M^{3}Searcher: Modular Multimodal Information Seeking Agency with Retrieval-Oriented Reasoning

Anonymous ACL submission

Abstract

Recent advances in DeepResearch-style agents have demonstrated strong capabilities in autonomous information acquisition and synthesize from real-world web environments. However, existing approaches remain fundamentally limited to text modality. Extending autonomous information-seeking agents to multimodal settings introduces critical challenges: the specialization-generalization trade-off that emerges when training models for multimodal tool-use at scale, and the severe scarcity of training data capturing complex, multi-step multimodal search trajectories. To address these challenges, we propose M^{3}Searcher, a modular multimodal information-seeking agent that explicitly decouples information acquisition from answer derivation. M^{3}Searcher is optimized with a retrieval-oriented multi-objective reward that jointly encourages factual accuracy, reasoning soundness, and retrieval fidelity. In addition, we develop MMSearchVQA, a multimodal multi-hop dataset to support retrieval centric RL training. Experimental results demonstrate that M^{3}Searcher outperforms existing approaches, exhibits strong transfer adaptability and effective reasoning in complex multimodal tasks.

1 Introduction

DeepResearch-style agents have recently demonstrated striking proficiency in acquiring and synthesizing information from real-world web environments, as exemplified by OpenAI DeepResearch (OpenAI, 2025) and Gemini DeepResearch (Google, 2025). These advances have spurred a growing research effort to equip large language models (LLMs) with reasoning-intensive search capabilities (Shao et al., 2025). Most approaches leverages reinforcement learning (RL) to train models to interact with web search engines (e.g. Google Search), planning, gathering and synthesizing information through multi-step deliberation (Jin et al., 2025; Zheng et al., 2025). However, these approaches remain confined to text modality, even though real-world user information needs are inherently multimodal (e.g. visual perception).

Extending autonomous information-seeking agents to multimodal inputs is therefore an essential step for building general intelligent systems. Nevertheless, this transition introduces several fundamental challenges: (i) **Specialization-Generalization Trade-off**: Training models to internalize multimodal tool-use policies comes at the expense of general reasoning capacity (Kalajdzievski, 2024; Li et al., 2024a), yet the effectiveness of multimodal RAG systems critically relies on a backbone model whose core reasoning performance remains robust and uncompromised. (ii) **Training Data Scarcity**: Existing datasets that capture complex, multi-step search trajectories are primarily designed for evaluation purposes (Wei et al., 2025), whereas large-scale training corpora provide only shallow reasoning paths, such as InfoSeek (Chen et al., 2023). This discrepancy hinders models from developing long-horizon information-seeking strategies.

To resolve these challenge, and inspired by the modular design of Jiang et al. (2025), we decouple the information-seeking process from answer derivation. Specifically, we introduce a lightweight and trainable MLLM, termed **M^{3}Searcher**, that serves as a dedicated modular multimodal information seeking agency. Its role is to execute a mulitmodal reasoning-intensive information seeking process (Shao et al., 2025). Specifically, it interprets non-textual inputs (e.g. visual recognition, OCR) and dynamically coordinating search strategies across heterogeneous modalities to assemble comprehensive and contextually relevant evidence. The gathered information is subsequently provided to a downstream answer generator, which performs reasoning over the curated evidence and formulates

[2] ✉ Corresponding author.

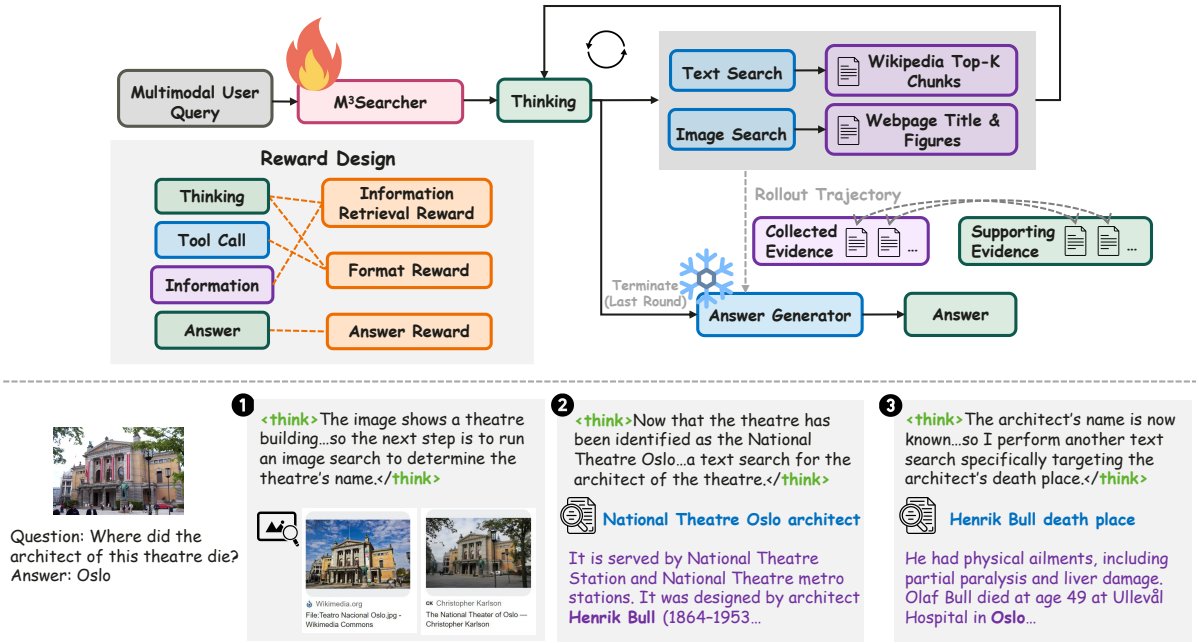

Figure 1: The architecture of M^{3}Searcher.

the final response to the user query. To effectively train M^{3}Searcher, we further propose a decoupled reinforcement learning framework with the following contributions:

1. **Dataset Construction:** We introduce **MM-SearchVQA**, a dataset demanding rigorous multimodal information seeking. Each instance enforces answer uniqueness and is accompanied by automatically extracted supporting evidence. By encompassing a broad spectrum of domains, difficulty levels and search intensities, the dataset encourages the model to learn the distinct control policies required for determining *when* to search, *what* to query, and *how* to integrate external knowledge.

2. **Decoupled Multimodal Information Seeking:** M^{3}Searcher focuses exclusively on optimizing heterogeneous search scheduling for maximizing information acquisition. To realize this, we introduce a specialized "expert answer generator" tool, which is triggered only once the context is deemed sufficient and well-grounded for the following reasoning process. This modularity allows the search strategies to remain highly adaptive while maintaining the reasoning capacity of a robust backbone within the MRAG system. It also renders the generator modality-agnostic, accommodating both pure textual LRMs (e.g., DeepSeek-R1) and MLLMs (e.g., GPT-4o).

3. **Retrieval-Oriented Multi-Objective Reward:** We employ a multi-objective reward modeling framework that jointly optimizes answer accuracy, reasoning validity, and retrieval quality. To ensure that the model genuinely grounds its inferences in retrieved evidence rather than exploiting spurious shortcuts, we incorporate a retrieval reward that evaluates the completeness of textual information gathering and the accuracy, and interpretive soundness of visual reasoning.

We conduct comprehensive evaluations of M^{3}Searcher across real-world benchmarks to assess its effectiveness. M^{3}Searcher outperforms both prompt-engineered agents and end-to-end trained counterparts. Moreover, it exhibits strong robustness and adaptability, as evidenced by stable performance under multiple transfer scenarios involving variations in search engines and answer generators.

2 M^{3}Searcher

2.1 Task Formulation

We consider a multimodal query (v, q) where v is the visual component and q is the textual component, with its ground-truth answer a. A trainable MLLM is formalized as an information-seeking agent $\mathcal{F}$, which engages in iterative interaction with a multimodal tool set $\mathcal{T}$. Through a sequence of tool invocations and intermediate reasoning steps,

the agent incrementally acquires task-relevant evidence and integrates the retrieved information to derive the final answer:

$$\mathcal{F}(q, v, \mathcal{T}) \rightarrow a. \quad (1)$$

2.2 Decoupled Agentic MRAG

Existing MRAG approaches either leverage a large-scale MLLM with elaborate prompt engineering, or train a smaller MLLM end-to-end (Geng et al., 2025; Wu et al., 2025b). This dichotomy introduces a fundamental dilemma. Large-scale MLLMs (e.g. GPT-4o) excel in emergent reasoning, but lack optimization for real-world web integration. Conversely, smaller models (e.g., Qwen2.5-VL-7B) optimized for web search exhibit a concurrent degradation in general reasoning, limiting their utility as a primary backbone for the overall system. To address this dilemma, and drawing inspiration from modular architectures (Jiang et al., 2025), we introduce a decoupled MRAG architecture that separates the information-seeking process from the answer generation. As depicted in Figure 2, M^{3}Searcher is solely responsible for comprehending multimodal queries, formulating iterative search strategies across heterogeneous search tools, and determining the optimal search termination point. The final evidential data is then passed to a dedicated answer generation for synthesis. This architectural decoupling offers two key advantages: it preserves the reasoning fidelity of the large-scale backbone while allowing for targeted optimization of information-seeking capabilities, and simultaneously removes modality constraints, enabling the use of modality-agnostic generators (e.g., GPT-4o, DeepSeek-R1).

2.3 Multimodal Tools Implementation

We equip M^{3}Searcher with three essential tools: an image search tool, a text search tool, and an answer generator tool. To enable effective RL, we developed a stable and high-concurrency tool environment. For `image search`, we integrate the Serper API[1] to perform reverse image retrieval. Given an input image, the API returns visually similar images, together with their corresponding website titles and URLs. Since the Serper API produces highly stable results, we incorporate a caching mechanism to reduce resource consumption and accelerate the search process. For `text search`, we utilize the 2025 wikipedia dump[2] as knowledge source. A retrieval–reranking pipeline, built upon the E5 models (Wang et al., 2022), is used to retrieve semantically relevant document chunks given a user query. Finally, the `answer expert` tool employs a high-capacity LRM which consumes the trajectory of information-seeking and synthesizes a final response.

2.4 Decoupled Multi-turn Rollout

M^{3}Searcher processes the query through three core operational states. In the **`Think`** state, the model conducts a fine-grained inspection of the visual component v and performs contextual inference across modalities, integrating visual and textual cues to construct a coherent situational understanding. When additional information is required, M^{3}Searcher transitions to the **`Tool_Call`** state, dynamically invoking external tools to retrieve supplementary evidence from real-world sources. The retrieved outputs are then encapsulated as **`Information`**, which re-enters the reasoning loop to refine and expand the model's understanding M^{3}Searcher operates iteratively upon these three states, allowing for progressive refinement of its understanding and retrieval strategy. Formally, at each time step t, the M^{3}Searcher execution can be represented as a tuple (α_t, C_t, I_t), where α_t represents the reasoning process, C_t is the tool invocation, and I_t is the tool response. The full rollout trajectory can thus be expressed as:

$$\mathcal{T} = \{O_1, \alpha_1, C_1, I_1, \ldots, O_t, \alpha_t, C_t, I_t\}. \quad (2)$$

Under the decoupled agentic design, the final tool invocation of M^{3}Searcher is required to invoke the answer expert. Consequently, the terminal tool response I_t provides the final answer to the user query q:

$$I_t = \mathcal{F}(q). \quad (3)$$

The prompt governing the rollout procedure is detailed in Appendix B.

2.5 Multi-Objective Rewrad Modeling

The goal of M^{3}Searcher is to perform a multimodal, reasoning-intensive information-seeking process. It must progressively and comprehensively gather relevant evidence across multiple hops to support the downstream generator in generating accurate

[1] https://serpapi.com/

[2] https://dumps.wikimedia.org/

answers. To achieve this, we formulate a multi-objective, retrieval-oriented RL reward function that jointly optimize accuracy, completeness and relevance of the information acquisition process.

Format Reward The format reward R_{format} enforces strict compliance with the syntactic and structural constraints specified in the prompt. For example, tool invocations are required to follow the correctly structured parsing format with valid parameterization; and the trajectory must terminate with a call to the answer generator tool. Any deviation from these requirements incurs a strong penalty of an absolute reward of -1.

Answer Reward The answer reward, R_{answer}, measures the semantic correctness of the final output I_t with respect to the reference solution. Rather than relying on brittle exact string matching, we employ an LLM-as-Judge evaluation strategy, which confers both flexibility and robustness in cases where multiple equivalent phrasings or semantically consistent answers are acceptable. The complete scoring prompt used for this evaluation is provided in the Appendix.

Information Retrieval Reward The information retrieval reward is designed to assess the fidelity and completeness of the information acquired to solve a multi-hop user query, independent of the capabilities of the downstream answer generator. The evaluation of this information acquisition is divided by modality.

For the visual modality, when processing retrieved images or relying on internal knowledge within the model's training cutoff, it may exhibit three distinct behaviors during the `Think` state: (1) correctly identifying the key visual elements, (2) demonstrating uncertainty and refraining from explicit recognition while offering a descriptive interpretation, or (3) producing an incorrect recognition. To shape this behavior, we assign graded rewards $R_{ImgRetrieval}$ of 0.5, 0.25, and 0, respectively. This reward structure encourages the model to adopt a more cautious and self-aware strategy when reasoning over visual inputs. For the textual modality, we assess the degree to which the information conveyed to the answer generator aligns with the reference evidence in MMSearchVQA. This metric quantifies whether M^{3}Searcher identifies all necessary pieces of information required for solving the query, thereby enhancing information completeness and mitigating reasoning shortcuts that may yield correct answers without genuine verification (e.g., a builder's place of death is not always the same as the building's location). The textual retrieval reward, denoted as $R_{TextRetrieval}$, is defined as a percentage score ranging from 0 to 0.5, representing the proportion of reasoning hops successfully supported by retrieved evidence. Specifically, we compare each reference evidence against both the `Information` and `Think` states. To ensure robust evaluation, we employ a LLM-as-Judger method to assess both modality reward score:

$$R_{ImgRetrieval} = LLM(\alpha_i), \tag{4}$$

$$R_{TextRetrieval} = LLM(\alpha_i, C_i). \tag{5}$$

The detailed judging prompt is provided in the Appendix. The final reward is:

$$R = R_{format} + R_{answer} + R_{Retrieve}, \tag{6}$$

$$R_{Retrieve} = R_{TextRetrieval} + R_{ImgRetrieval}. \tag{7}$$

2.6 RL Training

To enhance the model's capability for information seeking and web-environment interaction within the MRAG framework, we adopt Group-Relative Policy Optimization (GRPO) (Shao et al., 2024). For each input multimodal question q, the current policy π_θ samples a group of trajectories $y_1, \ldots, y_G\}$. Then the optimization objective of GRPO is formulated as:

$$\mathcal{J}(\theta) = \mathbb{E}_{i,t} \min\Big[\rho_t^i A_t^i, \text{clip}(\rho_t^i,\ 1-\epsilon,\ 1+\epsilon) A_t^i\Big] - \beta\, \mathbb{D}_{\text{KL}}[\pi_\theta || \pi_{\text{ref}}], \tag{8}$$

where ρ_t^i represents the importance sampling ratio between the updated and previous policies and A_t^i is an estimator of the advantage at time step t:

$$A_t^i = \frac{R_i - mean(\{R_i\})}{std(\{R_i\})}. \tag{9}$$

The hyperparameter β controls the KL divergence penalty, constraining the deviation from the reference policy to ensure stable updates. The context for policies includes both model-generated outputs and tool responses. To prevent external knowledge sources from biasing policy learning, we apply a loss mask over all tool-response tokens. This ensures that policy gradients are computed exclusively for LLM-generated tokens, enabling precise

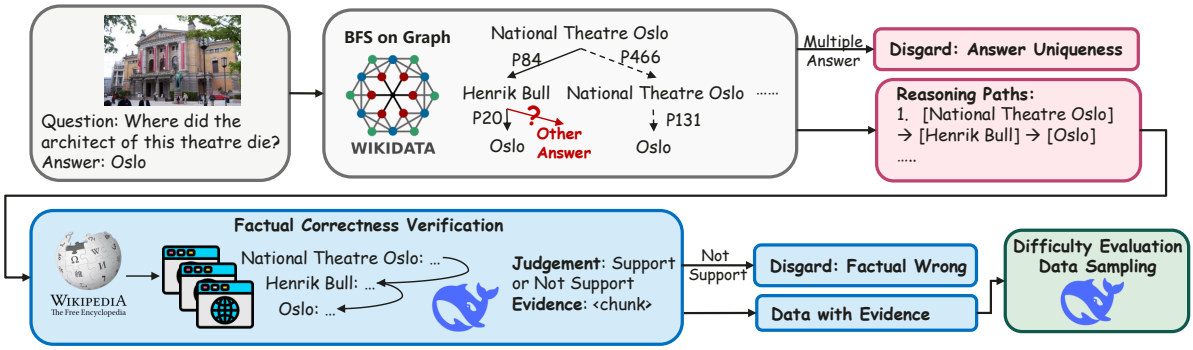

Figure 2: The MMSearchVQA data construction pipeline.

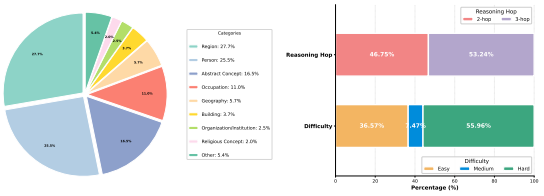

Figure 3: Overview of MMSearchVQA dataset statistics. The left figure summarizes the domain distribution, and the right figure reports the distribution of question difficulty levels and reasoning hops.

optimization of search planning and multimodal information-seeking capabilities within the MRAG system.

3 MMSearchVQA Dataset

Existing Visual Question Answering (VQA) datasets typically fall into two categories. Automatically constructed datasets (Chen et al., 2023; Cheng et al., 2025; Wu et al., 2025b; Fu et al., 2025), such as InfoSeek, involve shallow reasoning chains — often limited to two-hop queries solvable through a simple sequence of image search followed by text search. In contrast, manually curated datasets such as MM-BrowseComp (Li et al., 2025b) feature more complex, multi-step reasoning but are expensive and difficult to scale. To address this limitation, we introduce MMSearchVQA, a dataset designed to foster the development of models for advanced information-seeking reasoning. MMSearchVQA not only requires deeper search and reasoning but also provides explicit supporting evidence that underpin the reasoning and answering processes.

As illustrated in Figure 2, our dataset is constructed upon ReasonVQA (Tran et al., 2025), which is derived from the Wikidata. We first perform a BFS traversal on the Wikidata graph, identifying all potential reasoning chains associated with each question. During this traversal, we discard questions that yield multiple valid answers to ensure answer uniqueness, and we retain only those samples that require at least two reasoning hops. Following the extraction of candidate reasoning paths, we conduct cross-validation against Wikipedia using the DeepSeek models. Each reasoning hop, including the final answering, must be consistently supported by evidence drawn from relevant Wikipedia content to be both factually accurate and temporally valid. Instances that fail to meet these criteria are excluded. During this verification process, we also extract fine-grained supporting evidence from the corresponding Wikipedia passages for each reasoning step, thus enhancing the interpretability and traceability of the reasoning process. To further characterize the cognitive difficulty of the resulting dataset, we employ DeepSeek-V3 (Guo et al., 2025) to answer each question three times and categorize questions into three levels: easy (all correct), medium (partially correct), and hard (all incorrect). This procedure yields a principled estimate of reasoning complexity across samples. To cultivate an information seeker capable of performing deep and precise searches, we prioritize training data that exhibit deeper information needs and greater reasoning complexity. Accordingly, we downsample easy questions to half the number of hard examples, ensuring a balanced yet challenging dataset. In total, the curated dataset contains 6,000 questions, with comprehensive statistics presented in Figure 3.

4 Experiments

4.1 Experimental Setup

Datasets We adopt MMSearchVQA dataset as the training corpus. We evaluate performance on

Table 1: The overall performance of M^{3}Searcher compared with baseline approaches. MMSearchVQA is in-domain benchmark with wikipedia based search and other benchmarks are transferred to Google Search Engine. The best performance is highlighted in **bold**, and the second-best performance is underlined.

Method	Backbone	In-Domian (Wiki Search)	→ Out-Domain (Google Search)		
		MMSearchVQA	InfoSeek	MMSearch	MRAG-Bench
No Agency					
Direct	Qwen3-VL-235B-A22B	40.42	40.16	28.23	9.22
	Qwen2.5-VL-72B	43.12	35.22	15.29	14.12
	Qwen2.5-VL-7B	31.12	23.50	11.69	8.57
RAG	Qwen3-VL-235B-A22B	29.79	31.75	30.83	**33.67**
	Qwen2.5-VL-72B	40.37	33.26	46.15	20.08
	Qwen2.5-VL-7B	30.00	31.75	38.09	15.68
Prompt Engineered Agents					
OmniSearch	Qwen2.5-VL-72B	45.65	40.60	15.00	27.07
	Qwen2.5-VL-7B	22.91	25.17	22.22	23.96
CogPlanner	Qwen2.5-VL-72B	48.37	41.72	39.77	29.12
	Qwen2.5-VL-7B	22.12	26.22	27.48	28.23
End-to-End Agents					
MMSearch-R1-7B$_{Retrain}$		31.63	20.20	7.02	27.60
MMSearch-R1-7B$_{Release}$		20.50	37.06	12.28	19.20
Decoupled Agents w/o Training					
Qwen3-30B-A3B		31.50	31.20	36.69	24.20
Qwen2.5-VL-7B		34.12	33.80	36.09	27.20
M^{3}Searcher					
Qwen3-30B-A3B		54.75	39.61	55.62	24.91
→ Transfer LRM Answer Generator					
DeepSeek-V3		56.87	40.33	60.95	29.12
DeepSeek-R1		59.25	**42.50**	**63.30**	30.00
→ Transfer MLLM Answer Generator					
Qwen2.5-VL-7B		57.00	39.44	61.54	19.95
Qwen2.5-VL-72B		**59.50**	40.20	59.17	27.20

both in-domain and out-of-domain benchmarks. The in-domain evaluation uses the MMSearchVQA test set, while out-of-domain evaluation is conducted on three publicly available VQA datasets: MMSearch (Jiang et al., 2024), Infoseek (Chen et al., 2023), and MRAG-Bench (Hu et al., 2024).

Baselines and Metrics We compare M^{3}Searcher against four categories of methodologies: (1) No-agency: We directly prompt MLLMs and use a fixed RAG pipeline comprising image retrieval, query rewriting, text retrieval, and answer generation. (2) Prompt-engineered agents: We select OmniSearch (Li et al., 2024b) and CogPlanner (Yu et al., 2025), both of which coordinate multiple agents via hand-crafted prompts for multimodal reasoning and retrieval. (3) End-to-end agents: We include MMSearch-R1 (Wu et al., 2025b) as a representative method optimized through end-to-end RL training. (4) Decoupled agents without specialized training: We employ a decoupled architecture in which Qwen2.5-VL-7B is used for information-seeking operations, without any tuning. We adopt LLM-as-Judge as the evaluation metric, which is well-aligned with the answer accuracy reward.

Transfer Experiment Settings For M^{3}Searcher, we conduct two sets of transfer experiments: (1) Search engine transfer: For the MMSearchVQA benchmark, which is built on Wikipedia-based content, we employ our in-house text search tool (described in Section 2.3) to retrieve the top 10 most relevant text chunks. For other benchmarks based on open-domain web data, we switch to Google Search via the Serper API[3] , also keeping the top 10 retrieved results. (2) Answer generator transfer: We explore the transfer of answer generators by incorporating models from both the DeepSeek series (Guo et al., 2025; Liu et al., 2024) and the Qwen-VL series (Bai et al., 2025).

[3] https://serper.dev/

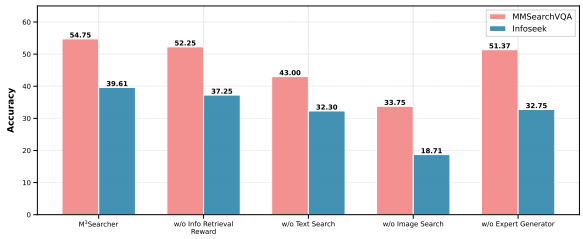

Figure 4: Ablation study.

Implementation Details For the baseline methodologies, we adopt Qwen3-VL-30B-A3B, Qwen2.5-VL-72B, and Qwen2.5-VL-7B as backbone models. Specifically, MMSearch-R1 employs Qwen2.5-VL-7B for both end-to-end training and inference. For M^{3}Searcher, we utilize Qwen2.5-VL-7B as the trainable planner and Qwen3-30B-A3B (Yang et al., 2025) as the answer generator during training. Verl (Sheng et al., 2024) is used for multi-turn RL training.

5 Main Results

The overall performance of M^{3}Searcher across multiple benchmarks is summarized in Table 1. Several insights can be drawn from these results: (1) **Baseline methodologies exhibit unstable variability in performance across benchmarks.** In most cases, either fixed RAG pipelines or prompt-engineered agents attain the strongest results. This suggests that explicit, hand-crafted prompt engineering provides a competitive advantage that decoupled, untrained agents fail to surpass. Agents with specialized training display inconsistent performance and unstable generalization: for example, MMSearch-R1 performs competitively on Infoseek, but its performance drops sharply on out-of-distribution tasks. (2) **M^{3}Searcher demonstrates robust and strong performance across various generalization and transfer settings.** It provides high-quality, correctly excavated evidence and both multimodal and purely textual backbones can reliably synthesize accurate answers, lifting the modality constraints. Notably, DeepSeek-R1 answer generator emerges as the top performer, underscoring the critical role of the inherent reasoning capability of the backbone model in the overall MRAG system effectiveness. M^{3}Searcher also maintains stable performance under search-engine transfer, exhibiting no degradation when switching search tools. This robustness highlights its high robustness to variations in the underlying information source, and further indicates that a self-built textual search engine is fully sufficient for on-policy RL training — particularly important given the prohibitive cost of commercial search engines.

6 Analysis

Ablation study. We evaluate the contribution of each core component in M^{3}Searcher by removing the Information Retrieval Reward, the image search tool, the text search tool and the answer generator tool. As shown in Figure 4, each component provides a measurable performance gain, underscoring their collective importance to the overall system effectiveness.

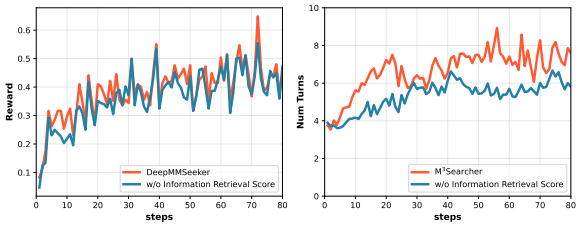

Figure 5: Training dynamics of reward and rollout turn counts with and without the information-retrieval reward.

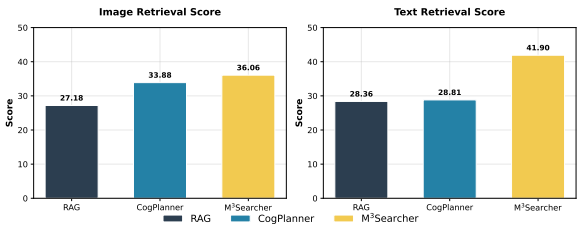

Figure 6: Text retrieval score and image retrieval score of M^{3}Searcher compared with CogPlanner and RAG pipeline baseliness.

Retrieval-oriented rewards enhance the breadth and completeness of information seeking. To rigorously evaluate the impact of retrieval-oriented reward design, we analyze the training dynamics presented in Figure 5. The results indicate that incorporating an information retrieval reward leads to consistently higher reward signals. Consequently, M^{3}Searcher engages in a greater number of information-seeking turns. This enables a broader coverage of relevant information and yielding final evidence that is both more complete and reliable, as demonstrated in Figure 6.

RL enhances heterogeneous tool coordination and improves the model's ability to leverage the

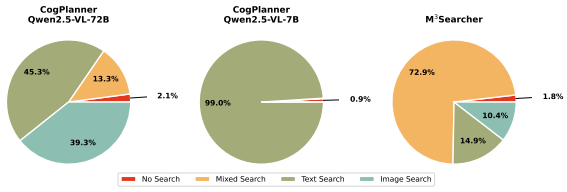

Figure 7: Tool usage statistics on MMSearchVQA.

image-search tool We analyze tool usage patterns on the MMSearchVQA benchmark in Figure 7. The results reveal a pronounced bias in the pretrained Qwen2.5-VL-7B model as it overwhelmingly favors text search tool while almost never invoking the image search tool. After RL this imbalance is substantially mitigated. It invokes a more diverse mixture of search actions, with a notably increased reliance on the image-search tool, indicating that RL helps the model internalize when visual external information is necessary for successful reasoning.

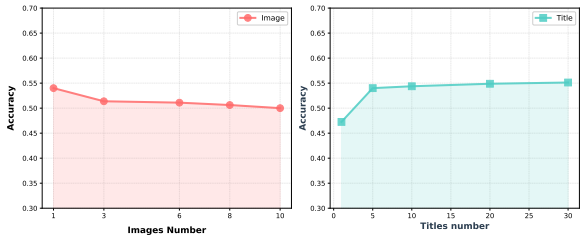

Figure 8: Performance with varying numbers of returned relevant images and associated webpage titles in the image search tool.

The Design of Image Search Tools as a Core Factor in Agentic MRAG Performance Our empirical observations indicate that the design of image search tools constitutes a critical determinant of performance. Specifically, we analyzed the influence of the returned relevant images and their associated webpage titles. As depicted in Figure 8, increasing the number of returned images lead to performance degradation, potentially attributed to the redundancy or near-similarity of multiple image inputs, which may introduce noise or confusion into the model's visual feature extraction process. Conversely, increasing the volume of textual returned information (webpage titles) demonstrates a positive correlation with performance since the webpage titles provide crucial context for interpreting the visual query. Based on this quantitative analysis, we adopt a design choice for the image search tool that returns the top-1 image along with the top-30 associated webpage titles, which balances informative context with minimal visual redundancy.

7 Related Work

With the introduction of DeepResearch by leading AI organizations, including OpenAI (OpenAI, 2025), Google (Google, 2025), and Perplexity (Perplexity, 2025), these systems have demonstrated strong potential in solving complex multi-step reasoning tasks. Recent advances highlight reinforcement learning (RL) as a promising paradigm and OpenAI's technical report explicitly demonstrates the effectiveness of employing RL to strengthen the multi-step decision-making and retrieval abilities (Jaech et al., 2024). Notable works such as Search-R1 (Jin et al., 2025; Song et al., 2025) mark an early milestone by incorporating web-search tool interaction into textual question-answering scenarios, achieving substantial performance gains. Following this, the Web Agents series developed by the Qwen team (Li et al., 2025a; Wu et al., 2025a) further optimizes information-seeking behaviors in complex, non-linear reasoning tasks. However, despite these advances, rare attention has been given to the optimization of MRAG systems, where reasoning must integrate and synthesize heterogeneous modalities. Exsiting work (Geng et al., 2025; Wu et al., 2025b; Narayan et al., 2025) employ a end-to-end RL paradigm for VQA tasks, which inadvertently restrict the MRAG backbone to relatively small models (e.g., Qwen2.5-VL-7B). This constraint imposes a substantial performance ceiling, limiting the practical effectiveness of these systems in real-world deployments.

8 Conclusion

We present M^{3}Searcher, a lightweight and trainable multimodal information seeker that decouples retrieval from answer generation in MRAG systems. By focusing on adaptive, reasoning-intensive search over heterogeneous sources, M^{3}Searcher preserves the reasoning capacity of downstream generators while efficiently aggregating contextually relevant evidence. Experiments on MMSearchVQA and real-world benchmarks demonstrate strong performance and robustness across different search engines and generators.

Limitations

We discuss several limitations of M^{3}Searcher as follows. First, although M^{3}Searcher adopts a modular architecture, its effectiveness is inherently constrained by the scale and diversity of the available tool set. Extending the agent to operate over a broader and more heterogeneous collection of real-world tools would substantially enlarge the action space and increase planning complexity. Second, while MMSearchVQA facilitates retrieval-centric multimodal training, the constructed queries are predominantly characterized by relatively long reasoning trajectories compared to those in existing training corpora. More complex scenarios that require substantially deeper multi-step search and decision-making processes remain underexplored. Extending the dataset construction pipeline therefore represents an important direction for future research.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923.

Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. 2023. Can pre-trained vision and language models answer visual information-seeking questions? arXiv preprint arXiv:2302.11713.

Xianfu Cheng, Wei Zhang, Shiwei Zhang, Jian Yang, Xiangyuan Guan, Xianjie Wu, Xiang Li, Ge Zhang, Jiaheng Liu, Yuying Mai, and 1 others. 2025. Simplevqa: Multimodal factuality evaluation for multimodal large language models. arXiv preprint arXiv:2502.13059.

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

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 frontiers of vision-language deep research agent. arXiv preprint arXiv:2508.05748.

Google. 2025. Gemini deep research system card.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948.

Wenbo Hu, Jia-Chen Gu, Zi-Yi Dou, Mohsen Fayyaz, Pan Lu, Kai-Wei Chang, and Nanyun Peng. 2024. Mrag-bench: Vision-centric evaluation for retrieval-augmented multimodal models. arXiv preprint arXiv:2410.08182.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. arXiv preprint arXiv:2412.16720.

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

Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. s3: You don't need that much data to train a search agent via rl. arXiv preprint arXiv:2505.14146.

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.

Damjan Kalajdzievski. 2024. Scaling laws for forgetting when fine-tuning large language models. arXiv preprint arXiv:2401.05605.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024a. Revisiting catastrophic forgetting in large language model tuning. arXiv preprint arXiv:2406.04836.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, and 1 others. 2025a. Websailor: Navigating super-human reasoning for web agent. arXiv preprint arXiv:2507.02592.

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

Yangning Li, Yinghui Li, Xinyu Wang, Yong Jiang, Zhen Zhang, Xinran Zheng, Hui Wang, Hai-Tao Zheng, Pengjun Xie, Philip S. Yu, Fei Huang, and Jingren Zhou. 2024b. Benchmarking multimodal retrieval augmented generation with dynamic vqa dataset and self-adaptive planning agent.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437.

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

OpenAI. 2025. Openai deep research system card. OpenAI Blog.

Perplexity. 2025. Perplexity deep research system card.

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, and 1 others. 2025. Reasonir: Training retrievers for reasoning tasks. arXiv preprint arXiv:2504.20595.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. arXiv preprint arXiv: 2409.19256.

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.

Duong T Tran, Trung-Kien Tran, Manfred Hauswirth, and Danh Le Phuoc. 2025. Reasonvqa: A multi-hop reasoning benchmark with structural knowledge for visual question answering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 18793–18803.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533.

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.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, and 1 others. 2025a. Webdancer: Towards autonomous information seeking agency. arXiv preprint arXiv:2505.22648.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. arXiv preprint arXiv:2505.09388.

Xiaohan Yu, Zhihan Yang, and Chong Chen. 2025. Unveiling the potential of multimodal retrieval augmented generation with planning. arXiv preprint arXiv:2501.15470.

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

A Implementation Details

A.1 RAG

For the RAG baseline, we adopt the prompt design and processing pipeline proposed in (Wu et al., 2025b). Specifically, we first perform image retrieval using the Serper API, followed by query refinement based on the image search context. The refined query is then used to conduct a subsequent text search, and the final response is generated using the combined information. It is worth noting that our implementation differs slightly from (Wu et al., 2025b) in that we do not utilize the Jina API to fetch the full webpage content.

A.2 OmniSearch

For the OmniSearch baseline, we leverage the publicly available implementation[4] and adapt the search tool interface to match our experimental setup. Apart from this minor adjustment, we retain the original workflow and prompt design to ensure a fair comparison.

A.3 CogPlanner

For CogPlanner, we develop a multi-agent planning framework built upon the `llama-index` library[5]. This implementation integrates dynamic query reformulation and retrieval strategy selection to facilitate efficient multimodal information synthesis.

A.4 MMSearch-R1

We adopt two implementation strategies. First, we build upon the publicly available codes[6] and substitute the original image and text search components with our custom-built tools. However, under

[4] https://github.com/Alibaba-NLP/OmniSearch

[5] https://github.com/run-llama/llama_index

[6] https://github.com/EvolvingLMMs-Lab/multimodal-search-r1

this setting, the model's tool invocation frequency rapidly diminishes to nearly zero. To ensure a fair and stable evaluation, we additionally employ the released model checkpoint[7] for our experiments.

B Prompts

Multimodal RAG Rollout

You are an expert in information seeking and reasoning. You will be given a question with a image. You need to collect information for the question step by step.
Follow these instructions carefully:
1. If you need external knowledge, call search tools.
2. Enclose your entire reasoning process within <think> ... </think> tags.
3. If you find no further external knowledge needed, stop the search process and call the answer model tool.

Answer Reward

You are an expert evaluator. You will be given:
- A question
- Several correct (golden) answer candidates
- My provided answer

Your task:
Strictly judge whether my answer is correct compared to the golden answers.

Judgement rules:
1. The meaning of my answer **must match** one of the golden answer candidates.
2. Reject or fail to answer is wrong answer.

Output format:
<reason> The reason of judgement. <Judgement> Yes or No.

Question: {question}
Golden Answer: {cand_ans}
My Answer: {gen}

[7] https://huggingface.co/lmms-lab/MMSearch-R1-7B