Towards Long-horizon Agentic Multimodal Search

Anonymous ACL submission

Abstract

Multimodal deep search agents can solve complex tasks by iteratively gathering textual and visual evidence, but long-horizon reasoning is hindered by context explosion and high multimodal token costs. To address this, we propose **LMM-Searcher**, a long-horizon multimodal deep search framework built on a file-based visual representation mechanism. By storing visual assets externally and referencing them with lightweight textual identifiers (UIDs), our approach reduces context overhead while preserving multimodal information for future retrieval. We further introduce a data synthesis pipeline for generating complex cross-modal multi-hop reasoning data. Using 12K synthesized trajectories, we fine-tune Qwen3-VL-Thinking-30A3B into a specialized multimodal deep search agent. Experiments on four benchmarks show that LMM-Searcher scales to 100-turn search horizons and achieves state-of-the-art performance among open-source models on challenging benchmarks such as MM-BrowseComp and MMSearch-Plus, while maintaining strong cross-model generalization. Code, data, and models will be released.

1 Introduction

Deep search agents (Jin et al., 2025; Song et al., 2025; Li et al., 2025b) have achieved strong performance on challenging real-world information-seeking tasks (Wei et al., 2025; Mialon et al., 2023). Unlike traditional search systems, they iteratively reason, retrieve evidence, and accumulate knowledge over long horizons to solve complex problems. Recent work further extends this paradigm to multimodal search agents (Geng et al., 2025; Huang et al., 2026; Chu et al., 2026) by incorporating visual tools such as image search.

However, multimodal deep search (Mei et al., 2025a) differs fundamentally from text-only search. The retrieved information is heterogeneous (Yin et al., 2024) and leads to severe context explosion due to the high token cost of multimodal inputs such as images and videos (Yao et al., 2026; Wen et al., 2025), especially in long-horizon tasks with many interactions. Existing context management methods, which mainly condense textual histories, do not generalize well to multimodal search (Mei et al., 2025b; Wu et al., 2025b; Chen et al., 2025a), since multimodal data have fundamentally different representations (Liang et al., 2022; Li and Tang, 2024; Shu et al., 2025). In practice, heuristic methods (Hong et al., 2025) often discard intermediate visual information, risking the loss of critical signals and limiting scalability. This raises a central question: *How can we effectively manage accumulated multimodal contexts during deep search?*

Inspired by recent progress in the planning-with-files paradigm (Othman, 2024; Merrill et al., 2026), we offload multimodal information from the context into external files that can be selectively retrieved, analyzed, or manipulated during reasoning. This preserves full multimodal content for future use while reducing context overhead via on-demand loading. To implement this idea, we propose a long-horizon multimodal context management framework, **LMM-Searcher**, which represents all visual assets—retrieved or generated—as external files indexed by unique textual identifiers (UIDs), optionally enriched with compact thumbnail-based semantics. These textual proxies enable efficient long-term tracking of multimodal information. To support fine-grained access, we redesign multimodal search tools and introduce a dedicated *fetch-image* operator, enabling a progressive search workflow that loads visual content only when needed.

Furthermore, to enhance the agent's ability in long-horizon multimodal search, we develop a data synthesis pipeline that constructs queries requiring complex cross-modal multi-hop reasoning. Combined with open-source deep search queries, these synthesized tasks are used to collect high-quality

trajectories from a strong teacher model for agentic training. Based on this training data, we fine-tune Qwen3-VL-Thinking-30A3B into a specialized multimodal deep search agent.

To validate our approach, we conduct extensive experiments on four multimodal search benchmarks. On challenging long-horizon benchmarks, MM-BrowseComp (MMBC) (Li et al., 2025a) and MMSearch-Plus (Tao et al., 2025), our method achieves success rates of 22.3 and 32.9, respectively. Equipped with the context management strategy, our model can scale to 100 turns and achieve performances of 30.1 and 34.8, establishing state-of-the-art results among open-source models. Besides the trained models, our approach also demonstrates superiority in enhancing the model's agentic search capabilities. Specifically, when applied to the same base models, our framework significantly outperforms the prior framework (Chu et al., 2026). Based on Seed-1.8, we achieve 46.7 on MMSearch-Plus, demonstrating the strong generalizability of our approach.

Our contributions are summarized below:

- **Long-horizon multimodal deep search framework:** We propose a novel framework based on file-based visual representation and a specialized agentic tool interface. By offloading visual assets to an external file system and fetching them on demand, our method mitigates the problem of context explosion and scales effectively.
- **Data synthesis pipeline for long-horizon search:** We develop a pipeline for complex cross-modal reasoning, producing queries that require more interaction turns and heavier vision-related tool use than existing datasets.
- **Long-horizon multimodal deep search agent:** Using the proposed framework and synthesized data, we distill 12K high-quality trajectories to fine-tune Qwen3-VL-30A3B-Thinking into a multimodal deep search agent that scales to 100 turns and achieves SOTA performance among open-source models across four benchmarks.

2 Related Work

Language-based Deep Search Agent. Language-based deep search agents aim to tackle the bottleneck of knowledge boundaries for large language models (LLMs) (Brown et al., 2020; Zhao et al., 2023) by introducing external search and retrieval mechanisms (Lewis et al., 2020; Gao et al., 2023). Early research typically adopts the retrieval-augmented generation (RAG) paradigm (Guu et al., 2020; Asai et al., 2023) to achieve knowledge enhancement by retrieving relevant documents from static databases (Karpukhin et al., 2020; Reimers and Gurevych, 2019). Subsequent work overcomes the limits of pre-built databases by integrating search tools (Li et al., 2025b; Sun et al., 2025), granting models direct internet search capability and improving open-domain question answering (Wei et al., 2025; Chen et al., 2025c). However, the language-based agents only support textual search inputs and feedback, limiting their capacity to resolve multimodal queries in real-world applications.

Multimodal Deep Search Agent. Similar to LLMs, MLLMs (Liu et al., 2023) also require external tools to handle real-world tasks. Early research (Wu et al., 2023) equips models with extensive visual plugins, including plugins for object detection (Zou et al., 2023) and image segmentation (Long et al., 2015). This enables MLLMs to autonomously invoke tools based on complex instructions. Building on this, recent work internalizes such interactive capabilities into the model's reasoning process, leading to the thinking-with-image paradigm. (OpenAI, 2025). Such frameworks treat visual operations as explicit reasoning steps, facilitating significant gains in spatial reasoning and fine-grained VQA. Building upon these advancements, recent work (Geng et al., 2025; Hong et al., 2025) integrates search tools into the MLLM's reasoning process. By combining internal visual reasoning with external search tools, models are empowered to perform complex fact-checking and open-domain multimodal exploration.

3 Multimodal Context Management

We first propose a context management mechanism that combines file-based multimodal representation with an extended agentic tool interface. Our core motivation is to decouple perception from reasoning. Specifically, instead of directly inserting multimodal contents into the model context—which risks token explosion and noise accumulation across multi-turn interactions, we store visual assets externally and reference them via textual identifiers (UIDs). Based on this design, we equip the agent with specialized tools that actively retrieve and process relevant visual content on demand. This design enables long-horizon multimodal search, preserving fine-grained perception

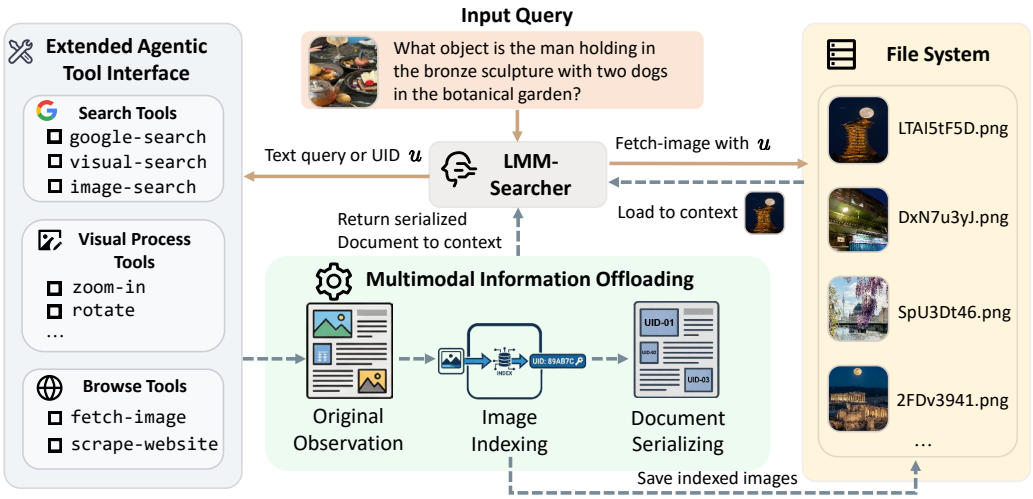


Figure 1: An illustration of LMM-Searcher. For simplicity, we employ simple strings as uids in this figure. In real implementation, we use URLs.

while avoiding excessive context consumption.

3.1 File-based Multimodal Data Management

Throughout the multimodal search process, each image returned by the environment is persistently stored in an external file system. To ensure the agent can reliably locate and load target images via UIDs on demand, it's essential to establish a strict one-to-one mapping between UIDs and images in the file system. Formally, let $\mathcal{I}$ denote the visual space and $\mathcal{U}$ denote the texutal UID space. The file system defines a persistent mapping function $f : \mathcal{I} \rightarrow \mathcal{U}$, such that each visual asset $i \in \mathcal{I}$ is uniquely associated with a UID $u = f(i)$.

Through this proxy representation u, visual content is converted into a lightweight UID that can be efficiently maintained in context. When fine-grained visual perception is required, the agent can actively retrieve the corresponding image via UID. To reduce storage overhead, if an image already exists in an external file system (*e.g.*, internet) with a valid identifier (*e.g.*, a URL), we directly reuse the existing UID rather than assigning a new one.

3.2 Extended Agentic Tool Interface

Previous multimodal search frameworks (Chu et al., 2026; Huang et al., 2026) are designed to load images into the model context immediately upon retrieval, leading to rapid context explosion. To enable long-horizon multimodal search, we redesign conventional search-related tools to operate over UID-based visual references. The designed interface includes three categories of tools: *search tools* for internet searches, *browse tools* for handling web page extraction and visual perception, and *visual processing tools* for the perception and editing of extracted images. We present a more detailed description below:

Search Tools. We integrate open-domain search tools built upon existing search engines (*i.e.*, Serper) for multimodal search. This suite of tools includes `google_search`, which accepts textual queries and retrieves relevant documents; `image_search`, which takes textual input and returns related images; and `visual_search`, which accepts visual queries and returns visually similar results. These tools return a set of retrieved items, including textual snippets, image links, thumbnails, and corresponding webpage URLs.

Browse Tools. We introduce two tools for accessing detailed content from web pages and images. The first, `scrape_website`, employs a summarization model to summarize webpage contents while concurrently extracting and storing all image URLs based on a given query. The second, `fetch_image`, is designed for active visual perception. Acting as a bridge between the UID space $\mathcal{U}$ and the visual space $\mathcal{I}$, this tool retrieves the image i corresponding to a given UID u from the external file system and provides it to the model for detailed inspection.

Visual Processing Tools. To support fine-grained visual reasoning, we incorporate an image processing tool (*i.e.*, `zoom_in`). Let g denote a visual transformation. Given an input UID u_{in}, the tool applies

the transformation to the underlying image, producing a new visual asset $i_{\text{new}} = g(f^{-1}(u_{\text{in}}))$. The resulting image is then uploaded to the file system and assigned a new identifier $u_{\text{new}} = f(i_{\text{new}})$. Because such operations involve active perception and generate a focused visual result, both i_{new} and its corresponding UID u_{new} are inserted into the context simultaneously.

3.3 Long-horizon Search Workflow

Building upon the aforementioned designs, the agent executes a long-horizon search workflow that simulates human information acquisition by memorizing spatial pointers rather than high-resolution visual details. Specifically, our middleware intercepts interleaved text-image documents from search or browse tools, offloading raw images to the file system and substituting them with UIDs. This lightweight proxy resolves context explosion, effectively decoupling long-horizon reasoning from intensive visual perception. When fine-grained analysis is required, the agent autonomously invokes the `fetch_image` function and subsequent visual processing tools via the UID to achieve strictly on-demand perception. Crucially, unlike heuristic methods that aggressively discard visual data (Hong et al., 2025; Chu et al., 2026), this UID-based framework guarantees zero information loss. The UID serves as a persistent, low-cost semantic pointer, ensuring the agent can reliably trace back to the uncompressed original visual evidence at any stage of the reasoning chain.

4 Agentic Training for Multimodal Search

To enhance long-horizon multimodal search based on our context management design, we introduce a comprehensive agentic training pipeline that encompasses query synthesis, trajectory data distillation, and model training. A major bottleneck in prior data synthesis efforts (Chu et al., 2026) is the scarcity of high-quality queries. Prior datasets (Chu et al., 2026) typically restrict multimodal inputs to the initial search stage, inherently limiting task complexity and subsequent reasoning. To address this limitation, we synthesize complex queries that demand continuous cross-webpage multimodal comprehension throughout the entire search process. Specifically, our pipeline proceeds through webpage VQA synthesis (Section 4.1), reasoning chain extension (Section 4.2), and agent trajectory distillation (Section 4.3) to generate final data for model training (Section 4.4).

4.1 Multimodal Webpage Query Synthesis

To ensure that the synthesized queries strictly require the model to read multimodal webpage information, we select visually-rich multimodal websites (*e.g.*, news and movie websites) as a starting point to synthesize single-hop visual questions. This process involves webpage content extraction and question synthesis.

Webpage Content Extraction. Given a multimodal webpage W, we parse it using Jina[1] and input it into an MLLM to extract a unique, unambiguous core entity E from the webpage, along with a context-rich image I_E related to E. Detailed prompts can be found in the Appendix.

Visual Question Synthesis. We prompt a MLLM to synthesize a text clue linking E and I_E based on W, while generating a visual question for I_E that cannot be answered by W's text alone. Merging these two components yields a single-hop question q_0 that requires the search agent to navigate to W and continuously verify multimodal information against the given clues to answer it.

4.2 Multi-hop Reasoning Chain Extension

To increase the difficulty of the previously obtained queries, we extend the reasoning chain based on E. The basic idea is to use E as a starting point to construct a multi-hop knowledge graph denoted as G, and then fuzzify the key nodes in G to ultimately generate an extended multi-hop visual question q. Specifically, this is divided into the following steps.

Multi-hop Graph Construction. We design a workflow to construct a multi-hop knowledge graph iteratively. We iteratively expand a directed knowledge graph $G = (\mathcal{V}, \mathcal{E})$, initialized with a single unexpanded root node $G^{(0)} = (\{v_{\text{root}}\}, \emptyset)$ where $v_{\text{root}} = E$. At iteration t, we select an unexpanded node $v_t \in \mathcal{V}^{(t-1)}$ (out-degree of 0) representing entity E_t. Based on its search results from knowledge sources (offline databases or internet webpages), an LLM extracts an attribute set $\mathcal{R}_t$, and filters it into a subset $\hat{\mathcal{R}}_t$ conditioned on graph density and node depth. Each attribute in $\hat{\mathcal{R}}_t$ links to a new target entity u_t^i. Finally, these new nodes and their directed links are appended to update the graph

[1] https://jina.ai

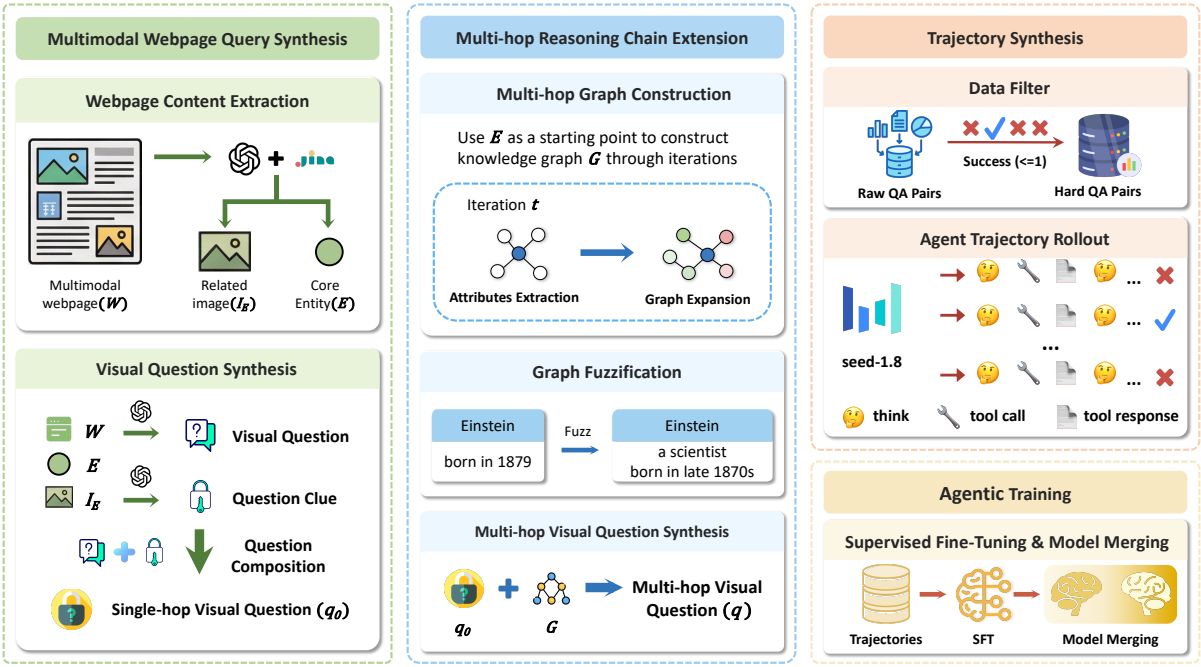


Figure 2: Overview of automated Visual Question-Answer (VQA) data synthesis pipeline. The pipeline constructs multimodal deep search data by synthesizing VQA pairs based on multimodal webpages and subsequently extending the reasoning chain.

to $G^{(t)}$. The graph expansion process at the t-th iteration can be represented as follows:

$$\begin{aligned} \mathcal{V}^{(t)} &= \mathcal{V}^{(t-1)} \cup \{u_t^i \mid r_t^i \in \hat{\mathcal{R}}_t\} \\ \mathcal{E}^{(t)} &= \mathcal{E}^{(t-1)} \cup \{(v_t, r_t^i, u_t^i) \mid r_t^i \in \hat{\mathcal{R}}_t\} \end{aligned} \tag{1}$$

Upon updating the graph, v_t is marked as expanded. To enhance reasoning complexity, we impose an obfuscation constraint: no single attribute $r_t^i \in \hat{\mathcal{R}}_t$ can independently deduce the source entity v_t in reverse. This irreversibility ensures that each newly added edge is indispensable for the final multi-hop reasoning, preventing shortcuts during answer retrieval.

Graph Fuzzification. After completing the graph construction, we select entities of leaf nodes and nodes with low in-degrees/out-degrees in the current knowledge graph G, and group them into a set of entities to be fuzzified, denoted as $\{F_k\}_{k=1}^{n}$. These entities lack the necessary edges for reference and thus must be represented in a fuzzified manner. We randomly select an attribute r_k^j of the target entity F_k that are not used during graph construction and prompt an LLM to fuzzify the node.

Multi-hop Visual Question Synthesis. We sample the constructed graph to obtain a subgraph G' containing the core entity E, where a randomly selected leaf node E_i is replaced with its corresponding image. An LLM then converts G' into a natural language reasoning text that concludes with E. Finally, the LLM integrates this text into the previously synthesized single-hop question q_0, yielding the final multi-hop visual question q.

4.3 Trajectory Synthesis

To construct training trajectories, we combine our synthesized queries with open-source datasets, including FVQA (Wang et al., 2017), LiveVQA (Fu et al., 2025), and REDSearcher (Chu et al., 2026). For quality control (Du et al., 2025; Liu et al., 2025b), we use Qwen2.5-VL-7B (Bai et al., 2025) to filter queries answerable without search. We then generate trajectories via rejection sampling with Seed-1.8 (Seed, 2026), keeping only those that successfully answer within 40 turns and a 64K context limit. This yields 12,736 high-quality samples. Compared to existing datasets, our pipeline produces significantly longer trajectories, highlighting stronger long-horizon search characteristics (See details in Table 5 in the Appendix).

4.4 Model Training

To validate our framework and synthesized dataset, we perform multi-turn supervised fine-tuning on Qwen3-VL-30B-A3B-Thinking (Bai et al., 2025),

Model	MMBC	MMSearch+	VisBrowse	MMSearch	Avg.
Direct Answer					
GPT-5	10.3	**19.1**	26.0	33.3	22.2
Seed-1.8	**13.0**	8.6	18.9	31.0	17.9
Kimi-K2.5	2.7	12.6	18.3	**47.0**	20.2
Gemini-2.5-Pro	10.3	14.5	**27.2**	39.8	**23.0**
Gemini-2.5-Flash	5.4	8.1	16.0	30.4	15.0
Qwen3-VL-30B-A3B-Thinking	7.1	2.7	13.0	17.7	10.1
Agent Workflow					
GPT-5	23.7	34.8	35.5	72.2	41.6
Seed-1.8	25.5	**46.7**	**58.0**	**73.2**	**50.9**
Kimi-K2.5	**25.9**	39.2	50.3	72.3	46.9
Gemini-2.5-Pro	12.1	28.1	16.0	66.3	30.6
Gemini-2.5-Flash	8.0	14.0	14.2	59.8	24.0
Qwen3-VL-30B-A3B-Thinking	9.8	14.4	16.0	62.0	25.6
Multimodal Search Agent					
MMSearch-R1-7B	-	-	-	53.8	-
Webwatcher-7B	-	-	-	49.1	-
Webwatcher-32B	-	-	-	55.3	-
DeepEyesV2-7B	-		-	63.7	-
Vision-DeepResearch-30B	-	28.5	-	69.6	-
REDSearcher-MM-30B	23.5	26.6	-	**72.9**	-
LMM-Searcher-30B	22.3/**30.1***	32.9/**34.8***	42.0/**48.3***	71.0/72.3*	42.1/**46.4***

Table 1: The performance comparison between our model and baseline methods. The performance marked with * is evaluated with 100 turns and the context management technique. MMBC shorts for MM-BrowseComp, and MMSearch+ shorts for MMSearch-Plus.

masking tool responses in the cross-entropy loss to optimize only reasoning and tool calls. Despite high data quality, multimodal search trajectories inherently lag behind pure-text ones in interaction scale, bottlenecking the model's scaling capabilities. Inspired by previous studies on cross-modal capability transfer (Chen et al., 2025b), we address this by merging our model with MiroThinker-1.7-mini (Team et al., 2026), a text model sharing the same backbone that exhibits strong deep-search capabilities from large-scale mid-training. The merging is applied only to shared language components. Denoting the SFT multimodal model as Θ_V and MiroThinker-1.7-mini as Θ_T, the final model is obtained by parameter interpolation:

$$\Theta_{\text{final}} = \alpha \cdot \Theta_V + (1 - \alpha) \cdot \Theta_T \tag{2}$$

We set $\alpha = 0.8$ to preserve multimodal capabilities while incorporating stronger search abilities. A more detailed study of model merging is left for future work.

5 Experiment

5.1 Experiment Setup

We evaluate our model on various challenging visual search benchmarks. The evaluation benchmarks include: MM-BrowseComp (Li et al., 2025a), VisBrowse-Bench (Zhang et al., 2026), MMSearch-Plus (Tao et al., 2025), and MMSearch (Jiang et al., 2024). For the baseline methods, we consider three categories: direct answer, agent workflow, and multimodal search agents. The baseline methods and implementation details are in the Appendix.

5.2 Main Results

5.2.1 Overall Performance

Table 1 reports the overall performance comparison across four benchmarks. Direct answer methods consistently underperform agent-based approaches, highlighting the necessity of tool use and external search for complex multimodal tasks. Moreover, our LMM-Searcher-30B achieves competitive or superior performance compared with existing multimodal search agents, reaching 28.7 on the challenging MMSearch-Plus benchmark while maintaining strong performance on MMBC and MMSearch. Furthermore, scaling to 100 turns with context management further improves performance across all benchmarks, achieving SOTA results on MMBC and MMSearch-Plus. Overall, these results demonstrate the effectiveness of our approach in combining strong search capability, multimodal

Model	Evaluation Method	MMBC	MMSearch+	VisBrowse	MMSearch
GPT-5	Direct Answer	10.3	19.1	26.0	33.3
	w. Previous Framework	-	17.2	-	63.7
	w. Our Framework	**36.5**	**34.8**	**35.5**	**72.2**
Seed-1.8	Direct Answer	13.0	8.6	18.9	31.0
	w. Previous Framework	21.4	11.0	-	69.7
	w. Our Framework	**35.1**	**46.7**	**58.0**	**73.2**
Qwen3-VL-30B-A3B-Thinking	Direct Answer	7.1	2.7	13.0	17.7
	w. Previous Framework	**10.7**	13.6	-	53.2
	w. Our Framework	9.8	**14.4**	**16.0**	**62.0**

Table 2: Performance comparison among different frameworks.

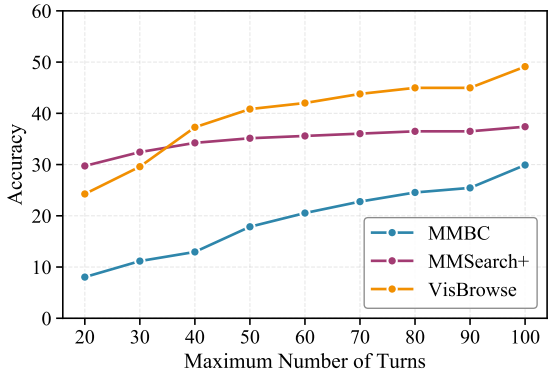
reasoning, and scalable long-horizon interaction.

5.2.2 Comparison with Other Frameworks

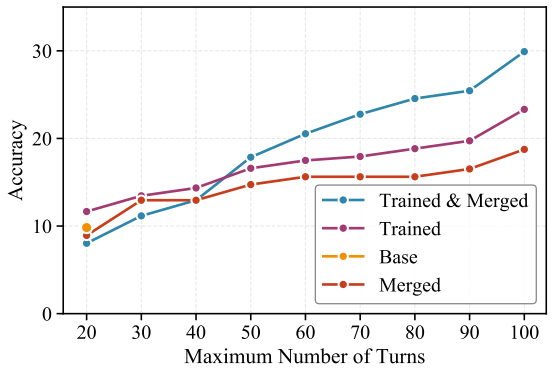
To evaluate the generalization of our context management design, we deploy identical models under our framework and prior frameworks, including REDSearcher (Chu et al., 2026) and Vision-DeepResearch (Huang et al., 2026), with a maximum of 50 interaction turns for fair comparison. As shown in Table 2, our framework consistently improves average performance across all models. Gains are smaller for weaker visual agents (*e.g.*, Qwen3-VL-30B-A3B-Thinking), suggesting that fixed search-and-look workflows are sufficient for simpler agents, while stronger models benefit substantially more. For instance, Seed-1.8 improves by 13.7% on MMBC and 35.7% on MMSearch-Plus. The largest gains occur on challenging tasks, including a 17.6% improvement for GPT-5 on MMSearch-Plus. These results highlight the effectiveness of our framework for complex visual multi-hop reasoning.

5.3 Further Analysis

Tool Distribution. To demonstrate that our synthesized queries inherently require more intensive multimodal search and browsing, we analyze the distribution of tool calls in agent trajectories and compare it with those induced by existing open-source multimodal search queries. As shown in Figure 5, our queries trigger substantially more "visual search" and "image search". More importantly, they require significantly more frequent "fetch image" steps during problem solving. This indicates that our synthesized queries demand deeper inspection of multimodal content on web pages, rather than relying on superficial retrieval alone.



(a) Scaling results on different benchmarks.



(b) Scaling results on model training and merging.

Figure 3: Interactive scaling results under different interaction-turn thresholds. A sample is counted as successful if the agent autonomously terminates and produces the correct answer within the specified number of turns.

Interaction Scaling. To evaluate the interactive scaling behavior of LMM-Searcher, we impose varying turn limits, truncating interactions that exceed the threshold and marking them as incomplete. Our evaluation focuses on two aspects: (1) the scaling performance of LMM-Searcher-30B across diverse benchmarks, and (2) the comparison of different model variants (*i.e.*, base, trained, merged,

	MMBC	MMSearch+	VisBrowse	MMSearch	Avg.
Qwen3-VL-30B-A3B-Thinking	9.8	14.4	16.0	62.0	25.6
+ Open-source Visual Query	20.7	**33.8**	39.5	70.4	41.1
+ Open-source Textual Query	21.6	32.4	39.1	70.3	40.9
+ Our Synthesized Query	**22.3**	32.9	**42.0**	**71.0**	**42.1**

Table 3: Data ablation results. Each dataset is incrementally added on top of the previous training set.

	MMBC	MMSearch+	VisBrowse	MMSearch	Avg.
Seed-1.8	9.8	14.4	16.0	62.0	25.6
w/ fetch-image tool	**35.1**	**46.7**	**58.0**	**73.2**	**53.3**
wo/ fetch-image tool	29.5	43.7	48.5	71.0	48.2
Δ	-5.6	-3.0	-9.5	-2.2	-5.1

Table 4: Tool ablation results. "w/ fetch-image" represents equipping the agent with a full tool interface, while "wo/ fetch-image tool" represents removing the fetch-image tool.

and final) on MMBC. The results in Figure 3 show that our model consistently benefits from increased interaction turns across all benchmarks, although the magnitude of improvement varies, suggesting different tasks require different reasoning depths. Notably, MMBC and VisBrowse continue to improve even at 100 turns, indicating strong scaling potential for multimodal search. Furthermore, Figure 3b shows that while the base model saturates around 20 turns, both synthetic data training and language model merging substantially enhance its scaling behavior, with their combination yielding further gains. These results demonstrate that our approach effectively transfers long-horizon capabilities from language-based search to multimodal settings.

Data Ablation. To evaluate our data synthesis pipeline, we conduct ablations using training data from different sources and modalities, including open-source multimodal search queries (FVQA, LiveVQA, and REDSearcher-MM), open-source textual queries (REDSearcher-Text), and our synthesized multimodal queries. Each dataset is added incrementally, and the resulting checkpoint is merged with MiroThinker-1.7-mini. As shown in Table 3, open-source multimodal queries already substantially improve agentic search performance. Adding textual queries mainly benefits the long-horizon benchmark MMBC but slightly degrades average performance, likely because language-based search capabilities are already inherited through model merging. Further incorporating our synthesized queries yields additional gains on MMBC and VisBrowse, achieving the best overall performance across benchmarks.

Tool Ablation. A key difference between our approach and prior multimodal deep search frameworks is that all multimodal information encountered during search is saved as files for flexible loading, enabled primarily by the "fetch-image" tool. To evaluate its effectiveness, we compare Seed-1.8 with and without this tool in our framework. As shown in Table 4, removing "fetch-image" degrades performance across all benchmarks. The largest drop occurs on VisBrowse (58.0 $\rightarrow$ 48.5), indicating strong reliance on webpage image retrieval, while MMSearch only decreases from 73.2 to 71.0, suggesting greater dependence on search engine results rather than webpage browsing.

6 Conclusion

In this work, we present LMM-Searcher, an open-source multimodal deep search agent for complex multimodal queries. Our framework combines file-based visual representation with agentic visual tools to support efficient multimodal context management and long-horizon interaction. We further develop a dedicated data synthesis pipeline, including multimodal query synthesis and trajectory rollout, to construct high-quality training data for SFT. Evaluations on four multimodal deep search benchmarks show that LMM-Searcher-30B achieves strong performance among open-source search agents, validating the effectiveness of our framework and data synthesis approach for multimodal deep search.

Limitations

While our proposed LMM-Searcher framework and data synthesis pipeline demonstrate significant advancements in long-horizon multimodal search, there remain a few notable limitations. First, due to computational resource constraints, the current model is optimized solely through supervised fine-tuning and lacks the advanced exploration, decision-making, and self-correction enhancements that reinforcement learning could provide. Second, our automated data synthesis pipeline currently focuses exclusively on image-related queries, omitting other modalities such as video and audio streams that are prevalent in real-world search scenarios. In future work, we plan to address these constraints by improving the model through reinforcement learning, as well as expanding our data synthesis and context management frameworks to seamlessly support and evaluate diverse modalities including video and audio.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Guoxin Chen, Zile Qiao, Xuanzhong Chen, Donglei Yu, Haotian Xu, Wayne Xin Zhao, Ruihua Song, Wenbiao Yin, Huifeng Yin, Liwen Zhang, and 1 others. 2025a. Iterresearch: Rethinking long-horizon agents via markovian state reconstruction. *arXiv e-prints*, pages arXiv–2511.

Shiqi Chen, Jinghan Zhang, Tongyao Zhu, Wei Liu, Siyang Gao, Miao Xiong, Manling Li, and Junxian He. 2025b. Bring reason to vision: Understanding perception and reasoning through model merging. *arXiv preprint arXiv:2505.05464*.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, and 1 others. 2025c. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, Shengchao Hu, and 1 others. 2026. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*.

Yifan Du, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, Jinpeng Wang, Chuyuan Wang, Mingchen Cai, Ruihua Song, and Ji-Rong Wen. 2025. What makes for good visual instructions? synthesizing complex visual reasoning instructions for visual instruction tuning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8197–8214.

Mingyang Fu, Yuyang Peng, Benlin Liu, Yao Wan, and Dongping Chen. 2025. Livevqa: Live visual knowledge seeking. *arXiv e-prints*, pages arXiv–2504.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, Haofen Wang, and 1 others. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1):32.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, and 1 others. 2025. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

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

Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, and 1 others. 2026. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, and 1 others. 2024. Mmsearch: Benchmarking the potential of large models as multi-modal search engines. *arXiv preprint arXiv:2409.12959*.

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 6769–6781.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, and 1 others. 2025a. Mm-browsecomp: A comprehensive benchmark for multimodal browsing agents. *arXiv preprint arXiv:2508.13186*.

Songtao Li and Hao Tang. 2024. Multimodal alignment and fusion: A survey. *arXiv preprint arXiv:2411.17040*.

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

Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. 2022. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems*, 35:17612–17625.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025a. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Zikang Liu, Kun Zhou, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji-Rong Wen. 2025b. Less is more: High-value data selection for visual instruction tuning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 3712–3721.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Lang Mei, Siyu Mo, Zhihan Yang, and Chong Chen. 2025a. A survey of multimodal retrieval-augmented generation. *arXiv preprint arXiv:2504.08748*.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, and 1 others. 2025b. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334*.

Mike A Merrill, Alexander G Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E Kelly Buchanan, and 1 others. 2026. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. *arXiv preprint arXiv:2601.11868*.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*.

OpenAI. 2025. Thinking with images. https://openai.com/index/thinking-with-images/.

Adi Othman. 2024. planning-with-files. https://github.com/othmanadi/planning-with-files. GitHub repository.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.

Bytedance Seed. 2026. Seed1.8 model card: Towards generalized real-world agency. *arXiv preprint arXiv:2603.20633*.

Dong Shu, Haiyan Zhao, Jingyu Hu, Weiru Liu, Ali Payani, Lu Cheng, and Mengnan Du. 2025. Large vision-language model alignment and misalignment: A survey through the lens of explainability. *arXiv preprint arXiv:2501.01346*.

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

Shiqian Su, Sen Xing, Xuan Dong, Muyan Zhong, Bin Wang, Xizhou Zhu, Yuntao Chen, Wenhai Wang, Yue Deng, Pengxiang Zhu, and 1 others. 2026. Miroflow: Towards high-performance and robust open-source agent framework for general deep research tasks. *arXiv preprint arXiv:2602.22808*.

Shuang Sun, Huatong Song, Yuhao Wang, Ruiyang Ren, Jinhao Jiang, Junjie Zhang, Fei Bai, Jia Deng, Wayne Xin Zhao, Zheng Liu, and 1 others. 2025. Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis. *arXiv preprint arXiv:2505.16834*.

Xijia Tao, Yihua Teng, Xinxing Su, Xinyu Fu, Jihao Wu, Chaofan Tao, Ziru Liu, Haoli Bai, Rui Liu, and Lingpeng Kong. 2025. Mmsearch-plus: Benchmarking

provenance-aware search for multimodal browsing agents. *arXiv preprint arXiv:2508.21475*.

MiroMind Team, S Bai, L Bing, L Lei, R Li, X Li, X Lin, E Min, L Su, B Wang, and 1 others. 2026. Mirothinker-1.7 & h1: Towards heavy-duty research agents via verification. *arXiv preprint arXiv:2603.15726*.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. 2017. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2413–2427.

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

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. 2025. Token pruning in multimodal large language models: Are we solving the right problem? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15537–15549.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.

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

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, and 1 others. 2025b. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*.

Linli Yao, Long Xing, Yang Shi, Sida Li, Yuanxin Liu, Yuhao Dong, Yi-Fan Zhang, Lei Li, Qingxiu Dong, Xiaoyi Dong, and 1 others. 2026. Towards efficient multimodal large language models: A survey on token compression.

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

Zhengbo Zhang, Jinbo Su, Zhaowen Zhou, Changtao Miao, Yuhan Hong, Qimeng Wu, Yumeng Liu, Feier Wu, Yihe Tian, Yuhao Liang, and 1 others. 2026. Visbrowse-bench: Benchmarking visual-native search for multimodal browsing agents. *arXiv preprint arXiv:2603.16289*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2):1–124.

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

Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. 2023. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276.

A Case Study

To concretely demonstrate the workflow of our framework, we select a case from VisBrowse-Bench and present some detailed step-level illustrations in Figure 6, and its complete reasoning trajectory in Table 6. As we can observe, our model exhibits the following capabilities: (1) Visual agentic tool use: The model can autonomously select tools to facilitate enhanced perception. In Iteration 3, it successfully delineates the precise zoom-in region, aiding in the identification of the correct brand. (2) Alternating reasoning and perception: From Iteration 5 to Iteration 13, through continuous reasoning, the model alternately employs text search and visual search tools to locate relevant images. In Iteration 14, the model proactively loads the intermediate search images into the context for perception, successfully deriving the correct answer. (3) Reflection: In Iterations 3 and 12, drawing upon previous failed tool calls, the model reflects to formulate new reasoning paths, thereby advancing the search process. These capabilities effectively enhance the model's cross-modal reasoning performance, enabling it to scale search turns within a limited context.

B Experimental Setup

Baselines. We consider three categories of baseline methods: direct answer, agent workflow, and multimodal search agents.

- Direct Answer. The model generates responses solely based on its parametric knowledge, without performing any image manipulation or external search.
- Agent Workflow. The model is integrated into our agent framework, where it can invoke a suite of tools to assist in answering queries.
- Multimodal Search Agents. We compare against existing open-source multimodal agents, including MMSearch-R1 (Wu et al., 2025a), WebWatcher (Geng et al., 2025), DeepEyesV2 (Hong et al., 2025), Vision-DeepResearch (Huang et al., 2026), and REDSearcher-MM (Chu et al., 2026). These methods typically combine perception, reasoning, and external search to address complex multimodal queries.

Implementation Details. We build our framework based on MiroFlow (Su et al., 2026), and utilize it for both trajectory rollout and answer verification. We utilize LLaMA-Factory (Zheng et al., 2024) as the training framework. We train the model for 3 epochs, with a global batch size of 64 and a learning rate of 1e-5. During evaluation, we set the maximum length as 128K and the maximum number of turns as 30 to make a fair comparison with previous methods. We also report the success rate when extending the turns to 100 and only keep the recent 5 tool call results, which is a context management strategy introduced by DeepSeek-V3.2 (Liu et al., 2025a). When evaluating on MMSearch-Plus, we only evaluate on the single-image subset to make a fair comparison with previous work (Huang et al., 2026).

Dataset	FVQA	LiveVQA	REDSearcher-MM	REDSearcher-Text	Ours
Num. of Samples	2301	1672	3366	3808	1589
Avg. Turns	5.63	7.16	13.21	21.97	17.26

Table 5: Dataset statistics, including the number of samples and the average number of tool-use turns per sample.

In *Percy Jackson And The Olympians* Season 2 Episode 8 (the finale):
- Percy Jackson is riding in a blue Prius with Sally and his friends.
- The Pegasus Blackjack appears, landing in front of the blue Prius.

Images:
- **Image URL**: https://images.squarespace-cdn.com/content/v1/5fbc4a62c2150e62cfcb09aa/1769036046809-0WFJWM5XTN0KMM9AZIO1/PercyJacksonSeason2Episode8.jpg
- **Caption**: Annabeth, Percy, and Grover back at Camp Half-Blood
- **Image URL**: https://images.squarespace-cdn.com/content/v1/5fbc4a62c2150e62cfcb09aa/79058544-8368-476a-b2e6-49939bb6737b/Percy-Jackson-Season-2-Episode-8-Thalia%2B1280.jpg
- **Caption**: Thalia with her spear
- **Image URL**: https://images.squarespace-cdn.com/content/v1/5fbc4a62c2150e62cfcb09aa/fd2497a0-85f0-4531-92b3-93210cacf068/Percy-Jackson-Season-2-Episode-8-Poseidon-and-Percy%2B1280.jpg
- **Caption**: Poseidon visits Percy in a dream.

Figure 4: The webpage returned to the agent. Its content is reorganized into a structured representation, where textual information is summarized into key bullet points and visual elements are converted into image–caption pairs, where the images are replaced with their URLs.

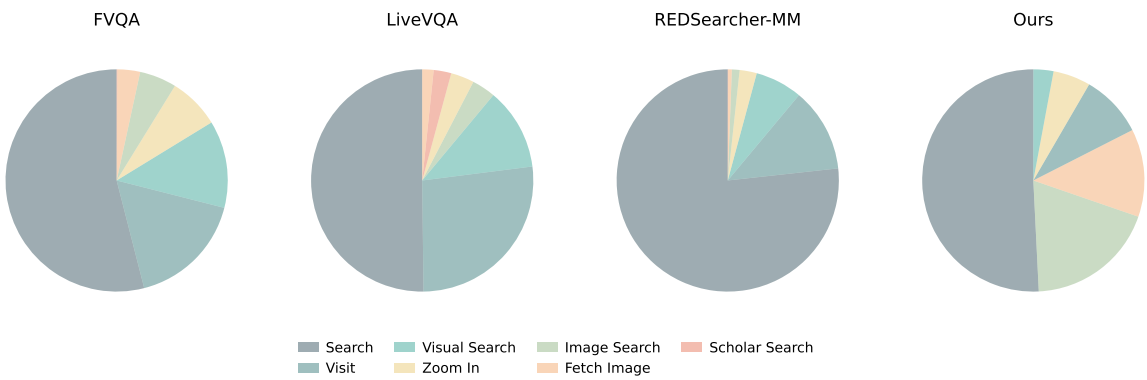


Figure 5: Tool call distribution of the training data. Our synthesized data require diverse types of tool calls and a higher proportion of “visual search” and “fetch image”.

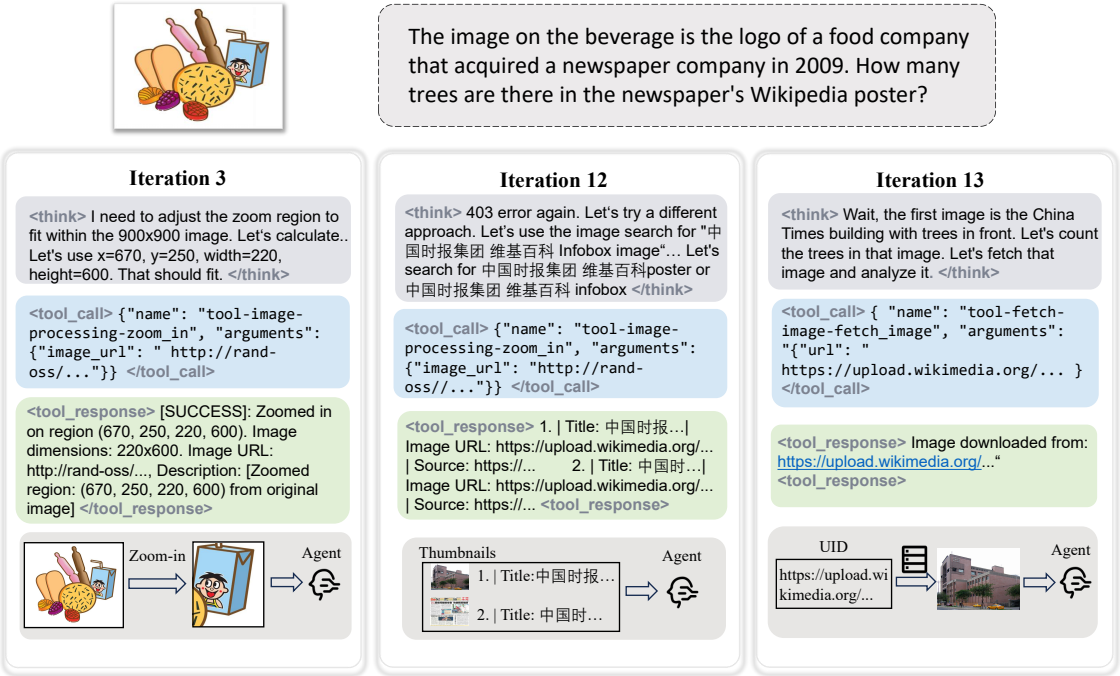


Figure 6: Detailed illustration of key steps within the model's search trajectory.

Examples of our LMM-Searcher on VisBrowse-Bench (Part 1/4)

Data Source: VisBrowse-Bench
Problem: The image on the beverage is the logo of a food company that acquired a newspaper company in 2009. How many trees are there in the newspaper's Wikipedia poster? Image URL: [http://rand-oss//2F%2F9bXW.jpg] Description: [Original input image]
Ground Truth: 5

Iteration 1

<think> First step is to identify the food company logo in the beverage image. Let's use visual search on the beverage (the blue carton with the cartoon character). </think>
<tool_call> {"name": "tool-google-search-visual_search", "arguments": {"image_url": "http://rand-oss//2FP2F9bXW"}} </tool_call>
<tool_response> 1. | Title: Giay tu cong bo san pham FROZEN COLA | Image URL: https://wantwant.vn/vnt..| Source: https://wantwant.vn/cn/gi.. 2. | Title: Facts For Kids- France | Image URL: https://www.oocities.org/... | Source: https://www.oocities.org/... </tool_response>

Iteration 2

<think> The visual search didn't directly identify the logo. Let's try zooming in on the blue beverage carton part of the image, since that's the logo area. Let's find the coordinates of the blue carton. Looking at the image, the blue carton is on the right side. Let's define the region: x around 670, y around 250, width around 300, height around 400. Let's zoom in on that area. </think>
<tool_call> { "name": "tool-image-processing-zoom_in", "arguments": {"image_url": "http://rand-oss/..."} } </tool_call>
<tool_response> [{"error": "Your cropped region extends beyond image bounds (image size: 900x900, your cropped region: (670+300)x(250+400))"}] </tool_response>

Iteration 3

<think> I need to adjust the zoom region to fit within the 900x900 image. Let's calculate: max x + width <=900, max y + height <=900. So if x=670, width can be up to 230 (900-670). If y=250, height can be up to 650 (900-250). Let's use x=670, y=250, width=220, height=600. That should fit. </think>
<tool_call> {"name": "tool-image-processing-zoom_in", "arguments": {"image_url": "http://rand-oss/..."}} </tool_call>

<tool_response> [SUCCESS]: Zoomed in on region (670, 250, 220, 600). Image dimensions: 220x600. Image URL: http://rand-oss/..., Description: [Zoomed region: (670, 250, 220, 600) from original image] </tool_response>

Table 6: A case study illustrating the complete search trajectory of the model.

Examples of our LMM-Searcher on VisBrowse-Bench (Part 2/4)

Iteration 4

<think> Now I can see the logo clearly: it's Want Want's mascot (the boy with the red cheeks). Want Want is a food company. The question says this company acquired a newspaper company in 2009. Let me confirm: Want Want acquired a newspaper? Let me search for "Want Want acquired newspaper 2009". </think>
<tool_call> {"name": "tool-google-search-google_search", "arguments": "{"q": "Want Want acquired newspaper 2009"}"} </tool_call>
<tool_response> 1. "title": "China Times Group is sold to Want Want", "link": "https://www.taipeitimes.com/News/biz/...", "snippet": "It owns the Chinese-language newspapers China Times..." 2. "title": "China Times Joins a Snack-food Empire|Industry|2009-03-05", "link": "https://english.cw.com.tw/article/...", "snippet": "A month after Want Want Holdings chairman Tsai Eng-meng ..." ..... </tool_response>

Iteration 5

<think> So the newspaper company is China Times Group (中时集团), acquired by Want Want in 2008 (the deal closed in 2009? The news says completed in March 2009). Now, the question asks: "How many trees are there in the newspaper's Wikipedia poster?" ... Let me search for the Wikipedia page of the newspaper acquired, then check the image (poster) for tree count. First, let's get the Wikipedia page for China Times Group. Let me search for "China Times Group Wikipedia" or "旺旺中时媒体集团 Wikipedia" (since it's in Chinese). </think>
<tool_call> { "name": "tool-google-search-google_search", "arguments": "{"q": "Want Want acquired newspaper 2009"}" } </tool_call>
<tool_response> 1. "title": "旺旺中时媒体集团- 维基百科", "link": "https://zh.wikipedia.org/zh-cn/..." 2. "title": "可靠来源/常见/旺旺中时媒体集团- 维基百科", "link": "https://zh.wikipedia.org/zh-cn/..." ..... </tool_response>

Iteration 6

<think> Now, let's open the Wikipedia page for 旺旺中时媒体集团 (https://zh.wikipedia.org/zh-cn/旺旺中时媒体集团). We need to find the "poster" (infobox image) and count the number of trees in it. Let's use the scrape_website tool to get the content of this page, ... Let's call the scrape_and_extract_info tool with the URL and ask to count the trees in the Wikipedia poster (the infobox image). </think>
<tool_call> { "name": "jina_scrape_llm_summary-scrape_and_extract_info", "arguments": "{"url": "https://zh.wikipedia.org/zh-cn/...", "info_to_extract": "Count the number of trees in the Wikipedia poster of 旺旺中时媒体集团"}" } </tool_call>
<tool_response> {"success": false, "url": "https://zh.wikipedia.org/zh-cn/旺旺中时媒体集团", "extracted_info": "", "error": "Scraping failed (both Jina and Python): Python Scrape: Unexpected error occurred: Client error '403 Forbidden' for url 'https://zh.wikipedia.org/zh-cn/..."} </tool_response>

Iteration 7

<think> 403 Forbidden error. Let's try a different approach. Let's first find the infobox image URL for this Wikipedia page. ... Let's search for "旺旺中时媒体集团维基百科图片". </think>
<tool_call> { "name": "tool-google-search-image_search", "arguments": "{"q": "旺旺中时媒体集团维基百科图片"}" } </tool_call>
<tool_response> 1. | Title: 旺旺中时媒体集团- 维基百科，自由的百科全书 | Image URL: https://upload.wikimedia.org/wikipedia/zh/thumb/0/05/... | Source: https://zh.wikipedia.org/zh-tw/... 2. | Title: 旺旺集团- 维基百科，自由的百科全书 | Image URL: https://upload.wikimedia.org/wikipedia/commons/7/74/... | Source: https://zh.wikipedia.org/zh-tw/... </tool_response>

Examples of our LMM-Searcher on VisBrowse-Bench (Part 3/4)

Iteration 8

<think> Wait, the first image is the logo of Want Want China Times Media Group. Let's check that image. The logo has the Want Want kid, "中时集团" in red squares, and "CHINA TIMES GROUP". Is there a tree in this logo? Let's fetch this image and analyze it. </think>
<tool_call> { "name": "tool-fetch-image-fetch_image", "arguments": "{"url": "https://upload.wikimedia.org/wikipedia/zh/thumb/0/05/..."}" } </tool_call>
<tool_response> Image downloaded from: https://upload.wikimedia.org/wikipedia/.. </tool_response>

Iteration 9

<think> Looking at this logo, I don't see any trees. Wait, maybe the "newspaper's Wikipedia poster" refers to the China Times (China Times Group) poster, not the media group's poster. Let's check the Wikipedia page for China Times Group (中国时报集团). Let's search for "中国时报集团维基百科" and get its poster. Alternatively, maybe the "poster" is the Wikipedia infobox image for China Times Group. Let's search for the China Times Group Wikipedia page. Let me call the google search for "中国时报集团维基百科". </think>
<tool_call> { "name": "tool-google-search-google_search", "arguments": "{"q": "中国时报集团维基百科"}" } </tool_call>
<tool_response> 1. "title": "讨论:中国时报集团- 维基百科，自由的百科全书", "link": "https://zh.wikipedia.org/zh-my/..." 2. "title": "中时新闻网", "link": "https://www.wikiwand.com/zh-hant/..." </tool_response>

Iteration 10

<think> Wait, maybe the Wikipedia page for China Times Group is under a different name? Let me try "中时集团维基百科". </think>
<tool_call> { "name": "tool-google-search-google_search", "arguments": "{"q": "中时集团维基百科"}" } </tool_call>
<tool_response> 1. "title": "中国时报集团- 维基百科，自由的百科全书", "link": "https://zh.wikipedia.org/zh-cn/..." 2. "title": "旺旺中时媒体集团- 维基百科", "link": "https://zh.wikipedia.org/zh-cn/..." </tool_response>

Iteration 11

<think> Okay, so the newspaper company is China Times Group (中国时报集团), which was acquired by Want Want in 2008-2009. Now, the question is asking for the number of trees in the newspaper's Wikipedia poster. Let's find the Wikipedia poster (infobox image) for China Times Group. Let's scrape the Wikipedia page for China Times Group and look for the infobox image. Let's use the jina_scrape tool on that page, asking to find the infobox image and count the trees in it. </think>
<tool_call> { "name": "jina_scrape_llm_summary-scrape_and_extract_info", "arguments": "{"url": "https://zh.wikipedia.org/zh-cn/...", "info_to_extract": "Find the infobox image (poster) on the Wikipedia page for 中国时报集团 and count the number of trees in it."}" } </tool_call>
<tool_response> {"success": false, "url": "https://zh.wikipedia.org/zh-cn/...", "extracted_info": "", "error": "Scraping failed (both Jina and Python): Python Scrape: Unexpected error occurred: Client error '403 Forbidden' for url 'https://zh.wikipedia.org/zh-cn/..."} </tool_response>

Examples of our LMM-Searcher on VisBrowse-Bench (Part 4/4)

Iteration 12

<think> 403 error again. Let's try a different approach. Let's use the image search for "中国时报集团维基百科 Infobox image". Or maybe directly find the infobox image URL via Google search image. Let's search for "中国时报集团维基百科 poster" or "中国时报集团维基百科 infobox". </think>
<tool_call> { "name": "tool-google-search-image_search", "arguments": "{"q": "中国时报集团维基百科 infobox"}" } </tool_call>
<tool_response> 1. | Title: 中国时报集团- 维基百科，自由的百科全书 | Image URL: https://upload.wikimedia.org/wikipedia/commons/e/ee/... | Source: https://zh.wikipedia.org/zh-tw/... 2. | Title: 中国时报- 维基百科，自由的百科全书 | Image URL: https://upload.wikimedia.org/wikipedia/zh/9/9b/... | Source: https://zh.wikipedia.org/zh-cn/... </tool_response>

Iteration 13

<think> Wait, the first image is the China Times building with trees in front. Let's count the trees in that image. Let's fetch that image and analyze it. </think>
<tool_call> { "name": "tool-fetch-image-fetch_image", "arguments": "{"url": "https://upload.wikimedia.org/wikipedia/commons/e/ee/..."}" } </tool_call>
<tool_response> Image downloaded from: https://upload.wikimedia.org/..." </tool_response>

Iteration 14

<think> Now let's count the trees in this image. Let's see: in front of the building, there are several trees. Let's list them: left side: one tree, then two more, ... Wait, but let me confirm once more. Let me list each tree with coordinates: 1. (x: 20, y: 800) - small tree 2. (x: 200, y: 850) - tree near pole 3. (x: 450, y: 880) - tree near first taxi 4. (x: 550, y: 890) - tree near second taxi 5. (x: 900, y: 850) - large tree on right Yes, that's 5 trees. So the answer is 5. </think>
The food company is Want Want, which acquired the **China Times Group** (中时媒体集团) in 2009. The Wikipedia infobox poster for this newspaper shows **5 distinct trees** in front of its headquarters building."

Iteration 15

<User> Summarize the above conversation, and output the FINAL ANSWER to the original question. If a clear answer has already been provided earlier in the conversation, do not rethink or recalculate it — simply extract that answer and reformat it to match the required format below. If a definitive answer could not be determined, make a well-informed educated guess based on the conversation...
<think> Okay, let's summarize the conversation and extract the final answer. First, we identified the logo on the beverage as Want Want's mascot. Then we found that Want Want acquired China Times Group in 2009. We looked at the Wikipedia poster (infobox image) for China Times Group, which shows the headquarters building with trees. Counting the distinct trees, we found 5.The original question asks for the number of trees in the newspaper's Wikipedia poster. So the answer is boxed{5}. </think>