VSearcher: Long-Horizon Multimodal Search Agent via Reinforcement Learning

Ruiyang Zhang [1,2] **Qianguo Sun**[2] **Chao Song**[2] **Yiyan Qi**[2] **Zhedong Zheng**[1*]
[1] FIC, University of Macau [2] IDEA

Abstract

Large models are increasingly becoming autonomous agents that interact with real-world environments and use external tools to augment their static capabilities. However, most recent progress has focused on text-only large language models, which are limited to a single modality and therefore have narrower application scenarios. On the other hand, large multimodal models, while offering stronger perceptual capabilities, remain limited to static knowledge and lack the ability to access and leverage up-to-date web information. In this paper, we propose VSearcher, turning static multimodal model into multimodal search agent capable of long-horizon, multi-turn tool use in real-world web environments, including text search, image search, and web browsing, via reinforcement learning. Specifically, we introduce Iterative Injection Data Synthesis pipeline to generate large-scale, complex multimodal QA questions, which are further filtered with comprehensive metrics to ensure high quality and sufficient difficulty. We then adopt an SFT-then-RL training pipeline to turn base multimodal models to agent capable of multi-turn tool calling in real-world web environments. Besides, we propose a multimodal search benchmark MM-SearchExam dedicated to evaluating search capabilities of multimodal search agents, which proves highly challenging for recent proprietary models. Extensive evaluations across multiple multimodal search benchmarks reveal effectiveness of our method. VSearcher achieves superior performance compared to recent multimodal search agents and even surpasses several proprietary models on multimodal web search tasks. Code is released at `https://github.com/Ruiyang-061X/VSearcher`.

1 Introduction

Since the introduction of strong reasoning models such as OpenAI o1 [13] and DeepSeek-R1 [9], foundation models [44, 2] are increasingly equipped with inherent step-by-step reasoning capabilities [34, 4, 5, 24]. Moreover, with paradigms such as ReAct [36], these reasoning models are developed into agents that iteratively think and act to break down complex problems [26]. Notably, large models are increasingly becoming autonomous agents [11] that can interact with real-world environments and harness external tools to augment their static, fixed knowledge [17, 30]. However, recent progress in building autonomous agents has primarily focused on large language models [7, 19, 21]. This single-modality constraint greatly limits their application scenarios and prevents them from meeting real-world user needs.

In contrast, large multimodal models [3, 15, 32, 42], with their stronger perceptual abilities, often remain restricted to static knowledge and cannot effectively harness external tools to access up-to-date information from real-world web environments [37]. Although there have been preliminary attempts to equip large multimodal models with web search tools, these works mainly focus on short-horizon search capabilities or rely on rigid workflows. MMSearch-R1 [33] pioneers the development of

*Correspondence to zhedongzheng@um.edu.mo.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

multimodal search agents, however, it is only capable of short-horizon web search including at most one round of image search and one round of text search, and follows a fixed tool calling procedure. More recently, DeepMMSearch-R1 [23] removes this fixed tool calling procedure and encourages more flexible tool use behavior, but it relies on simple VQA data for training and mainly exhibits short-horizon web search ability. Similarly, VRAG-RL [31] adopts RL to encourage adaptive web search behavior, but limited difficulty of its training data leads to search behavior spanning only a few steps. OmniSearch [18] leverages a planning agent and a sub-question solver to iteratively address complex multimodal questions, while it still relies on rigid agent workflow. Overall, short-horizon search behavior and rigid workflows hinder the full potential of large multimodal models when interacting with real-world web tools.

In this paper, we propose VSearcher, transforming static multimodal models into autonomous agents capable of long-horizon, multi-turn tool use in real-world web environments, including image search, text search, and web browsing, via reinforcement learning. Notably, our model is trained in real-world web environments with challenging multimodal tasks, resulting in agent that can navigate extensive web sources and solve real-world multimodal user requests. Specifically, we propose a comprehensive post-training framework for building multimodal search agents, spanning data synthesis, rejection fine-tuning, and reinforcement learning. Moreover, we introduce a challenging multimodal browsing benchmark to enable more thorough evaluation of multimodal search agents. For data synthesis, we propose Iterative Injection-based Data Synthesis to fully automatically generate large-scale, challenging multimodal browsing tasks. We also propose several filtering metrics to ensure the difficulty and quality of the synthesized questions. The synthesized challenging multimodal browsing tasks serve as the foundation for facilitating long-horizon web browsing ability in our search agent. For rejection fine-tuning, we aim to instill strong browsing ability from powerful teacher model into base multimodal model to provide an initial long-horizon tool-use capability. Specifically, we build a ReAct [36] inference framework and utilize Gemini-3-Pro-Thinking as the teacher model. We generate trajectories for our synthesized multimodal browsing questions with Gemini-3-Pro-Thinking and reject those trajectories that end with incorrect answers. Finally, we fine-tune base multimodal models on the remaining high-quality trajectories. For reinforcement learning, we perform GRPO [28] on the fine-tuned model to further generalize long-horizon search ability. Notably, the entire training process is conducted in real web environments to enhance real-world browsing performance. We combine answer-level reward and trajectory-level reward as agent reward signal, providing supervision for both final answer correctness and effectiveness of information seeking process. Strict format checking is enabled during rollout to ensure correct formatting throughout the whole rollout process. We utilize our synthesized questions as the training data for the reinforcement learning stage. For the benchmark, we curate highly challenging questions from our data synthesis pipeline. We apply more extensive iterative generation process to ensure the difficulty of these questions. Several recent proprietary models achieve low accuracy on our benchmark.

Thorough evaluation on several multimodal search benchmarks validates the effectiveness of our trained model. VSearcher surpasses recent multimodal browsing agents and even exceeds the performance of several proprietary models. In summary, our contributions are as follows:

- **Data synthesis pipeline for multimodal browsing task.** We propose Iterative Injection-based Data Synthesis, together with comprehensive filtering metrics, enabling fully automatic, large-scale generation of challenging multimodal browsing tasks.
- **A comprehensive post-training pipeline for multimodal search agent.** We propose a thorough post-training pipeline for transforming base multimodal model into long-horizon multimodal search agent. Rejection sampling fine-tuning instills initial multi-turn tool-use capability, and reinforcement learning generalizes long-horizon search ability in real-world web environments.
- **Challenging benchmark for evaluation of multimodal search agent.** We curate a challenging benchmark MM-SearchExam for evaluating multimodal browsing ability, on which strong proprietary models achieve low performance.
- **Strong performance on various multimodal browsing benchmarks.** VSearcher achieves stronger results on several multimodal search benchmarks than recent multimodal browsing agents, and surpasses performance of several proprietary models.

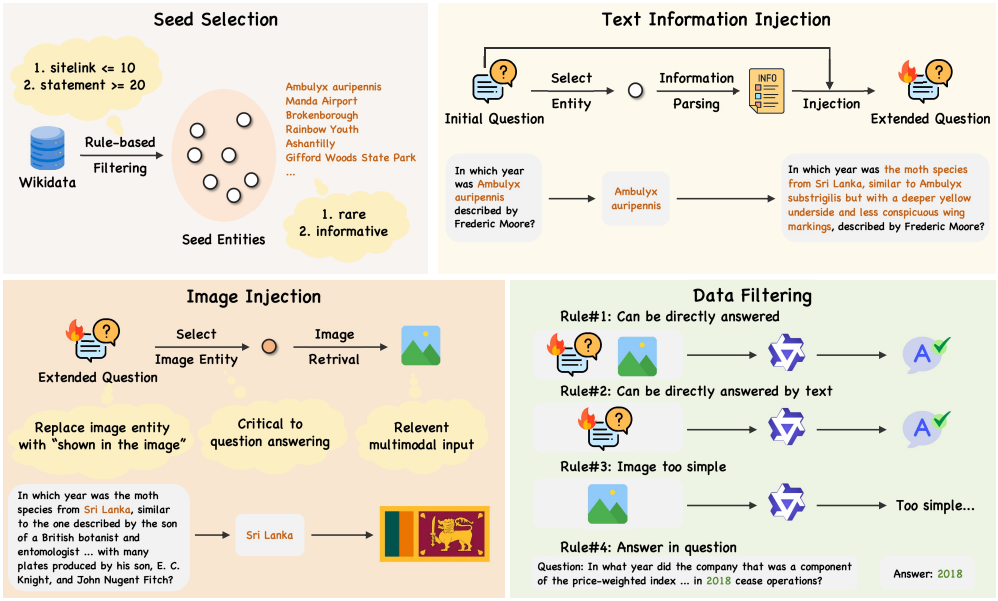


Figure 1: Our proposed Iterative Injection-based Data Synthesis.

2 Method

In this section, we provide an overview of our post-training framework for VSearcher, including Iterative Injection-based Data Synthesis, Rejection Sampling Fine-tuning, and Reinforcement Learning. We also describe the curation details of our proposed multimodal search benchmark, MM-SearchExam.

2.1 Iterative Injection-based Data Synthesis.

Challenging browsing tasks form the foundation for training and building long-horizon multimodal search agents. We propose Iterative Injection-based Data Synthesis, together with comprehensive data-filtering metrics, to synthesize challenging multimodal browsing tasks fully automatically. In summary, we begin with a text-only seed entity and iteratively extend it into complex multimodal QA questions (see Fig. 1).

Seed Selection. Seed selection is critical to the difficulty of the final synthesis questions. Seeds for browsing tasks should be as rare as possible to encourage the browsing agent to rely on external tools rather than its static knowledge. We utilize Wikidata as the data source, which serves as the structured knowledge base behind Wikipedia. In Wikidata, a 'sitelink' refers to the links of an entity across different languages; fewer sitelinks typically indicate a rarer entity. A 'statement' denotes factual triples associated with an entity; more statements usually imply that the corresponding Wikipedia page contains richer information. During seed selection, we explicitly select entities with sitelinks below a threshold and statements above a threshold, yielding seeds that are both rare and sufficiently documented for subsequent iterative extension. We use SPARCLE to perform rule-based filtering over Wikidata to obtain seed entities satisfying our requirements.

Initial QA Generation. For each selected seed, we parse its Wikipedia content from an offline Wikipedia index. Due to the design of the seed selection stage, the corresponding content tends to be both rarely-known and sufficiently detailed. We then prompt an off-the-shelf LLM to generate an initial text-only QA pair from the content. We instruct the LLM to generate a one-sentence question about the provided entity. The question should be specific enough to have a unique answer. The generated answer should be a single word or a short phrase, making correctness easy to verify.

Text Information Injection. After obtaining the initial QA pair, we perform multiple rounds of text information injection to gradually extend the initial question into a complex browsing problem. In each round, we select an entity from the question, hide it, and replace it with its parsed information. This process increases question complexity and the need for web browsing. Specifically, in each round, we first prompt an LLM to select an entity from the initial question and obtain the corresponding

Wikipedia content of the selected entity. We then prompt an LLM to extract one specific piece of information from the Wikipedia content. Notably, we instruct the LLM to extract rarely known information rather than commonly known facts, as the latter are less effective at increasing difficulty. Moreover, we encourage the LLM to include as many entities as possible in the extracted information, which facilitates subsequent rounds of iterative injection by providing additional entities to extend. Finally, we prompt an LLM to transform the original question by hiding the selected entity, replacing it with the parsed information, and ensuring that the transformed question remains logically coherent.

Image Injection. After several rounds of text information injection, the initial QA pair becomes a complex browsing problem. We then perform image injection to turn the extended question into multimodal question. For image injection, we first prompt an LLM to select an entity in the question that is critical for correct answering of the question. We then retrieve the Wikipedia image of the selected entity. The retrieved image serves as the input image for the question. Next, we prompt an LLM to transform the question by replacing the entity mention with a phrase such as 'shown in the image'. After multiple rounds of text information injection, the question tends to be complex and the selected image entity tends to be rare. As a result, the image of this entity also tends to be rare, which encourages the model to call image search tool to obtain additional information of the image. Moreover, because we choose an entity that is crucial for answering the question and replace it with its image, the image plays a larger role in solving the question, avoiding cases where the model can solve the browsing task solely from text.

Difficulty Level Design. To train a multimodal browsing agent that is proficient across tasks with varying difficulty and browsing requirements, we generate multimodal browsing questions at different difficulty levels and mix them during training. This strategy is beneficial for building an agent with adaptive and robust browsing skills. Specifically, we synthesize 3 levels of browsing tasks: easy, medium, and hard. We conduct 1 round, 3 rounds, and 5 rounds of text information injection for easy, medium, and hard browsing tasks respectively.

Data Filtering Strategy. We design comprehensive filtering metrics for our synthesized multimodal browsing tasks to ensure both high difficulty and high quality. (1) We filter out questions that can be answered directly by a weaker LVLM. Even with multiple rounds of iterative extension, the combination of injected information could create reasoning shortcuts that degrade the task to simple QA question. We remove such samples because they typically lead to no tool use during subsequent rejection sampling fine-tuning and reinforcement learning, and therefore do not improve long-horizon tool-use capability. (2) We filter out samples that can be answered directly from the textual content by a weaker LLM. Although we explicitly select an entity that is critical for answering the question during image injection, there are still cases where the answer can be inferred from text alone. In such samples, the multimodal information plays a minor role, so we remove them because they are less beneficial for improving multimodal perception ability in browsing agents. (3) We filter out samples with too simple image. During case studies, we observe image that contains little information and are overly simple, *e.g.*, the flag of a commonly known country. Such samples are not beneficial for improving multimodal browsing skills such as calling image search, extracting information from image, and perform text search based on those information to obtain more context. (4) We filter out samples where the answer is already present in the question. These occur as corner cases: during information injection, the extracted information could accidentally include the ground-truth answer and be injected into the question. We remove these corner cases.

2.2 Rejection Sampling Finetuning

In this stage, we instill strong multi-turn browsing skills from proprietary models to base multimodal models (see Fig. 2). In general, we utilize standard ReAct framework to let a proprietary model tackle our synthesized multimodal browsing tasks, perform rejection sampling over the generated trajectories, and fine-tune the base multimodal model on the resulting high-quality trajectories. This stage cold-starts the base model with long-horizon, multi-turn tool-use capabilities and prepares it for the subsequent reinforcement learning stage.

ReAct Loop. In standard ReAct framework, the agent iteratively performs reasoning, action, and observation to solve challenging tasks. In each round, the agent first reasons based on the previous context. It then either calls a tool if it needs additional information or terminates the trajectory by producing a final answer. When a tool call is issued, the agent waits for the observation returned by the tool and continues once the observation is available. In our scenario, we provide 3 real-world web

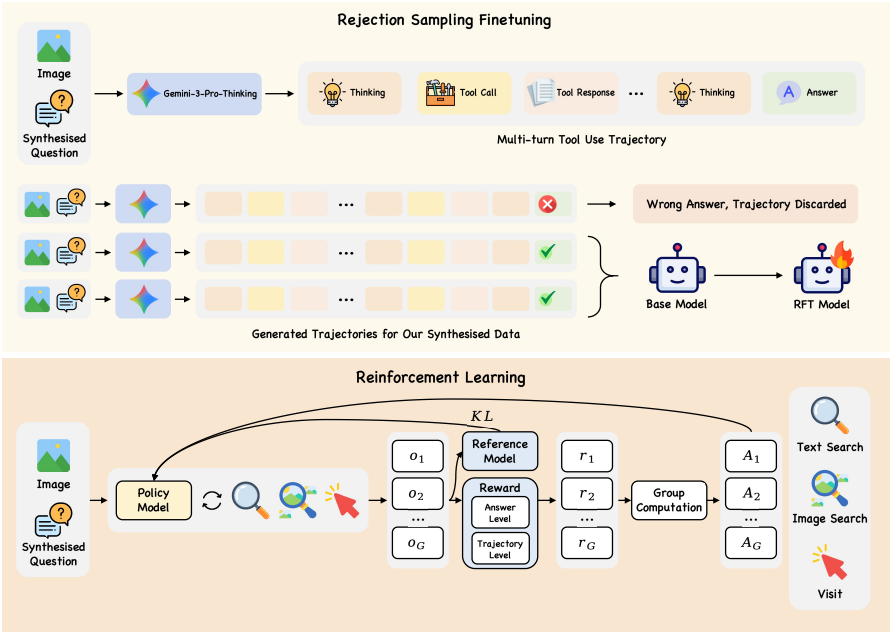


Figure 2: Rejection sampling finetuning and reinforcement learning process for VSearcher.

tools, including image search, text search, and visit. For image search, we use the Google Vision Web Detection API on the image in the question; the tool response includes 5 similar image links and corresponding page information that contains those images, including page titles and page links. For text search, we utilize Google Custom Search API; the tool response includes 5 pages related to the query, including the page link, title, and snippet. For visit, the tool returns a summary of the web content based on the provided link and visiting goal describing what to extract from the page. A complete trajectory with T iterations can be defined as:

$$\mathcal{H}_T = (\tau_0, a_0, o_0, \ldots, \tau_i, a_i, o_i, \ldots, \tau_T, a_T), \tag{1}$$

where τ_i, a_i, o_i represent thought, action, and observation in the i-th round, respectively. At step t, the thought τ_t and a_t are sampled from a policy based on all previous context, i.e., $\pi(a, t|\mathcal{H}_{t-1})$.

Rejection Sampling. For the generated ReAct trajectories from the proprietary model, we perform rejection sampling based on answer correctness. Following HLE, we utilize LLM-as-a-Judge to verify whether the final answer is correct, and we keep only trajectories that end with correct answer.

Supervised Fintuning. Finally, we perform supervised fine-tuning on the base multimodal model with the rejection-sampled full trajectories to instill initial long-horizon tool-use capability. The supervised fine-tuning process minimizes the following objective across all training trajectories:

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\log \pi_\theta\big(o_{i,t} \mid o_{i,<t}, q_i\big). \tag{2}$$

where N is the number of training trajectories, $o_{i,t}$ denotes the t-th token of the output sequence for the i-th trajectory, T_i is the total length of the output sequence of the i-th training trajectory, $o_{i,<t}$ represents the tokens preceding the t-th token of the i-th training trajectory, q_i is the input query for the i-th training trajectory, and π_θ is the model policy parameterized by θ.

2.3 Reinforcement Learning

In this stage, we perform reinforcement learning on the fine-tuned multimodal model to further generalize its long-horizon tool-use capabilities. Notably, we conduct RL in real-world web environments, enabling agent to develop more robust browsing skills and to better solve real-world multimodal user requests. In practice, we adopt the off-the-shelf Group Reward Proximal Optimization (GRPO) [28] algorithm. Specifically, GRPO performs multiple rollout samplings and optimizes the policy to favor responses with higher assigned rewards, training objective of GRPO is as follows:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}$$
$$\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(\frac{\pi_\theta(o_i\mid q)}{\pi_{\theta_{\text{old}}}(o_i\mid q)}A_i,\ \text{clip}\Big(\frac{\pi_\theta(o_i\mid q)}{\pi_{\theta_{\text{old}}}(o_i\mid q)},\ 1-\epsilon,\ 1+\epsilon\Big)A_i\right) - \beta\, D_{\text{KL}}(\pi_\theta\,\|\,\pi_{\text{ref}})\right]. \tag{3}$$

$$D_{\text{KL}}(\pi_\theta\,\|\,\pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i\mid q)}{\pi_\theta(o_i\mid q)} - \log\frac{\pi_{\text{ref}}(o_i\mid q)}{\pi_\theta(o_i\mid q)} - 1, \tag{4}$$

where π_θ is the current model policy, $\pi_{\theta_{\text{old}}}$ is the old policy, G is the rollout group size, q is the query, o_i is the i-th sampled reponse, π_{ref} is the reference model, ϵ is the clipping hyper-parameter controlling updating degree, and β is the coefficient of Kullback–Leibler (KL) penalty. A_i is the normalized advantages computed based on rewards $\{R_1, R_2, \cdots, R_G\}$.

Our reward shaping is as follow:

$$\begin{aligned} R_i =& 0.9 \times R_{\text{answer_level}}(o_i) + 0.1 \times R_{\text{trajectory_level}}(o_i) + \\ & 0.1 \times R_{\text{deep_search}}(o_i) + 0.1 \times R_{\text{wide_search}}(o_i), \end{aligned} \tag{5}$$

where $R_{\text{answer_level}}(o_i)$ denotes the answer correctness reward: a trajectory that produces the correct answer receives a reward of 1, and an incorrect answer receives a reward of 0. Following HLE [25], we use an LLM-as-a-Judge to evaluate answer correctness. $R_{\text{trajectory_level}}(o_i)$ is our proposed trajectory-level reward. During our data synthesis process, all entities used for text information injection and the image entity form a set of key entities. We utilize the hit ratio of these key entities in the model trajectory as this reward. This reward reflects the effectiveness of the agent information seeking process. Our two-level reward design not only emphasizes final answer correctness, but also provides supervision over the entire agent trajectory, which facilitates the development of more effective long-horizon search ability. We conduct strict format checking at each rollout step; trajectories that fail the format checks are terminated immediately.

We also propose to facilitate two new search strategies in our multimodal search agent. Specifically, we explore two directional search strategies, including deep search and wide search. In particular, we realize this through reward shaping. For a given search trajectory, we collect all issued tool-call parameters, including image search, text search, and visit. We then adopt MLLM-as-Judge and utilize GPT-5.6 Sol as the judge model, asking the judge to determine whether the search pattern belongs to (1) continuously digging into information on the same topic and discovering more specific details (Deep Search Pattern), or (2) collecting diverse information on the same topic and reaching a comprehensive conclusion (Wide Search Pattern). If the search pattern is judged to belong to one of these patterns, the trajectory will be assigned a positive search pattern reward. $R_{\text{deep_search}}(o_i)$ is the deep search pattern reward, which is 1 if MLLM-as-Judge determines that the search process matches the deep search pattern and 0 otherwise, and $R_{\text{wide_search}}(o_i)$ is the wide search reward, with a value of either 1 or 0, as determined by MLLM-as-Judge.

2.4 Curation of MM-SearchExam

To enable more thorough evaluation of multimodal search agents, we curate a challenging multimodal search benchmark, MM-SearchExam. MM-SearchExam is constructed with our proposed Iterative Injection-based Data Synthesis pipeline. Specifically, we perform 10 rounds of iterative extension to make the browsing tasks highly challenging. We also apply data filtering procedure to ensure high quality and difficulty. Additionally, we explicitly remove seeds that appear in our training data to avoid data leakage. In total, MM-SearchExam contains 283 synthesized browsing tasks. Quantitative experiments show that several proprietary models achieve low performance on MM-SearchExam, highlighting difficulty of the benchmark and the effectiveness of our Iterative Injection-based Data Synthesis pipeline in generating highly challenging browsing tasks.

3 Experiment

3.1 Settings

Implementation Details. For rejection sampling fine-tuning, we leverage Gemini-3-Pro-Thinking as the teacher model to generate trajectories for our synthesized browsing tasks due to its strong

Table 1: Comparison between different baselines across 5 multimodal search benchmarks. BC-VL refers to BrowseComp-VL and MM-BC refers to MM-BrowseComp.

Model	Type	MMSearch	BC-VL	MM-BC	SimpleVQA	MM-SearchExam
Open-source Models with Web Tools						
Qwen3-VL-4B-Thinking	Short-Horizon	35.0	21.5	1.6	41.1	2.4
Qwen3-VL-8B-Thinking	Short-Horizon	39.1	24.8	5.7	45.5	4.4
Qwen3-VL-30B-A3B-Thinking	Short-Horizon	43.7	27.5	6.7	46.3	6.0
Qwen2.5-VL-7B-Instruct	Short-Horizon	36.1	14.8	2.1	35.8	3.9
Qwen2.5-VL-72B-Instruct	Short-Horizon	41.5	24.3	2.4	37.7	11.6
InternVL3.5-8B	Short-Horizon	29.2	17.8	1.6	38.8	3.1
InternVL3.5-38B	Short-Horizon	33.3	19.3	4.9	39.7	6.3
Proprietary Models with Web Tools						
GPT-4o	Short-Horizon	13.4	17.3	0.8	40.0	8.1
GPT-5	Long-Horizon	45.2	29.0	9.3	62.3	20.8
Gemini-3-Flash	Long-Horizon	57.3	31.3	4.9	60.8	20.1
Gemini-3-Pro	Long-Horizon	59.0	35.0	10.8	64.1	23.8
Qwen3-VL-Plus	Long-Horizon	50.7	28.0	6.0	58.4	18.8
Agentic Multimodal Models						
MMSearch-R1-7B	Short-Horizon	37.5	16.0	4.9	40.0	4.5
DeepMMSearch-R1-7B	Short-Horizon	-	-	-	55.8	-
WebWatcher-7B	Long-Horizon	38.3	25.5	6.5	37.7	10.2
DeepEyesV2-7B	Short-Horizon	40.9	22.0	8.1	44.4	4.5
VSearcher-7B	Long-Horizon	45.3	30.2	5.9	45.3	17.8
VSearcher-8B	Long-Horizon	47.2	30.8	6.2	46.6	19.3

web-browsing capabilities. We obtain 1308 high-quality trajectories for fine-tuning and use LLaMA-Factory to conduct training. For reinforcement learning, we utilize rllm [29] as the training framework. We use a batch size of 32 and perform 8 rollouts per sample. We set the maximum prompt length to 2K and the maximum response length to 28K. The maximum number of steps in the ReAct loop is set to 30, and the maximum trajectory time is set to 2 hours 30 minutes. The learning rate is set to 2e-6. The KL-penalty coefficient is set to 0.001. We use Qwen2.5-72B-Instruct as the judge model for verifying answer correctness. We utilize Qwen3-VL-8B-Thinking as our base models. We use 16 H100 GPUs for all experiments. In this work, we define search agents capable of yielding 10 or more tool calls as long-horizon search agents, and those yielding fewer than 10 tool calls as short-horizon search agents.

Tool Implementation. For text search, we use the Google Custom Search API as the backend. For each query, we return 5 relevant search results, including the page link, page title, and page snippet which provides a brief summary of the page. For image search, we use the Google Vision Web Detection API. For each input image, we return the 5 most similar image URLs and the corresponding pages that contain these images, including their page links and page titles. For the visit tool, the inputs are a webpage URL and a specific visit goal. We first call the JINA API on the URL, which parses the webpage content and returns it in clean markdown format. We then use Qwen2.5-72B-Instruct as a summarization model to extract content relevant to the visit goal. The resulting summary is returned as the final response of the visit tool.

Baselines. MMSearch-R1 [33] is a pioneering work that utilizes RL to train multimodal search agent. It is equipped with image search and text search. Specifically, it mainly focuses on a two-step tool-use pattern: first performing image search, followed by text search. DeepMMSearch-R1 [23] is a follow-up work to MMSearch-R1. It primarily improves image-search effectiveness by incorporating an image-cropping tool and enabling self-reflection and self-correction during rollouts. It emphasizes multi-turn tool-use capabilities. The agent is equipped with image search, text search, and image cropping tool. DeepEyesV2 [10] is a comprehensive study on building multimodal agentic models. It utilizes Python code to perform image operations such as cropping, and it is also equipped with image search and text search.

3.2 Main Results

We present comparisons with various baselines in Tab. 1. Specifically, VSearcher-8B denotes model trained with Qwen3-VL-8B-Thinking as base model. Since recent agentic multimodal models are

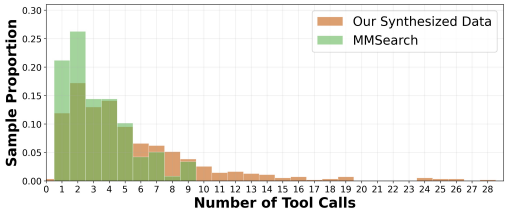


Figure 3: Tool call distribution of Gemini-3-Pro-Thinking in solving our synthesized data and MMSearch questions.

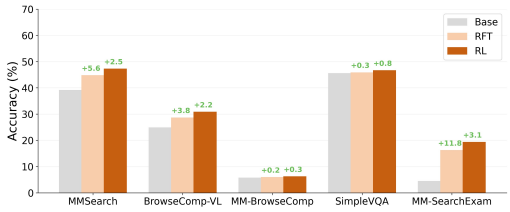


Figure 4: Performance comparison between base model, RFT model, and RL model.

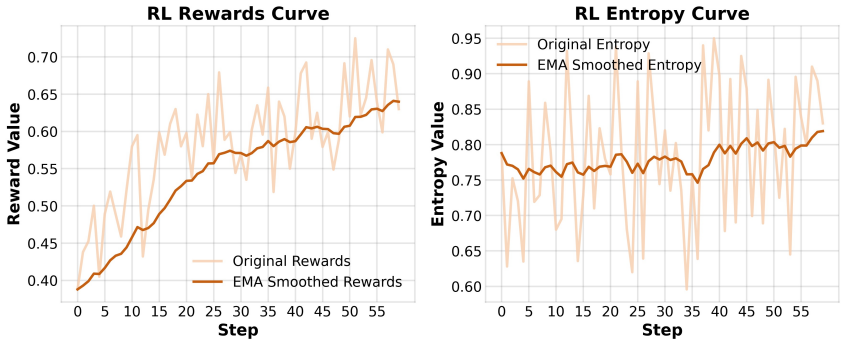


Figure 5: Our RL training dynamic, including the reward curve and RL entropy.

trained with Qwen2.5-VL-7B-Instruct as base model, we also present results trained with Qwen2.5-VL-7B-Instruct, referred to as VSearcher-7B. We observe that VSearcher-8B achieves superior performance compared with various agentic multimodal models. VSearcher-7B also surpasses strong agentic multimodal models by clear margins across various benchmarks. Notably, VSearcher-8B also outperforms strong proprietary models, *e.g.*, GPT-5, on several benchmarks, including MMSearch and BrowseComp-VL. These results illustrate the strong multimodal browsing capabilities of VSearcher. Moreover, the strong performance of VSearcher validates the effectiveness of rejection-sampling fine-tuning with high-quality teacher trajectories for instilling initial multi-turn tool-use abilities, as well as reinforcement learning for further enhancing multi-turn multimodal browsing skills. Moreover, it is clear that long-horizon search agents typically achieve higher performance than short-horizon search agents, illustrating the superiority of long-horizon search capabilities in effectively addressing challenging search problems.

4 Detailed Analysis

Synthesized Data Quality. We assess the quality of our synthesized data by comparing the number of tool calls required to solve each browsing task (see Fig. 3). We perform this analysis by utilizing Gemini-3-Pro-Thinking to solve both our synthesized data and MMSearch [14] under the ReAct framework. We observe that (1) the distribution of tool-call counts for our synthesized data is closely calibrated to that of MMSearch. This strong alignment between the complexity of our agent training data and the target evaluation benchmark helps build appropriate multi-turn multimodal browsing skills for real-world tasks. (2) The tool-call distribution for our synthesized data exhibits a long-tail pattern: some synthesized tasks require Gemini-3-Pro-Thinking to utilize more than 20 tool calls to solve. These highly challenging samples can help VSearcher develop more complex multi-turn browsing skills during rejection-sampling fine-tuning and RL.

Performance Improvement Analysis. We analyze accuracy improvements across three stages: the base model, the model after rejection-sampling fine-tuning (RFT), and the model after RL (see Fig. 4). We observe a progressive improvement from the base model to RFT and then to RL. During RFT stage, the teacher model, Gemini-3-Pro-Thinking, transfers its strong browsing behaviors to the base model, which helps the base model learn to solve complex browsing tasks. After RL, model obtains further improvements in answer accuracy. This suggests that RL further generalizes the initial multi-turn browsing skills learned during RFT.

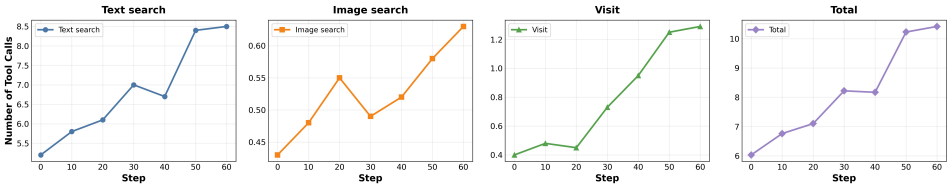


Figure 6: Analysis of tool call count for different tools during the RL process.

RL Training Dynamics. We present the reward curve and RL entropy during training (see Fig. 5). We observe that the reward exhibits a steady increasing trend throughout RL, indicating that RL in real-world web environment effectively improves the multi-turn browsing skills of our model. Regarding entropy, it remains stable over the entire RL process. The agent maintains a reasonable level of exploration, avoiding both entropy collapse and overly aggressive exploration.

Tool Call Number Analysis. We analyze tool-use dynamics during RL (see Fig. 6). We observe text search is primary source of information during information seeking. Average number of image search is around 0.6. Since we only conduct image search for image provided in question, agent perform at most one time of image search for one question. Number of visit calls increases steadily, indicating that model increasingly learns to leverage visit to gather more detailed evidence.

5 Related Work

Deep Research Agent. The construction of deep research agents can be categorized as static or dynamic [12, 35, 43]. Static deep research agents follow explicitly predefined pipelines [1, 6, 8]. Dynamic workflows focus on adaptive task planning: they dynamically design the solution pipeline based on the task and user feedback. Planning-only methods directly generate plans from initial user query, which is used by most deep research systems. The intent-to-planning strategy first clarifies user intent through targeted questions and then generates task plan, which is utilized by OpenAI DR. Different from existing works, our method focuses on multimodal domain, solving complex multimodal questions by dynamically using various tools, such as image search, text search, and visit.

Multimodal QA Benchmark. Most existing multimodal QA benchmarks focus on evaluating single-step prediction or limited retrieval ability, with little emphasis on long-horizon reasoning and planning [16, 16]. Benchmarks such as OK-VQA [22] and A-OKVQA [27] emphasize evaluation of direct answer prediction, without requiring complex retrieval or reasoning. Recent efforts have begun to shift attention toward model reasoning and tool-use capabilities. MMT-Bench [38] covers large-scale, planning-oriented tasks across multiple domains; however, its multiple-choice output format restricts thorough evaluation of long-horizon reasoning. Dyn-VQA [18] constructs adaptive query tasks but remains limited in both multimodal diversity and scale. In contrast to existing benchmarks, we propose a challenging benchmark designed to evaluate long-horizon tool-use capabilities of MLLMs, which is large-scale and constructed in fully automated manner.

Large Vision-Language Model. Large Vision-Language Models (LVLMs) have demonstrated strong general performance in multimodal understanding [39, 20, 40, 41]. Current LVLMs achieve high accuracy across a wide range of vision tasks, including general visual capabilities, visual question answering, OCR, grounding, and visual reasoning. Despite these advances, such models typically rely on static knowledge and are unable to incorporate up-to-date information. In contrast to existing approaches, our method enhances base LVLMs with both text search and image search tools in a real-world web environment and facilitates long-horizon, multi-turn tool usage, significantly improving the ability of model to solve complex multimodal problems.

6 Limitations

One limitation of this work is the high API cost associated with tool calling, which reduces the ease of reproducing our experimental results and conducting follow-up research. This cost is difficult to avoid when building long-horizon agents that interact with real-world web environments. To mitigate this issue, we will open-source the cached tool-response data, thereby reducing the cost of relevant tool calls and facilitating future research. In addition, the evaluation of our model is not yet fully comprehensive, as benchmarks designed to assess multimodal search capability remain relatively

scarce. We will conduct more thorough evaluations as more benchmarks for multimodal search agents become available, in order to better understand the weakness of our model.

7 Conclusion

In this paper, we propose VSearcher, a long-horizon, multi-turn multimodal search agent trained with reinforcement learning in real-world web environment. VSearcher can adaptively leverage text search, image search, and visit tools to retrieve web information and solve challenging tasks. We propose a systematic post-training framework to convert static multimodal model into multimodal search agent. In particular, we propose Iterative Injection-based Data Synthesis pipeline, together with rigorous data filtering strategies, to automatically synthesize highly challenging multimodal browsing tasks. We further adopt rejection-sampling fine-tuning to transfer multi-turn browsing ability from proprietary teacher model to base multimodal models, and apply reinforcement learning to further generalize the tool-use capabilities. For more thorough evaluation of multimodal search agent, we curate a highly challenging benchmark, MM-SearchExam. Across extensive evaluations on multiple multimodal search benchmarks, VSearcher consistently outperforms recent multimodal search agents and even surpasses several proprietary models.

Acknowledgment

We acknowledge supports from Guangdong Basic and Applied Basic Research Foundation 2025A1515012281, Nanjing Municipal Science and Technology Bureau 202401035 and University of Macau MYRG-GRG2024-00077-FST-UMDF.

References

[1] Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. A survey on rag with llms. *Procedia computer science*, 246:3781–3790, 2024.

[2] Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundation models defining a new era in vision: a survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4): 2245–2264, 2025.

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

[4] Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025.

[5] Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert Van Rooij, Kun Zhang, and Zhouchen Lin. Empowering llms with logical reasoning: A comprehensive survey. *arXiv preprint arXiv:2502.15652*, 2025.

[6] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729, 2024.

[7] Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. From llm reasoning to autonomous ai agents: A comprehensive review. *arXiv preprint arXiv:2504.19678*, 2025.

[8] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

[9] 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.

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

[11] Sihao Hu, Tiansheng Huang, Gaowen Liu, Ramana Rao Kompella, Fatih Ilhan, Selim Furkan Tekin, Yichang Xu, Zachary Yahn, and Ling Liu. A survey on large language model-based game agents. *arXiv preprint arXiv:2404.02039*, 2024.

[12] Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Huichi Zhou, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, et al. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*, 2025.

[13] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

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

[15] 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*, 2024.

[16] Jian Li, Weiheng Lu, Hao Fei, Meng Luo, Ming Dai, Min Xia, Yizhang Jin, Zhenye Gan, Ding Qi, Chaoyou Fu, et al. A survey on benchmarks of multimodal large language models. *arXiv preprint arXiv:2408.08632*, 2024.

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

[18] 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*, 2024.

[19] Yi-Cheng Lin, Kang-Chieh Chen, Zhe-Yan Li, Tzu-Heng Wu, Tzu-Hsuan Wu, Kuan-Yu Chen, Hung-yi Lee, and Yun-Nung Chen. Creativity in llm-based multi-agent systems: A survey. *arXiv preprint arXiv:2505.21116*, 2025.

[20] Daizong Liu, Mingyu Yang, Xiaoye Qu, Pan Zhou, Yu Cheng, and Wei Hu. A survey of attacks on large vision–language models: Resources, advances, and future trends. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[21] Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*, 2025.

[22] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

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

[24] Avinash Patil and Aryan Jadon. Advancing reasoning in large language models: Promising methods and approaches. *arXiv preprint arXiv:2502.03671*, 2025.

[25] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[26] Aske Plaat, Max van Duijn, Niki Van Stein, Mike Preuss, Peter van der Putten, and Kees Joost Batenburg. Agentic large language models, a survey. *Journal of Artificial Intelligence Research*, 84, 2025.

[27] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pages 146–162. Springer, 2022.

[28] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[29] Sijun Tan, Michael Luo, Colin Cai, Tarun Venkat, Kyle Montgomery, Aaron Hao, Tianhao Wu, Arnav Balyan, Manan Roongta, Chenguang Wang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. rllm: A framework for post-training language agents. `https://pretty-radio-b75.notion.site/rLLM-A-Framework-for-Post-Training-Language-Agents-21b81902c146819db63cd98a54ba5f31`, 2025. Notion Blog.

[30] Zhengwei Tao, Haiyang Shen, Baixuan Li, Wenbiao Yin, Jialong Wu, Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Liwen Zhang, et al. Webleaper: Empowering efficiency and efficacy in webagent via enabling info-rich seeking. *arXiv preprint arXiv:2510.24697*, 2025.

[31] Qiuchen Wang, Ruixue Ding, Yu Zeng, Zehui Chen, Lin Chen, Shihang Wang, Pengjun Xie, Fei Huang, and Feng Zhao. Vrag-rl: Empower vision-perception-based rag for visually rich information understanding via iterative reasoning with reinforcement learning. *arXiv preprint arXiv:2505.22019*, 2025.

[32] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S Yu. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256. IEEE, 2023.

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

[34] Fengli Xu, Qianyue Hao, Chenyang Shao, Zefang Zong, Yu Li, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, et al. Toward large reasoning models: A survey of reinforced reasoning with large language models. *Patterns*, 6(10), 2025.

[35] Renjun Xu and Jingwen Peng. A comprehensive survey of deep research: Systems, methodologies, and applications. *arXiv preprint arXiv:2506.12594*, 2025.

[36] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

[37] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024.

[38] Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, et al. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi. *arXiv preprint arXiv:2404.16006*, 2024.

[39] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024.

[40] Ruiyang Zhang, Hu Zhang, and Zhedong Zheng. Vl-uncertainty: Detecting hallucination in large vision-language model via uncertainty estimation. *arXiv preprint arXiv:2411.11919*, 2024.

[41] Ruiyang Zhang, Hu Zhang, Hao Fei, and Zhedong Zheng. Uncertainty-o: One model-agnostic framework for unveiling uncertainty in large multimodal models. *arXiv preprint arXiv:2506.07575*, 2025.

[42] Ruiyang Zhang, Dongzhan Zhou, and Zhedong Zheng. Sketchthinker-r1: Towards efficient sketch-style reasoning in large multimodal models. *arXiv preprint arXiv:2601.02825*, 2026.

[43] Wenlin Zhang, Xiaopeng Li, Yingyi Zhang, Pengyue Jia, Yichao Wang, Huifeng Guo, Yong Liu, and Xiangyu Zhao. Deep research: A survey of autonomous research agents. *arXiv preprint arXiv:2508.12752*, 2025.

[44] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, 16(12):9851–9915, 2025.

A Discussion

MM-SearchExam is synthesized with the same pipeline as the training data. Does VSearcher overfit to specific patterns, or does it learn general search abilities? We have evaluated VSearcher on 4 additional benchmarks, including MMSearch, MM-BrowseComp, SimpleVQA, which are human-annotated, and BrowseComp-VL, which is synthesized with a distinct pipeline from our data synthesis process. Across all of these benchmarks, VSearcher consistently shows clear improvements over the baselines. These results suggest that our synthesized challenging multimodal browsing tasks do not simply causing agent to overfit to specific pattern but facilitate the construction of general long-horizon web search ability.

What are the contributions over previous works, *e.g.*, Search-R1, MMSearch-R1, and DeepMMSearch-R1? Three key features distinguish our method from prior work: (1) a comprehensive data synthesis pipeline capable of generating complex multimodal browsing tasks at large scale; (2) a focus on incentivizing long-horizon, multi-turn tool-use capability; and (3) a thorough post-training pipeline for large multimodal model. For Search-R1, this work focuses on building reasoning and tool-use capability in large language model. Search-R1 utilizes off-the-shelf QA data for RL training. Its training data is relatively simple, and it only learns short-horizon search behavior, e.g., 2–3 times of tool calls. Moreover, it is trained entirely in offline environment and is equipped only with text search. For MMSearch-R1, this work conducts at most one round of image search and one round of text search. Moreover, its inference process follows a rigid procedure: first text search, then image search, then answer generation. Additionally, its proposed data synthesis pipeline creates simple VQA questions, which are insufficient for facilitating long-horizon search abilities such as consistent evidence gathering and adaptive search strategy adjustment. Quantitatively, VSearcher surpasses this baseline across all five evaluated benchmarks by clear margins (see Tab. 1). For DeepMMSearch-R1, this work utilizes InfoSeek for model training and does not propose any data synthesis method. Notably, the relatively simple training data leads to short-horizon search behavior (see Figures G.1 and G.2 in the appendix of its paper). Quantitatively, VSearcher also surpasses this strong baseline (see Tab. 1).

What is the difference between SFT data and RL data? We first utilize our data synthesis pipeline to generate a large pool of multimodal browsing tasks, including around 9K Easy questions, 3K Medium questions, and 2K Hard questions. Specifically, we perform 1, 3, and 5 rounds of text information injection to synthesize the Easy, Medium, and Hard questions, respectively. We then apply our data filtering strategy to all synthesized questions to ensure their quality and difficulty. From this pool, we sample 3K questions, including 1K questions from each difficulty level. This subset is fed into Gemini-3-Pro-Thinking to perform ReAct inference to generate trajectories. We then conduct rejection sampling on those generated trajectories based on answer correctness. The rejection sampled 1,308 trajectories are utilized as SFT data. We further sample 1K questions with no overlap with SFT training data to serve as RL training data. The RL training data has an Easy:Medium:Hard ratio of 4:3:3.

Why is the average tool call count for the visit tool around 1.2 (see Fig. 6)? How does the agent access URLs from text search and image search with so few visit tool calls? VSearcher calls the visit tool only when it identifies a specific URL as critically important. Notably, the tool responses from text search and image search contain information such as page titles and page snippets, which often already include key information from the corresponding pages. As a result, VSearcher can obtain sufficient information from the tool responses of text search and image search, and then continue the information seeking process without frequently calling the visit tool.

B Implementation Detail

B.1 Tool Implementation

Image Search. We utilize Google Vision Web Detection API to implement our image search tool. The input to this service is the URL of the query image, and it returns web pages that contain images similar to the input. We then post-process the returned pages. Specifically, we iterate over the pages and, for each page, verify whether the similar images it contains are reachable from our server network and correspond to valid image files. If so, we add the image and its source page to the returned list. Specifically, we only select one image for one returned page. For each input image,

we collect information of five similar web images, including the image URL, the URL of the page containing the image, and the page title.

Text Search. Our text search tool is implemented based on Google Custom Search API. The text search tool input is the text query to be searched. For each query, the tool returns the top 5 most relevant web pages. Each result includes the page URL, title, and a snippet that provides a concise preview of the content. The Google Custom Search API has a daily rate limit of 10,000 requests; we use multiple API keys to ensure sufficient daily budget for RL training. The API is provided through Google Cloud, which offers a $300 free quota and is convenient for initial usage.

Visit. We leverage the Jina API to implement our visit tool. The input to the visit tool is a target URL and a specific goal for visiting the page. We first query the Jina API with the URL, which returns the cleaned full content of the web page in Markdown format. Because the returned content can be extremely long, directly leveraging the full content as the tool response in the ReAct process can quickly exhaust the model context limit. Therefore, we further utilize an LLM, Qwen2.5-72B-Instruct, as a summarization model: it takes the full web content and the visit goal as input and summarizes the relevant information conditioned on the goal. This summary is returned as the final tool response.

B.2 Iterative Injection-based Data Synthesis

Seed Selection. We obtain rare entities from Wikidata to serve as seeds, which helps synthesize challenging multimodal browsing tasks. For an entity in Wikidata, 'sitelink' refers to the number of Wikipedia page links in different languages, and 'statement' refers to the number of atomic facts associated with the entity. We control entity rarity by requiring the number of sitelinks to be below a threshold, while ensuring sufficient coverage by requiring the number of statements to be above a threshold. In practice, we select seed entities with sitelink ≤ 10 and statement ≥ 20. We utilize SPARQL queries to access the official Wikidata Query Service and retrieve seed entities.

Initial Question Generation. For each seed entity, we retrieve its Wikipedia page content and generate a simple QA pair based on the content, which serves as the starting point for subsequent iterative extensions. Specifically, we download an offline Wikipedia dump (2025-08) and serve Wikipedia locally for more convenient content acquisition. After obtaining the Wikipedia content for the seed entity, we utilize Qwen2.5-72B-Instruct to generate a simple QA pair about the entity.

Text Information Injection. Starting from the initial QA pair, we perform several rounds of text information injection to transform it into a complex browsing task. Each round contains three steps: entity selection, information parsing, and question transformation. In each round, we first utilize Qwen2.5-72B-Instruct to select an entity from the current question. We provide examples in the prompt to encourage selecting meaningful entities (*e.g.*, World War II) rather than ambiguous ones (*e.g.*, theory). After selecting an entity, we retrieve its Wikipedia content from the offline index and parse a specific fact about the entity with Qwen2.5-72B-Instruct. For information parsing, we explicitly prompt the LLM to extract less well-known information, which increases the challenge of the extended question and improves the difficulty of the final browsing task. We also encourage the extracted information to include as many entities as possible, so that subsequent injection rounds have additional entities to extend. We also prompt LLM to extract information that is concise and in one sentence, to avoid over-long question. Finally, we prompt Qwen2.5-72B-Instruct to transform the current question by hiding the selected entity and replacing it with the parsed information.

Image Injection. After the text information injection process, we perform image injection to transform the question into a multimodal browsing task. We first prompt Qwen2.5-72B-Instruct to select an image entity from the question. Notably, we explicitly prompt the LLM to choose an entity that is critical for answering the question. We then prompt the LLM to replace the selected entity with a phrase such as 'shown in the image', and retrieve Wikipedia image of the selected entity as the image in the multimodal browsing task. This design ensures that the injected image is critical for answering the question correctly, which faciliates multimodal search agent to focus on the multimodal input.

Difficulty Level Design. We explicitly design 3 difficulty levels for the synthesized data to ensure that our trained search agent can handle tasks requiring different degrees of browsing effort. Specifically, we construct 3 levels of multimodal browsing tasks: Easy, Medium, and Hard. For Easy questions, we perform one round of text information injection. For Medium questions, we perform three rounds. For Hard questions, we perform five rounds.

Data Filtering. We design 4 filtering criteria to ensure the difficulty and quality of the synthesized data. First, we feed the synthesized question to Qwen2.5-VL-72B-Instruct and prompt it to answer directly; we filter out questions that it can answer correctly without tool use. These questions typically require few tool calls and are ineffective for improving multi-turn tool-use capabilities. Second, we input only the text part of the synthesized question to Qwen2.5-72B-Instruct and filter out questions that can be answered correctly from text alone. Third, we filter questions with overly simple images (*e.g.*, images that contain only easily recognizable text). Specifically, we utilize Qwen2.5-VL-72B-Instruct as an image judge and filter samples that it deems too simple. Finally, we filter corner cases where the answer is explicitly revealed in the question, which can happen when text information injection injects the answer into the question.

B.3 Rejection Sampling Finetuning

Teacher Model We utilize Gemini-3-Pro-Thinking as the teacher model to generate multi-turn tool-calling trajectories. We choose it for two reasons. (1) We observe strong multi-turn browsing capabilities in Gemini-3-Pro-Thinking, including adaptive image search for complex image inputs, dynamically adjusting text search queries in creative ways to retrieve useful information, and issuing appropriate visit actions to obtain more concrete evidence. (2) We find that it achieves strong performance on various browsing-focused benchmarks, surpassing other proprietary models by clear margin. These advantages make it well-suited as a teacher model for generating high-quality information-seeking trajectories.

Trajectory Generation We utilize a standard ReAct process with Gemini-3-Pro-Thinking to generate information-seeking trajectories. We set the max number of tool calls to 30. The trajectory timeout is set to 2 hours 30 minutes. The max context length is set to 200K. We set the rollout temperature to 0.6, top_p to 0.95, the max new tokens to 10,000, and the presence penalty to 1.5 to reduce repetition. Notably, we enable strict format checking during each round of rollout process. For format checking, each response must contain exactly one pair of <think> and </think> tags. After the thinking section, the response must contain either <tool_call></tool_call> or <answer></answer>; having both is not allowed. All XML tags must be properly closed. The tag order must be either <think>...</think><tool_call>...</tool_call> or <think>...</think><answer>...</answer>. Each response must contain exactly one tool call. If a tool call is present, its content must be a valid JSON object. If a model response fails format checking, we terminate the rollout immediately. This strict format control ensures that the cold-started model learns the correct response format.

Rejection Sampling We perform rejection sampling on the generated trajectories to retain high-quality trajectories as effective supervision signals for the base models during finetuning process. Rejection sampling is based on the final answer correctness. We evaluate answer correctness with an LLM-as-a-judge approach, following the practice in BrowseComp. We utilize Qwen2.5-72B-Instruct as the judge model. We retain only trajectories that yield correct final answer.

Supervised Finetuning After rejection sampling, we collect 1,301 high-quality trajectories from Gemini-3-Pro-Thinking. We then perform supervised fine-tuning on the base models with these trajectories to instill initial multi-turn tool-calling abilities. We utilize LLaMA-Factory as the SFT framework. Notably, we perform full fine-tuning to improve instruction-following ability. We set the per-device batch size to 1 and the gradient accumulation steps to 2. The learning rate is set to 1e-5. The number of epochs is set to 5. We utilize a warmup ratio of 0.1. The cutoff length is set to 20,480.

B.4 RL Training Details

We utilize rllm as the RL training framework. We choose this framework because it is specifically designed for agentic RL training. In the latest version of rllm, it is convenient to convert a multi-turn tool using ReAct process into multi-turn tool-use RL training.

For RL, the training batch size is set to 32, and number of rollouts is 8 per training sample. The max prompt length is set to 2K, and the max new token is set to 28K, resulting in max model length of 30K. The max number of tool calls is set to 30. The total trajectory timeout is set to 2 hours 30 minutes, since the high degree of parallelism during RL training can slow down the rollout process, especially for trajectories with extensive tool calls. The learning rate is set to 2e-6. We disable the KL penalty to encourage more exploration during RL. The rollout temperature is set to 0.6. We filter out rollout trajectories that exceed the max prompt length, exceed the max response length, exceed

the max tool-call limit, time out, or encounter unexpected errors. These trajectories are utilized only for advantage computation and are excluded from the loss calculation. Finally, we use 1K synthesized samples from our proposed Iterative Injection-based Data Synthesis for RL training, with Easy:Medium:Hard ratio of 4:3:3. We utilize Qwen3-VL-2B-Thinking, Qwen3-VL-4B-Thinking, and Qwen3-VL-8B-Thinking after rejection-sampling fine-tuning as the RL base models. We utilize 16 H100s (80G) for all experiments.

B.5 Benchmark

MMSearch. MMSearch is a pioneering benchmark for evaluating multimodal browsing capabilities. It contains 300 QA questions spanning 14 topics, such as finance and entertainment. It includes 171 samples with image input and 129 samples without image. Since we focus on multimodal search, we evaluate on the 171 samples with image input.

BrowseComp-VL. BrowseComp-VL is a recently proposed and challenging multimodal search benchmark. It contains two subsets, Level1 and Level2. The Level1 subset includes 199 samples that already require challenging multi-hop reasoning. The Level2 subset includes 200 samples and is more difficult than Level1 because key information is further fuzzed to increase question difficulty. We utilize all samples in this benchmark for evaluation since they all include image input.

MM-BrowseComp. MM-BrowseComp contains 224 challenging, hand-crafted questions designed to evaluate multimodal retrieval ability. The questions are constructed in order to make sure text alone is insufficient to answer correctly. We filter out samples without image input, resulting in 122 samples for evaluation.

SimpleVQA. SimpleVQA is designed to evaluate the factual question-answering ability of multimodal models. It consists of questions with short natural-language descriptions of factual knowledge. It contains 2,025 samples spanning 9 topics and is designed to have short, concrete answers that are easy to verify. We randomly sample 180 questions from its English subset for evaluation.

B.6 API Cost Analysis

Table 2: Details of the API costs for our utilized tools, including image search, visit, and text search.

	Image Search	Visit	Text Search
Total	21$	100$	500$
RFT	17.5$	5$	50$
RL	3.5$	95$	450$

We provide detailed budget breakdown in Tab. 2. For image search, agent only conducts image search on image provided in the multimodal browsing task. Consequently, it calls image search at most once per training sample, and total number of image search calls is relatively small. For visit, agent invokes it only when it finds a URL of critical importance. As a result, number of visit calls is also relatively small, averaging around 1.2 calls per trajectory (see Fig. 6). The majority of tool cost comes from text search, with total text search cost being approximately $500. Google Cloud provides $300 initial free credits, which helps offset overall budget.

C More Analysis

Tool Call Distribution. We present analysis of the tool call distribution of VSearcher across different benchmarks (see Fig. 7). Notably, text search serves as the primary source of information. In addition, we observe that on benchmarks where images play more important role in correct question answering, *e.g.*, MM-BrowseComp, VSearcher conducts substantially more image search than on benchmarks where images are typically simple, *e.g.*, SimpleVQA.

Sensitivity Analysis of Judge Model. We present a sensitivity analysis of our utilized LLM judge for answer correctness evaluation (see Tab. 3). Specifically, we also use GPT-5.4 as the judge model and evaluate the performance of VSearcher on our adopted benchmarks. We observe that the evaluation results from GPT-5.4 align closely with those from Qwen2.5-72B-Instruct. This suggests that the influence of bias from specific judge model could be limited.

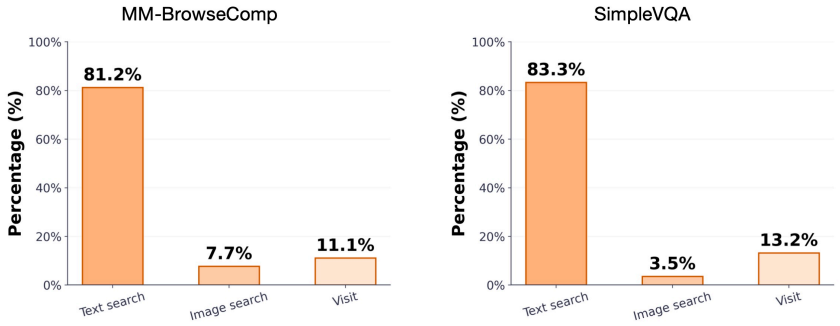


Figure 7: Tool call distribution of VSearcher across different benchmarks.

Table 3: Sensitivity analysis of the judge model utilized for answer evaluation.

Judge Model	MMSearch	BrowseComp-VL	MM-BrowseComp	SimpleVQA	MM-SearchExam
Qwen2.5-72B-Instruct	46.2	19.3	12.5	56.2	24.9
GPT-5.4	46.4	19.3	12.4	56.5	24.9

Table 4: Time cost comparison between various models.

Model	Avg. Time per Trajectory	Avg. Tool Call Count	Accuracy
MMSearch-R1	14.3s	1.8	16.2
DeepEyesV2	35.8s	4.2	17.8
VSearcher	74.2s	10.7	19.3

Time Cost Analysis. We present time cost analysis of various agentic multimodal models (see Tab. 4). Specifically, we conduct this analysis on BrowseComp-VL. We observe that VSearcher yields higher time cost, mainly due to more rounds of inference loop in ReAct inference process. However, VSearcher surpasses performance of all baseline models by clear margin.

Additional Benchmark Evaluation. We present additional evaluation of our model on MMSearch-Plus and BrowseComp-V^3 to illustrate the generalization ability of our search agent (see Tab. 5). We observe that VSearcher still consistently outperforms recent multimodal search agents by a clear margin. This illustrates the generalization and robustness of VSearcher search capabilities and verifies that our post-training pipeline indeed equips the base model with genuine search abilities.

Ablation of Trajectory Reward. We present the ablation results of our proposed trajectory reward on both 7B and 8B scale models (see Tab. 6). We observe a clear positive impact of our proposed trajectory-level reward on model performance across various benchmarks. This demonstrates that dense trajectory-level supervision is effective in facilitating stronger long-horizon capabilities, encouraging the model to optimize toward more effective information-seeking process.

Ablation of Search Strategies. We present ablation results of our proposed two search strategies (see Tab. 7). Through experimental validation, we observe that this reward design can facilitate the emergence of directional search patterns beyond the previous vanilla search capability and lead to overall performance improvements. Through case studies, we also observe that the search agent exhibits stronger capabilities in solving tasks such as "Count the number of times a specific hero is used by an e-sports player."

Ablation of Reward Weight for Trajectory Reward. We present ablation results of different weights for trajectory reward (see Tab. 8). We observe that adding a small trajectory reward already yields a positive influence on model performance. This illustrates that dense supervision signals over the agent full trajectory are highly beneficial for long-horizon rollouts. On the other hand, assigning an excessively large weight to the trajectory reward, such as 0.5, can compromise the model final performance. We analyze that this could weaken the priority of final-answer correctness during the training process and consequently degrade model performance. Through case studies, we observe that, in some trajectories, the model can identify several key entities but fail to synthesize the correct answer from the extensive search context. This pattern could lead to lower model performance.

Table 5: Comparison between different baselines across MMSearch-Plus and BrowseComp-V^3.

Model	Type	MMSearch-Plus	BrowseComp-V^3
Open-source Models with Web Tools			
Qwen3-VL-4B-Thinking	Short-Horizon	9.3	5.3
Qwen3-VL-8B-Thinking	Short-Horizon	12.7	7.2
Qwen3-VL-30B-A3B-Thinking	Short-Horizon	13.5	10.4
Qwen2.5-VL-7B-Instruct	Short-Horizon	4.3	4.1
Qwen2.5-VL-72B-Instruct	Short-Horizon	6.1	7.3
InternVL3.5-8B	Short-Horizon	3.8	3.2
InternVL3.5-38B	Short-Horizon	5.7	6.2
Proprietary Models with Web Tools			
GPT-4o	Short-Horizon	5.9	6.8
GPT-5	Long-Horizon	18.6	21.5
Gemini-3-Flash	Long-Horizon	21.1	17.8
Gemini-3-Pro	Long-Horizon	25.8	22.9
Qwen3-VL-Plus	Long-Horizon	16.4	18.4
Agentic Multimodal Models			
MMSearch-R1-7B	Short-Horizon	10.8	9.3
DeepMMSearch-R1-7B	Short-Horizon	-	-
WebWatcher-7B	Long-Horizon	12.6	13.7
DeepEyesV2-7B	Short-Horizon	13.5	14.2
VSearcher-7B	Long-Horizon	15.1	16.3
VSearcher-8B	Long-Horizon	16.6	17.5

Table 6: Ablation of our proposed trajectory reward.

Model	MMSearch	BC-VL	MM-BC	SimpleVQA	MM-SearchExam
VSearcher-7B wo. Trajectory Reward	44.5	29.1	5.7	45	17.2
VSearcher-7B	45.3	30.2	5.9	45.3	17.8
VSearcher-8B wo. Trajectory Reward	46	29.3	5.8	45.8	17.9
VSearcher-8B	47.2	30.8	6.2	46.6	19.3

Table 7: Ablation of our proposed search strategies. The baseline result is trained with both accuracy reward and trajectory reward applied.

Model	MMSearch	BC-VL	MM-BC	SimpleVQA	MM-SearchExam
VSearcher-8B	47.2	30.8	6.2	46.6	19.3
VSearcher-8B w. DeepSearch	47.8	32	6.5	47.1	19.8
VSearcher-8B w. WideSearch	48.3	31.5	7.2	47.3	19.6
VSearcher-8B w. DeepSearch & WideSearch	**48.4**	**32.2**	**7.6**	**47.8**	**20.1**

Comparison with Zero-Style Baseline. We present comparison between VSearcher-8B and a baseline trained only with RL (see Tab. 9). We observe a clear performance drop compared with the model trained using the RFT-to-RL pipeline. This illustrates that RFT process can instill basic long-horizon search ability into base model and elevate the performance upper-bound of trained agent.

Ablation of Main Training Stages. We present detailed ablation results of main training phases for our VSearcher (see Tab. 10). Specifically, Qwen3-VL-8B-Thinking w. RFT (1K & 16 Epochs) represents the SFT experiment with compute matched to RL. During RL, we train on 1K samples for 2 epochs with the rollout number set to 8, resulting in approximately 16 epochs over the 1K samples. During evaluation, we randomly vary the seed within [0, 1000], run each setting four times, and report the final mean and standard deviation. We observe that (1) both the RFT-only and RL-only variants surpass the performance of Base + Tools, illustrating that both RFT and RL are effective training methods for equipping the model with long-horizon search capabilities. (2) The RFT + RL variant outperforms the RL-only variant. This demonstrates that RFT and RL form an effective post-training combination for eliciting the upper bound of long-horizon search capabilities. Conducting RFT first and instilling initial long-horizon search capabilities into the base model can effectively raise the model capability upper bound. (3) Relying only on RFT, including scaling the RFT data and

Table 8: Ablation of the weight of trajectory reward. The ablation is conducted on VSearcher-8B.

Weight	MMSearch	BC-VL	MM-BC	SimpleVQA	MM-SearchExam
0	46	29.3	5.8	45.8	17.9
0.05	46.3	29.4	6	46.1	18.2
0.1	**47.2**	**30.8**	**6.2**	**46.6**	**19.3**
0.2	46.9	30.2	5.9	46.3	18.4
0.3	46.5	29.1	5.9	46	18.2
0.5	44.2	28.7	5.4	45.2	16.4

Table 9: Comparison with model trained directly with RL.

Model	MMSearch	BC-VL	MM-BC	SimpleVQA	MM-SearchExam
VSearcher-8B-Zero	42.8	26.1	5.5	45.9	8.7
VSearcher-8B	47.2	30.8	6.2	46.6	19.3

Table 10: Detailed ablation of main training stages.

Model	MMSearch	BC-VL	MM-BC	SimpleVQA	MM-SearchExam
Qwen3-VL-8B-Thinking w. Tools	39.1±0.3	24.8±0.2	5.7±0.2	45.5±0.3	4.4±0.3
Qwen3-VL-8B-Thinking w. RFT	44.7±0.5	28.6±0.5	5.9±0.1	45.8±0.8	16.2±0.2
Qwen3-VL-8B-Thinking w. RL	42.8±0.7	26.1±1.3	5.5±0.2	45.9±0.2	8.7±0.2
Qwen3-VL-8B-Thinking w. RFT & RL	47.2±0.4	30.8±0.1	6.2±0.3	46.6±0.1	19.3±0.2
Qwen3-VL-8B-Thinking w. RFT (2k)	45.2±0.2	29.1±0.3	5.9±0.2	46.1±0.3	16.8±0.8
Qwen3-VL-8B-Thinking w. RFT (3k)	45.5±0.3	29.3±0.8	6±0.1	46.3±0.5	17.2±0.3
Qwen3-VL-8B-Thinking w. RFT (1k & 16 Epoch)	43.2±0.5	27.8±0.2	5.8±0.5	45.6±0.2	14.8±1.5

conducting RFT with compute matched to RL, cannot surpass the variant with RL training. This reveals that RFT alone is insufficient to facilitate strong long-horizon capabilities in the base model. The RL training process is essential for further generalizing long-horizon search capabilities and achieving higher performance.

Statistics of MM-SearchExam. We present time cost analysis of various agentic multimodal models (see Tab. 11). Specifically, the distribution of tool calls is obtained by evaluating Gemini-3-Pro-Thinking on MM-SearchExam with the tool-call budget extended to 100. During the data filtering process, we already filtered out synthesized samples that could be correctly answered by Qwen2.5-VL-72B-Instruct using only the textual questions, resulting in an image modality participation rate of 100%.

Category Distribution of MM-SearchExam. We present the category distribution analysis of questions in MM-SearchExam (see Fig. 8). MM-SearchExam contains diverse questions spanning across 8 different categories.

Necessity of Long-Horizon Behavior. We present results of VSearcher with different tool call limits during inference time (see Tab. 12). We observe that when VSearcher tool call count is capped at 5, there is a clear drop in evaluation performance. This directly illustrates the necessity of long-horizon search behavior. Through case studies, we also find that the browsing tasks in these benchmarks are quite challenging, and even humans using a computer browser cannot locate the final answer within only 5 search actions, which further verifies the importance of long-horizon search capabilities for search agents. Moreover, recent proprietary models such as GPT-5 and Gemini-3-Pro typically conduct more than 10 web searches when solving these tasks, even though these models have larger scales and more extensive prior knowledge. This further validates the reasonableness of developing long-horizon search capabilities in open-source models.

Challenge of Our Synthesized Questions. We present average tool call count of Gemini-3-Pro-Thinking on our synthesized questions from different difficulty level (see Tab. 13). We observe that the actual number of tool calls made by Gemini-3-Pro-Thinking clearly exceeds the number of injection rounds used during the data synthesis process. This illustrates that our synthesized data cannot be directly solved through simple reverse search and are quite challenging from a problem-solving perspective, thereby effectively facilitating robust reasoning capabilities and complex browsing abilities.

Table 11: Various statistics of MM-SearchExam.

Type	Metric	Value
Data Length	Average Question Length	156.2 tokens
	Average Answer Length	3.5 tokens
Distribution of Tool Calls	# Tool Calls (Mean)	15.3
	# Tool Calls (Max)	62
	# Tool Calls (Min)	6
Multimodal	Image Modality Participation Rate	100%
Topic	# Major Topic	8

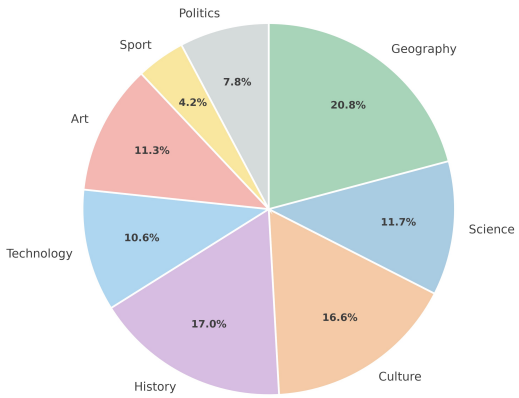


Figure 8: Distribution analysis of question categories in MM-SearchExam.

Table 12: Analysis of VSearcher performance under different tool call thresholds. VSearcher-8B (Original) refers to results of VSearcher with 30 tool call time limit.

Model	MMSearch	BC-VL	MM-BC	SimpleVQA	MM-SearchExam
VSearcher-8B w. Cap at 5	38.3	17.2	5.1	41.8	5.8
VSearcher-8B w. Cap at 10	40.8	21.4	5.4	43.7	8.7
VSearcher-8B w. Cap at 15	45.3	26.4	5.8	45.3	16.4
VSearcher-8B (Original)	**47.2**	**30.8**	**6.2**	**46.6**	**19.3**

Table 13: Average tool call count of Gemini-3-Pro-Thinking on our synthesized questions.

Level	# Avg. Tool Calls
Easy	3.1
Medium	8.5
Hard	12.8

D Prompt

This is the system prompt for our multimodal search agent, used for both the ReAct inference and RL training processes.

System Prompt

You are a multimodal deep research agent. Given a user question, you should conduct thorough searches across various information sources on the real-world internet, perform analysis and reasoning, and give accurate answers to the user question.

Workflow:

- You should first conduct reasoning within <think>< /think> tags. This includes analysis of the given question, interpretation of tool-returned information, and analysis of what actions need to be taken next.
- If you think you need to call a tool to provide you with additional information, you should call the tool within <tool_call>< /tool_call> tags.
- The returned information from tools will be returned to you within <tool_response>< /tool_response> tags.
- If you think you have gathered enough information and are confident you can answer the question, provide your final answer within <answer>< /answer> tags (e.g., <answer>Titanic< /answer>). Do not provide explanations in <answer>< /answer> tags.

Guidelines:
- If you need to call a tool, you should call only one tool at a time. Do not call multiple tools at the same time.
- You should not provide tool responses yourself. You should only call tools and wait for the tool response.

Tool set:
<tools>
{"type": "function", "function": {"name": "text_search", "description": "Perform a Google web search and return a string of the top search results.", "parameters": {"type": "object", "properties": {"query": {"type": "string", "description": "The search query."}}, "required": ["query"]}}}
{"type": "function", "function": {"name": "image_search", "description": "Perform a Google image search on the given image and return a list of URLs of similar images, along with the titles and links of the pages where they appear. Note that the image search is only conducted on the initial image provided by the user, so no parameters are needed for this tool."}}
{"type": "function", "function": {"name": "visit", "description": "Visit a webpage and return a summary of its content.", "parameters": {"type": "object", "properties": {"url": {"type": "string", "description": "The URL of the webpage to visit."}, "goal": {"type": "string", "description": "The specific information goal for visiting the webpage."}}, "required": ["url", "goal"]}}}
< /tools>

For each function call, return a JSON object with the function name and arguments inside <tool_call></tool_call> tags:
<tool_call>
{"name": <function-name>, "arguments": <args-json-object>}
</tool_call>

Current date:

This is the LLM-as-Judge prompt we adopt, following the prompt in HLE and BrowseComp.

LLM-as-Judge Prompt

Judge whether the following [response] to [question] is correct or not based on the precise and unambiguous [correct_answer] below.

[question]: {question}

[response]: {response}

Your judgement must be in the format and criteria specified below:

extracted_final_answer: The final exact answer extracted from the [response]. Put the extracted answer as 'None' if there is no exact, final answer to extract from the response.

[correct_answer]: {correct_answer}

reasoning: Explain why the extracted_final_answer is correct or incorrect based on [correct_answer], focusing only on if there are meaningful differences between [correct_answer] and the extracted_final_answer. Do not comment on any background to the problem, do not attempt to solve the problem, do not argue for any answer different than [correct_answer], focus only on whether the answers match.

correct: Answer 'yes' if extracted_final_answer matches the [correct_answer] given above, or is within a small margin of error for numerical problems. Answer 'no' otherwise, i.e. if there if there is any inconsistency, ambiguity, non-equivalency, or if the extracted answer is incorrect.

confidence: The extracted confidence score between 0% and 100% from [response]. Put 100 if there is no confidence score available.

This is the prompt for the summarization model to extract relevant information from the full web page content.

Summary Prompt in Visit Tool

You are an expert in summarizing web page information related to a user goal.

Given the following web page content and a specific user goal for visiting this page, you should carefully read the web page content, collect relevant details with respect to the user goal, and generate a paragraph-level summary of those details in response to the user goal. The generated summary should be accurate, clear, and logically coherent.

Web page content:
[WEB PAGE CONTENT]

User goal:
[USER GOAL]

This is the prompt used to generate the initial QA pair for seed entity based on its Wikipedia content.

Prompt for Initial QA Pair Parsing

Generate a question and answer pair based on the page content and the provided entity.

Guidelines:
- The question should be about the provided entity.
- Question should be one sentence.
- The question should be specific so that there exist only one answer.
- The answer should be a word or a short phrase. which is easy to verify correctness.
- The answer should be concise and specific.
- Return the question and answer pair in json format.

The json format should be like:
{
"question": ...,
"answer": ...
}
Do not provide any other information.

Page content:
[PAGE CONTENT]

Entity:
[ENTITY]

This is the prompt for selecting entity for text information injection.

Prompt for Entity Selection

Select a entity from the given question.

Guidelines:
- Only output the entity, no need for other explantion.
- The entity should be substring of the given question, do not answer the question.

Example:
Ukraine, World War II, Ivy League, PageRank, Sverker Johansson, Virginia Woolf, Reykjavík, Borobudur

Do not select entity like this:
German-born theoretical physicist, Polish-born, theory, Roman emperor from 98 to 117 AD,

Text:
[TEXT]

This is the prompt for parsing specific information of selected entity from its Wikipedia content.

Prompt for Entity Information Parsing

Parse information of the entity from the given text.

Guidelines:
- The information should be rarely known.
- The information should contain more entities as possible.
- The information should be short and concise, in one sentence.
- Only output the information, no need for other explantion.

Parse information like this:
First-lap contact with Lewis Hamilton dropped Lando Norris to the back at the Spanish Grand Prix 2023.
A member of the McLaren Young Driver Programme since 2017, Lando Norris joined McLaren in 2019 to partner Carlos Sainz Jr..

Do not parse information like this:
Vladimir Putin has served as President of Russia since 2012.
As observational evidence for a dynamic universe was lacking at the time, Einstein introduced a new term.
Following the discovery of the recession of the galaxies by Edwin Hubble in 1929, Einstein abandoned his static model of the universe, and proposed two dynamic models of the cosmos, the Friedmann–Einstein universe of 1931 and the Einstein–de Sitter universe of 1932. In each of these models, Einstein discarded the cosmological constant, claiming that it was "in any case theoretically unsatisfactory".

Text:
[TEXT]

Entity:
[ENTITY]

This is the prompt for injecting the parsed information into original question and hiding the selected entity.

Prompt for Text Information Injection

Transform the question based on the entity and its information.

Guidelines:
- The entity should be hidden from the question.
- Replace the entity with its information.
- Only output the transformed question, no need for other explantion.

Use the part of the information that is less known.
Example:
Full Information:
Founded by Jimmy Wales and Larry Sanger in 2001, Wikipedia began as a complementary project for Nupedia, switching from its own license to the GNU Free Documentation License at the urging of Richard Stallman.
Use information like this:
switching from its own license to the GNU Free Documentation License at the urging of Richard Stallman.
Do not use information like this:
Founded by Jimmy Wales and Larry Sanger in 2001

Use the part of the information that contains more entities.
Example:
Full Information:
The Mojave phone booth, originally set up in 1948 for volcanic cinder miners, became an internet sensation in 1997 after being featured in a New York Times article, leading to its removal by Pacific Bell in 2000 due to environmental concerns.
Use information like this:
became an internet sensation in 1997 after being featured in a New York Times article, leading to its removal by Pacific Bell in 2000 due to environmental concerns.
Do not use information like this:
The Mojave phone booth, originally set up in 1948 for volcanic cinder miners

Question:
[QUESTION]

Entity:
[ENTITY]

Information:
[INFORMATION]

This is the prompt for selecting image entity from extended question.

Prompt for Image Entity Selection

Select a entity from the given question.

Guidelines:
- Only output the entity, no need for other explantion.
- The entity should be substring of the given question, do not answer the question.
- Select the entity that is more critical to the correct answering of the question.

Example:
Ukraine, World War II, Ivy League, PageRank, Sverker Johansson, Virginia Woolf, Reykjavík, Borobudur

Do not select entity like this:
German-born theoretical physicist, Polish-born, theory, Roman emperor from 98 to 117 AD,

Text:
[TEXT]

This is the prompt for hiding image entity and replacing it with phase like ‘shown in the image’.

Prompt for Image Injection

Transform the question based on the entity.

Guidelines:
- The entity should be hidden from the question.
- Replace the entity with phase like ‘shown in the image’.
- The question should be rephrased to keep logically coherent and easy to understand.

Question:
[QUESTION]

Entity:
[ENTITY]

This is the prompt for obtaining direct answer from model.

Prompt for Direct Answer

[QUESTION]

Please directly answer this question. Answer with a short phrase. Do not provide any other information.

This is the prompt for evaluating the image complexity during data filtering process.

Prompt for Image Evaluation

I am currently building a training dataset for a multimodal deep-research agent. I want the images in the training samples to be complex so that they encourage the agent to perform image search to obtain more information. I want to filter out samples with images that are too simple.

Now, evaluate the given image.

Answer “yes” if the image is simple (e.g., it is the flag of the United States or contains only text that is easy to parse). Answer “no” if the image is complex, information-rich, or hard to interpret.

Just provide the final answer, do not provide any other information.

E Synthesized Data

Case for Easy Data

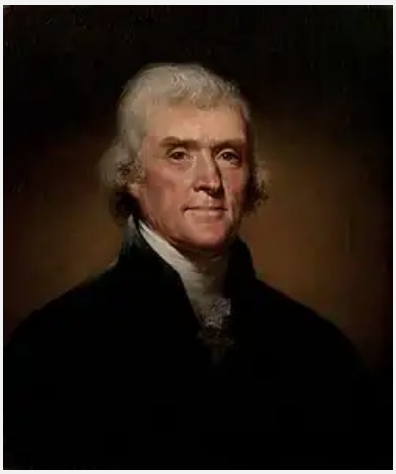
Question: In which county is the state park located that was the first and largest park acquired under the Project 70 Land Acquisition and Borrowing Act, featuring the river shown in the image, which is the busiest whitewater east of the Mississippi River?
Image:



Answer: Fayette County

Case for Medium Data

Question: What is the common name of the grass that was described by explorers during the expedition commissioned by the third President of the United States, who served from 1801 to 1809 and orchestrated the Louisiana Purchase, which included the Corps of Discovery led by Captain Meriwether Lewis and Second Lieutenant William Clark, as shown in the image?
Image:



Answer: needle-and-thread grass

Case for Hard Data

Question: In which year was the moth species, similar to the one described by the son of a British botanist and entomologist who was the first to describe the insect order Thysanura in 1840, and Mary, daughter of Edward Betts, in 1847, but with a deeper yellow underside and less conspicuous wing markings, described by the British entomologist who compiled the major work L̈epidoptera Indicaön butterflies from the region shown in the image, with many plates produced by his son, E. C. Knight, and John Nugent Fitch?
Image:



Answer: 1879

F Case Study

Case Trajectory in MM-SearchExam

Question: In what year did the population of the historic French port commune in Ille-et-Vilaine, Brittany, more than double with the merging of the town known for its corsairs who forced English ships to pay tribute and brought wealth from overseas adventures, with a former town and commune on the north coast of Brittany that merged with the commune where British troops captured and burned 30 privateers and a hundred other ships during the Seven Years' War in June 1758, to form the commune in the 8th year of the 1960s decade and is now a district known for its long sandy beach and sea spa, as shown in the image?
Image:



Step 1

<think>
I need to perform image search on the image in the question to obtain more information.
</think>
<tool_call>
{"name": "image_search", "arguments": {}}
</tool_call>
<tool_response>



1. [Page Link] https://gratefulreader.home.blog/2021/07/05/the-librarian-of-saint-malo-by-mario-escobar/ [Page Title] The Librarian of Saint-Malo by Mario Escobar - The Grateful Reader
</tool_response>

Step 2

<think>
I need to find information about the merger of Saint-Malo and Brest in 1964.
</think>
<tool_call>
{"name": "text_search", "arguments": {"query": "Saint-Malo merger Brest 1964"}}
</tool_call>
<tool_response>
1. [Link] https://en.wikipedia.org/wiki/Battle_of_Saint-Malo [Title] Battle of Saint-Malo - Wikipedia [Snippet] The Battle of Saint-Malo was fought between Allied and German forces for control of the French coastal town of Saint-Malo in Brittany during World War II.
2. [Link] http://channelislandsshipping.je/page32.html [Title] Channel Islands to France Services [Snippet] When British Railways finally withdrew their services to Saint Malo at the end of the season in 1964 the French decided to replace the services ...
3. [Link] https://www.history.navy.mil/research/library/online-reading-room/title-list-alphabetically/a/account-operations-american-navy-france-during-war-germany.html [Title]

Account of the Operations of the American Navy in France During ... [Snippet] At various times, naval port officers were assigned to certain ports—to Brest, Havre, Cherbourgh, Rouen, St. Malo, Granville, St. Nazaire, Nantes. Quiberon Bay.
4. [Link] https://bibliotheque.idbe.bzh/data/cle_119/Breton_News_1964_nA_44_.pdf [Title] [PDF] Breton News 1964 n° 44 [Snippet] Meetings were to be held at Gwengamp, Brest, Lorient, St-Brieuc, and St-Malo between Dec. 12 and 19. Their theme would be the future of the young Bretons is.
5. [Link] https://www.facebook.com/groups/DoverHistoryPages/posts/8240998352588815/ [Title] FALAISE 1947 UK TSS PACKET LATER IN 1964 REBUILD AS CAR ... [Snippet] ... Saint-Malo, France. The short cruises would be scheduled for both long weekends and midweek, and would include cruises along the Seine to ...
< /**tool_response**>

Step 3

<**think**>
I need to find information about the Saint-Malo Brest merger in 1964 and its relation to the population doubling.
< /**think**>
<**tool_call**>
{“name”: “text_search”, “arguments”: {“query”: “Saint-Malo Brest merger 1964 population double”}}
< /**tool_call**>
<**tool_response**>
1. [Link] https://bibliotheque.idbe.bzh/data/cle_119/Breton_News_1964_nA_44_.pdf [Title] [PDF] Breton News 1964 n° 44 [Snippet] Meetings were to be held at Gwengamp, Brest, Lorient, St-Brieuc, and St-Malo between Dec. ... increase its population from 82 millions to 12 ... It originated from ...
2. [Link] https://en.wikipedia.org/wiki/Saint-Malo [Title] Saint-Malo - Wikipedia [Snippet] The population of the commune more than doubled in 1967 with the merging of three communes: Saint-Malo, Saint-Servan (population 14,963 in 1962) and Paramé (...
3. [Link] https://theses.hal.science/tel-05101404v1/file/2024CHALLIS220341.pdf [Title] [PDF] Temporal tidal changes on the north-west European shelf [Snippet] Temporal tidal changes on the north-west European shelf. Summary : Potential drivers of temporal tidal changes are well identified, ...
4. [Link] https://www.jstor.org/content/pdf/oa_book_monograph/10.7591/j.ctt1vjqrc3.pdf [Title] A COLONIAL AFFAIR - jstor [Snippet] The definitive work on the St. Malo merchant association is André Lespag- nol, Messieurs de Saint-Malo: Une élite négociante au temps de Louis XIV (Saint Malo:.
5. [Link] https://dokumen.pub/the-corsairs-of-saint-malo-network-organization-of-a-merchant-elite-under-the-ancien-regime-9780231542661.html [Title] The Corsairs of Saint-Malo: Network Organization of a Merchant ... [Snippet] Henning Hillmann examines the merchant community of Saint-Malo, Brittany, a key port in the French Atlantic economy, to...
< /**tool_response**>

Step 4

<**think**>
I need to find information about the merger of Saint-Malo, Saint-Servan, and Paramé in 1967.
< /**think**>
<**tool_call**>
{“name”: “text_search”, “arguments”: {“query”: “Saint-Malo Saint-Servan Paramé merger 1967”}}
< /**tool_call**>
<**tool_response**>
1. [Link] https://en.wikipedia.org/wiki/Saint-Servan [Title] Saint-Servan [Snippet] The commune of Saint-Servan was merged with Paramé, into the commune of Saint-Malo in 1967. Originally, the area was known as Aleth, whose first bishop was the ...
2. [Link] https://www.ot-montsaintmichel.com/en/discover/extend-the-discovery-of-the-

bay/the-bay-brittany-side/saint-malo/ [Title] Saint-Malo - Mont Saint-Michel Normandy Destination [Snippet] Since 1967, with the merger of the three former communes of Saint-Servan, Paramé ... Malo, Greater Saint-Malo has become a city of nearly 50,000 inhabitants.
3. [Link] https://droneandslr.com/travel-blog/france/saint-malo-guide/ [Title] Discover Saint-Malo, France — A Charming, Coastal ... [Snippet] The Solidor Tower is located in the former town of Saint-Servan, which merged with Saint-Malo and Paramé in 1967. Perched at the estuary's ...
4. [Link] https://artsandculture.google.com/entity/param%C3%A9/m04czd53?hl=en [Title] Paramé — Google Arts & Culture [Snippet] Paramé merged with Saint-Servan to form the commune of Saint-Malo in 1967. Paramé is now a district of Saint-Malo and its seaside resort. The town is known ...
5. [Link] https://citypages.neocities.org/saintmalo [Title] Saint-Malo [Snippet] The commune of Saint-Servan was merged, together with Paramé, and became the commune of Saint-Malo in 1967. Saint-Malo was the site of an Anglo-French ...
< /**tool_response**>

Step 5

<**think**>
I need to search for information about the British troops' capture and burning of 30 privateers and a hundred other ships at Brest in June 1758.
< /**think**>
<**tool_call**>
{"name": "text_search", "arguments": {"query": "British troops captured burned 30 privateers and a hundred other ships June 1758 Brest"}}
< /**tool_call**>
<**tool_response**>
1. [Link] https://en.wikipedia.org/wiki/Raid_on_St_Malo [Title] Raid on St Malo - Wikipedia [Snippet] Instead the British occupied St Servan, a nearby port, and burned 30 privateers and 100 other vessels. Troops were sent on a reconnaissance eastwards towards ...
2. [Link] https://www.facebook.com/groups/thisdaythisbattle/posts/1143313949495109/ [Title] The french frigate L'Hermione defeats a British fleet at the Battle of ... [Snippet] In all, the battle involves 72 French guns against 128 British guns. On July 21, 1781, the two French ships attacked the British convoy that was ...
3. [Link] https://www.masshist.org/beehiveblog/2014/11/the-siege-of-louisbourg-1758-the-daring-capture-of-the-prudent-and-bienfaisant/ [Title] The Siege of Louisbourg, 1758: The Daring Capture of the Prudent ... [Snippet] In early June 1758, ground forces under the command of Jeffery Amherst landed outside the city and began a siege operation.
4. [Link] https://www.historicnavalfiction.com/general-hnf-info/this-day-in-history/june [Title] This Day in Naval History - June [Snippet] First day of The Four Days Battle, off the North Foreland between the English fleet of 56 ships, under George Monck, 1st Duke of Albemarle, and the Dutch fleet ...
5. [Link] https://www.jstor.org/stable/44224761 [Title] THE BRITISH COASTAL EXPEDITIONS TO FRANCE, 1757-1758 [Snippet] The route to St Servan was twisting and the British did not get possession until nightfall. In the basin they found thirty privateers and a hundred other.
< /**tool_response**>

Step 6

<**think**>
I need to find information about the merger of Saint-Malo, Saint-Servan, and Paramé in 1967, focusing on the population doubling aspect.
< /**think**>
<**tool_call**>
{"name": "text_search", "arguments": {"query": "Saint-Malo merger 1967 population doubled Saint-Servan Paramé"}}
< /**tool_call**>
<**tool_response**>
1. [Link] https://en.wikipedia.org/wiki/Saint-Malo [Title] Saint-Malo - Wikipedia [Snippet]

The population of the commune more than doubled in 1967 with the merging of three communes: Saint-Malo, Saint-Servan (population 14,963 in 1962) and Paramé (...
2. [Link] https://citypages.neocities.org/saintmalo [Title] Saint-Malo [Snippet] The population of the commune more than doubled in 1968 with the merging of three communes: Saint-Malo, Saint-Servan (population 14,963 in 1962) and Paramé (...
3. [Link] https://kids.kiddle.co/Saint-Malo [Title] Saint-Malo Facts for Kids [Snippet] The city's population more than doubled in 1967. This happened when Saint-Malo joined with two other towns: Saint-Servan and Paramé. People who live in Saint- ...
4. [Link] https://alchetron.com/Saint-Malo [Title] Saint Malo - Alchetron, The Free Social Encyclopedia [Snippet] The population of the commune more than doubled in 1968 with the merging of three communes: Saint-Malo, Saint-Servan (population 14,963 in 1962) ...
5. [Link] https://en.wikipedia.org/wiki/Saint-Servan [Title] Saint-Servan - Wikipedia [Snippet] The commune of Saint-Servan was merged with Paramé, into the commune of Saint-Malo in 1967. Originally, the area was known as Aleth, whose first bishop was the ...
< /**tool_response**>

Step 7

<**think**>
I have gathered enough information. Now I am confident to answer the question.
< /**think**>
<**answer**>1967< /**answer**> ✓

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly elaborate our contributions and main scope in the abstract and introduction, see Lines 7–10 and 78–91.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of this work in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This work does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all details needed to reproduce our experimental results in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We currently do not provide the URL to the implementation code in this paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the core settings in the main text, see Section 3.1. We also provide more comprehensive experimental settings in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not conduct repeated experiments to obtain error bars, as the API cost of repeated evaluations is relatively high.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information about the GPUs used in Section 3.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We strictly followed the NeurIPS Code of Ethics throughout our research process.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper presents a multimodal search agent, which we think does not have malicious effects on any disadvantaged social group.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not pose a high risk of misuse. Therefore, we do not present specific safeguards for this work.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We explicitly cite all utilized benchmarks in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: In this paper, we release a new benchmark for evaluating multimodal search agents. We provide details on the benchmark curation process, but do not currently provide full documentation for the benchmark.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not use crowdsourcing in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not use crowdsourcing in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We only use LLM to revise the English writing of this paper.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.