Seeking and updating with live visual knowledge. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025b.

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

Google. Gemini Deep Research. *Google blog*, 2024. URL https://gemini.google/overview/deep-research/?hl=en.

Google. Gemini 3 flash: frontier intelligence built for speed. https://blog.google/products-and-platforms/products/gemini/gemini-3-flash/, December 2025. Accessed: 2026-01-29.

Google. Gemini, 2026. URL https://gemini.google.com/. Accessed: 2026-01-29.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

Bohan Hou, Jiuning Gu, Jiayan Guo, Ronghao Dang, Sicong Leng, Xin Li, Xuemeng Song, and Jianfei Yang. Interlv-search: Benchmarking interleaved multimodal agentic search. *arXiv preprint arXiv:2605.07510*, 2026. URL https://arxiv.org/abs/2605.07510.

Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, et al. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*, 2026.

OpenAI Aaron Hurst, Adam Lerer, Aditya Ramesh, Alec Radford, et al. Gpt-4o system card. *ArXiv*, abs/2410.21276, 2024. URL https://api.semanticscholar.org/CorpusID:273662196.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Pengshuo Qiu, Pan Lu, Zehui Chen, Guanglu Song, Peng Gao, Yu Liu, et al. Mmsearch: Unveiling the potential of large models as multi-modal search engines. In *The Thirteenth International Conference on Learning Representations*, 2024.

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

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

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

Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, Chuanhao Li, Jiayi Tian, Chenchen Zhang, Tianhao Peng, Yancheng He, Jihao Gu, Yuanxing Zhang, Jian Yang, Ge Zhang, Wenhao Huang, Wangchunshu Zhou, Zhaoxiang Zhang, Ruizhe Ding, and Shilei Wen. Mmbrowsecomp: A comprehensive benchmark for multimodal browsing agents, 2025b. URL https://arxiv.org/abs/2508.13186.

Yangning Li, Yinghui Li, Xinyu Wang, Yong Jiang, Zhen Zhang, Xinran Zheng, Hui Wang, Hai-Tao Zheng, Philip S Yu, Fei Huang, et al. Benchmarking multimodal retrieval augmented generation with dynamic vqa dataset and self-adaptive planning agent. *arXiv preprint arXiv:2411.02937*, 2024b.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Lang Mei, Zhihan Yang, Xiaohan Yu, Huanyao Zhang, and Chong Chen. Ai-searchplanner: Modular agentic search via pareto-optimal multi-objective reinforcement learning. *arXiv preprint arXiv:2508.20368*, 2025.

Thomas Mensink, Jasper Uijlings, Lluis Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araujo, and Vittorio Ferrari. Encyclopedic vqa: Visual questions about detailed properties of fine-grained categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3113–3124, 2023.

Meta. Llama 4 Herd. *Meta blog*, 2025. URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

OpenAI. Introducing deep research. OpenAI blog, February 2025. URL https://openai.com/index/introducing-deep-research/.

OpenAI. Introducing GPT-5.2. https://openai.com/index/introducing-gpt-5-2/, December 2025a. Accessed: 2026-01-29.

OpenAI. Introducing OpenAI o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/, April 2025b. Accessed: 2026-01-29.

OpenAI. ChatGPT: smart and simple AI, 2026. URL https://chatgpt.com/. Accessed: 2026-01-29.

Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. A new era of intelligence with gemini 3. Google Blog, November 18 2025. URL https://blog.google/products-and-platforms/products/gemini/gemini-3/. Accessed: 2026-02-12.

Qwen Team. Qwen3.6-Plus: Towards Real World Agents. https://qwen.ai/blog?id=qwen3.6, 2026. Alibaba Qwen Studio blog. Accessed: 2026-05-06.

Yang Shi, Yuhao Dong, Yue Ding, Yuran Wang, Xuanyu Zhu, Sheng Zhou, Wenting Liu, Haochen Tian, Rundong Wang, Huanqian Wang, et al. Realunify: Do unified models truly benefit from unification? a comprehensive benchmark. *arXiv preprint arXiv:2509.24897*, 2025a.

Yang Shi, Jiaheng Liu, Yushuo Guan, Zhenhua Wu, Yuanxing Zhang, Zihao Wang, Weihong Lin, Jingyun Hua, Zekun Wang, Xinlong Chen, et al. Mavors: Multi-granularity video representation for multimodal large language model. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10994–11003, 2025b.

Yang Shi, Huanqian Wang, Wulin Xie, Huanyao Zhang, Lijie Zhao, Yi-Fan Zhang, Xinfeng Li, Chaoyou Fu, Zhuoer Wen, Wenting Liu, et al. Mme-videoocr: Evaluating ocr-based capabilities of multimodal llms in video scenarios. *arXiv preprint arXiv:2505.21333*, 2025c.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

Xijia Tao, Yihua Teng, Xinxing Su, Xinyu Fu, Jihao Wu, Chaofan Tao, Ziru Liu, Haoli Bai, Rui Liu, and Lingpeng Kong. Mmsearch-plus: A simple yet challenging benchmark for multimodal browsing agents. *arXiv preprint arXiv:2508.21475*, 2025a.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, et al. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*, 2025b.

Core Team, Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, et al. Mimo-v2-flash technical report, 2026a. URL https://arxiv.org/abs/2601.02780.

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

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond images and language. *arXiv preprint arXiv:2511.21395*, 2025.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

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

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*, 2025b.

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.

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

Yu Zeng, Wenxuan Huang, Zhen Fang, Shuang Chen, Yufan Shen, Yishuo Cai, Xiaoman Wang, Zhenfei Yin, Lin Chen, Zehui Chen, et al. Vision-deepresearch benchmark: Rethinking visual and textual search for multimodal large language models. *arXiv preprint arXiv:2602.02185*, 2026.

YiFan Zhang, Yang Shi, Weichen Yu, Qingsong Wen, Xue Wang, Wenjing Yang, Zhang Zhang, Liang Wang, and Rong Jin. Debiasing multimodal large language models via penalization of language priors. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 4232–4241, 2025.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025.

List of Appendices

A Additional Experiments and Analyses **15**

A.1 Cross-Benchmark Tool Ablations 15

A.2 Staged Oracle Analysis of Failure Sources 16

A.3 Sub-goal Stability across Rollouts 19

A.4 Human Performance under Extended Protocols 20

A.5 Detailed Results by Sub-domain 21

A.6 Agent Behavior and Trajectory Analysis 21

A.6.1 Tool Transition Matrices 21

A.6.2 Tool-Use Calibration 22

A.6.3 Sub-goal Progression 23

A.6.4 Rollout Trajectory Statistics 25

B Dataset Details and Annotation Quality **25**

B.1 Definition of Core Concepts 25

B.2 Sub-goal Statistics 27

B.3 Examples across Five Domains 27

B.4 Inter-Annotator Agreement 27

B.5 Human Annotation Cost 29

C Implementation and Evaluation Details **30**

C.1 OmniSeeker Framework 30

C.2 Tool Definitions 31

C.3 Experimental Setup 31

C.4 Evaluation 33

C.5 Prompts 34

C.6 Failure Taxonomy 43

D Case Study **43**

E Long-term Maintenance and Reproducibility **48**

F Limitations and Future Work **48**

A Additional Experiments and Analyses

This section reports four experiments that complement the main results. Section A.1 ablates visual tools under a fixed agent across four benchmarks to characterize *which visual-search capabilities* each benchmark depends on. Section A.2 supplies gold intermediate information stage by stage to localize *where* the agent loses performance. Section A.3 examines whether correct trajectories that follow different routes complete the same gold sub-goals, i.e., whether Process Score is stable across rollouts. Section A.4 evaluates humans under relaxed time and assistance protocols to examine what the human baseline measures. Unlike the failure taxonomy (Figure 5) and the trajectory statistics in Section A.6, which describe observed trajectories, the first two experiments intervene on the agent's tools or inputs while holding all other factors fixed.

A.1 Cross-Benchmark Tool Ablations

Setup. We evaluate the same GPT-5.2-powered OmniSeeker agent on BrowseComp-V^3 (300 questions), MM-BrowseComp (Li et al., 2025b) (the 243 questions with multimodal inputs), MMSearch-Plus (Tao et al., 2025a) (310 questions), and VDR-Bench (Zeng et al., 2026) (500 questions). Across all runs, we fix the OmniSeeker code version, system prompt, model configuration, tool implementations and back-ends, the $T = 20$ interaction budget, and the GPT-4o judge. Within each benchmark, we compare the full toolset with four ablations that remove ReverseImageSearch, ImageSearch, or CropImage individually, or remove all visual tools and retain only TextSearch and WebVisit. The available toolset is therefore the only factor changed within each benchmark. Each setting is run three times and we report the mean final-answer accuracy. The relative drop of ablation t is $(1 - \mathrm{Acc}_{-t}/\mathrm{Acc}_{\mathrm{full}}) \times 100\%$.

Table 4: **Cross-benchmark tool ablations with a fixed GPT-5.2 OmniSeeker agent.** Values are final-answer accuracy (%); parentheses give the relative drop from the full toolset. Boldface marks single-tool drops below 10% for readability only and does not indicate statistical significance. The last row retains only TextSearch and WebVisit.

Tool configuration	BrowseComp-V^3	MM-BrowseComp	MMSearch-Plus	VDR-Bench
Full toolset	36.00	28.80	33.55	37.00
− ReverseImageSearch	24.00 (−33.3%)	**27.16 (−5.7%)**	22.58 (−32.7%)	19.20 (−48.1%)
− ImageSearch	28.00 (−22.2%)	23.05 (−20.0%)	**32.90 (−1.9%)**	**36.20 (−2.2%)**
− CropImage	30.33 (−15.7%)	**27.57 (−4.3%)**	**31.29 (−6.7%)**	30.40 (−17.8%)
− All visual tools	15.33 (−57.4%)	15.64 (−45.7%)	23.23 (−30.8%)	18.20 (−50.8%)

Distinct capability-dependence profiles. Although full-toolset accuracy lies in a similar range across the four benchmarks (28.80%–37.00%), the single-tool ablations produce clearly different profiles (Table 4 and Figure 7). MM-BrowseComp is sensitive to ImageSearch (−20.0%) but only weakly sensitive to ReverseImageSearch and CropImage (−5.7% and −4.3%). Part of its visual evidence comes from video frames, PDFs, or specialized documents that standard image search engines may not index reliably, which may limit the utility of ReverseImageSearch. MMSearch-Plus is sensitive to ReverseImageSearch (−32.7%) but much less so to ImageSearch and CropImage (−1.9% and −6.7%), consistent with tasks that mainly require retrieving the source or entity of a given image rather than acquiring new images from textual clues. VDR-Bench is sensitive to ReverseImageSearch and CropImage (−48.1% and −17.8%) but hardly affected by removing ImageSearch (−2.2%). BrowseComp-V^3 is clearly affected by each single-tool ablation (−33.3%, −22.2%, and −15.7%). Its distinguishing property is therefore not the largest drop for every tool, but its joint sensitivity to all three visual-search capabilities. Similar full-toolset accuracy combined with distinct ablation profiles points to differences in capability coverage rather than a uniform shift in overall difficulty.

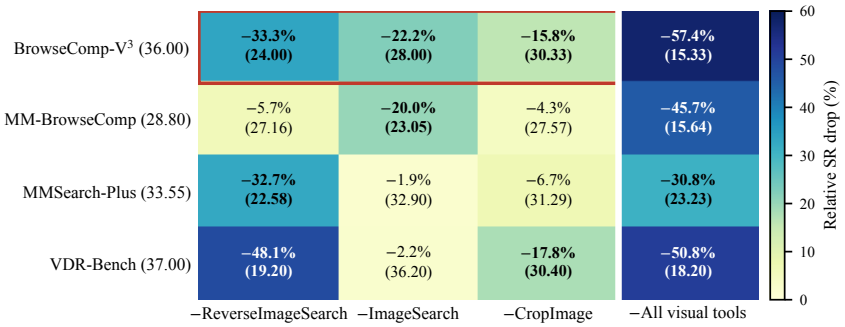


Figure 7: **Tool-dependence profiles of four benchmarks.** Each cell shows the relative SR drop when a tool is removed, with the absolute SR in parentheses; the full-toolset SR is given next to each benchmark name. Single-tool drops of at least 10% are shown in bold. The red box highlights that BrowseComp-V^3 is the only benchmark with a drop of at least 10% for all three single-tool ablations.

Interpretation. The three ablations operationalize three complementary capabilities, and hence three failure types an agent can exhibit: **image-conditioned retrieval** (ReverseImageSearch), i.e., using a given image, a retrieved image, or a local crop to find the corresponding source, entity, or related visual instance; **text-conditioned image acquisition** (ImageSearch), i.e., turning textual clues into an effective query and identifying the target visual evidence among candidate images; and **fine-grained visual localization** (CropImage), i.e., locating the decisive person, object, or region and constructing a crop for subsequent visual queries. The vector of drops thus characterizes which of these difficulties a benchmark exposes. In this comparison, the three existing benchmarks each emphasize a different subset, whereas BrowseComp-V^3 brings all three into a single evaluation. These drops reflect how much this particular agent relies on each tool; a small drop does not imply that the underlying capability is irrelevant to a benchmark, since other tools may partially substitute for the removed one. We compare benchmarks using final-answer accuracy only, because the other benchmarks adopt different trajectory-annotation schemes and do not provide gold sub-goal DAGs from which a matched Process Score could be computed. Within BrowseComp-V^3, the `key_info`, `ability`, and dependency annotations additionally align trajectories with individual evidence steps, supporting the finer-grained diagnosis in Section A.2.

A.2 Staged Oracle Analysis of Failure Sources

Motivation. End-to-end accuracy is jointly affected by retrieval strategy, planning, tool use, webpage interaction, and multimodal evidence processing, so a low SR alone does not reveal which of these is responsible. We distinguish three related but distinct abilities: **multimodal understanding**, recognizing the content of a target image once it is specified; **multimodal reasoning**, using visual and textual evidence to infer the next step and advance a multi-step evidence chain; and **multimodal integration**, selecting, aligning, and combining evidence from multiple webpages, images, and text passages in a long context to derive the answer. Understanding a specified image does not imply that a model can reliably acquire, select, and combine visual evidence during open-ended search. We therefore supply gold intermediate information stage by stage and measure how much performance is recovered.

Setup. We evaluate GPT-5.2 + OmniSeeker on the same 300 questions, fixing the system prompt, model configuration, tool back-ends, sampling parameters, the $T = 20$ budget, and the GPT-4o judge. Each setting is run three times and we report the mean, and every condition is evaluated on all 300 questions, so SR values share the same denominator. Oracle information changes only the model input; the gold sub-goal set, final-answer evaluation, and Process Score computation are unchanged. Supplied evidence is not automatically counted

as a completed sub-goal: the model must still use it and state the corresponding `key_info`. All supplied captions and `key_info` were verified not to contain the final answer.

Cumulative oracle conditions. Conditions (a)–(e) in Table 5 are cumulative: each contains all oracle information of the preceding condition and adds further gold evidence. Because the amount of supplied information differs across conditions, we use SR as the primary cross-condition metric and Process Score as an auxiliary diagnostic. Conditions (d) and (e) receive exactly the same evidence and differ only in whether the target image is identified, which makes their Process Score difference the most direct controlled comparison.

Table 5: **Cumulative staged oracle conditions** (GPT-5.2 + OmniSeeker, 300 questions, mean of three runs). ΔSR is relative to the preceding condition.

Condition	Information provided	SR	ΔSR	PS	Stages still required
(a) No oracle	Question and initial image(s) only	36.00	–	57.70	Full end-to-end search and reasoning
(b) Pre-visual text evidence	All gold text evidence before the first visual step	37.00	+1.00	57.70	Visual evidence acquisition, reasoning, integration
(c) (b) + first visual evidence	First gold visual evidence added	42.33	+5.33	58.10	Continuing the search from visual evidence; integration
(d) Full-chain evidence	Complete gold evidence and webpages; target image not identified	50.33	+8.00	60.70	Target-image selection; evidence integration
(e) (d) + target image specified	Same evidence as (d); target image identified	68.00	+17.67	82.55	Image understanding, integration, answer derivation

Matched visual-input interventions. Condition (e) still leaves 32% of questions unsolved. To separate the remaining difficulties, we construct matched interventions on top of (e), denoted F1 (Table 6). Under full-chain evidence, we replace the specified full image with the gold target crop (F2) or with a verified caption stating the task-relevant visual fact (F3). Under concise gold `key_info` instead of full-chain evidence, we again compare the gold crop (F5) with the visual-fact caption (F4). Each comparison changes a single factor.

Table 6: **Matched visual-input interventions** built on condition (e) of Table 5 (300 questions, mean of three runs). Each ΔSR compares two conditions that differ in a single factor.

ID	Visual information	Context	SR	Comparison	Factor isolated
F1	Specified full image	Full-chain evidence	68.00	–	Baseline (condition (e))
F2	Gold target crop	Full-chain evidence	73.00	F2−F1: +5.00	Within-image localization and visual distraction
F3	Gold visual-fact caption	Full-chain evidence	89.00	F3−F2: +16.00	Visual-fact extraction (full evidence)
F5	Gold target crop	Concise gold `key_info`	79.00	F5−F2: +6.00	Long-context burden (with crop)
F4	Gold visual-fact caption	Concise gold `key_info`	97.00	F4−F3: +8.00	Long-context burden (with caption)
				F4−F5: +18.00	Visual-fact extraction (concise evidence)

Findings. The results (Figure 8) localize difficulties at several stages.

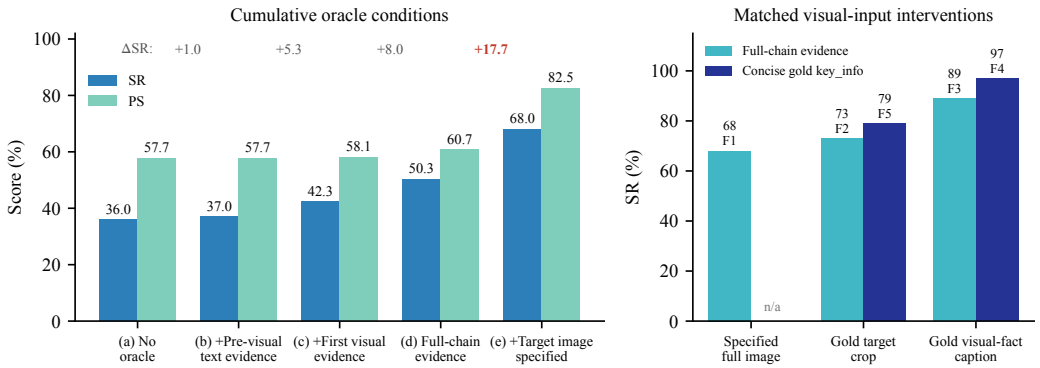


Figure 8: **Staged oracle analysis of GPT-5.2 + OmniSeeker.** *Left:* SR and PS under the cumulative conditions (a)–(e) of Table 5; numbers on top give the SR gain over the preceding condition. *Right:* SR under the matched interventions of Table 6, grouped by the form of visual information and colored by the evidence context; the full image was not evaluated with concise `key_info`.

- **An initial retrieval direction alone does not resolve the task.** Supplying all gold text evidence before the first visual step raises SR only from 36.00% to 37.00%, and adding the first gold visual evidence raises it to 42.33%. Failures therefore do not arise solely from early textual retrieval: even with the search direction and the first visual clue given, the agent still struggles to use that clue to advance the rest of the evidence chain.
- **Complete gold evidence does not close the gap.** In (d), the full evidence chain and its webpages largely remove the need for open-web retrieval, planning, and tool orchestration, yet SR reaches only 50.33% and PS 60.70%. Retrieval and orchestration are thus important but do not fully explain the remaining errors.
- **Target-image selection is a distinct difficulty.** Identifying the target image within the same evidence raises SR from 50.33% to 68.00% and PS from 60.70% to 82.55%, the largest gain in the cumulative sequence.
- **Extracting task-relevant facts from visual input is the most pronounced remaining difficulty.** Replacing the full image with the gold crop yields a moderate gain (+5 pp), showing that within-image localization and visual distraction matter. Replacing the crop with a verified visual-fact caption yields much larger gains under both full-chain evidence (+16 pp) and concise `key_info` (+18 pp). Converting visual input into facts usable for subsequent cross-modal reasoning is therefore the largest single operational difficulty we isolate for the current system.
- **Long-context evidence integration remains a further difficulty.** Compressing full-chain evidence into concise gold `key_info` consistently improves SR, from 73% to 79% with the gold crop and from 89% to 97% with the caption, indicating that selecting and integrating evidence over long contexts is an additional source of error.

Scope of the conclusions. Oracle gains measure the performance recoverable under a given intervention; they are not exact shares of each error type. A single question may fail at several stages, so we do not assign additive causal percentages to failure sources. Within this scope, the results support a qualified conclusion: under the current agentic search setting, the model continues to face substantial difficulty in selecting decisive visual evidence, extracting task-relevant facts from it, and integrating evidence from multiple sources in long contexts, beyond what retrieval and orchestration alone account for. This complements the test-time scaling results (Figure 4): additional interaction turns or samples give the agent more chances to reach relevant evidence, whereas the oracle conditions show that selecting and using visual evidence remains difficult even when the gold evidence is already in context.

A.3 Sub-goal Stability across Rollouts

Motivation. Process Score counts how many gold sub-goals a trajectory completes, regardless of the order in which they are completed or the tools used to reach them (Section 5.1). A natural concern is whether a single set of gold sub-goals can fairly score a question that admits several valid search routes. If the annotated sub-goals were tied to one particular route, correct trajectories that take a different route would complete fewer of them and receive lower scores. We test this directly by examining whether correct trajectories that follow different routes still complete the same gold sub-goals.

Setup. We draw a 50-question subset of BrowseComp-V^3 stratified by domain and difficulty and sample each question 8 times with Qwen3.6-Flash + OmniSeeker, yielding 400 trajectories. All other settings, including the $T = 20$ budget and the GPT-4o judge, follow the main evaluation. We restrict the analysis to trajectories whose final answer is correct, so that a low Process Score cannot be explained by the model simply failing to find the evidence. We then focus on questions answered correctly at least twice whose correct trajectories differ in tool-call sequence or sub-goal order, and compute the Process Score of each of these trajectories exactly as in the main evaluation.

Table 7: **Process Score of correct trajectories that follow different routes** (Qwen3.6-Flash + OmniSeeker, 50 questions × 8 samples). The last two rows are computed over the correct trajectories of the 16 route-divergent questions.

Statistic	**Value**
Sampled trajectories (50 questions × 8)	400
Trajectories with a correct final answer	98 (24.5%)
Questions answered correctly at least twice	18
of which correct trajectories differ in tool sequence or sub-goal order	16
Mean Process Score of these correct trajectories	97.1
Share of these correct trajectories with full Process Score	89.8%

Results. Among the 400 trajectories, 98 reach the correct answer, and 16 of the 18 questions solved at least twice are solved along different routes (Table 7). On these route-divergent correct trajectories, the mean Process Score is 97.1 and 89.8% of trajectories complete every gold sub-goal. Different routes therefore converge on the same set of gold sub-goals rather than being penalized for departing from the annotated one. Manual inspection of the remaining non-full cases shows that the missing sub-goals had been reached during the trajectory but were not stated in the emitted `<SUBGOAL>` tags under sampling randomness, rather than being skipped because of the route taken.

Example. Table 8 shows an Expert question from Technology/Machinery with 12 gold sub-goals organized into three evidence branches that merge into a final step: A = *RailGun/Mjölnir → Quake → Trent Reznor*, B = *San Francisco → Sharon Tate/Roman Polanski → The Pianist → Adrien Brody*, C = *GTX 980 → 2014*, and D = *The Grand Budapest Hotel → 5*. A sampled trajectory starts from a different branch and visits the branches in a different order, yet completes all 12 sub-goals and receives the same full score as the annotated trajectory.

Table 8: **Two correct trajectories with different routes on the same question.** Both complete all gold sub-goals and receive the same Process Score.

Trajectory	**Branch order**	**Completed sub-goals**	**Process Score**
Annotated trajectory	A → B → C → D	12 / 12	100%
Sampled trajectory	C → A → B → D	12 / 12	100%

Scope. This analysis covers the correct trajectories of one model on 16 route-divergent questions. It shows that, among the successful routes we observe, the gold sub-goals are completed consistently and Process Score does not depend on the order of sub-goals or tool calls. It does not establish that the gold sub-goals cover every conceivable way of solving a question; the sub-goals are annotated as the evidence that valid solutions share, and their reliability across annotators is reported in Section B.4.

A.4 Human Performance under Extended Protocols

Motivation. Under the main protocol (30 minutes per question, a standard browser, public webpages only, and no commercial LLM services; Section C.3), human participants reach an SR of 68.03% and a Process Score of 82.93% (Table 2). The large gap between the two indicates that participants often recover much of the evidence chain but do not reach the final answer within the time limit. To examine whether the human SR mainly reflects the difficulty of the tasks under these constraints, we evaluate humans on a subset of questions under relaxed protocols.

Setup. We draw a 50-question subset stratified by domain and difficulty and recruit 5 new experts who took part in neither dataset construction nor the original human baseline, together with one additional independent verifier. We compare three protocols on the same subset: **(i) Main protocol**, a re-run of the original 30-minute, no-AI setting on this subset, which serves as the within-subset control; **(ii) Extended time**, identical to (i) except that the limit is raised to 60 minutes; and **(iii) Human–AI collaboration**, in which participants may use AI assistants without a time limit, with the time spent recorded. In (iii), questions are rotated across experts so that no expert answers a question they already attempted in (ii). SR and Process Score are computed exactly as for the main human baseline.

Table 9: **Human performance under different protocols** on a stratified 50-question subset. All three protocols use the same questions; results are therefore comparable within this table but not directly with the full-benchmark human baseline in Table 2.

Protocol	Time limit	AI	SR (%)	PS (%)	Avg. time
(i) Main protocol (control)	30 min	No	52 (26/50)	70.5	26.5 min
(ii) Extended time	60 min	No	80 (40/50)	89.4	49.6 min
(iii) Human–AI collaboration	None	Yes	94 (47/50)	98.6	10.5 min

Results. Doubling the time limit raises SR from 52% to 80% and Process Score from 70.5% to 89.4% (Table 9), and participants use most of the additional time (49.6 minutes on average). With AI assistance, SR reaches 94% and Process Score 98.6%, while the average time per question drops to 10.5 minutes. The trajectories suggest a complementary division of labor: without assistance, participants spend much of their time reading long, text-heavy pages, whereas they are effective at verifying visual evidence; AI assistants filter candidate pages and entities quickly, leaving humans to confirm the decisive visual details. The remaining errors in (iii) arise mainly from AI hallucinations that misdirect retrieval or from human misjudgment among closely matching candidates.

Interpretation and scope. On this subset, relaxing the time limit or allowing AI assistance lets participants solve nearly all questions, which indicates that the answers are recoverable from public web evidence and that the human SR under the main protocol primarily reflects retrieval difficulty under the stated time and tool constraints. These results are obtained on 50 of the 300 questions with a separate group of participants, so they characterize this subset rather than establishing recoverability for every item in the benchmark.

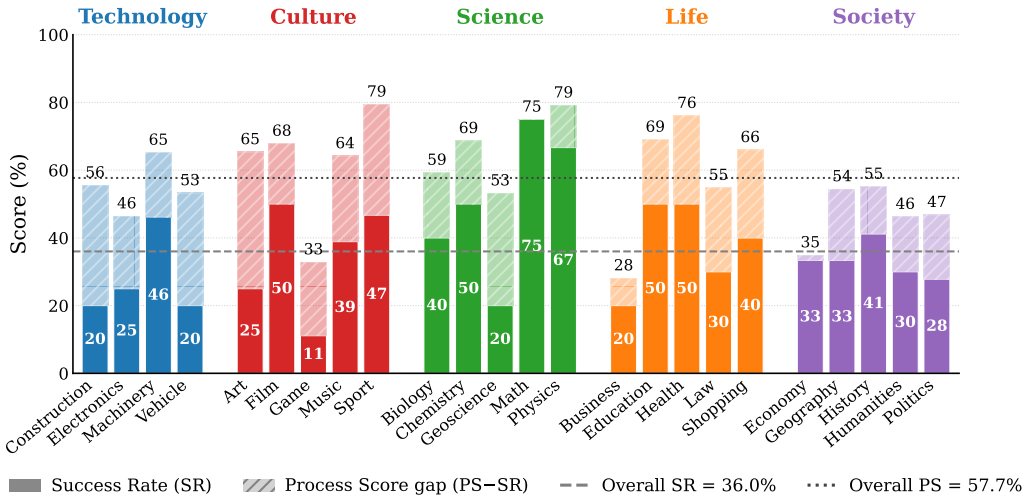


Figure 9: Performance of the OmniSeeker-augmented GPT-5.2 across all subtasks.

A.5 Detailed Results by Sub-domain

Figure 9 presents a detailed performance breakdown across all 24 sub-domains of BrowseComp-V^3 for the top-performing model GPT-5.2, reporting both Success Rate (SR) and Process Score (PS). The results reveal a balanced distribution of difficulty across these tasks: with the exception of Science/Math (SR=75.0%) and Science/Physics (SR=66.7%) —two sub-domains where symbolic reasoning alone can often suffice—no sub-domain exceeds an SR of 50%, and the majority fall within the 20%–45% range. More strikingly, we observe a systematic gap between PS and SR (overall $\Delta = +21.7$ pt), with 19 of 24 sub-domains exhibiting $\Delta \geq +17$ pt. The gap is most pronounced in Culture/Art (+40.4), Technology/Construction (+35.5), Technology/Vehicle (+33.3), Science/Geoscience (+33.2) and Culture/Sport (+32.8), suggesting that the model frequently recovers most intermediate sub-goals yet fails at the "last-mile" step of producing the exact fine-grained entity (e.g., an artwork title, a vehicle model, a building name).

A.6 Agent Behavior and Trajectory Analysis

To characterize how different models behave—rather than merely how well they score—we select three systems that span complementary points in the model-family landscape: **GPT-5.2** (the strongest proprietary model), **Kimi-K2.5** (a long-context proprietary model), and **Qwen3-VL-235B** (an open-source representative), yielding a total of 900 rollout trajectories for analysis.

A.6.1 Tool Transition Matrices

We estimate first-order transition frequencies from each rollout's tool-invocation sequence, yielding a 5×6 matrix $P(\text{next} \mid \text{current})$ per model (Figure 10); the absorbing *Answer* state appears only as a column. The matrices reveal systematic cross-model differences at several key entries. On the "search-then-read" link $P(\text{Web} \mid \text{Text})$, GPT-5.2 reaches 0.44, Kimi-K2.5 0.31, and Qwen3-VL-235B only 0.13, indicating that the last model rarely opens a webpage after receiving search results. The TextSearch self-loop $P(\text{Text} \mid \text{Text})$ climbs to 0.61 on Qwen3-VL-235B and is noticeably higher than GPT-5.2's 0.42, reflecting a tendency to rephrase keywords and re-query the same modality rather than switch tools. On the propensity to answer directly, $P(\text{Answer} \mid \text{Text})$ is 0.15 for Qwen3-VL-235B—far above the 0.04–0.05 observed on the other two models—mirroring the early-stopping texture seen in §6.4. Notably, the CropImage self-loop $P(\text{Crop} \mid \text{Crop})$ lies in the range 0.38–0.45 across *all three* models and is substantially higher than any other tool-to-itself transition, suggesting

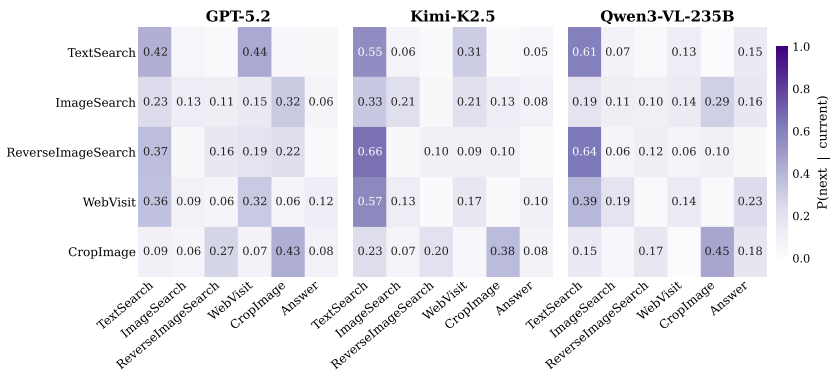


Figure 10: **One-step tool transition matrices.** For each focus model, cell (i, j) reports $P(\text{tool}_j \mid \text{tool}_i)$ aggregated over all 300 rollouts. Rows are the five search tools (each row sums to 1); columns additionally include *Answer*, corresponding to emitting `<FINAL_ANSWER>` immediately after the row tool. The *Answer* row is omitted since it is an absorbing state.

that even the strongest models often need multiple attempts to localize the correct visual region: fine-grained spatial grounding remains a shared bottleneck for multimodal browsing agents. Overall, the transition matrices provide a quantitative account of the textures observed in §6.4: **GPT-5.2 transitions fluently across modalities, Kimi-K2.5 loops within text tools but still reads pages, and Qwen3-VL-235B stays within textual self-loops and commits to an answer once evidence seems sufficient.**

A.6.2 Tool-Use Calibration

Each question in BrowseComp-V^3 is accompanied by a *human-annotated expected tool usage*, which we use to assess whether each model's tool invocations conform to the benchmark's requirements from two complementary perspectives: Figure 11 contrasts, for each tool, the model's invocation rate against the benchmark's required rate, while Figure 12 contrasts, for each question, the expected and actual totals and encodes final-answer correctness by color.

Figure 11 surfaces three markedly different tool-use profiles. GPT-5.2 meets or exceeds the expected rate on every tool, with WebVisit (+21 pt) and CropImage (+21 pt) systematically over-invoked, indicating a *comprehensive-coverage, redundant- verification* profile. Kimi-K2.5 exhibits a clearly *biased* pattern: ImageSearch (+24 pt) and WebVisit (+21 pt) are heavily over-invoked whereas ReverseImageSearch (−11 pt) and CropImage (−3 pt) are materially skipped, concentrating its surplus budget on familiar text- and web-oriented tools. Qwen3-VL-235B shows the opposite pattern: aside from TextSearch and ImageSearch, it under-invokes ReverseImageSearch (−27.5 pt), WebVisit (−33.4 pt) and CropImage (−13.6 pt), and is the only model to exhibit systematic search insufficiency.

Figure 12 then reveals whether such redundancy or insufficiency translates into final correctness. The point cloud of GPT-5.2 is consistently above $y = x$ and dominated by green, indicating that the extra search budget is converted into improved accuracy. Kimi-K2.5's cloud is more dispersed, with many dots at $y \approx 18$–20 that are predominantly red, showing that its over-invocation fails to translate into correct answers. Qwen3-VL-235B's cloud lies almost entirely below the diagonal, and the under-invoked points are overwhelmingly red, indicating that skipping benchmark-required tools directly leads to incorrect answers.

Taken together, the two figures locate the three models along a clear calibration spectrum: **GPT-5.2 exhibits disciplined over-invocation with substantial accuracy gains; Kimi-K2.5 shows a modality-biased redundancy that fails to yield commensurate returns; and Qwen3-VL-235B suffers from systematic under-invocation accompanied by a high rate of incorrect answers.** This calibration ordering aligns exactly

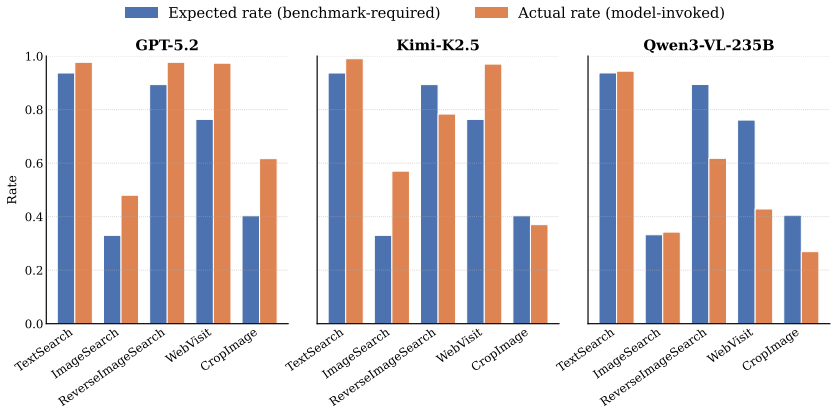


Figure 11: **Per-tool expected vs. actual usage rate.** For each model and each tool k, the *Expected rate* is $P(\text{expected}_k > 0)$, the fraction of questions whose human-annotated trajectory requires the tool, and the *Actual rate* is $P(\text{actual}_k > 0)$, the fraction of rollouts in which the model actually invoked the tool at least once.

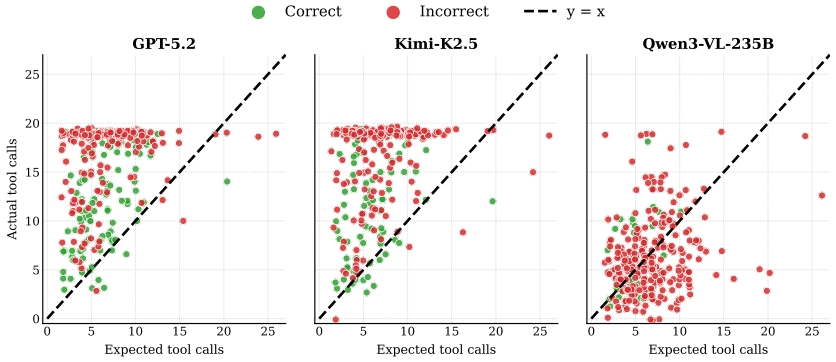


Figure 12: **Per-question total tool calls: expected vs. actual.** Each dot is one question; the x-axis is the total number of tool calls required by the human-annotated trajectory, and the y-axis is the total number of calls the model actually made. Color encodes final-answer correctness; the dashed line marks $y = x$.

with the SR ranking of the three models, indicating that the balance and sufficiency of tool use constitute a central behavioral factor in explaining cross-model performance differences.

A.6.3 Sub-goal Progression

To obtain an online estimate of how well each step progresses toward the ground-truth sub-goals, we concatenate the assistant text and tool-return text accumulated up to step t for every rollout and compute the fraction of ground-truth *key_info* tokens that have appeared in this accumulated context; the resulting curves are then averaged separately over correct (green) and incorrect (red) trajectories for each model (Figure 13). To avoid misleading averages in sparsely supported regions, step indices at which fewer than $n = 5$ trajectories are available are shown as a dashed extrapolated continuation, marked at the first sparse point by an open circle and an "$n < 5$" annotation; only the solid segments are used for quantitative comparison.

All three models exhibit an early-stage divergence between correct and incorrect trajectories. For GPT-5.2 and Kimi-K2.5, the correct curve pulls visibly ahead starting around $t \approx 3$ and quickly stabilizes above 0.80, whereas the incorrect curve plateaus in the 0.65–0.75 range; the persistent gap of 0.10–0.15 between the two indicates that **incorrect trajectories are**

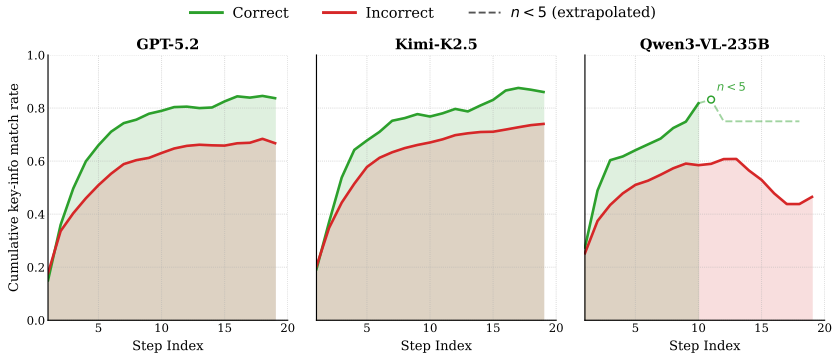


Figure 13: **Cumulative key-info match rate vs. step.** For each focus model and each rollout, we track the running fraction of ground-truth sub-goal *key_info* tokens that have appeared in the accumulated assistant and tool-return text up to step t, and average the curve separately over correct (green) and incorrect (red) trajectories. Solid segments cover steps supported by $n \geq 5$ trajectories; segments with $n < 5$ are rendered as a dashed extrapolated continuation marked by an open circle.

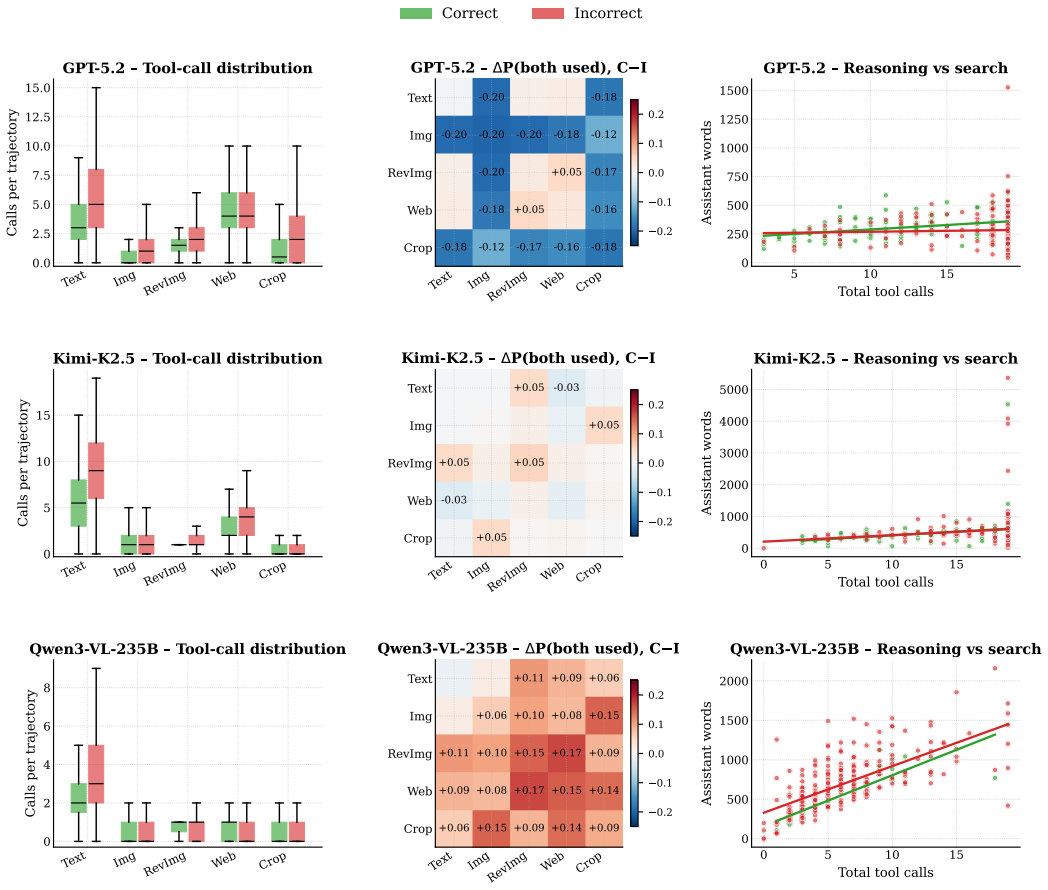


Figure 14: **Rollout trajectory statistics of the three focus models (GPT-5.2, Kimi-K2.5, Qwen3-VL-235B).** *Left:* box plots of per-tool call counts for correct (green) and incorrect (red) trajectories. *Middle:* heatmaps of pairwise tool co-occurrence difference $\Delta P = P_{\text{correct}} - P_{\text{incorrect}}$, where a positive (blue) cell indicates that the pair is more likely to be jointly used in correct trajectories. *Right:* scatter of assistant word count versus total tool calls per trajectory, with per-class OLS regression lines.

associated with insufficient early-stage evidence collection—when critical facts fail to enter the context early, subsequent steps rarely recover them. Because this analysis is correlational, it does not rule out downstream errors: the staged oracle analysis (Section A.2) shows that even with the complete gold evidence in context, SR reaches only 50.33%, so the use and integration of evidence remain additional sources of failure.

Qwen3-VL-235B departs from this pattern in two notable ways. First, the correct curve enters the dashed extrapolation region ($n < 5$) after $t \approx 10$, because **its correct trajectories almost invariably terminate within ten steps**; beyond this point the number of correct trajectories is too small for the per-step mean to be statistically reliable, so the segment is excluded from quantitative comparison. This behavior is fully consistent with the "early-stopping, thin-trajectory" texture observed in §6.4. Second, the incorrect curve exhibits a pronounced decline after $t \approx 12$, falling from 0.60 to 0.47. This indicates that when the model continues searching despite insufficient evidence, **additional steps tend to introduce irrelevant content and dilute the existing key-info density**—longer search trajectories drift further from the correct answer rather than closer to it.

Taken together, the cumulative progression curves close the loop on the preceding behavioral analyses at the information level: **the critical facts underlying a correct answer are predominantly collected within the first handful of steps, rather than recovered through late-stage reversal**; in parallel, the early termination of Qwen3-VL-235B's correct trajectories and the late-stage information dilution in its incorrect trajectories jointly corroborate the systematic under-invocation hypothesis identified in §A.6.2.

A.6.4 Rollout Trajectory Statistics

Figure 14 summarizes the trajectory-level statistics of the three focus models under the full rollout setting. The *left* column shows the per-tool call-count distributions as box plots, colored by final-answer correctness; the *middle* column shows pairwise tool co-occurrence differences ΔP between correct and incorrect trajectories, which compares which tool combinations tend to co-occur in correct trajectories; and the *right* column shows, for each trajectory, the relation between assistant word count and total tool calls, with OLS regression lines fitted separately on the two classes.

B Dataset Details and Annotation Quality

B.1 Definition of Core Concepts

Difficulty Level (Hop-based). Each sample in BrowseComp-V^3 is assigned to one of four difficulty tiers $\mathcal{D} = \{\text{Easy}, \text{Medium}, \text{Hard}, \text{Expert}\}$ according to the number of reasoning hops required to reach the correct answer. A *hop* corresponds to an atomic information-retrieval or inference operation in which the agent must consume the output of a preceding step as input and initiate a new round of search or reasoning to advance the task. Greater hop counts entail longer reasoning chains, stronger inter-step dependencies, and higher overall task difficulty. The assignment integrates four complementary factors: (i) *reasoning depth*—the number of dependent inference steps; (ii) *ambiguity*—the presence of distractors or competing hypotheses; (iii) *domain knowledge requirement*—the degree of specialized expertise needed; and (iv) *evidence integration load*—the extent to which information must be aggregated from heterogeneous sources. In BrowseComp-V^3, the four tiers are distributed as follows: Easy accounts for 15.0% (45 questions), Medium for 46.3% (139 questions), Hard for 28.7% (86 questions), and Expert for 10.0% (30 questions), forming a pyramid-shaped distribution centered on medium difficulty that balances discriminative power across model capabilities with sufficient challenge for state-of-the-art systems.

Cross-modal Complexity Level. We define three cross-modal complexity levels $\mathcal{L} = \{L1, L2, L3\}$ based on the breadth of visual–textual interaction and the coupling strength across modalities:

- **Level 1 (Intra-region Alignment).** Critical information is concentrated within a single visual region. The task requires only localized vision–language alignment (e.g.,

Table 10: Distribution of sub-goal counts by difficulty and level.

Difficulty				**Level**			
Group	Mean	Max	Min	Group	Mean	Max	Min
Easy	2.60	5.00	2.00	Level 1	4.35	11.00	2.00
Medium	4.45	7.00	2.00	Level 2	5.35	15.00	2.00
Hard	6.59	15.00	4.00	Level 3	6.10	21.00	3.00
Expert	8.90	21.00	5.00	–	–	–	–

identifying a single entity in one image), and cross-modal interaction remains at the level of weak reference.

- **Level 2 (Inter-region Integration).** Critical information is distributed across multiple distinct visual regions within the same image or page. The agent must explicitly align and synthesize information from several local areas (e.g., cropping and independently recognizing multiple sub-regions before integrating their semantics).
- **Level 3 (Inter-image Reasoning).** Critical information is dispersed across multiple independent images. The agent must perform associative cognition and compositional reasoning that spans separate visual sources (e.g., identifying a sculptured figure from one image and a geographic location from another, then jointly querying a third fact).

Progressing from $L1$ to $L3$ increases both the alignment difficulty and the compositional reasoning depth. In BrowseComp-V^3, Level 1 comprises 89 questions, Level 2 comprises 140 questions, and Level 3 comprises 71 questions, ensuring broad coverage of cross-modal interaction patterns encountered in real-world multimodal deep search.

Sub-goal. For a given task q, the sub-goal set $\mathcal{G}_q = \{g_1, g_2, \ldots, g_n\}$ is an expert-annotated sequence of critical intermediate reasoning milestones that must be achieved to solve the task. Each sub-goal g_i is formally characterized as a tuple $g_i = (\textit{description}_i,\ \textit{key_info}_i,\ \textit{ability}_i)$, where:

- **description**: The semantic objective of this intermediate step (e.g., "Identify the location shown in Figure 2").
- **key_info**: The core factual conclusion that should be obtained upon successful completion of this sub-goal (e.g., "Vatican"). This field serves as the sole criterion for determining whether the sub-goal has been achieved.
- **ability**: The set of tools or cognitive capabilities required to accomplish this sub-goal (e.g., ReverseImageSearch, TextSearch, Reasoning).

Sub-goals are connected by explicit dependency relations, forming a directed acyclic graph (DAG): $g_i \to g_j$ indicates that the execution of g_j requires the *key_info* of g_i as a prerequisite input. This structure enables a fine-grained, process-level evaluation via the *Process Score*:

$$\text{ProcessScore}(q) = \frac{|\hat{\mathcal{G}}_q|}{|\mathcal{G}_q|}, \tag{2}$$

where $\hat{\mathcal{G}}_q \subseteq \mathcal{G}_q$ denotes the subset of sub-goals successfully achieved by the model. Unlike end-to-end metrics that assess only final-answer correctness, the sub-goal mechanism enables systematic tracking of intermediate evidence acquisition, thereby permitting rigorous diagnostic analysis of failure modes and precise delineation of model capability boundaries.

Key Evidence. For a sample x, let $\mathcal{E}(x)$ denote the full set of obtainable evidence units (text spans, visual regions, tables, formulas, etc.). A subset $\mathcal{K}(x) \subseteq \mathcal{E}(x)$ is defined as *key evidence* if it satisfies the following two conditions simultaneously:

- **Necessity:** Removing any single element from $\mathcal{K}(x)$ renders the correct answer underivable or unsupported.

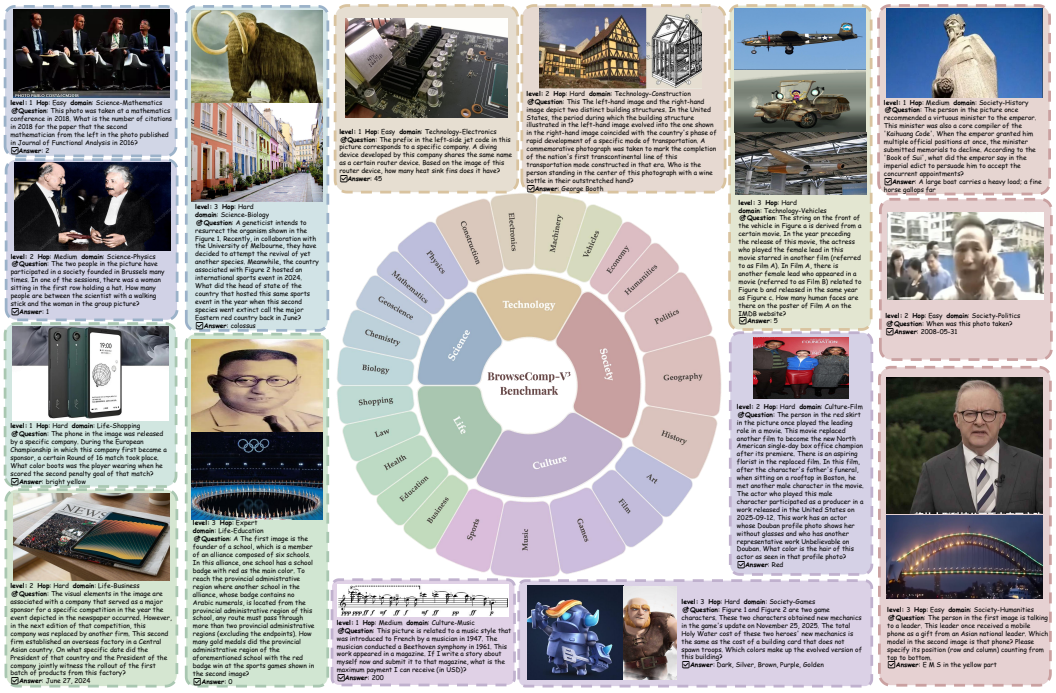


Figure 15: Representative samples from the five task categories of BrowseComp-V^3.

- **Sufficiency:** The elements in $\mathcal{K}(x)$ alone are logically sufficient to derive the correct answer.

In other words, $\mathcal{K}(x)$ constitutes a *minimally sufficient* evidence set for answering the question correctly. In addition, BrowseComp-V^3 enforces a **public searchability** constraint: every element of $\mathcal{K}(x)$ must be accessible through standard public search tools (e.g., text search engines, image search engines) without requiring access to videos, proprietary documents, or restricted resources. In annotation practice, key evidence is instantiated as the *key_info* fields of the sub-goals together with their corresponding source URLs, collectively forming a fully reproducible and auditable evidence chain.

B.2 Sub-goal Statistics

In Table 10, the number of sub-goals shows a clear increasing trend with level: the mean rises from 4.35 at Level 1 to 5.35 at Level 2 and 6.10 at Level 3, while the maximum also increases from 11 to 21. This indicates that higher-level tasks generally require longer decomposition chains. Across difficulty groups, the mean sub-goal counts for Easy, Medium, Hard, and Expert are 2.60, 4.45, 6.59, and 8.90, respectively, demonstrating a monotonic increase with difficulty. Notably, the Expert group has a maximum of 21 and a minimum of 5, suggesting that high-difficulty samples not only involve more sub-goals on average but also exhibit a substantially higher upper bound of complexity.

B.3 Examples across Five Domains

Figure 15 presents representative samples from the five task categories of BrowseComp-V^3.

B.4 Inter-Annotator Agreement

Each item in BrowseComp-V^3 is authored by a single primary annotator and then independently solved and labeled by two domain-matched annotators. The label-bearing fields comprise *difficulty tier*, *cross-modal level*, *hop count*, *answer correctness*, *evidence sufficiency*,

and *sub-goal decomposition*. To ensure that the released benchmark follows a consistent, measurable rubric despite being produced by 20+ experts, we adopt a two-phase IAA protocol combining iterative pilot calibration, a multi-annotator formal-stage pass, and a four-expert reconciliation loop.

Phase 1 – Iterative Pilot Calibration. Before formal annotation, all annotators study the instruction document and the gold exemplars introduced in Stage 1, and then enter iterative pilot rounds. In each round, 10 questions are randomly drawn from a held-out pool (non-overlapping across rounds and disjoint from the formal data) and are labeled *independently* by two annotators along all six structured fields. After each round, we compute pairwise agreement, hold a disagreement-resolution meeting on every divergent label, and update the instruction document with newly clarified boundary cases, so that successive rounds operate under an increasingly tight rubric. Pilot rounds continue until the per-field agreement satisfies our convergence criterion (Table 11). The protocol converges in 5 rounds, yielding a pilot-stage agreement of $\kappa_{\text{pre}} = 0.90$ averaged across the six fields. Only annotators who have completed pilot convergence are admitted to formal annotation.

Phase 2 – Formal-Stage Multi-Annotator Labeling. During formal annotation, two domain-matched experts independently solve every question, record their trajectories, and label all six fields, yielding two parallel label sets per question across the entire 300-question benchmark. Per-field first-pass agreement, denoted $\kappa_{\text{first-pass}}$, is computed from these paired labels and constitutes our primary measure of annotation-system reliability. Crucially, $\kappa_{\text{first-pass}}$ is computed *before* any reconciliation, so that it reflects the intrinsic clarity of the rubric rather than post-hoc adjudication.

Four-Expert Reconciliation Loop. Whenever any field of any question receives divergent first-pass labels, the question enters an arbitration loop. *Two additional senior annotators* from the same domain—disjoint from the original two—are recruited to independently re-label the question. The four experts then convene a structured reconciliation meeting in which the discussion proceeds field by field, returning to the instruction document and the underlying evidence as needed, until they produce a *unanimously signed-off* label per disputed field. If unanimity cannot be reached, the question is escalated to the senior expert panel (Stage 5) for terminal adjudication. A question is admitted to the released benchmark only after every disputed field has been resolved to unanimous consensus, so the released dataset attains a 100% post-arbitration consensus rate by construction.

Per-Field Thresholds. We deliberately impose a strict, field-specific threshold strategy, reflecting our position that release-grade benchmarks should permit no ambiguity on verifiable facts and minimal ambiguity on judgment-based labels. Objective fields (answer correctness, hop count, evidence sufficiency) are required to reach near-perfect agreement ($\kappa \geq 0.95$); subjective ordinal fields (difficulty tier, cross-modal level) are required to reach almost-perfect agreement ($\kappa \geq 0.90$), corresponding to the upper bound typically attainable for ordinal subjective annotation. The set-valued sub-goal field is evaluated under a semantic F1 anchored on the ground-truth `key_info`, with paraphrastic equivalence resolved by GPT-4o-judged matching, and is required to reach $F_1 \geq 0.90$. This agreement measures whether two annotators identify the same sub-goals for a question; whether solvers following different routes complete these sub-goals is examined separately in Section A.3.

Results. Table 11 reports per-field statistics. Formal-stage agreement ranges from 0.91 to 0.96 across all six fields, and $\kappa_{\text{first-pass}} \geq \kappa_{\text{pre}}$ holds on every dimension—confirming that the iterative pilot effectively transmitted a unified standard to the full annotation team and that the boundary cases accumulated during pilot rounds further sharpened the rubric. Approximately 32% of questions trigger the arbitration loop on at least one field, of which roughly 3% require escalation to the senior expert panel; the average reconciliation meeting converges in 1.2 rounds. These numbers indicate that the rubric is essentially clear at the population level while still admitting a non-trivial fraction of legitimately ambiguous boundary cases, which the four-expert reconciliation loop is designed to absorb.

Table 11: Per-field IAA thresholds and statistics for BrowseComp-V^3. *Threshold* reports the convergence target adopted in the iterative pilot; κ_{pre} is the pilot-stage agreement at convergence; $\kappa_{\text{first-pass}}$ is the formal-stage first-pass agreement on all 300 questions before any reconciliation; *Arb. rate* is the field-level fraction of questions entering the four-expert reconciliation loop. After arbitration, every question reaches unanimous consensus, so the post-arbitration consensus rate is 100% by construction.

Field	**Type**	**Metric**	**Threshold**	κ_{pre}	$\kappa_{\text{first-pass}}$	**Arb. rate**
Answer correctness	Objective binary	Cohen's κ	≥ 0.95	0.94	0.96	3.0%
Hop count	Objective count	Pearson r	≥ 0.95	0.93	0.96	3.0%
Evidence sufficiency	Objective binary	Cohen's κ	≥ 0.95	0.92	0.95	5.0%
Sub-goal decomposition	Set-valued	Semantic F_1 (GPT-4o)	≥ 0.90	0.88	0.92	10.0%
Difficulty tier	Subjective ordinal	Weighted Cohen's κ	≥ 0.90	0.87	0.91	9.0%
Cross-modal level	Subjective ordinal	Weighted Cohen's κ	≥ 0.90	0.86	0.91	10.0%
Mean (6 fields)	—	—	—	0.90	0.94	6.7%

In summary, $\kappa_{\text{first-pass}}$ characterizes the intrinsic clarity of the rubric, the arbitration rate characterizes the distribution of legitimate boundary cases, and the post-arbitration consensus rate characterizes the release readiness of the dataset. Together, these three quantities provide an orthogonal characterization of annotation consistency that is more informative than any single agreement number.

B.5 Human Annotation Cost

The construction of BrowseComp-V^3 involves two distinct human-in-the-loop processes, for which we report compensation costs separately.

(1) Trajectory and sub-goal annotation. As described in Section 3, the benchmark is constructed by a dedicated team of **more than 20 Master's and Ph.D. researchers** in artificial intelligence and related disciplines. Unlike single-point Q&A verification, each sample in BrowseComp-V^3 requires the annotator to (i) record a complete tool-augmented browsing trajectory, (ii) decompose the trajectory into fine-grained sub-goals, and (iii) annotate the key information required at every intermediate step. Because this protocol is substantially more labor-intensive than conventional Q&A labeling (typically paid at \$1/question in prior work), we compensated annotators at the equivalent of **\$6 per question**, amounting to approximately **\$1,800** for the 300-question benchmark.

(2) Human-baseline collection. To estimate human performance, we additionally recruited **12 participants with Ph.D.-level expertise**, none of whom had been involved in the construction pipeline. Each question was assigned a 30-minute time budget, and participants were allowed to browse publicly accessible web resources but forbidden from consulting any commercial LLM service. The per-question compensation was **\$2**, yielding a total of approximately **\$600** across the 300 questions.

(3) Facilities and infrastructure. In addition to per-question compensation, we provided free meals, a comfortable working environment, and all necessary computing equipment—including GPU workstations as well as GPT-4o API credits consumed during the verification loop—at an estimated cost of **\$1,000**.

Total. Summing the three components above, the total human-related expenditure for constructing BrowseComp-V^3 is approximately **\$3,400** in US dollars.

Algorithm 1: OmniSeeker Agent Loop

Input : question q, images I, max turns T, tool set $\mathcal{T}$
Output : final answer a, sub-goal list S

msgs $\leftarrow$ [SystemPrompt(), User(q, I)]
for $t \leftarrow 1$ **to** T **do**
  if $t < T$ **then**
    tools $\leftarrow \mathcal{T}$
  else
    tools $\leftarrow \varnothing$ `// force textual answer on the last turn`
  $r_t \leftarrow$ LLM.generate(msgs, tools)
  if <FINAL_ANSWER> $\in r_t$ **then**
    $S \leftarrow$ parse all <SUBGOAL>…</SUBGOAL> accumulated so far
    $a \leftarrow$ parse <FINAL_ANSWER>…</FINAL_ANSWER> from r_t
    return (a, S)
  if r_t *contains a tool call* **then**
    $(\tau, \text{args}) \leftarrow$ the tool call in r_t
    obs $\leftarrow$ MCP.execute(τ, args)
    msgs $\leftarrow$ msgs $\cup$ { r_t, ToolObs(τ, obs) }
  else
    msgs $\leftarrow$ msgs $\cup$ { r_t }
return ("Failed", S)

C Implementation and Evaluation Details

C.1 OmniSeeker Framework

Framework Overview. OmniSeeker is a lightweight agent framework that we develop to support a unified evaluation of multimodal browsing agents. It organizes multi-turn reasoning–acting interactions in a ReAct-style loop (Yao et al., 2023), in which the evaluated large model serves as the decision-making policy while a curated set of multimodal tools is exposed through the Model Context Protocol (MCP) (Anthropic, 2024). The design goal is to provide *identical* tool interfaces, prompt templates, and interaction protocols across all evaluated models, so that cross-model comparisons faithfully reflect intrinsic model capability rather than idiosyncrasies of the tool back-end or the orchestration strategy. The overall interaction logic is summarized in Algorithm 1.

Multi-turn Interaction Loop. For each question, OmniSeeker allows the agent to perform at most $T = 20$ turns of interaction. In every turn, the model produces a response based on the current dialogue history, optionally accompanied by a tool invocation; the tool's execution result is then appended to the dialogue as a new observation that informs the next decision. When the model decides that the gathered evidence is sufficient, it emits a final answer in a dedicated tag and the loop terminates. If the turn budget is exhausted without a final answer, the last turn disables tool access and forces the model to produce a textual answer conditioned on the current context.

Structured Extraction of Answers and Sub-goals. To enable the process-level evaluation described in Section 5.1, OmniSeeker requires the model to *incrementally* emit intermediate sub-goals as XML-tagged segments (`<SUBGOAL>…</SUBGOAL>`), and to output its final answer inside a dedicated tag (`<FINAL_ANSWER>…</FINAL_ANSWER>`). After every turn, the framework applies pattern matching to the assistant message: once a `<FINAL_ANSWER>` tag is detected, the final answer is extracted and all previously emitted `<SUBGOAL>` entries along the trajectory are collected as the sub-goal sequence associated with that rollout, which is subsequently used for joint outcome-level and process-level evaluation.

Multimodal Message Format. To keep the multimodal input specification consistent across models, OmniSeeker adopts a *placeholder-driven* image injection scheme. Question-side images are embedded into the user message in order as `<image_1>`, `<image_2>`, …, whereas images produced in the middle of the trajectory (e.g., results of image search or cropped sub-regions) are tagged as `<obs_1>`, `<obs_2>`, …, so that the model can explicitly distinguish *question evidence* from *retrieved evidence*. Images are injected inline by default; crops generated on the fly can additionally be uploaded to a cloud object store (Cloudflare, Inc., 2022) in order to obtain a publicly accessible URL, which is then fed seamlessly into the downstream reverse image search. This "crop–reverse-search" closure is a key enabler for fine-grained visual entity recognition.

Fault Tolerance and Termination. Tool-side exceptions are handled under an "inject-and-continue" policy: error messages are wrapped as observations and returned to the model so that it can self-diagnose and adapt its strategy, rather than aborting the rollout. Model-side exceptions, such as empty responses or malformed outputs, trigger up to two retries; if no valid response is produced thereafter, the sample is recorded as failed. Together these mechanisms prevent long-tail anomalies from contaminating the aggregate statistics.

C.2 Tool Definitions

Tool Suite Overview. OmniSeeker equips the agent with a standardized suite of five multimodal browsing tools: **TextSearch**, **WebVisit**, **ImageSearch**, **ReverseImageSearch**, and **CropImage**. The first four mirror mainstream information-seeking behaviors on the open web, while CropImage enables fine-grained exploitation of visual evidence. All tools are exposed to the evaluated model through a uniform MCP interface, ensuring that every model operates under identical tool specifications. Table 12 summarizes the input, output, back-end, and key notes of each tool.

Back-ends and Default Parameters. For the retrieval-oriented tools (TextSearch, ImageSearch, ReverseImageSearch), we route all queries through Serper as a Google Search API proxy so that results remain consistent with Google Web search in both ranking and content. For page access, we employ Jina Reader to convert HTML into clean, noise-free long-form text, preventing stylistic clutter from interfering with the model's reading. For image cropping, we perform the pixel-level operation locally and optionally upload the result to a cloud object store so that it can be re-consumed by the downstream reverse image search. All retrieval-oriented tools return the top $k = 5$ results by default, balancing recall against context length.

Tool-Chain Closure. The five tools are complementary in capability: TextSearch and ImageSearch support coarse-grained retrieval; WebVisit enables deep reading of page content; and CropImage paired with ReverseImageSearch forms a "zoom-in-then-locate-source" closure for visual retrieval. This closure is especially critical for sub-domains that demand fine-grained visual entity recognition, such as artwork titles, vehicle models, and building names.

C.3 Experimental Setup

Evaluation Scale and Sampling. We evaluate all models on the full set of 300 questions in BrowseComp-V^3. Each question is handled by an independent rollout, rollouts are fully isolated from one another to guarantee reproducibility, and failed or timed-out samples are recorded as such without retry. For every model we adopt the *officially recommended* sampling and generation parameters released by the corresponding provider, without imposing any unified override on parameters such as `temperature` or `top_p`.

Human Baseline. Each participant has up to 30 minutes per question with access to a standard web browser and publicly accessible resources only; participants are forbidden from consulting any commercial LLM service. If a participant cannot reach a reliable conclusion within the time limit, they may terminate the task and document their key exploration

Table 12: Tool specifications in the OmniSeeker framework. All retrieval-style tools return the top $k = 5$ results by default.

Tool	Input	Output	Backend	Notes
TextSearch	Natural-language `query`	List of items with `title` / `url` / `snippet`	Serper (Google Search)	Top-k=5, region=us
WebVisit	Target `url`	Cleaned page text with embedded image URLs	Jina Reader	Single page per call
ImageSearch	Natural-language `query`	List of items with `title` / `link` / `image_url` / `thumbnail`	Serper (Google Images)	Top-k=5
ReverseImageSearch	Image placeholder or image URL	List of visually similar pages and source metadata	Serper (Google Lens)	Top-k=5; Base64 auto-uploaded to obtain a public URL
CropImage	Image placeholder + crop box (l, t, r, b)	Cropped sub-image (inline or public URL)	Local image processing + cloud object store	Cropped images cannot be re-cropped

steps as a partial trajectory. Results under a longer time limit and with AI assistance are reported in Section A.4.

Deep Research Systems. For OpenAI Deep Research and Gemini Deep Research, we submit each question through the corresponding official product interface with its dedicated deep-research mode enabled. These systems are evaluated as end-to-end, product-level research services: their internal models, search tools, planning strategies, and orchestration pipelines are left unchanged. Each session is allowed to run until the system returns its final research response.

Tool-Augmented MLLMs. For GPT-5.2-Thinking, Gemini-3-Pro-Preview, and Claude-Sonnet-4.5-Thinking, we interact with each service through its official web platform with the most advanced reasoning mode enabled, so the comparison reflects each vendor's strongest publicly accessible agentic configuration. We do not impose any turn or wall-clock budget on these settings; every session is allowed to run until the model emits its final answer.

Tool-Free MLLMs. Each model receives only the question text and the associated original images and must directly produce the final answer; no external tool access, no in-context retrieval results, and no scratchpad images are provided.

Open-Source Agents. For each of the four open-source agents—MMSearch-R1-7B (Wu et al., 2025b)[1], WebWatcher-7B and WebWatcher-32B (Geng et al., 2025)[2], and Vision-DeepResearch-8B (Huang et al., 2026)[3]—we reuse the officially released implementation and follow the tool-use and interaction logic prescribed in the corresponding paper; hyperparameters and prompt templates are set as specified by the original authors. Each agent is capped at $T = 20$ reasoning–acting turns per question, matching the OmniSeeker budget.

OmniSeeker. OmniSeeker organizes each rollout as a ReAct-style loop with at most $T = 20$ reasoning–acting turns per question. The agent is exposed to a fixed suite of five multimodal tools—TextSearch, WebVisit, ImageSearch, ReverseImageSearch, and CropImage—whose framework-level behavior is detailed in Section C.1 and whose individual interfaces are specified in Section C.2.

[1] https://github.com/EvolvingLMMs-Lab/multimodal-search-r1
[2] https://github.com/Alibaba-NLP/DeepResearch/tree/main/WebAgent/WebWatcher
[3] https://github.com/Osilly/Vision-DeepResearch

Fair Comparison across Agent Frameworks. For a fair cross-framework comparison, both OmniSeeker and the Open-Source Agents use Serper[4] for text and image search and Jina[5] for webpage retrieval, with a maximum of $T = 20$ reasoning–acting turns per question.

C.4 Evaluation

Evaluation Dimensions. We adopt a joint outcome-level and process-level evaluation protocol (see the formal definitions in Section 5.1 of the main text). At the outcome level, the *Success Rate* (SR) measures whether the final answer matches the ground truth. At the process level, the *Process Score* (PS) quantifies the extent to which the model fulfills the critical sub-goals along its multi-step search–reasoning trajectory. Together, the two metrics characterize the gap between "answering correctly" and "reasoning correctly", enabling fine-grained diagnosis of failure modes.

LLM-as-a-Judge. We use GPT-4o as the judge model. For each rollout, the judge receives the ground-truth answer, the ground-truth sub-goal list (each carrying a `key_info` tag), the model's final answer, and the incremental `<SUBGOAL>` sequence produced along the trajectory. It then emits a structured evaluation record containing (i) a binary `answer_correct`; (ii) a continuous `process_score`$\in [0, 1]$; (iii) a 0/1 vector `subgoal_vector` aligned with the ground-truth sub-goals; and (iv) a `confidence` level in $\{$`HIGH`, `MEDIUM`, `LOW`$\}$.

Process Score Computation. The process score is defined as the fraction of *covered* sub-goals. For each ground-truth sub-goal, the judge performs a semantic match against the model's sub-goal sequence to decide whether the associated `key_info` is covered; the process score is then the number of covered sub-goals divided by the total number of ground-truth sub-goals. Compared with token-overlap metrics, this formulation is more robust to surface-form variation and more faithful to the underlying question of whether the model's reasoning actually captures the key evidence. Matching is content-based rather than position-based: a ground-truth sub-goal counts as completed if any emitted sub-goal matches it, regardless of the order in which sub-goals appear or the tools used to reach them, and additional emitted sub-goals neither add credit nor enter the denominator. Reported averages are question-level macro-averages: each question's score is computed first and every question receives equal weight, both over the full benchmark and within each category or difficulty subset. Human Process Scores are computed in the same way, with the judge applied to the exploration steps recorded by each participant. The stability of this score across rollouts that follow different routes is analyzed in Section A.3.

Human-in-the-Loop Anomaly Detection. Beyond the automatic metrics, the judge additionally runs a positive anomaly check on every rollout whose *final answer is correct*, in order to identify cases where "the answer is right but the process is not grounded". Such anomalies include: (a) the answer is an unsupported guess not backed by the emitted sub-goals; (b) the reasoning chain is visibly inconsistent; or (c) the model's sub-goals directly contradict the ground-truth `key_info`. Rather than automatically flipping such samples to `incorrect`, our evaluation system tags them with an `anomaly_flag` and **routes them to human adjudication**. A human reviewer then inspects the full trajectory, the associated tool invocations, and the ground-truth sub-goals, and renders the final correct/incorrect verdict. This human-in-the-loop mechanism avoids the misclassifications that rigid rule-based overrides can introduce, while keeping the human effort focused on the small subset of genuinely ambiguous cases.

[4] https://serper.dev
[5] https://jina.ai

C.5 Prompts

System Prompt

```
============================================================
SYSTEM PROMPT: MULTI-MODAL SEARCH & REASONING AGENT
============================================================

ROLE
----
You are a Multi-Modal Search & Reasoning Agent equipped with
capabilities in multimodal search, analysis, and reasoning.


============================================================
REACT FRAMEWORK
============================================================

You must operate using a ReAct-style reasoning and action loop.
The reasoning process must be explicit, structured, and visible
in text. Hidden or internal reasoning is strictly forbidden.

Required loop:

  1. Think step-by-step about what information is missing.
  2. Decide whether a tool is required.
  3. Call a tool to act.
  4. Observe the tool result.
  5. Continue reasoning and potentially call additional tools.
  6. Repeat this Thought -> Tool -> Observation cycle until you
     reach a verifier-supported conclusion.

You must not skip the loop or jump to the final answer directly.
Reasoning must precede every tool call. You may not produce a
final answer until the needed evidence has been collected from
tools.


OVERALL GOAL
------------
  - Use tools to collect verifiable information.
  - Perform cross-modal analysis and reasoning based on tool
    outputs.
  - Explicitly tag key intermediate reasoning results to support
    process-level scoring.
  - Provide concise, deterministic, and reviewable final answers.


============================================================
AVAILABLE TOOLS
============================================================

1. TextSearch
   Searches the web for relevant information based on a text query.

   Input
     - query: text query

   Return
     - List of search results, each including:
         title, link, snippet


2. WebVisit
```

```
   Visits a specified webpage and returns structured text and
   images on the page.

   Input
     - url: target webpage URL

   Return
     - The main textual content of the page and available image
       information (URL / description)


3. ImageSearch
   Searches for related images based on a text query.

   Input
     - query: text query

   Return
     - List of images, each including:
         title, thumbnailUrl, link


4. ReverseImageSearch
   Performs reverse searches for images based on similarity to
   confirm origin or semantics.

   Input
     - A tagged image <image_k> or <obs_i>, or a direct image_url

   Return
     - List of related webpages or matching images, each including:
         title, thumbnailUrl, link


5. CropImage
   Crops a designated region from a tagged image to focus on key
   information, supporting more accurate image understanding and
   image search.

   Input
     - crop_config:
         A dictionary mapping existing image tokens to crop
         coordinates.

   Return
     - The cropped image, in a new <obs_i> form.

   Constraints
     - Cropped images cannot be cropped again.
     - Crop coordinates must be integers.
     - Token keys must be EXISTING image tokens from the
       conversation. Refer to "Image Token Rules" below.

   Format Example
     Single crop:

       {
         "crop_config": {
           "<image_1>": [100, 200, 500, 600]
         }
       }

   CRITICAL RULES
```

```
     1. Dictionary keys MUST be existing image tokens
        (<image_1>, <image_2>, <obs_1>, <obs_2>, etc.) that
        already appear in the conversation. See "Image Token
        Rules" below.
     2. Do NOT invent new token names like "thumb_1",
        "crop_region", "dumpster_area", or any custom identifiers.
     3. Do NOT use nested objects. The format MUST be a flat
        dictionary:
          {"<existing_token>": [left, top, right, bottom]}
     4. Each value must be a list of exactly 4 integers:
          [left, top, right, bottom]
        NOT a nested object like {"coordinates": [...]} or
        {"token": "...", "coordinates": [...]}.
     5. Do NOT add extra metadata fields like "type", "properties",
        "code", "human_readable_name", etc.


============================================================
IMAGE TOKEN RULES
============================================================

Input images to the task:
  - <image_1> ... </image_1>
  - <image_2> ... </image_2>

Tool-generated images:
  - <obs_1> ... </obs_1>
  - <obs_2> ... </obs_2>

Tool usage requirements:
  - Explicitly specify correct image tokens or URLs.
  - Reuse existing image tokens to avoid redundant searches.


============================================================
TOOL-USE STRATEGY
============================================================

  - Do NOT call multiple tools in the same turn.
  - Breadth-first then narrow:
      first build a candidate fact space, then gradually narrow
      the scope.
  - Cross-verification:
      when ambiguity exists, a second source or image evidence
      must be employed.
  - Visual priority:
      for image-related questions, relevant images must be
      processed and referenced.
  - Efficiency:
      avoid redundant queries and fully utilize existing results.
  - Image focus:
      when searching only part of an image or focusing on a
      region, use the CropImage tool.
  - If no new useful information can be obtained from further
    tool calls, conclude the task using available evidence or
    return the mandatory failure output.
  - Per interaction turn, only one tool may be invoked. Never call
    multiple tools in the same assistant turn. If another tool is
    needed, wait for the next turn after receiving the previous
    tool's output.
  - You can only view the images returned by a tool call at the
    moment that tool responds. Images returned in earlier turns
    will be replaced with their corresponding titles in the
```

```
    conversation history.


============================================================
SUBGOAL TAGGING
============================================================
For Process-Level Evaluation

TAGGING PRINCIPLES
------------------
Use <SUBGOAL> and </SUBGOAL> tags to wrap key intermediate
reasoning conclusions or confirmed important information.

<SUBGOAL> tags are intended to capture fact-level, verifiable
intermediate conclusions, such as name, time, location, entities,
events, relations, and quantities, that directly support the final
answer. Sub-goals must not include guesses, common-sense
assumptions, or reasoning shortcuts.

<SUBGOAL> tags should be generated dynamically and incrementally
throughout the reasoning process. As soon as an intermediate fact
or validated insight is established, regardless of which reasoning
step you are in, output the corresponding <SUBGOAL> tag
immediately. Do not wait until the final response turn to output
all subgoals.

Before outputting <FINAL_ANSWER>, you must output ALL subgoals
that have not been output yet.


STRICT REQUIREMENTS
-------------------
Each subgoal must be:
  - An intermediate reasoning result or a fact confirmed through
    tool usage.

Each subgoal must not be:
  - Operational steps, tool invocation descriptions, or low-level
    thoughts.

Additional constraints:
  - Typically one sentence only, marking nodes of key information
    directly supporting the final answer.
  - Avoid redundancy, repetition, or irrelevant tagging.
  - Do NOT output <SUBGOAL> in the final turn unless it appears
    BEFORE the <FINAL_ANSWER>.

Examples:
  <SUBGOAL>Confirm that the building in the image is the National Stadium
(Bird's Nest) in Beijing.</SUBGOAL>
  <SUBGOAL>Confirm that construction of the National Stadium in Beijing began
on December 24, 2003.</SUBGOAL>


============================================================
FINAL ANSWER FORMAT
============================================================
Mandatory

  - Wrap the final conclusion (maximum 1-2 sentences) strictly
    within:
      <FINAL_ANSWER>Your final answer</FINAL_ANSWER>
```

```
  - The <FINAL_ANSWER> content must not contain reasoning steps,
    subgoals, or source citations.

  - If you determine that the required final answer cannot be
    found through available tool outputs, you must return:
      <FINAL_ANSWER>Failed. I cannot answer this question.</FINAL_ANSWER>

  - Do not fabricate or infer information beyond verified evidence.
  - Do NOT output multiple <FINAL_ANSWER> blocks.


============================================================
ANSWERING RULES
============================================================

  1. All factual information must be based on tool outputs.
  2. When citing webpage information, specify the source (URL).
  3. Place reasoning process and subgoal tags outside the final
     answer.
  4. Final answers must be concise and clear.
  5. You have at most 20 interaction rounds and must provide the
     final answer before round 20. If this cannot be done, you
     must declare failure using the required final-answer format:
       <FINAL_ANSWER>Failed. I cannot answer this question.</FINAL_ANSWER>
```

Eval Prompt

```
============================================================
EVAL PROMPT: MODEL PERFORMANCE JUDGE
============================================================

ROLE
----
You are a professional AI evaluation agent. Your task is to
evaluate a model's performance on multi-step reasoning tasks
along two dimensions:

  1. Correctness of the final answer.
  2. Quality of the reasoning process.


INPUT FIELDS
------------
You will be given the following input fields:

  - Question:
      {question}

  - Ground Truth (Reference Annotation):
      - Gold final answer: {gt_answer}
      - Ground-truth sub-goals: {formatted_sub_goals}

        Each ground-truth sub-goal includes a key_info field
        (critical facts or intermediate conclusions) that
        specifies the condition for that sub-goal to be
        considered achieved.

  - Model Prediction:
      - Predicted final answer: {pred_answer}
      - Predicted sub-goals: {formatted_pred_subgoals}


EVIDENCE BOUNDARY
```

```
-----------------
Your evaluation must be based strictly on this provided
information. Do not introduce external knowledge beyond what is
necessary for consistency checking.


============================================================
EVALUATION RULES
============================================================

EVALUATION PRIORITY
-------------------
  1. First, evaluate answer_correct based solely on final
     answer comparison.
  2. Second, calculate process_score based on sub-goal
     achievement.
  3. Third, IF AND ONLY IF answer_correct is true, check for
     obvious anomalies for flagging. This does NOT change
     answer_correct.


I. ANSWER CORRECTNESS
---------------------
Determine whether the model's predicted final answer matches
the ground-truth final answer.

  CORRECT
    - The predicted final answer and the ground-truth final
      answer are equivalent in core meaning.
    - Differences in wording, format, or minor stylistic
      variations are allowed.

  INCORRECT
    - The predicted final answer and the ground-truth final
      answer do not match in core meaning.
    - If multiple possible answers are acceptable according to
      the ground truth, treat the prediction as CORRECT only if
      it is semantically compatible with at least one acceptable
      answer.

  IMPORTANT
    The answer_correct value is determined SOLELY by comparing
    the predicted final answer with the ground-truth final
    answer. Anomaly detection (Section III) does NOT affect
    this value.


II. PROCESS SCORE
-----------------
Reasoning Process Quality

1. Ground-truth anchored evaluation

   - The list of ground-truth sub-goals defines the authoritative
     set of sub-goals to be evaluated.
   - For each ground-truth sub-goal i, you must determine whether
     this sub-goal is achieved by searching over ALL predicted
     sub-goals.
   - The matching between ground-truth sub-goals and predicted
     sub-goals is content-based (by key_info), not position-based.
     The order of predicted sub-goals may differ from the ground
     truth.
   - The number of elements in subgoal_vector and subgoal_details
```

```
     MUST equal the number of ground-truth sub-goals.
   - If there are more predicted sub-goals than ground-truth
     sub-goals, these extra predicted sub-goals MUST NOT create
     additional entries in subgoal_vector or subgoal_details.
     However, you may still use them for anomaly detection
     (fabrication, contradictions, faulty logic).

2. Scoring each ground-truth sub-goal

   For each ground-truth sub-goal G_i with its key_info:

   ACHIEVED (1 point)
     - There exists at least one predicted sub-goal whose content
       correctly identifies and recovers the key_info of G_i
       (semantic equivalence is allowed).
     - The information is reasonable, does not involve fabricated
       facts, and does not conflict with any key_info in the ground
       truth.

   NOT_ACHIEVED (0 points)
     - No predicted sub-goal correctly captures the key_info of G_i.
     - All candidate predicted sub-goals that mention related
       content contain clear fabrication or contradict the
       ground-truth key_info.
     - The prediction effectively omits this sub-goal, i.e., no
       predicted sub-goal provides the required key_info.

3. Process score calculation

   - PROCESS_SCORE =
       (number of ACHIEVED ground-truth sub-goals)
       /
       (total number of ground-truth sub-goals)

   - You must also output a subgoal_vector such as [1,0,1,...],
     where each entry is 1 if the corresponding ground-truth
     sub-goal is ACHIEVED and 0 otherwise.
   - The length of subgoal_vector MUST equal the total number of
     ground-truth sub-goals.
   - subgoal_details MUST contain one entry per ground-truth
     sub-goal, in ground-truth order (index 1..N), describing
     whether that ground-truth sub-goal is achieved and why.


III. ANOMALY DETECTION RULE
---------------------------
For Human Review

TRIGGER CONDITION
  This rule applies ONLY when answer_correct is true, i.e., the
  predicted final answer matches the ground-truth final answer.

CRITICAL CLARIFICATION
  - Anomaly detection is for FLAGGING purposes only.
  - anomaly_flag does NOT change the answer_correct value.
  - answer_correct reflects only whether the final answer is
    correct.
  - Flagged anomalies indicate cases that may need human review
    to verify the reasoning quality.

When the final answer is correct, perform a careful check for
potential reasoning anomalies. However, you must be CONSERVATIVE
and set anomaly_flag to true ONLY when there is CLEAR and
```

```
UNMISTAKABLE evidence of severe issues.

Flag anomaly_flag as true ONLY in these cases:

1. Explicit Guessing or Unsupported Answering

   - The prediction explicitly states guessing behavior
     (phrases like "I guess," "I'll try," "randomly choosing").
   - The final answer has absolutely zero connection to the
     reasoning provided in sub-goals.
   - There is no plausible logical path from the sub-goals to
     the answer.

   DO NOT flag if:
     The reasoning is weak but present, or the connection is
     unclear but possible.

2. Severe Factual Contradiction with Ground Truth

   - A predicted sub-goal states a fact that directly,
     unmistakably contradicts a key_info in the ground-truth
     sub-goals.
   - Example of flaggable contradiction:
       Predicted says "Event occurred in 1990" but ground-truth
       key_info clearly states "Event occurred in 2000."
   - This must be a clear factual conflict, not a different
     interpretation or reasoning approach.

   DO NOT flag if:
     The prediction uses different reasoning, omits information,
     or has minor inconsistencies.

3. Logically Impossible Reasoning Chain

   - The sub-goals contain statements that are internally
     contradictory in a way that makes the reasoning logically
     impossible.
   - Example:
       "X > Y and Y > Z, therefore X < Z" (violates transitivity).

   DO NOT flag if:
     The reasoning has gaps, is incomplete, or uses a suboptimal
     approach.

WHEN IN DOUBT, DO NOT FLAG
  - If you are uncertain whether an anomaly exists, set
    anomaly_flag to false.
  - Missing or incomplete reasoning is already captured by
    process_score.
  - Different reasoning approaches are acceptable.
  - The bar for flagging should be HIGH: only flag obvious,
    severe issues.

IV. CONFIDENCE LEVEL
--------------------
You must output a confidence level describing how reliable your
evaluation is, based on clarity and sufficiency of the
information given.

  HIGH
    - Evidence is clear and sufficient; sub-goal alignment and
      correctness judgments are straightforward.
```

```
  MEDIUM
    - Some minor ambiguity exists, e.g., slightly unclear wording,
      but you can still make a reasonably stable judgment.

  LOW
    - The ground truth and/or prediction are highly ambiguous,
      incomplete, or under-specified, making the evaluation
      unstable or speculative.

============================================================
OUTPUT SPECIFICATION
============================================================

You MUST output a single JSON object that conforms to the
provided JSON schema. Do not include any additional text outside
of this JSON object.

The JSON object must include:

  - answer_correct (boolean)
      - true if the predicted final answer matches the
        ground-truth final answer in core meaning.
      - false if the predicted final answer does not match the
        ground-truth final answer.
      - NOTE: This value is NOT affected by anomaly_flag. It
        reflects only the final answer comparison.

  - process_score (number in [0.0, 1.0])
      - The PROCESS_SCORE value defined above.

  - subgoal_vector (array of 0/1 integers)
      - Binary vector where each element corresponds to a
        ground-truth sub-goal, in order.
      - Length = number of ground-truth sub-goals.

  - subgoal_details (array)
      - Each element is an object describing the evaluation of
        one ground-truth sub-goal.
      - One element per ground-truth sub-goal, with strictly
        aligned indices.

  - anomaly_flag (boolean)
      - true if any anomaly (fabrication, faulty-logic success,
        contradiction of key_info) is detected.
      - false otherwise.

  - anomaly_reason (string)
      - If anomaly_flag is true, concisely describe the anomaly.
      - If anomaly_flag is false, set this to "None".

  - confidence (string)
      - One of "HIGH", "MEDIUM", or "LOW".

  - judge_reasoning (string)
      - A concise but explicit explanation of why you reached
        this evaluation, referencing both process and final answer.

Your response must be valid JSON and satisfy the JSON schema
given in the response_format configuration.
```

Table 13: Taxonomy of failure modes with representative failure cases.

	Error Type	Definition	Failure Case
	Visual Grounding Error	The model failed to identify the specific entity or region in the image that corresponded to the textual description, or incorrectly localized the target within the visual input.	The model cropped the wrong region using incorrect bounding box coordinates; the cropped area did not contain the target entity.
	Image Perception Failure	The model inaccurately perceived visual details in the image, such as colors, counts, shapes, textures, or spatial relationships.	The model counted 13 people in a painting; the correct answer was 3.
	Candidate Entity Confusion	The model encountered multiple plausible entities or intermediate results but selected one that did not satisfy the question's specific constraints.	The model identified two paintings by the same artist but selected the one from 1890 instead of the earlier 1889 work.
	Reasoning Error	The model possessed correct intermediate facts but made a logical error in connecting them or chose an incorrect operation during the reasoning process.	The model identified the Nebula Award logo but miscounted its circles as 5 instead of 4.
	Tool Call Failure	The model invoked an incorrect tool or populated tool parameters with wrong values, such as passing an invalid image token or a malformed query string.	The model passed an image filename to the TextSearch tool instead of calling ReverseImageSearch with the proper image token.
	Answer Vague	The model produced an imprecise or approximate answer that did not satisfy the exact requirements specified in the question.	The model answered "approximately 82.9%" instead of the exact ratio.
	Hallucination	The model generated information that was not present in the image or search results, relying instead on its internal parametric knowledge.	The model claimed a tree had yellow leaves when the ground truth was that the tree had no leaves.

C.6 Failure Taxonomy

While Figure 5 visualizes the error distribution, this appendix provides the formal taxonomy and definitions for the seven canonical failure modes in BrowseComp-V^3. As detailed in Table 13, each category is paired with a concise definition and a representative case study. These labels are assigned via an automated LLM-as-a-judge (GPT-5.2), which evaluates complete trajectories—including tool invocations, observations, and sub-goal sequences—to select the most appropriate category without human intervention. To ensure traceability, the color-coded markers in Table 13 map directly to the legend in Figure 5, linking quantitative results to their qualitative definitions.

D Case Study

Case 1: `036_Life_Law`

Case Summary: The agent follows a tool-based trajectory to identify the legal dispute, verify the sports-equipment clue, and inspect the final podium image.

Domain: Life / Law.

Question: The image shows the courtroom scene of a well-known trademark-infringement case. Counsel for the defense claimed that the object held by the brand's spokesperson was, in fact, another piece of sports equipment. Within the two years following that trial, the sport represented by that equipment hosted a major international tournament in South America. In the singles event of that tournament, a player who is 172 cm tall finished third. What color were the shoes that the player wore on the podium during the award ceremony?



Input image: courtroom scene.

Ground Truth: Red. **Prediction:** Red.
Evaluation. Final score: **1.0**; Process score: **1.0**.

Round 1

Assistant reasoning. The first step is to identify the courtroom image and recover the trademark-dispute context before following the sports-equipment clue.

Tool call

```
assistant to=functions.ReverseImageSearch
{
  "arguments": {
    "image_token": "<image_1>"
  }
}
```

Observation. Reverse image search results identify the scene as a hearing, where Michael Jordan's lawyer displayed the dunking-player logo used by the Chinese sportswear company Qiaodan Sports. Related Sina and Weibo reports confirm the trademark-infringement context.



(1) Sina/Pengpai report: Jordan winning part of the Qiaodan trademark dispute.



(2) Related Chinese-language report for the same dispute context.



(3) Supporting context: Qiaodan / Michael Jordan visual matching.



(4) Supporting context: Qiaodan mark and related image cluster.



(5) Supporting context: image-search cluster.

Round 2

Assistant reasoning. The reverse image search returned a Sina news link among the top results. I will visit the page to confirm the case details and trial timeline.

Tool call
assistant to=functions.WebVisit

```
{
  "arguments": {
    "url": "http://www.china.org.cn/business/2016-12/08/content_39876607.htm"
  }
}
```


Observation. The report describes the long-running Michael Jordan / Qiaodan Sports trademark dispute. The Supreme People's Court held public hearings on April 19 and 26, 2016, and issued its ruling in December 2016. The case lasted nearly 5 years from the initial filing in 2012.

Round 3

Assistant reasoning. The key clue is the defense-side claim that the logo action involved sports equipment, so the next step is to verify what equipment and sport were claimed.

Tool call
assistant to=functions.TextSearch

```
{
  "arguments": {
    "query": "Qiaodan Sports logo table tennis paddle action not basketball"
  }
}
```


Observation. Search results identify a Sina Sports report about Qiaodan Sports' "table-tennis-paddle" theory.

Round 4

Assistant reasoning. The search results returned a Sina Sports article link about Qiaodan's "table-tennis-paddle" defense. I will visit the page to read the full claim.

Tool call
assistant to=functions.WebVisit

```
{
  "arguments": {
    "url":
"https://sports.sina.cn/nba/other/2020-04-08/detail-iirczymi5126392.d.html?vt=4"
  }
}
```


Observation. The article confirms that during a hearing, Qiaodan Sports argued its logo depicted a table-tennis-paddle swing rather than a basketball dunk. The defense further claimed the name "Qiaodan" meant "southern vegetation" and was unrelated to Michael Jordan.

Round 5

Assistant reasoning. The relevant sport is table tennis. The trial took place in April 2016, so the "two years following" spans 2016–2018. I need to identify which major international table tennis tournament was held in South America during this period.

Tool call
assistant to=functions.TextSearch

```
{
  "arguments": {
    "query": "major international table tennis tournament South America 2016 2017 2018"
  }
}
```


Observation. Search results highlight the 2016 Rio de Janeiro Summer Olympics (August 2016, Brazil) as the only major international table tennis event held in South America during 2016–2018. No World Championship or World Cup in table tennis was hosted in South America in 2017 or 2018.

Round 6

Assistant reasoning. The Rio 2016 Olympics is confirmed as the target event. The question mentions "singles" but does not specify gender. I need to check both men's and women's singles bronze medalists and match the 172 cm height clue.

Tool call

```
assistant to=functions.TextSearch
{
  "arguments": {
    "query": "Rio 2016 Olympics table tennis singles bronze medalist men women"
  }
}
```

Observation. Results show: Men's singles bronze—Jun Mizutani (Japan); Women's singles bronze—Kim Song-i (North Korea).

Round 7

Assistant reasoning. Two candidates: Jun Mizutani and Kim Song-i. The question specifies 172 cm. I need to search each player's height to determine the correct target.

Tool call

```
assistant to=functions.TextSearch
{
  "arguments": {
    "query": "Jun Mizutani table tennis height"
  }
}
```

Observation. Multiple sources (Wikipedia, pongworld.com, sport.de) consistently list Jun Mizutani's height as 172 cm. This matches the clue.

Round 8

Assistant reasoning. Jun Mizutani matches 172 cm. To be rigorous, I should also verify Kim Song-i's height to confirm she does not match.

Tool call

```
assistant to=functions.TextSearch
{
  "arguments": {
    "query": "Kim Song-i table tennis player height"
  }
}
```

Observation. Wikipedia and the official Rio 2016 athlete profile both record Kim Song-i's height as 161 cm. This does not match the 172 cm clue. The target player is confirmed as Jun Mizutani (men's singles bronze).

Round 9

Assistant reasoning. The target player is confirmed. The remaining question is purely visual: what color shoes did Jun Mizutani wear on the podium? I need to find medal-ceremony photographs.

Tool call

```
assistant to=functions.ImageSearch
{
  "arguments": {
    "query": "Rio 2016 table tennis men's singles medal ceremony podium"
  }
}
```

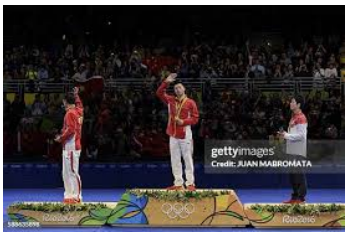
Observation. Image search returns multiple medal-ceremony photographs from Getty Images and Alamy, showing Zhang Jike, Ma Long, and Jun Mizutani on the podium at Riocentro on August 11, 2016.



(1) Rio 2016 men's singles table tennis podium context.



(2) Podium image: Ma Long, Zhang Jike, and Jun Mizutani.



(3) Medal-ceremony result cluster.



(4) Podium view: bronze medalist's shoes visible.

Round 10

Assistant reasoning. The image search returned a clear podium photograph showing all three medalists. Jun Mizutani is on the right side (bronze position), but the full image is too wide to clearly determine shoe color. I will crop the lower-right region around his feet.

Tool call

```
assistant to=functions.CropImage
{
  "arguments": {
    "image": "<obs_podium_image>", "bbox": [97, 171, 177, 285]
  }
}
```

Observation. The crop focuses on Jun Mizutani's lower body and feet on the podium. A pair of red sports shoes is clearly visible.



(1) Crop output: focused shoe-area view confirming red shoes.

Final Answer and Subgoal Summary

Subgoal trace.

1. <SUBGOAL> The courtroom image is associated with the Michael Jordan / Qiaodan Sports trademark dispute. </SUBGOAL>
2. <SUBGOAL> The defense-side sports-equipment clue points to a table-tennis racket and therefore table tennis. </SUBGOAL>
3. <SUBGOAL> The relevant South American tournament is the table tennis event at the 2016 Rio Summer Olympics. </SUBGOAL>
4. <SUBGOAL> Jun Mizutani is listed at 172 cm and won bronze in Rio 2016 men's singles table tennis. </SUBGOAL>
5. <SUBGOAL> In the Rio 2016 men's singles medal-ceremony podium image, Jun Mizutani's shoes appear red. </SUBGOAL>

Final-answer extraction. <FINAL_ANSWER> Red </FINAL_ANSWER>

E Long-term Maintenance and Reproducibility

Because BrowseComp-V^3 is grounded in publicly searchable web evidence, link rot and content drift are intrinsic risks. We therefore convert the "temporal stability" design principle (Section 3) into a sustained operational commitment, organized as four complementary pillars.

(1) Snapshot Archiving at Release. At submission time, every URL referenced in a gold-standard trajectory is archived to the Wayback Machine and additionally mirrored to a release-controlled object store. Each question is shipped with both the *live* URL and a fallback `archive_url` field, so that even if the original source becomes unreachable, the originally-seen content remains verifiable. The dataset schema is augmented with three explicit fields: `url`, `archive_url`, and `snapshot_hash` (SHA-256 of the archived page), enabling deterministic re-evaluation against the exact content seen by annotators.

(2) Automated Quarterly Link-Health Audits. We run a CI-driven crawler every three months that probes, for every URL referenced by a question: (a) URL reachability (HTTP status, redirect chain), (b) content-hash drift against the original `snapshot_hash`, and (c) text-level similarity (TF–IDF cosine) between the live page and the archived snapshot, to detect substantive content rewrites that do not change the hash. Each question is then assigned a public `health_status` flag with three values: `ok` (live and stable), `drift` (live but materially changed), or `dead` (unreachable). The latest health report is published alongside the leaderboard, allowing users to transparently distinguish intrinsic model errors from infrastructure-induced ones.

(3) Versioned Re-annotation Protocol. When a source is reclassified as `dead`, or its drift score crosses a predefined threshold, the affected question is *not* silently patched. Instead, we (i) freeze the current version (v$x.y$), (ii) re-annotate an equivalent question using a still-live source within the same sub-domain and difficulty tier, following the original construction pipeline (Section 3), and (iii) release the update as a numbered minor version with a public changelog. Leaderboard entries always declare the version they evaluated against, ensuring longitudinal comparability across years.

(4) Community Reporting Channel. We provide a public issue template and a structured web form for users to flag broken links, content drift, or contested ground-truth answers. All reports are triaged within two weeks by the maintainer team; accepted reports trigger an entry in the next minor-version changelog and are credited to the reporter.

Governance and Longevity. The authors will maintain the benchmark on an ongoing basis through the archiving, auditing, re-annotation, and community-reporting mechanisms above. All scripts, CI configurations, and the changelog are open-sourced in the same repository as the dataset, making the maintenance process auditable and transferable to future custodians if maintainership changes.

F Limitations and Future Work

While BrowseComp-V^{3}provides a systematic evaluation framework for multimodal agentic search, several limitations warrant discussion.

Benchmark scale and coverage. BrowseComp-V^{3}contains 300 expert-curated questions across 24 sub-domains, which is sufficient to expose clear performance gaps but still limited relative to the long-tail diversity of open-web multimodal search. Certain rare domains, query formulations, and evidence compositions may remain underrepresented.

Temporal dependency of web evidence. Our tasks rely on publicly accessible web sources, which are inherently dynamic. Although we include snapshots, link-health audits, and versioned re-annotation in the maintenance pipeline, source drift and content removal can still affect long-term stability and comparability across evaluation periods.

Protocol-induced constraints. For budget-comparable evaluation, OmniSeeker uses a unified tool interface and a fixed interaction cap (T=20). This improves fairness across models, but may also bias outcomes for systems that benefit from different tool granularities or longer horizons. Therefore, reported scores should be interpreted as performance under a controlled protocol rather than absolute upper bounds.

Metric granularity. Success Rate (SR) and Process Score (PS) capture complementary aspects of final correctness and intermediate progress, but they do not fully characterize evidence quality, calibration, or efficiency trade-offs (e.g., latency and tool-call cost). More fine-grained process diagnostics are still needed.

Future work. We plan to extend BrowseComp-V^{3}in four directions: (i) increasing benchmark scale and domain diversity, (ii) introducing adaptive-budget evaluation beyond fixed-turn settings, (iii) enriching process metrics with evidence faithfulness and efficiency-aware criteria, and (iv) strengthening continual maintenance with automated drift detection and community-driven updates.