

MMSearch-R1: Incentivizing LMMs to Search

Jinming Wu^{1,*}, Zihao Deng^{1,*}, Wei Li¹, Yiding Liu¹, Bo You¹,
Bo Li², Zejun Ma¹, Ziwei Liu²

¹ByteDance, ²S-Lab, NTU

 [Code](#)  [Model & Data](#)

Abstract

Robust deployment of large multimodal models (LMMs) in real-world scenarios requires access to external knowledge sources, given the complexity and dynamic nature of real-world information. Existing approaches such as retrieval-augmented generation (RAG) and prompt engineered search agents rely on rigid pipelines, often leading to inefficient or excessive search behaviors. We present **MMSearch-R1**, the first end-to-end reinforcement learning framework that enables LMMs to perform **on-demand**, multi-turn search in real-world Internet environments. Our framework integrates both image and text search tools, allowing the model to reason about when and how to invoke them guided by an outcome-based reward with a search penalty. To support training, We collect a multimodal search VQA dataset through a semi-automated pipeline that covers diverse visual and textual knowledge needs and curate a search-balanced subset with both search-required and search-free samples, which proves essential for shaping efficient and on-demand search behavior. Extensive experiments on knowledge-intensive and info-seeking VQA tasks show that our model not only outperforms traditional RAG-based baselines of the same model size, but also matches the performance of a larger RAG-based model while reducing search calls by over **30%**. We further analyze key empirical findings to offer actionable insights for advancing research in multimodal search.

1 Introduction

Scaling up visual-text pair data by leveraging large-scale, high-quality, and diverse datasets across different training stages (Radford et al., 2021; Awadalla et al., 2024; Zhang et al., 2025; Wang et al., 2025; Dong et al., 2025) has become a fundamental paradigm for acquiring grounded knowledge of the visual world, driving breakthroughs in

Large Multimodal Models (LMMs) (Hurst et al., 2024; Team et al., 2023, 2025b; Bai et al., 2025; Li et al., 2024a,b; Lin et al., 2024; Chen et al., 2024b). However, this paradigm faces inherent limitations when handling complex and dynamic real-world knowledge (Kandpal et al., 2023; Chang et al., 2024). Specifically, long-tail information such as facts that emerge after the model’s training cut-off or domain-specific knowledge constrained by privacy, copyright, or security considerations is difficult to capture through static training alone. As a result, leading LMMs continue to struggle with knowledge-intensive and information-seeking VQA tasks where external and up-to-date knowledge is often required (Li et al., 2023b; Jiang et al., 2024; Fu et al., 2025). When confronted with inputs beyond their internal knowledge boundaries, such as unfamiliar visual content or previously unseen textual information, models are prone to hallucinations (Liu et al., 2024a; Li et al., 2023a), which severely compromise their reliability in applications that demand factual accuracy and trustworthiness (Cui et al., 2023).

Integrating the ability to interact with search tools into LMMs to search for external information offers a promising solution to these limitations (Jiang et al., 2024; Hu et al., 2023a; Zhang et al., 2024d). Existing approaches can be broadly categorized into two paradigms: (1) Retrieval-Augmented Generation (RAG) (Chen et al., 2022; Yu et al., 2024a; Chen et al., 2024a), which retrieves external visual or textual information to guide model generation; and (2) Prompt-engineered agents (Hu et al., 2023a; Zhang et al., 2024d; Jiang et al., 2024), which prompt large models to iteratively perform web searches and reason over the retrieved results to generate answers. However, these approaches remain suboptimal in practice. RAG-based methods follow a fixed retrieve-then-generate workflow grounded in static knowledge bases, often resulting in over-retrieval, high computational cost, and the

*Equal contribution.

unrealistic assumption that all required information is already present in the corpus. Such a controlled setting fails to capture the dynamic and unpredictable nature of real-world scenarios, rendering these systems vulnerable in practical deployments. On the other hand, prompt-engineered agents interact with real-world search engines, but the model parameters are not optimized through learning. As a result, the models do not truly learn how to interact effectively with search tools or adapt their behavior to open world environments. This motivates the development of methods that teach models to search on demand and interact effectively with search tools, ensuring practical usability in dynamic real-world settings.

Recent advances, such as OpenAI’s o series (Jaech et al., 2024; OpenAI, b) and DeepSeek-R1 (Guo et al., 2025), have highlighted the potential of end-to-end reinforcement learning (RL) to enhance the reasoning capabilities of large-scale models. In addition, OpenAI introduced the Deep Research (OpenAI, a), claiming that training models via end-to-end RL to interact with search tools and web content can significantly improve their ability to solve complex open-ended tasks that require iterative reasoning and information seeking. In the open-source community, efforts such as DeepResearcher (Zheng et al., 2025), Search-R1 (Jin et al., 2025), and ReSearch (Chen et al., 2025) have followed this direction, applying end-to-end RL to improve models’ abilities in multiturn search and retrieval-augmented generation, aiming to boost performance on information-seeking question answering tasks. However, existing work mainly focuses on text-based search, while in the multimodal domain, challenges such as constructing suitable training data and designing RL frameworks to incentivize models to perform search in real-world environments remain underexplored.

In this work, we focus on training LMMs to learn three key search-related abilities: (1) *when to search*, (2) *what to search for*, and (3) *how to reason over search results to answer user queries*. By exploring these questions, we propose **MMSearch-R1**, the first end-to-end RL-based solution designed to equip LMMs with the capability to perform search on demand in real-world environments. Our efforts can be summarized as follows:

Datasets Construction We propose an semi-automated method for constructing a multimodal search VQA dataset by estimating the model’s familiarity with the visual and textual knowledge

required to answer each question. and curate a mixture of search-required and search-free VQA samples, which is crucial for shaping the model’s ability to perform on-demand search.

Multimodal Search Tool Integration We build a real-world search pipeline consisting of two tools: an image search tool, which allows the model to retrieve webpage thumbnails and titles related to a user-provided image to help identify unfamiliar visual content; and a text search tool, which enables the model to issue precise textual queries to retrieve relevant webpages and acquire textual knowledge.

Better Performance through Wiser Search We demonstrate that by incorporating an outcome-based reward with search penalty, the GRPO algorithm (Shao et al., 2024) can be directly applied to LMMs without cold-start initialization. This setup encourages models to perceive their knowledge boundaries and then perform search when necessary. As a result, the model learns to reason about when and how to execute multi-turn, complex search strategies. In knowledge-intensive and information-seeking VQA tasks, MMSearch-R1-7B not only outperforms RAG-based baselines of the same model size, but also achieves competitive performance compared to a 32B RAG-based model, while significantly reducing the number of search calls by over **30%**.

2 RL Training with Multimodal Search

2.1 Group Relative Policy Optimization

As shown in Figure 1, we adopt standard GRPO as our base RL algorithm, with modifications to allow search interactions with the real-world environment during the rollout process. Originally introduced in DeepSeekMath (Shao et al., 2024), GRPO is a variant of the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017). Unlike PPO, GRPO estimates the baseline directly from a group of rewards, without relying on a value function, which significantly reduces the computational burden during training. The details of the GRPO algorithm are provided in the Appendix D.

2.2 Multimodal Search Tools

A fully functional search toolkit is crucial for solving information-seeking VQA tasks. As illustrated in the bottom part of Figure 1, we equip the model with two types of search tools for interacting with real-world internet content. The first is an image

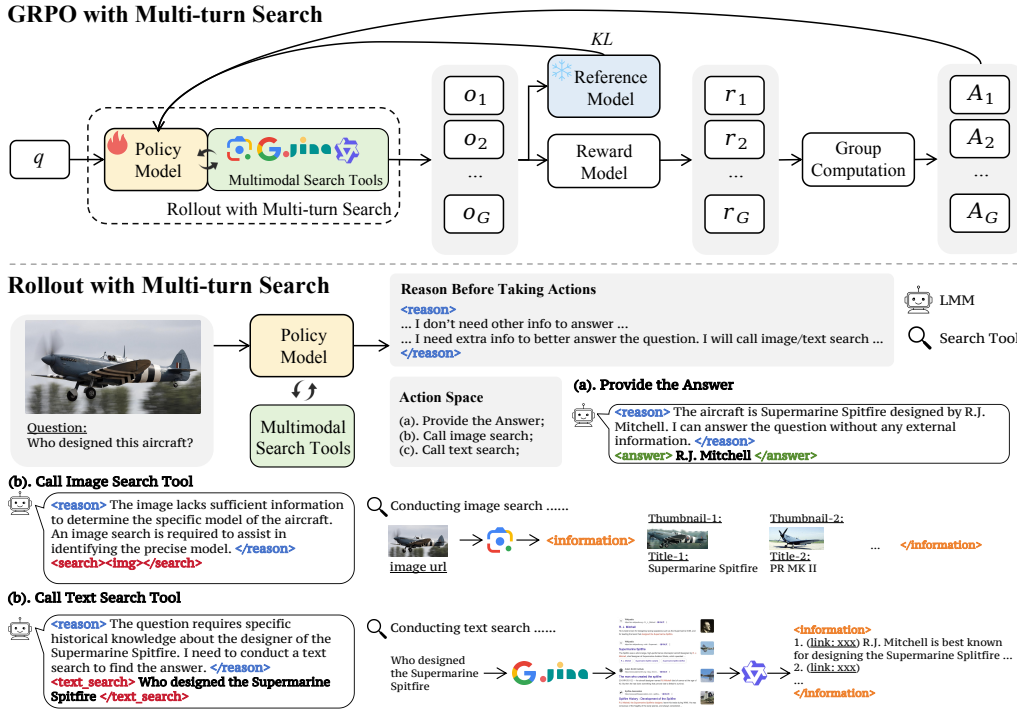


Figure 1: Illustration of training in MMSearch-R1. Top: The GRPO training pipeline integrated with multimodal search tools. Bottom: A detailed view of the rollout process and search tool execution.

search tool powered by SerpApi.¹ The model submits the image from the original question to the image search engine, which returns the top-5 visually matched webpages in an interleaved format, each represented by a thumbnail and a title. This helps the model identify the key visual entities present in the input image. The second is a text search pipeline composed of SerpApi, Jina Reader², and a webpage summarizer. The model autonomously generates a text query related to the original question and submits it to the search tool to retrieve relevant information. SerpApi returns the top-5 webpage URLs associated with the query. Jina Reader then fetches the full content of each webpage and converts it into a clean, readable format. A summarization model (Qwen3-32B (Qwen Team) in our implementation) processes each page and extracts the content most relevant to the user’s question into a concise summary. Implementation details can be found in the Appendix E.

2.3 Rollout with Multi-turn Search

To address complex information-seeking tasks, the model may engage in multiple rounds of interaction with the internet environment before providing a final answer. As illustrated in the bottom part

of Figure 1, the rollout process is multi-turn and iterative. We structure the prompt to guide the model to perform reasoning within the `<reason>` and `</reason>` tags whenever new information is received, including the original question or any retrieved search results, and then choose an action from a predefined action space (see Appendix F.2 for the full prompt). If the model decides to answer the question, it is required to place its response between `<answer>` and `</answer>`. If it chooses to invoke the image search tool, it should append `<search></search>` at the end of its response. For text search, the model should autonomously generate a natural language query and append it between `<text_search>` and `</text_search>`. All retrieved information is returned to the model enclosed within `<information>` and `</information>`, and is fed into the next round of dialogue. This iterative process continues until the model provides a final answer or reaches the maximum number of allowed turns. To prevent training bias from environment feedback, retrieved content from search tools is masked during loss computation and does not contribute to gradient updates.

2.4 Reward Modeling

Reward modeling plays a critical role in RL training, as it encodes the desired model behavior

¹SerpApi: <https://serpapi.com/>

²Jina Reader: <https://jina.ai/reader/>

and directly guides the optimization process. In MMSearch-R1, the reward consists of two components: an accuracy score with search penalty and a format score.

Accuracy Score with Search Penalty We use the exact string match to evaluate whether the final answer provided by the model is consistent with the ground truth. If the match is exact, the accuracy score is 1; otherwise, it is 0. For correct answers, we further check whether the model relied on search tools to arrive at the answer. A penalty factor (ranging from 0 to 1) is applied to the accuracy score when any search was performed. This design encourages the model to first exploit its internal knowledge and invoke search tools only when necessary, thereby shaping on-demand search behavior.

Format Score This component checks whether the multi-turn responses of the model strictly follow the predefined prompt format. Specifically, the model is required to reason between `<reason>` and `</reason>` before taking any action, take only one action per turn, place search patterns such as `<search></search>` or `<text_search> text query </text_search>` at the end of its response when a search is performed, and wrap the final answer between `<answer>` and `</answer>` when concluding. The format score is assigned a value of 1 only if all responses fully adhere to these formatting requirements; otherwise, it is set to 0. We combine the two components using a weighting coefficient α , and define the final reward as a weighted sum of the accuracy and format scores as follows:

$$\begin{aligned} \text{reward} = & \alpha \cdot \text{Acc_Score} \cdot \text{Search_Penalty} \\ & + (1 - \alpha) \cdot \text{Format_Score} \end{aligned} \quad (1)$$

3 Curating Search-balanced Datasets

We aim to explore the potential of training models for on-demand search using simple outcome-based reward reinforcement learning. Therefore, the datasets should satisfy three criteria: **(1). Coverage of Both Search-Required and Search-Free Questions** The datasets should include a mix of search-free and search-required questions. A search-free question can be answered solely using the model’s internal knowledge, whereas a search-required question involves information beyond the model’s existing knowledge and therefore requires access to external information sources.

Search-required questions can be further categorized into Visual Knowledge-required and Textual Knowledge-required types. Visual Knowledge refers to the model’s ability to recognize visual entities in an image (e.g., "What is the model of the aircraft shown in the image?"), while Textual Knowledge refers to factual information about the visual entity (e.g., "Who is the designer of Supermarine Spitfire?"). Existing datasets such as OK-VQA (Marino et al., 2019) and InfoSeek (Chen et al., 2023) typically use image sources already seen by current multimodal models and tend to focus on Textual Knowledge-required questions. Meanwhile, MMSearch (Jiang et al., 2024) collects images from recent news, aligning better with the Visual Knowledge-required criteria. However, it is relatively small in scale and mainly serve as benchmarks, leading to a lack of sufficient Visual Knowledge-required training data. **(2). Concise and Unambiguous Answers for Reliable Verification** VQA questions should be designed to elicit concise and unambiguous answers that can be easily and reliably verified through simple, rule-based reward mechanisms. Such questions typically revolve around factual knowledge, enabling automated evaluation during reinforcement learning. **(3). Diversity in Knowledge Categories and Question Difficulty** The dataset should cover diverse knowledge domains and difficulty levels to ensure broad real-world generalization.

To support our investigation, we construct a multimodal search VQA dataset, FactualVQA (FVQA), using a combination of automated pipelines and manual annotation, covering both training and evaluation needs, as illustrated in Figure 2(c).

3.1 Training Dataset Construction

To meet the above requirements, we select suitable training data from multiple sources through two main processes: VQA Collection and Search Balancing.

VQA Collection We first developed an automated annotation pipeline to collect Visual Knowledge-required training data, as illustrated in Figure 2(a). To explore visual concepts that models are relatively familiar or unfamiliar with, we begin with the Metadata distribution of MetaCLIP (Xu et al., 2023). This Metadata is constructed from multiple sources such as WordNet and Wikipedia, covering a wide range of Visual Concepts (VC) from common to rare. Designed to guide balanced dataset construction for vision-language pretrain-

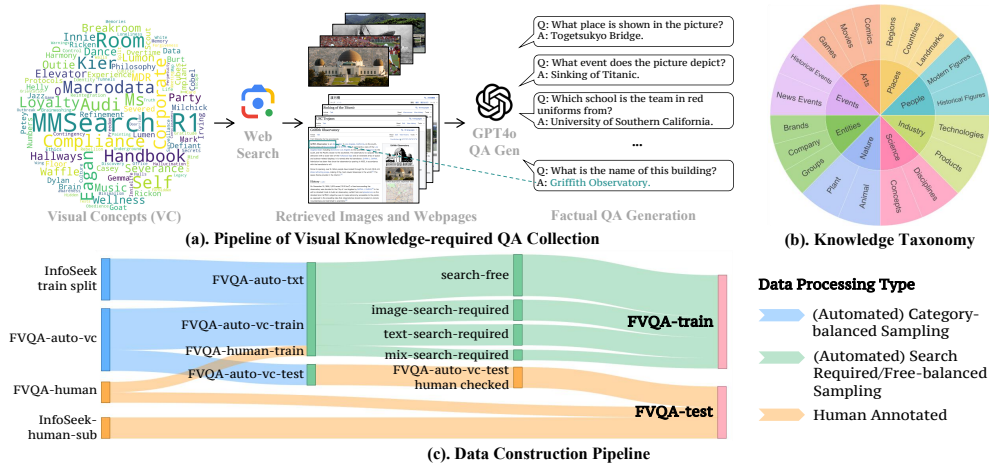


Figure 2: Illustration of data construction process of FVQA: (a). An automated pipeline for visual knowledge-required VQA samples collection; (b). Knowledge taxonomy; (c). Overall pipeline showing the composition and origin of FVQA from various automatic and manually curated sources.

ing, it exhibits a long-tailed distribution. Concepts in the head of the distribution correspond to commonly referenced items in the real world (e.g., car, tree), whereas those in the tail represent less common or niche concepts (e.g., thylacine, astrolabe). We randomly sampled 10,000 visual concepts from both the head and the tail of the Metadata distribution. For each concept, we performed a web search to retrieve the most relevant image and its associated webpage. These image–webpage pairs were then input to GPT-4o, which was prompted to generate a factual visual question-answer (VQA) pair centered around the given visual concept. The generated answers were required to be concise and precise. The prompts used for QA generation and two examples can be found in Appendix F.1 and J. Next, GPT-4o was employed to classify the knowledge type assessed by each question, resulting in a knowledge taxonomy as illustrated in Figure 2(b). Based on this taxonomy, we performed balanced sampling across categories and curated a final set of 6,000 VQA samples, referred to as FVQA–auto–vc, with 5,400 used for training and 600 for testing. In parallel, to enrich the dataset with textual knowledge-required examples, we sampled from the open-source InfoSeek (Chen et al., 2023) dataset. Specifically, we classified the questions in the InfoSeek training split by the type of knowledge they require and mapped them into the same taxonomy. After balanced sampling across categories, we obtained 7,000 samples, referred to as FVQA–auto–txt. Finally, to further diversify the dataset with real-user queries, we manually annotated an additional 800 samples, referred to as

FVQA–manual–train. Annotators were instructed to select a knowledge category from the taxonomy, locate a relevant image, and pose a factual question pertaining to that category. They were allowed to perform both visual and textual searches until sufficient information was gathered to formulate an accurate answer, from which a concise and precise response was extracted.

Search Balancing The goal of this stage is to distinguish between search-required and search-free questions within the collected data. To this end, we first trained a Qwen2.5-VL-Instruct-7B model using our training framework on the full set of samples obtained from the VQA collection process. This model was then used to classify the original questions: for each question, 8 rollouts were generated. If all 8 rollouts failed, the question was discarded due to insufficient training signal. A question was labeled as image-, text-, or mixed-search-required if the model produced correct answers only when the corresponding type of search behavior was performed. In particular, mixed-search indicates that both image and text searches needed to be executed for the model to answer correctly. If any rollout produced a correct answer without invoking search, the question was labeled as search-free. Finally, we constructed a search-balanced training set of 5,000 samples, referred to as FVQA–train, consisting of approximately 3,400 search-required and 1,600 search-free VQA examples. Maintaining a balanced distribution of search types is crucial to shaping the search behavior of the model during training.

3.2 Test Dataset Annotation

To better evaluate the model’s performance, we additionally constructed a high-quality test set, where all examples were either manually verified or fully human-annotated to ensure accuracy. The test set, referred to as FVQA-test, includes 1800 examples collected from three sources: (1) 600 samples drawn from FVQA-auto-vc, ensuring no overlap with the training set, with each example manually checked for correctness; (2) 600 samples selected from the InfoSeek Human Split, where answers were manually annotated, as the original human-labeled answers were not publicly available; (3) 600 samples collected directly from the manual annotation process described above.

4 Experiments

4.1 Setups

Implementation Details We built our training framework based on veRL (Sheng et al., 2024) and conduct experiments on Qwen2.5-VL-7B-Instruct (Bai et al., 2025). The model was trained using the dataset described in Section 3. At each training step, we sampled 512 examples, with each example undergoing 8 rollouts. Each rollout consists of up to three rounds of dialogue, during which the model can perform at most two search actions and is required to produce a final answer in the third round. Image search is only allowed in the first round, and each image search returns up to 5 top visual matched webpages in the form of interleaved thumbnails and titles. Text search, on the other hand, returns up to 5 summarized webpage contents per query. The search penalty factor is set to 0.9. The weighting coefficient α between the accuracy reward and the format reward is set to 0.1. Further details on the hyperparameter settings can be found in Appendix H.1.

Benchmark We selected FVQA-test, InfoSeek (Chen et al., 2023), MMSearch (Jiang et al., 2024), SimpleVQA (Cheng et al., 2025) and LiveVQA (Fu et al., 2025) as benchmark datasets to evaluate the model’s ability to handle both knowledge-intensive and information-seeking VQA tasks. Specifically, for InfoSeek, we randomly sampled 2k examples from its test split due to the large dataset size. For MMSearch, we filter and retain only the QA pairs that include images. For SimpleVQA, we extracted all QA examples written in English. Among these benchmarks, MMSearch, SimpleVQA and LiveVQA serve as out-of-distribution (OOD) test-

sets for our trained models. Details of the benchmark datasets are provided in Appendix G.

Baselines To validate the effectiveness of MMSearch-R1, we evaluated against both closed-source models (GPT-4o and Gemini 2.5 Pro) and open-source models from the Qwen2.5-VL series. The models are tasked with solving VQA problems in two different workflows. (1) Prompt-based Search Workflow: In this setup, the model is required to produce Thought and Action sequentially in each dialogue turn based on the Observation, following the standard ReAct (Yao et al., 2023) paradigm under the same prompt as RL training. (2) Answer under RAG Workflow: In this workflow, models are required to perform exactly two search operations using our multimodal search tools for each VQA example, first performing an image search and then a text search. Specifically, given an input image and question, the model is provided with the image search results and the original question in the first round and is prompted to generate a text query to assist in answering. In the second round, the retrieved results based on the text query are fed into the model, and the model is asked to produce the final answer. Under a fixed budget of search steps, the RAG workflow typically exposes the model to more external information compared to the react paradigm and the on-demand search strategy. The prompts used for the RAG workflow are provided in Appendix F.5.

Metric We adopt the LLM-as-Judge as our evaluation framework to assess the accuracy of model responses. We record the search ratio, the proportion of responses that invoke a search, allowing us to jointly evaluate answer correctness and search efficiency. We employ GPT-4o as the judging model, which receives the original image, the question, the ground-truth answer, and the model’s response to determine correctness. The prompt used for judgment is detailed in the Appendix F.6.

4.2 Main Results

In this section, we analyze the key results that emerged from our experiments.

Result 1: RL training enables models to better recognize the boundaries of their knowledge and perform on-demand search more effectively. As shown in Table 1, We attribute this deficiency to the model’s inadequate instruction-following capabilities for the specific search-triggering prompts, a result of insufficient adaptation during its training

Table 1: Performance of MMSearch-R1 across benchmarks. "Acc (%)" denotes the accuracy evaluated by LLM-as-Judge, while "SR (%)" represents the search ratio, defined as the percentage of total search calls made relative to the maximum allowed search steps for each method.

Model	Average		In-Domain				Out-of-Domain					
	Acc	SR	FVQA-test		InfoSeek		MMSearch		SimpleVQA		LiveVQA	
			Acc	SR	Acc	SR	Acc	SR	Acc	SR	Acc	SR
GPT4o *RAG	62.1	100	66.0	100	59.1	100	62.5	100	63.4	100	59.6	100
Gemini 2.5 Pro *RAG	61.8	100	66.1	100	56.7	100	62.5	100	65.9	100	57.8	100
Qwen2.5-VL-32B *RAG	55.1	100	57.0	100	56.8	100	57.9	100	54.5	100	49.6	100
Qwen2.5-VL-7B *RAG	51.6	100	52.9	100	53.7	100	52.2	100	51.6	100	48.0	100
GPT4o *ReAct	55.8	65.2	58.2	66.8	54.4	68.1	55.8	78.5	57.9	42.2	52.9	70.4
Gemini 2.5 Pro *ReAct	56.5	63.6	56.1	65.2	55.9	62.4	57.2	77.1	60.2	40.4	53.4	72.8
Qwen2.5-VL-32B *ReAct	47.9	67.3	49.6	68.8	50.1	77.2	44.4	72.5	56.0	52.5	39.2	65.4
Qwen2.5-VL-7B *ReAct	43.1	51.9	46.4	55.0	43.8	61.0	40.4	62.0	50.6	38.1	34.2	43.4
MMSearch-R1-7B	54.6	67.1	58.4	66.8	55.1	61.6	53.8	88.5	57.4	42.5	48.4	76.2

phase. This is further evidenced by its significantly lower search rate, which indicates that the model fails to initiate search actions in many instances where they are necessary. On both in-domain and out-of-domain test sets, MMSearch-R1-7B outperforms the RAG-based counterparts of the same size by an average of **3%** in accuracy, while reducing the average search rate by **32.9%**. This indicates that our RL-trained model not only achieves higher correctness but also relies less on external information, thereby exhibiting a more efficient and targeted use of search. Notably, MMSearch-R1-7B performs competitively with RAG-based Qwen2.5-VL-32B, a significantly larger model, further highlighting the benefits of learning adaptive search behavior, as opposed to executing fixed-stage retrieval regardless of necessity. These results validate the effectiveness of outcome-based RL in enabling intelligent, cost-aware search behavior.

Result 2: RL training enhances the model’s ability to generate effective text queries and summarize retrieved information. Our RL training jointly equips model with three key capabilities: deciding whether to search, determining what to search for, and extracting useful information from retrieved results. To evaluate the latter two abilities in isolation, we adopt the same RAG setting as in the baseline comparison, where both image and text search are executed for every question. In this setup, the search process is fixed, and the model is responsible only for generating effective text queries and reasoning over the retrieved content. This removes variability in search triggering and allows us to focus on evaluating how well the model interacts with external information.

As shown in Figure 3(a), MMSearch-R1-7B

demonstrates consistent improvements over the base model across both in-domain and out-of-domain tasks. These results indicate that reinforcement learning improves not only the decision of when to search, as shown earlier, but also enhances two core retrieval abilities: generating more accurate, contextually relevant queries and extracting useful information to support accurate answers. These gains appear across both image and text retrieval, highlighting the broader value of RL in strengthening retrieval and reasoning capabilities.

Result 3: RL improves the model’s ability to utilize its internal knowledge. As shown in Figure 3(b), we conduct a behavioral breakdown of responses on datasets InfoSeek and SimpleVQA to better understand how the model’s behavior changes after reinforcement learning. The results reveals that a clear upward trend in the *Correct without Search* proportion from the base model to the RL-trained model. These gains indicate that the RL-trained model can answer substantially more questions correctly without invoking the search tool, demonstrating improved recall and reasoning based on its internal knowledge. This shift suggests that reinforcement learning enhances the model’s ability to rely on its own parameters when sufficient, and to reserve external search for genuinely novel or long-tail queries. As a result, the model exhibits more accurate on-demand search behavior, engaging external tools only when internal knowledge is insufficient.

4.3 Ablation Study

Sensitivity Analysis of Reward Components: Search Penalty and Score Weight As shown in Figure 4(a), we performed a sensitivity analysis

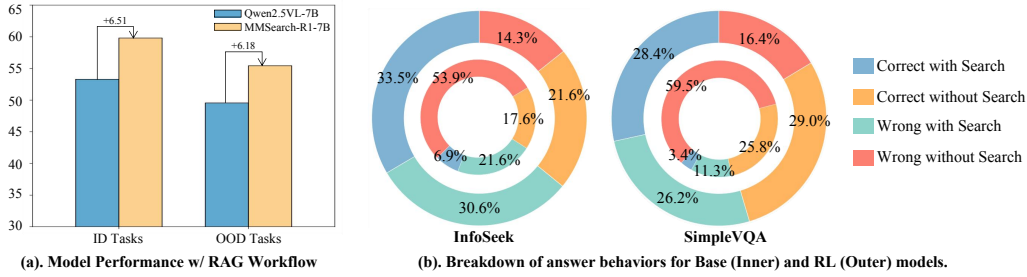


Figure 3: (a). Performance comparison between the Base model and the RL-trained model under the RAG workflow. (b). Answer behavior breakdown of Base (inner circle) and RL (outer circle) models in InfoSeek and SimpleVQA.

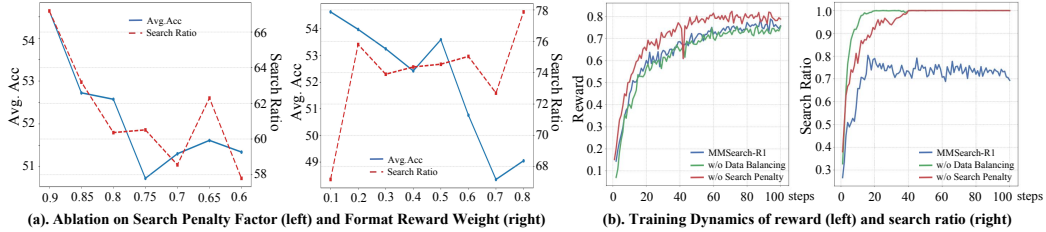


Figure 4: (a). Ablation study on search penalty factor and format reward weight. (b). Training dynamics of reward and search ratio for different strategies

on the search penalty and the format score weight, evaluating their impact on training stability and model performance. For the subsequent ablation studies, these parameters were held constant at 0.9 and 0.1, respectively. Moderate search penalties are optimal for convergence. Excessive penalties suppress searching and harm accuracy, necessitating a balance between performance and search efficiency tailored to specific applications. An overly high weight of format reward leads to suboptimal search (high search ratio, low accuracy) and hinders convergence. We recommend a minimal weight, as format adherence is a trivial task for well-sized models, allowing them to focus on developing effective search strategies.

Analysis of Search Penalty and Data Balancing Mechanisms Figure 4(b) illustrates the training dynamics of reward and search ratio during reinforcement learning. Removing either the search penalty or data balancing leads to distinct trade-offs. Although both ablated variants achieve slightly higher rewards, they do so at the cost of overusing the search tool, with search ratios rapidly converging to nearly 100%. In contrast, MMSearch-R1, trained with both data balancing and a search penalty, achieves a comparable reward while maintaining a significantly lower and more stable search ratio. This suggests that the model has learned to invoke the search tool only when necessary, enabling more efficient on-demand search behavior.

4.4 Other Analysis

We conducted further analyses of the behavior and performance of our model, which revealed several interesting emergent capabilities of MMSearch-R1. These additional studies, presented in the Appendix, suggest that MMSearch-R1 not only benefits from search at inference time, but also develops more generalizable behaviors through training. Specifically, we find that: (1) the model can internalize its search capability in a way that improves performance on more general multimodal tasks, such as producing image captions enriched with factual details beyond purely visual descriptions (Appx. B.2); (2) it exhibits a notable ability to resolve conflicts in retrieved evidence, cross-referencing visual features to identify credible information from conflicting image search results, while also leveraging a voting-like mechanism to infer consistent facts from multiple conflicting textual sources (Appx. C); (3) despite some degradation in performance, the model remains reasonably robust to the noise that is commonly present in real-world search results, indicating a degree of resilience under imperfect retrieval conditions (Appx. I.3); and (4) compared with a standard RAG workflow, MMSearch-R1 achieves similar token consumption while exhibiting lower end-to-end inference latency, suggesting a favorable efficiency profile in practical deployment settings (Appx. I.2).

5 Related Work

5.1 Search-Augmented Agents

Recent work has explored search-augmented paradigms that move beyond static corpora by enabling models to interact with real-time information sources. WebGPT (Nakano et al., 2021) is among the earliest systems to demonstrate this idea at scale: it augments a language model with Bing search results and trains it with human feedback to produce more factual, citeable answers. Toolformer (Schick et al., 2023) introduces a self-supervised framework where the model learns when and how to invoke search tools, by generating API call demonstrations and fine-tuning on helpful examples. This enables efficient and context-aware tool use without heavy supervision. SAIL (Luo et al., 2023) integrates web search into instruction tuning by pairing prompts with retrieved results, teaching models to identify relevant evidence, filter noise, and perform multi-hop reasoning. Although these methods demonstrate the promise of dynamic, tool-augmented reasoning, they still depend heavily on high-quality annotated data.

As language models continue to improve in knowledge retention and reasoning, a parallel line of training-free, prompt-based methods has emerged. These approaches use large models as agents guided by structured prompts. Recent studies (Zheng et al., 2025; Alzubi et al., 2025) incorporate web search directly into the reasoning process by decomposing complex queries into sub-tasks and selectively invoking external tools. This trend has also extended to the multimodal setting. AVIS (Hu et al., 2023a) uses a language model as a controller that iteratively issues vision-language queries to gather external visual and textual evidence for answering complex visual questions. VSA (Zhang et al., 2024d) equips vision-language models with retrieval capabilities, allowing them to proactively search for and integrate visual evidence from large-scale corpora. MMSearch (Jiang et al., 2024) further introduces an agentic pipeline with re-querying, reranking, and summarization to support multimodal information synthesis. Together, these works highlight the growing promise of interactive, tool-augmented multimodal reasoning. However, because the underlying models are typically not trained to use search tools through supervised learning or reinforcement learning, their interactions with external tools can remain suboptimal and unreliable in noisy real-world settings.

5.2 Reinforcement Learning-powered Search Agents

Recent advances such as OpenAI’s o series (Jaech et al., 2024; OpenAI, b), DeepSeek-R1 (Guo et al., 2025), and Kimi-K1.5 (Team et al., 2025a) have demonstrated the potential of end-to-end reinforcement learning (RL) for improving the reasoning abilities of large-scale models. Building on this momentum, organizations including OpenAI, Google, and Perplexity have introduced Deep Research agents (OpenAI, a; Claude; Google; Perplexity), which combine large reasoning models with real-time web search to complete open-ended research tasks such as report drafting and multi-source synthesis. Notably, OpenAI has reported training a highly capable Deep Research model through end-to-end RL, demonstrating the practicality of this approach in real-world applications.

Open-source efforts such as DeepResearcher (Zheng et al., 2025), Search-R1 (Jin et al., 2025), and ReSearch (Chen et al., 2025) have followed a similar direction, using end-to-end RL to improve multi-turn search and retrieval-augmented generation for information-seeking question answering. However, current research remains largely centered on text-based search. In the multimodal setting, key challenges—such as constructing suitable training data and designing RL frameworks that encourage effective search behavior in real-world environments—remain insufficiently explored. Concurrent with our work, efforts such as WebWatcher (Geng et al., 2025) and MM-BrowseComp (Li et al., 2025) have begun to address this gap by developing benchmarks for Multimodal Deep Research Agents.

6 Conclusion

In this work, we present MMSearch-R1, a RL-based framework that equips LMMs with the ability to perform on-demand search in real-world internet environments. MMSearch-R1 learns to recognize knowledge gaps, selectively invoke image or text search, and reason over retrieved content. It outperforms same-sized RAG baselines and approaches the performance of larger models while requiring significantly fewer search calls. Our framework, dataset, and findings offer practical insights into training LMMs with real-world interaction capabilities and lay the groundwork for building multimodal agents that are both adaptive and interactive.

Limitations

First, the interaction between the model and external multimodal search tools still has room for improvement in terms of stability and quality. For example, image search currently requires submitting the full image, which may not be optimal for retrieving localized visual content. The text search pipeline is composed of several components, including SerpAPI, Jina Reader, and a summarization model. Each component may introduce potential sources of variability. Specifically, the ranking of result returned by SerpAPI may vary over time; Jina Reader may fail to extract content from certain domains due to restrictions or errors; and the summarizer may generate hallucinations when extracting relevant information. Despite iterative refinement of the pipeline, our monitoring during training revealed an end-to-end failure rate of approximately 0.2% for image search and 1% for text search, where failure is defined as receiving no valid results. These numbers are higher when considering partial failures, such as retrieving fewer than the expected top five results. As training scales up, ensuring stability and consistency in tool outputs remains a nontrivial challenge.

Second, our reward design based on exact string match, while simple and scalable, has limited flexibility. This design is well-suited for short factual questions with unambiguous answers. However, it may penalize answers that are semantically correct but differ slightly in phrasing. This limitation affects the generalization of the reward function to more complex or open-ended QA tasks. Although our dataset focuses on fact-based QA that aligns well with this evaluation strategy, more flexible reward signals could help expand the framework to broader question types and reduce reliance on surface-form matching.

In general, improving the robustness of tool interactions and enhancing the expressiveness of the reward function will be essential for building reliable multimodal reasoning agents.

Acknowledgments

We would like to thank our data expert colleagues, Yifan Jiang, Ziyi Zheng, and Yuping Liu, for their valuable contributions to data annotation and quality checking. Their careful efforts in labeling, verification, and consistency review greatly supported the construction of our dataset and improved its overall quality.

References

- Salaheddin Alzubi, Creston Brooks, Purva Chiniya, Edoardo Contente, Chiara von Gerlach, Lucas Irwin, Yihan Jiang, Arda Kaz, Windsor Nguyen, Sewoong Oh, and 1 others. 2025. Open deep search: Democratizing search with open-source reasoning agents. *arXiv preprint arXiv:2503.20201*.
- Anthropic. Claude 3.5 Sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet/>. Technical Report, 2024.
- 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*.
- Anas Awadalla, Le Xue, Oscar Lo, Manli Shu, Hannah Lee, Etash Guha, Sheng Shen, Mohamed Awadalla, Silvio Savarese, Caiming Xiong, and 1 others. 2024. Mint-1t: Scaling open-source multimodal data by 10x: A multimodal dataset with one trillion tokens. *Advances in Neural Information Processing Systems*.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.
- Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. 2024. How do large language models acquire factual knowledge during pretraining? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Fan Yang, Zenan Zhou, Weipeng Chen, Haofen Wang, Jeff Z Pan, and 1 others. 2025. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*.
- Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W Cohen. 2022. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. *arXiv preprint arXiv:2210.02928*.
- 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*.
- Zhanpeng Chen, Chengjin Xu, Yiyan Qi, and Jian Guo. 2024a. Mllm is a strong reranker: Advancing multimodal retrieval-augmented generation via knowledge-enhanced reranking and noise-injected training. *arXiv preprint arXiv:2407.21439*.
- Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024b. Internvl: Scaling up vision foundation models and aligning

- for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Denvy Deng, and Qi Zhang. 2023. Uprise: Universal prompt retrieval for improving zero-shot evaluation. *arXiv preprint arXiv:2303.08518*.
- 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*.
- Claude. Claude takes research to new places. <https://www.anthropic.com/news/research/>. Technical Report, 2025.
- Chenhang Cui, Yiyang Zhou, Xinyu Yang, Shirley Wu, Linjun Zhang, James Zou, and Huaxiu Yao. 2023. Holistic analysis of hallucination in gpt-4v (ision): Bias and interference challenges. *arXiv preprint arXiv:2311.03287*.
- Hongyuan Dong, Zijian Kang, Weijie Yin, Xiao Liang, Chao Feng, and Jiao Ran. 2025. Scalable vision language model training via high quality data curation. *arXiv preprint arXiv:2501.05952*.
- Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.
- 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 frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*.
- Google. Try Deep Research and our new experimental model in Gemini, your AI assistant. <https://blog.google/products/gemini/google-gemini-deep-research/>. Technical Report, 2025.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Ziniu Hu, Ahmet Iscen, Chen Sun, Kai-Wei Chang, Yizhou Sun, David Ross, Cordelia Schmid, and Alireza Fathi. 2023a. Avis: Autonomous visual information seeking with large language model agent. *Advances in Neural Information Processing Systems*.
- Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A Ross, and Alireza Fathi. 2023b. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.
- 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, Chaoyou Fu, Guanglu Song, and 1 others. 2024. Mm-search: 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*.
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.
- Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*.
- 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*.
- Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.
- Dongxu Li, Yudong Liu, Haoning Wu, Yue Wang, Zhiqi Shen, Bowen Qu, Xinyao Niu, Fan Zhou, Chengen Huang, Yanpeng Li, and 1 others. 2024b. Aria: An open multimodal native mixture-of-experts model. *arXiv preprint arXiv:2410.05993*.
- Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, and 1 others. 2025. Mm-: A comprehensive benchmark for multimodal browsing agents. *arXiv preprint arXiv:2508.13186*.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023a. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.
- Yunxin Li, Longyue Wang, Baotian Hu, Xinyu Chen, Wanqi Zhong, Chenyang Lyu, Wei Wang, and Min Zhang. 2023b. A comprehensive evaluation of gpt-4v on knowledge-intensive visual question answering. *arXiv preprint arXiv:2311.07536*.
- Li, Bo and Zhang, Kaichen and Zhang, Hao and Guo, Dong and Zhang, Renrui and Li, Feng and Zhang, Yuanhan and Liu, Ziwei and Li, Chunyuan. LLaVA-NeXT: Stronger LLMs Supercharge Multimodal Capabilities in the Wild. <https://llava-vl.github.io/blog/2024-05-10-llava-next-stronger-llms/>. Technical Report, 2024.
- Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
- Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024a. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*.
- Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. 2024b. Ocr-bench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.
- Hongyin Luo, Tianhua Zhang, Yung-Sung Chuang, Yuan Gong, Yoon Kim, Xixin Wu, Helen Meng, and James Glass. 2023. Search augmented instruction learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.
- Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*.
- Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.
- Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.
- OpenAI. a. Introducing deep research. <https://openai.com/index/introducing-deep-research/>. Technical Report, 2025.
- OpenAI. b. OpenAI o3 and o4-mini System Card. <https://openai.com/index/o3-o4-mini-system-card/>. Technical Report, 2025.
- Perplexity. Introducing Perplexity Deep Research. <https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research/>. Technical Report, 2025.
- Qwen Team. Qwen3: Think Deeper, Act Faster. <https://qwenlm.github.io/blog/qwen3/>. Technical Report, 2025.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

- Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv preprint arXiv:2305.15294*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseek-math: 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*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025a. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.
- Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, and 1 others. 2025b. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*.
- Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Instructretro: Instruction tuning post retrieval-augmented pretraining. *arXiv preprint arXiv:2310.07713*.
- Huazheng Wang, Jinming Wu, Haifeng Sun, Zixuan Xia, Daixuan Cheng, Jingyu Wang, Qi Qi, and Jianxin Liao. 2024. Mdr: Model-specific demonstration retrieval at inference time for in-context learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4189–4204.
- Xiao Wang, Ibrahim Alabdulmohsin, Daniel Salz, Zhe Li, Keran Rong, and Xiaohua Zhai. 2025. Scaling pre-training to one hundred billion data for vision language models. *arXiv preprint arXiv:2502.07617*.
- Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. 2023. Demystifying clip data. *arXiv preprint arXiv:2309.16671*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.
- Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and 1 others. 2024a. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. *arXiv preprint arXiv:2410.10594*.
- Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiakuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in Neural Information Processing Systems*.
- Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and 1 others. 2024a. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024b. *arXiv preprint arXiv:2407.12772*.
- Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024b. Raft: Adapting language model to domain specific rag. In *First Conference on Language Modeling*.
- Wenqi Zhang, Hang Zhang, Xin Li, Jiashuo Sun, Yongliang Shen, Weiming Lu, Deli Zhao, Yueting Zhuang, and Lidong Bing. 2025. 2.5 years in class: A multimodal textbook for vision-language pretraining. *arXiv preprint arXiv:2501.00958*.
- Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024c. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*.
- Zhixin Zhang, Yiyuan Zhang, Xiaohan Ding, and Xiangyu Yue. 2024d. Vision search assistant: Empower vision-language models as multimodal search engines. *arXiv preprint arXiv:2410.21220*.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.
- 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. *arXiv preprint arXiv:2504.03160*.

A Related Work (Full)

A.1 Large Multimodal Models (LMMs)

The development of Large Multimodal Models (LMMs) marks a significant breakthrough in artificial intelligence, enabling unified processing of textual and visual information that aligns with the inherently multimodal nature of human perception. Recent advances have established a dominant paradigm where scaling up diverse, high-quality vision-text paired datasets across training stages equips LMMs with grounded visual understanding. This approach has yielded remarkable success, as demonstrated by state-of-the-art models including GPT-4o (Hurst et al., 2024), Gemini (Team et al., 2023), Qwen2.5-VL (Bai et al., 2025), the LLaVA series (Liu et al., 2023; Li et al., 2024a; Zhang et al., 2024c), and related systems (Anthropic; Li et al., 2024b; Chen et al., 2024b; Lin et al., 2024), which exhibit exceptional capabilities in visual comprehension, cross-modal reasoning, and instruction following. However, this training methodology inherently yields static knowledge, while real-world information is complex, dynamic, and constantly evolving. As a result, LMMs often struggle with long-tail concepts or newly emerging facts beyond their training cut-off, which can lead to hallucinations and ultimately compromise their stability and reliability in real-world applications. This limitation has led to increasing efforts to augment LMMs with retrieval and search tool-use capabilities.

A.2 Large Models with External Knowledge Access

Given the strong reliance of large models on external knowledge in practical applications, two primary approaches have emerged to enhance their factual reliability and knowledge coverage: retrieval-based methods and search-based methods.

A.2.1 Retrieval-Augmented Generation (RAG)

The RAG paradigm retrieves relevant information from external knowledge bases via dense vector search and incorporates it into the model input to help generate more factually grounded responses. In the field of natural language understanding, Petr Karpukhin et al. (Karpukhin et al., 2020) introduced Dense Passage Retrieval (DPR), which was the first to apply dense retrieval to open-domain question answering. DPR employs a dual-encoder architecture to separately encode queries and doc-

ument passages, enabling efficient semantic-level matching. Building on DPR’s strong retrieval capabilities, Lewis et al. (Lewis et al., 2020) proposed the RAG framework, which integrates pre-trained generative models with nonparametric document indices, forming a unified ‘retrieval-then-generation’ architecture that laid the foundation for subsequent research. As model training and inference techniques have advanced, RAG-based methods have gradually incorporated key techniques such as long-context modeling, chain-of-thought reasoning, self-reflection and document ranking. These enhancements have improved performance across pretraining (Wang et al., 2023), fine-tuning (Izacard et al., 2022; Zhang et al., 2024b; Cheng et al., 2023), and inference (Shao et al., 2023; Asai et al., 2023; Yu et al., 2024b; Wang et al., 2024) stages, significantly boosting RAG’s effectiveness in knowledge-intensive tasks.

The field of visual understanding has also seen notable progress through the adoption of retrieval-augmented methods, particularly in knowledge-intensive VQA tasks. By incorporating external knowledge, including text, images, and structured data, these approaches overcome the limitations of vision-only models that often lack access to broader world knowledge. MuRAG (Chen et al., 2022) integrates both image and text retrieval to enhance open-domain question answering over multimodal data. REVEAL (Hu et al., 2023b) utilizes a multi-source multimodal knowledge memory to augment visual-language pre-training, enabling better handling of knowledge-intensive queries. RagVL (Chen et al., 2024a) introduces a knowledge-enhanced reranking mechanism, improving retrieval precision by filtering out noisy data. VisRAG (Yu et al., 2024a) employs a vision-based approach to process multimodality documents, preserving visual information without relying solely on text extraction. Collectively, these advances have demonstrated the potential of retrieval-augmented generation to bridge visual understanding with external knowledge.

Although RAG methods have shown strong performance in practice, they face several critical limitations, the most notable being their reliance on a static knowledge base. These methods often assume that the required information can be found within the existing corpus. However, in real-world scenarios, information is frequently dynamic and constantly evolving, and the complexity of web-scale environments means that relevant content

may not always be effectively retrieved. This poses significant challenges to the applicability of RAG-based approaches in more general and open-ended settings.

A.2.2 Search-Augmented Approaches

Several recent efforts have explored search-augmented paradigms that go beyond static corpora by enabling models to interact with real-time information sources. In the field of natural language understanding, WebGPT (Nakano et al., 2021) is one of the earliest systems to demonstrate this idea at scale. It augments a language model with access to Bing search results and trains it using human feedback to quote sources and generate more factual, citeable answers. This setup significantly improves factual consistency, especially in domains requiring timely or less common knowledge. Toolformer (Schick et al., 2023) introduces a self-supervised framework where the model learns when and how to invoke external tools, such as search engines, by generating API call demonstrations and fine-tuning on helpful examples. This enables efficient and context-aware tool use without heavy supervision. SAIL (Luo et al., 2023) further integrates web search into instruction tuning. It constructs training examples by pairing user prompts with retrieved search results, teaching the model to select relevant information, filter noise, and perform multi-hop reasoning. While these approaches demonstrate strong potential for dynamic and tool-augmented reasoning, they still rely heavily on high-quality annotated data for training. As language models continue to improve in both knowledge retention and reasoning capabilities, a new line of training-free, prompt-based methods has emerged. These approaches leverage large models as agents orchestrated by structured prompts, without requiring additional fine-tuning. For example, Recent studies (Zheng et al., 2025; Alzubi et al., 2025) have introduced agentic workflows that incorporate web search tools directly into the reasoning process. By decomposing complex queries into sub-tasks and selectively invoking external tools through prompts, these systems can effectively tackle information-seeking tasks that require up-to-date and reliable web content. Such approaches highlight the growing viability of zero-shot and in-context tool use for knowledge-intensive applications.

Building on this trend, recent research has further extended tool-augmented, training-free

paradigms into the multimodal domain, exploring how large models can function as autonomous visual search agents. For example, AVIS (Hu et al., 2023a) proposes a framework in which a large language model acts as a controller, iteratively issuing vision-language queries to retrieve external visual and textual information in order to answer complex visual questions. VSA (Zhang et al., 2024d) takes this further by equipping vision-language models with retrieval capabilities, enabling them to behave like multimodal search engines that can proactively seek and integrate visual evidence from large-scale corpora. MMSearch (Jiang et al., 2024) presents a comprehensive pipeline that empowers large multimodal models with advanced search capabilities. Its agentic workflow encompasses querying, reranking, and summarization stages, enabling models to autonomously process and synthesize information from both visual and textual modalities. These approaches signify a shift towards interactive, tool-augmented processes in multimodal reasoning. However, a key limitation of these approaches is that the underlying models are typically not trained to interact with search tools in a supervised or reinforcement-based manner. As a result, the agent may not learn to engage with external tools in the most effective or reliable way, especially in real-world environments with noisy or ambiguous search results.

A.3 Reinforcement Learning-powered Search Agents

Recent advances, such as OpenAI’s o series (Jaech et al., 2024; OpenAI, b), DeepSeek-R1 (Guo et al., 2025) and Kimi-K1.5 (Team et al., 2025a), have highlighted the potential of end-to-end reinforcement learning (RL) to enhance the reasoning capabilities of large-scale models. These efforts have led to the emergence of Large Reasoning Models (LRMs) that are capable of solving complex, multi-step reasoning tasks beyond the reach of standard instruction-tuned language models. Building on this momentum, leading organizations such as OpenAI, Google, and Perplexity have introduced Deep Research agents (OpenAI, a; Claude; Google; Perplexity). These systems combine large reasoning models with real-time web search tools to complete open-ended, research-oriented tasks, such as drafting reports or synthesizing diverse information sources. Notably, OpenAI reports that it has successfully trained a highly capable Deep Research model through end-to-end rein-

forcement learning, demonstrating the feasibility and effectiveness of this approach in real-world applications. In the open-source community, efforts such as DeepResearcher (Zheng et al., 2025), Search-R1 (Jin et al., 2025), and ReSearch (Chen et al., 2025) have followed this direction, applying end-to-end RL to improve models’ abilities in multiturn search and retrieval-augmented generation, aiming to boost performance on information-seeking question answering tasks. However, existing work mainly focuses on text-based search, while in the multimodal domain, challenges such as constructing suitable training data and designing RL frameworks to incentivize models to perform search in real-world environments remain under-explored. Concurrent with our work, related efforts such as WebWatcher (Geng et al., 2025) and MM-BrowseComp (Li et al., 2025) have also begun to explore this direction. They have focused on constructing benchmarks suitable for evaluating Multimodal Deep Research Agents.

B Generalization of MMSearch-R1

We investigate the model’s generalization capabilities from two primary aspects. First, we evaluate its performance across a broad range of VQA benchmarks. These benchmarks are not strongly correlated with multimodal search, thereby offering a more general assessment of the model’s ability to solve common visual question answering tasks. Second, through case studies, we examine the model’s performance in video captioning scenarios. This allows us to investigate whether the search capabilities acquired during QA-based training can be effectively transferred to open-ended generation tasks.

B.1 Evaluation on General VQA Benchmarks

To assess whether RL training affects general VQA capabilities, we compare MMSearch-R1-7B with its backbone model Qwen2.5-VL-7B on a suite of standard VQA benchmarks, including AI2D (Kembhavi et al., 2016), ChartQA (Masry et al., 2022), LLaVA-Wilder (Li, Bo and Zhang, Kaichen and Zhang, Hao and Guo, Dong and Zhang, Renrui and Li, Feng and Zhang, Yuanhan and Liu, Ziwei and Li, Chunyuan), MathVista (Lu et al., 2023), MME (Fu et al., 2023), and OCRBench (Liu et al., 2024b). All experiments are conducted using LMMS-Eval (Zhang et al., 2024a). As shown in Table 2, MMSearch-R1-7B achieves compara-

ble performance across all benchmarks, slightly outperforming the base model on LLaVA-Wilder and MathVista while maintaining similar results on AI2D, ChartQA, OCRBench and MME. These results suggest that our reinforcement learning process, while enhancing search-related behavior, preserves the model’s general visual understanding and reasoning ability.

B.2 Case Study on Captioning Task

We further conducted case studies to demonstrate that RL training enables the model to internalize the use of search tools, allowing it to address a broader range of problems beyond those with fixed answers. For instance, in captioning tasks, MMSearch-R1 can utilize search tools to enrich captions with relevant knowledge, moving beyond superficial visual descriptions. This capability allows the model to deliver more informative responses to users. As shown in Figure 5, due to the lack of knowledge required to correctly identify information (e.g. name and location) about the museum, Qwen2.5-VL-7B can only attempt to describe the image based on its visual appearance. In contrast, MMSearch-R1 is able to enrich the caption by leveraging image search to obtain additional contextual knowledge.

C Search Behavior: Resolving Conflicting Results

Given that our model performs searches in a real-world environment and incorporates the top-5 search results into its context, the presence of conflicting information among these results is inevitable. We did not introduce explicit rewards or specialized components to resolve such conflicts; instead, handling them is considered part of the model’s general search result processing ability, implicitly optimized through RL training. As shown in Figure 6, We observed the model exhibiting the following behaviors when faced with conflicting results.

Image Search Results Conflicts: The model tends to compare the visual features of the user’s original image with those in the retrieved images to assess the credibility of each result and make a more informed decision.

Text Search Results Conflicts: When textual conflicts arise, the model implicitly evaluates the reliability of information by cross-referencing multiple sources and tends to favor facts that are consistently supported across different search results.

Table 2: Performance Comparison on General VQA Benchmarks

Model	AI2D test	ChartQA test	LLaVA-Wilder small	MathVista testmini	MME test	OCRBench -
Qwen2.5-VL-7B	93.0	86.6	73.5	68.2	623/1705	84.7
MMSearch-R1-7B	92.8	86.1	74.8	68.8	622/1707	84.5

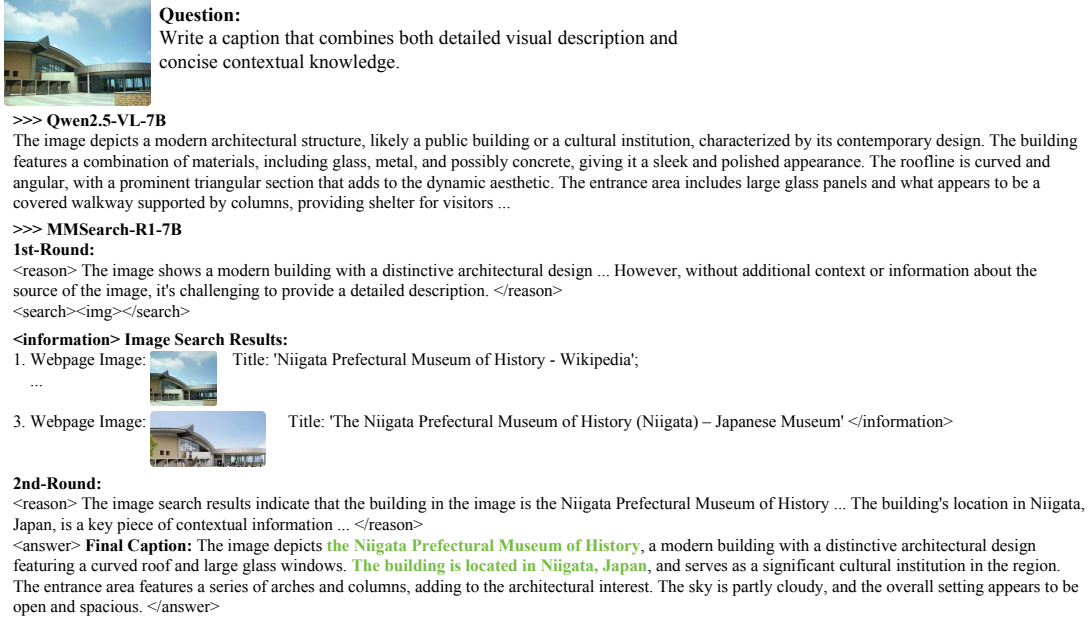


Figure 5: Case Study on Informative Image Captioning Task.

This process is similar to a majority-voting mechanism.

D Group Relative Policy Optimization (GRPO) Algorithm

GRPO updates the current policy model using a group of rollouts sampled from an old policy π_{old} , along with a reference π_{ref} used as a constraint. Specifically, given a question q sampled from the training set \mathcal{D} , GRPO generates a set of outputs from π_{old} , and optimizes the current policy π_θ by maximizing the following objective:

$$\begin{aligned} \mathcal{J}_{GRPO}(\theta) &= \mathbb{E}[q \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)] \\ &= \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left(R_{i,t} \hat{A}_{i,t}, \right. \right. \\ &\quad \left. \left. \text{clip}(R_{i,t}, 1 - \epsilon, 1 + \epsilon) \hat{A}_{i,t} \right) \right. \\ &\quad \left. - \beta \mathbb{D}_{KL} [\pi_\theta \| \pi_{ref}] \right\}, \end{aligned} \quad (2)$$

where $R_{i,t} = \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}$ is the policy ratio. ϵ and β are hyper-parameters that control the stability and constraint strength of the policy update, respectively. Given the sampled rollouts $\mathbf{o} = \{o_1, o_2, \dots, o_G\}$, a group reward $\mathbf{r} =$

$\{r_1, r_2, \dots, r_G\}$ can be obtained by a reward model. The advantage $\hat{A}_{i,t} = \tilde{r}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$ mitigates variance across samples and prevents any single reward signal from dominating the policy update.

E Multimodal Search Tools

To enhance the stability and efficiency of multimodal search services during both training and inference, we encapsulate the image and text search tools as independent HTTP services. These services are optimized through pipelined parallel processing, local caching, and multi-metric monitoring. The overall architecture of the search pipeline is illustrated in Figure 7.

Image Search Tool The image search tool is built on top of SerpAPI. Given an input image URL, SerpAPI returns a set of visually similar webpages, including metadata such as URLs, thumbnails, and titles. We sort the returned results by relevance and extract up to five valid results (thumbnail + title pairs). To avoid redundant SerpAPI calls during training and evaluation, we implement a cache

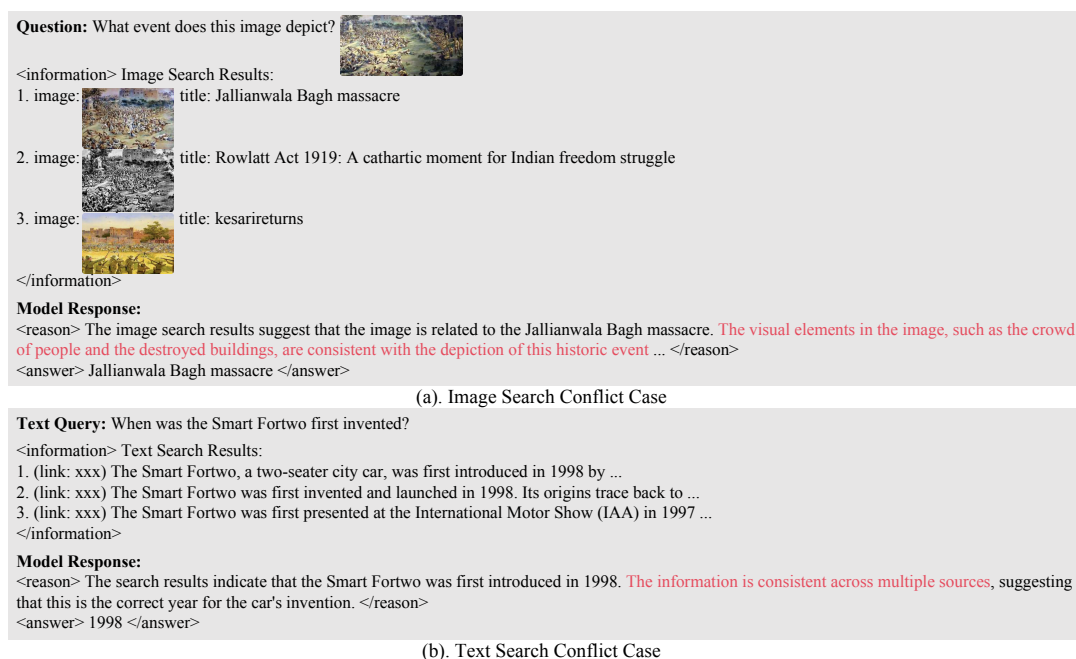


Figure 6: Case Study on Informative Image Captioning Task.

mapping image URLs to their corresponding search results.

Text Search Tool The text search pipeline consists of SerpAPI, JINA Reader, and a webpage summarization model, forming a search–parse–summarize chain. SerpAPI first retrieves a list of webpage URLs based on a given query. In each round, the top 5 URLs are processed using a pipelined parallel strategy: JINA fetches and converts the webpage content into clean, structured text, while Qwen3-32B acts as a summarizer, focusing on extracting information relevant to the original user question using a tailored prompt. The full prompt is provided in Tabel 5. This approach improves focus and reduces token usage. If some URLs fail to be processed, additional URLs from the SerpAPI result list are used to retry, until 5 successful summarizations are obtained or the list is exhausted. To improve efficiency and reduce redundant computation, the text search tool incorporates a multi-layer caching mechanism: (1). Mapping from query to webpage URLs (to skip repeated searches); (2). Mapping from URL to JINA parsing results (to avoid re-parsing); (3). Mapping from JINA outputs to Qwen3-32B summaries (to reduce summarization cost). We deployed Qwen3-32B on a cluster of 8 × 8 NVIDIA H100 GPUs to support our training and inference needs.

Cache Mechanism Caching is implemented us-

ing Redis to support high-throughput concurrent read/write access. For latency-critical components, caching is applied to reduce response time. Because JINA parsing results are often large, they are stored in object storage (TOS), while Redis only stores references (keys) to these results, reducing memory footprint. We adopt an LRU (Least Recently Used) policy for eviction under memory constraints. Additionally, a scheduled offline retry mechanism is in place to reprocess failed JINA URLs and update the cache accordingly.

Distributed Rate Limiting infrastructure-level constraints on concurrent request handling can bottleneck training efficiency. Limited parallelism may cause request backlogs or latency spikes, ultimately affecting throughput and stability. To manage system load during peak traffic, we implement distributed rate limiting using Redis. This allows us to smooth bursty traffic to JINA and webpage summary model over time windows, reducing the risk of system overload and improving service availability and stability.

F Prompts

F.1 Prompts for FVQA-auto-ac VQA Generation

To generate factual QA pairs for the FVQA–auto–vc set, we designed the prompt to guide GPT-4o in producing concise and grounded question–answer pairs based on given image–webpage pairs. The

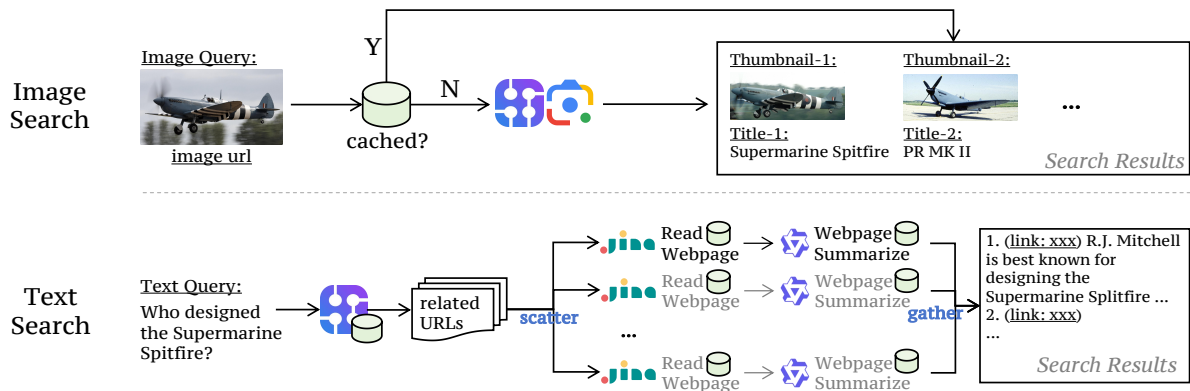


Figure 7: The overall architecture of the multimodal search pipeline.

prompt were tailored to encourage questions that focus on visual concepts and can be answered using observable or associated information in the webpage content. Details of the prompts are shown in Table 3.

F.2 Prompts for RL Training

During training, prompts are used to guide and constrain the model’s behavior, ensuring that it interacts with the search tools in a consistent and structured manner. Specifically, we design three types of prompts for different points in the dialogue: one used at the beginning of the conversation, one after the model performs an image search, and another after it performs a text search. These context-specific prompts play a key role in aligning the model’s actions with the intended workflow and ensuring proper use of the multimodal search tools. Details of the prompts are shown in Table 4.

F.3 Prompts for Webpage Summarization Model

The prompt shown in Table 5 is used to guide the webpage summarization model in the text search pipeline to effectively extract and summarize webpage content based on the given user question. The goal is to ensure that the returned content is more focused and concise, thereby reducing overall token consumption.

F.4 Prompt for Direct Answer Baseline

Table 6 presents the prompt template used in the Direct Answer baseline. In this setting, the model is expected to provide a concise answer to a given question based solely on the visual content of the image.

F.5 Prompts for RAG Workflow Baseline

Under the RAG workflow, models are required to perform exactly two search operations for each VQA example using our multimodal search tools: first an image search, followed by a text search. Under the RAG workflow, models are required to perform exactly two search operations for each VQA example using our multimodal search tools: first an image search, followed by a text search. This results in a maximum of three conversation rounds. Specifically, given an input image and question, the model is presented with the image search results and the original question in the first round and is prompted to generate a relevant text query to aid in answering. In the second round, the retrieved text results based on that query are provided to the model, which is then asked to generate the final answer. In our experimental setup, the RAG workflow reaches the maximum allowed number of search steps, making it a strong baseline that provides the model with the most external information. Details of the prompts of RAG Workflow are shown in Table 7.

F.6 Prompts for LLM-as-Judge Evaluation

In our experiments, we adopt the LLM-as-judge approach to evaluate whether a model’s response aligns with the ground truth answer. This method provides a more objective and generalizable way to assess performance on open-ended VQA tasks. Table 8 shows the full prompt used with GPT-4o, which serves as the judge LLM throughout all our experiments. Once GPT-4o determines whether each response is correct or incorrect, its judgments are used to compute the accuracy metric.

Table 3: Prompt used for FVQA-train VQA Generation.

Usage	Prompt
Factual QA Generation	<p>Your task is to generate a factual question–answer pair based on the given visual concept, the image, and the associated webpage content.</p> <p>The generated question must strictly follow the requirements below:</p> <ol style="list-style-type: none"> 1. The answer must contain the keyword visual concept. 2. The question must start with "Who", "What", or "Where". 3. The question must NOT include the visual concept itself, nor any background knowledge directly related to it. 4. The question should resemble something a curious human without prior knowledge about the image might ask. <p>In addition to the question, generate a concise and factual answer grounded in the visual concept, image, and webpage content.</p> <p>Visual Concept: {visual_concept} Image: {image} Webpage Content: {webpage_content}</p> <p>Respond only with the generated question and answer.</p>

G Details of Benchmark Datasets

FVQA-test As described in Section 3, the FVQA-test set is a manually curated test set consisting of 1800 examples collected from three sources: (1) 600 samples drawn from FVQA-auto-vc, with no overlap with the training set; each example was manually verified for correctness; (2) 600 samples selected from the InfoSeek Human Split, where we manually re-annotated the answers, as the original human-labeled answers were not publicly released; (3) 600 samples newly collected by human annotators. This dataset covers a wide range of visual and textual knowledge categories, enabling a comprehensive evaluation of model performance.

InfoSeek InfoSeek (Chen et al., 2023) is constructed through a semi-automated process that transforms Wikidata triples into natural language questions using human-authored templates. Annotators design question templates for 300 Wikidata relations, incorporating placeholders for visual entity types and units to ensure clarity. These questions are then paired with corresponding images and answers to form image, question, answer triplets. To ensure diversity and answerability, question-answer pairs that lack supporting evidence in Wikipedia are filtered out, and balanced sampling is applied across entities and relations. This design makes InfoSeek particularly suitable for evaluating information-seeking capabilities, as it emphasizes diverse, fact-based queries grounded in real-world knowledge and multimodal contexts. We randomly sampled 2000 examples from its test split due to the large dataset size.

MMSearch MMSearch (Jiang et al., 2024) con-

tains 300 manually collected examples spanning 14 subdomains, categorized into two types: News, which includes up-to-date events from August 2024 to ensure no overlap with LMM training data, and Knowledge, which focuses on rare, verified knowledge that current state-of-the-art LMMs (e.g., GPT-4o, Claude 3.5) fail to answer. Among these, 171 examples are visual questions (i.e., questions paired with images), while the remaining are purely textual. We use the 171 visual questions as our evaluation subset. This dataset is particularly well-suited for evaluating models’ real-world information-seeking abilities, especially in scenarios that require up-to-date retrieval or reasoning over rare knowledge.

SimpleVQA SimpleVQA (Cheng et al., 2025) is built from two main sources of seed examples. First, factual image-question pairs are selected from existing VQA datasets that meet real-world knowledge standards. These datasets were constructed after 2023 and focus on practical, application-driven content. Second, additional data is collected from search engines such as Google, Baidu, and Wikipedia, where expert annotators generate QA pairs based on retrieved images and factual knowledge. These examples span a wide range of entities and events, with answers focused on objective, fact-based information involving entity recognition and attribute extraction. Building on these seed samples, the dataset further undergoes reliable difficulty filtering and human-in-the-loop quality control, making it a strong candidate for evaluation of models’ factual and knowledge-based reasoning abilities. The final SimpleVQA benchmark contains 2,025 examples. To avoid language-related confounding factors, we select the 1,013 En-

Table 4: Prompts used during training at conversation start and after different search actions.

Usage	Prompt
1st Round	<p>Answer the user’s question based on the provided image. Examine the image carefully and identify any recognizable entities, such as faces, objects, locations, events, logos, or text. Determine whether you have sufficient knowledge to confidently recognize the main visual element and answer the user’s question. If so, first explain your reasoning, then provide a clear and direct answer.</p> <p>If you are unable to confidently identify the visual element, stop and invoke the image search tool by appending the string <code><search>img</search></code> at the end of your response. This will trigger a Google Lens search using the original image to retrieve relevant information that can help you confirm the visual content.</p> <p>Once you have sufficient visual understanding, combine it with the user’s question and assess whether you can confidently answer. If so, answer the question directly using your own knowledge. If not, invoke the text search tool by generating a concise and specific query, and output it in the format <code><text_search>your query here</text_search></code> at the end of your response. Carefully craft your query to accurately retrieve the information needed to help answer the question. The text search tool will then use Google Search to return relevant information based on your query. You must include your reasoning inside <code><reason>...</reason></code> before taking any action, whether it is calling the image search tool, generating a text search query, or providing a final answer. The reasoning may involve analysis of the original image and question, interpretation of search results, or logical steps leading to the final answer.</p> <p>All search results will be placed inside <code><information></code> and <code></information></code> and returned to you. When you are ready to answer the question, wrap your final answer between <code><answer></code> and <code></answer></code>, without detailed illustrations. For example: <code><answer>Titanic</answer></code>.</p> <p>Here is the image and the question: {image}{question}</p>
After image search	<p>Original question: {question}</p> <p>Please extract the visual element relevant to the user’s question (such as faces, objects, locations, events, logos, or text) from the search results. Then, combine this information with the user’s question and perform reasoning to determine whether a Google text search is needed to retrieve additional information to answer the question.</p> <p>If a text search is needed, output the string <code><text_search>your query here</text_search></code> at the end of your response. Please generate a well-crafted query based on the visual element that will help retrieve the most relevant information.</p> <p>If a text search is not needed, use your own knowledge to directly answer the user’s question. You must conduct your reasoning inside <code><reason></code> and <code></reason></code> before taking any action, whether it’s generating a text search query or providing a final answer. If you decide to give the final answer, place it inside <code><answer></code> and <code></answer></code>, without detailed explanation or illustration. For example: <code><answer>Titanic</answer></code></p>
After text search	<p>Original question: {question}</p> <p>Please analyze the search results and the user’s question and continue reasoning inside <code><reason></code> and <code></reason></code>.</p> <p>If you determine that additional knowledge is still required to answer the user’s question, stop responding to the question and instead report a warning by outputting the string "Unable to answer due to lack of relevant information" at the end of your response.</p> <p>If no further external information is needed, you should provide the final answer by placing it within <code><answer></code> and <code></answer></code>. The answer must be concise, clear, and to the point, without any additional explanation or elaboration.</p>

Table 5: Prompt used for Webpage Summarization Model.

Usage	Prompt
System Message	<p>You are a helpful assistant. Your task is to summarize the main content of the given web page in no more than five sentences. Your summary should cover the overall key points of the page, not just parts related to the user’s question.</p>
Prompt	<p>If any part of the content is helpful for answering the user’s question, be sure to include it clearly in the summary. Do not ignore relevant information, but also make sure the general structure and main ideas of the page are preserved. Your summary should be concise, factual, and informative. Webpage Content (first 30000 characters) is: {webpage_content}</p> <p>Question: {question}</p>

Table 6: Prompt used for Direct Answer baselines.

Usage	Prompt
1st-Round	Based on the image, answer the question with as few words as you can. Question: {question} Image: image

Table 7: Prompts used for RAG workflow baselines. Text in red indicates content returned by the multimodal search tool.

Usage	Prompt
1st Round	You are given a question accompanied by an image that cannot be answered without external knowledge. To assist your understanding, you are also provided with image search results consisting of five web pages related to the original image, ranked by relevance. Each result includes the main image from the web page and its title. Question: {question} Image: {image} Image Search Results: 1. Webpage Image: {image} Webpage Title: {title} 2. Webpage Image: {image} Webpage Title: {title} ... 5. Webpage Image: {image} Webpage Title: {title} Assume you have access to a search engine (e.g., google). Based on the question, image and image search results, please raise a text query to the search engine to search for what is useful for you to answer the question correctly. You need to consider the characteristics of asking questions to search engines when formulating your questions. Now give the text query to search engine directly (do not wrap it in quotes or add any explanation):
2nd Round	You should now read the text search result and answer the question based on the image provided. Text Search Results: ... Original question: {question} Answer the question with as few words as you can.

glish QA pairs for evaluating model performance.

LiveVQA LiveVQA (Fu et al., 2025) is built from six globally recognized news platforms, such as CNN, BBC, Yahoo, Forbes, AP News, and Variety, to ensure diversity and timeliness across visual and textual content. The dataset covers 14 major news categories, including sports, entertainment, science, economy, and health, and contains 3,602 VQA pairs. For each article, question–answer pairs are generated using GPT-4o, including both basic visual questions and more complex multi-hop questions that require reasoning over the accompanying text. This setup supports evaluation of real-world information-seeking and multimodal reasoning capabilities.

H Details of Experimental Settings

H.1 RL Training Setting

We adopt the Qwen2.5-VL-7B model as the backbone model, and the overall GRPO training is implemented based on the veRL (Sheng et al., 2024) framework. The training set is the FVQA-train dataset, which consists of approximately 1,600 search-free samples and 3,400 search-required sam-

ples. Training of 7B model was conducted on a total of 32 NVIDIA H100 GPUs (4 nodes with 8 GPUs each), requiring approximately 230 GPU hours to complete. The total batch size is set to 512, with a mini-batch size of 128. For each training prompt, we generate 8 rollouts, and each rollout allows up to 3 tool calls. Regarding the reward components, the search penalty and the weights for the format score are both set to 0.1. The learning rate is $2e-6$, and we fix the KL divergence coefficient β to 0.001 and the clip ratio ϵ to 0.2. For MMSearch-R1-7B, we use the checkpoint from step 50 (when training converges) for downstream evaluation.

H.2 SFT Training Setting

For the SFT model used in the RL vs. SFT performance comparison experiments presented earlier. We fine-tune the Qwen2.5-VL-7B model using LLaMA-Factory (Zheng et al., 2024), with a dataset of 8,000 samples generated by prompting GPT-4o to perform on-demand search. Each sample contains up to three rounds of dialogue. The GPT-4o responses are used as supervision signals,

Table 8: Full prompt used for GPT-4o as the judge LLM in all experiments.

Usage	Prompt
System Message	<p>You are an AI assistant tasked with evaluating the correctness of model responses based on an image, question, and ground truth answer. Your judgment should follow these principles:</p> <ol style="list-style-type: none"> 1. Consider the image, question, and ground truth answer holistically before evaluating the model's response. 2. Your decision should be strictly Yes or No, based on whether the model's response is factually accurate and aligns with the ground truth answer. 3. If the model response is a more specific form of the ground truth answer, it is correct. 4. If the model response includes all key information but adds minor details, it is correct as long as the extra details are factually correct. 5. If the model response contradicts, modifies, or omits critical parts of the answer, it is incorrect. 6. For numerical values, ensure correctness even when presented in different units. 7. For names, check for first and last name correctness. If the middle name is extra but correct, consider it correct. 8. For yes/no questions, the response must exactly match "Yes" or "No" to be correct. 9. If the judgment can be made based solely on the text, you may choose to ignore the input image, as some images may be unfamiliar to you and could affect your judgment. Refer to the image only when necessary to minimize misjudgment. 10. If there are multiple candidate answers, you can also evaluate the model's response against all of them. If the response aligns with at least one candidate according to the rules above, it should be considered correct. <p>Your output must be in the following format: <judge>Yes/No</judge> <reason>Explanation of why the answer is correct or incorrect.</reason></p>
Prompt	<p>Image, Question, and Model Response Evaluation Question: {question} Ground Truth Answer: {ground truth answer} Candidate Answers: {candidate answers} Model Response: {model response}</p> <p>Evaluation Instructions Evaluate whether the Model Response is correct based on the Image, Question, Ground Truth Answer and Candidate Answers. Follow the predefined judgment rules and provide a clear Yes/No answer along with a justification.</p> <p>Output Format <judge>Yes/No</judge> <reason>Detailed reasoning following the evaluation principles.</reason></p>

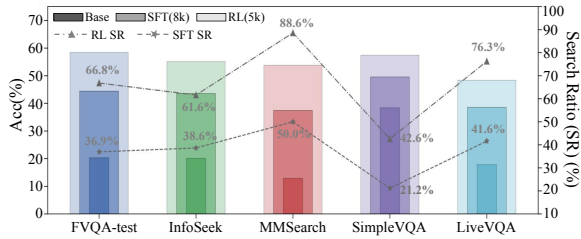


Figure 8: Performance improvements of SFT and RL over Base across five VQA datasets.

while the user question in the first turn and tool-returned content in later turns are masked out during loss computation. Fine-tuning is performed on a single machine with 8 Nvidia H100 GPUs. We use a per-device batch size of 1 and apply gradient accumulation over 2 steps. The model is trained for 1 epochs with a learning rate of $1e-5$. A cosine learning rate scheduler is adopted, with 10% of the total training steps used for warm-up.

H.3 Evaluation Setting

We conduct model inference using the *veRL* framework by setting the *val_only* parameter of the trainer to True. The inference engine is based on *vLLM*, with *top_p* set to 1.0 and temperature set to 0. A single response is generated for each sample in the test set.

For evaluation, we adopt an LLM-as-Judge approach. Specifically, we use GPT-4o-20241120 as the judge model and apply a predefined prompt template (as shown in Table 8) to assess each inference result. During judgment, *top_p* is set to 0.1 and temperature is set to 0. The model’s binary output ("Yes" or "No") is used to compute the final accuracy metric.

I Full Experimental Results

I.1 Comparison between SFT and RL

To investigate the effectiveness of RL and SFT as training paradigms, we compared their performance gains and data efficiency through controlled experiments. For RL, we use the aforementioned FVQA-train set, which contains 5k VQA samples in the form of (image, question, answer) triplets. For SFT, we first construct a dataset by aggregating three sources: FVQA-auto-vc, FVQA-auto-txt, and FVQA-human-train, resulting in 13.2k examples. We then distill GPT-4o’s behavior on this dataset by prompting it with the same instructions used in RL training, allowing it

to autonomously decide whether to invoke external tools when answering a question. This process yields 13.2k responses from GPT-4o, each including an image, question, and a response containing its reasoning process and any tool usage, often in a multi-turn dialogue format. During SFT training, we mask out the prompts and all tool-returned content from the multi-turn dialogues, using only GPT-4o’s responses as supervision signals. This ensures that the model learns to mimic GPT-4o’s reasoning and answering behavior without relying on intermediate tool outputs, thereby encouraging it to internalize the reasoning process demonstrated by the teacher model. To ensure training quality, we compare GPT-4o’s final predicted answers with the original ground truth answers and filter out 5.2k samples where GPT-4o produced incorrect answers. The remaining 8k examples constitute our SFT training set. The implementation details of SFT can be found in the Appendix H.2.

We fine-tune the base model, Qwen2.5-VL-7B, using the above datasets for both SFT and RL training. We then evaluate the performance of the models trained with SFT and RL on all downstream tasks and compare their improvements over the base model, as shown in Figure 8. The results show that the model trained with RL consistently outperforms the one trained with SFT across all tasks, despite being trained on only about half as much data. In addition, the behavior of the RL-trained model aligns more closely with the specific requirements of each task, especially in its use of search tools. For example, in the MMSearch and LiveVQA tasks, which are both recently collected and focus on information-seeking questions, the RL-trained model shows a higher frequency of search tool invocation. This is consistent with expectations, as these datasets contain many questions that require external visual or textual knowledge not encountered during pretraining. The RL-trained model learns to rely more effectively on search tools in such scenarios.

I.2 Cost-Profiling Study

As shown in Table 9, we conducted a cost profiling study on FVQA-test testset to analyze the latency and token consumption of our model: (1). **Token Efficiency Analysis.** MMSearch-R1 introduces additional token overhead due to generating intermediate Thought tokens. However, this overhead is relatively minor in our setting, averaging only about 100 extra output tokens per question

compared to the RAG baseline. Notably, when the model chooses not to invoke the search tool, it also avoids processing search results, which helps save input tokens for subsequent turns. As a result, the total token consumption of MMSearch-R1 and the RAG baseline remains very close. **(2). Latency Analysis.** The search tool pipeline accounts for the majority of overall inference time. By reducing search requests by approximately 33%, MMSearch-R1 achieves better end-to-end inference latency compared to the RAG baseline.

I.3 Study on the Impact of Noisy Search Results

In our implementation, we perform proactive noise filtering through the following methods. For image search, we directly extract the primary thumbnails and their corresponding titles from the top-5 relevant web pages, bypassing any additional quality assessment. For text search, we employ the Qwen3-32B model to summarize and extract information pertinent to the user’s query. This process effectively filters out irrelevant content and increases information density. To ensure the reliability of this summarization, we manually evaluated various prompts on a small dataset, verifying that the model’s outputs contain minimal hallucinations and noise. However, the presence of noisy search results in a real-world environment is unavoidable. Consequently, we proceeded to analyze the impact of such noisy content on our experimental results. We reviewed and categorized the model’s inference trajectories for each test question to identify cases impacted by noisy retrievals. Using GPT-4o, we evaluated both image and text search results: if image search titles referred to different entities, or if any text search summaries were irrelevant or unhelpful, the trajectory was marked as noisy. We then compared the model’s accuracy on these noisy cases to its accuracy on the full dataset to assess robustness to search noise. As shown in Table 10, The results indicate that search noise does negatively impact model performance, as accuracy on noisy samples is lower than on the full dataset. However, the difference is not substantial, suggesting that the model demonstrates a certain degree of robustness to search noise.

I.4 Study on Reward Modeling

Our main training framework adopts a rule-based reward design based on exact string match, which is simple, deterministic, and scalable. However,

this formulation may not fully capture the semantic correctness of answers, especially when multiple surface forms express the same factual content. To explore whether a more flexible and context-aware reward signal can better guide model learning, we conduct an additional experiment using GPT-4o as the reward model during training. GPT-4o provides semantic-level accuracy judgments, allowing us to assess the impact of richer reward feedback on model behavior and final performance. The prompt used to elicit reward signals from GPT-4o is the same as the one used in our evaluation setup, as shown in Table 8.

As shown in Table 11, training with GPT-4o reward leads to a clear improvement across all datasets. The *4o version achieves an average accuracy of 59.5%, which is 3.8 points higher than the *EM version. These results indicate that more general reward supervision holds greater potential for open-ended tasks. GPT-4o helps avoid false negatives caused by exact match and leads to improved robustness. However, it may introduce bias, higher training cost and struggles with questions beyond its knowledge scope. We believe that exploring more general reward modeling represents a promising direction for future research. Notably, this experiment is conducted on a subset of the FVQA-train dataset, in which search-free examples are relatively underrepresented. As a result, the model exhibits a higher average search rate at convergence compared to Table 1.

I.5 Experiment Statistical Significance

We conducted a statistical significance evaluation for the main 7B model experiments reported in the main text to verify the robustness of our findings. All experiments were repeated five times, and we report the mean and standard deviation (\pm std) across runs for both accuracy (Acc) and search ratio (SR) in Table 12. For the Direct Answer and RAG Workflow baselines, the primary source of randomness lies in slight fluctuations in GPT-based evaluation. For MMSearch-R1, variability arises from two factors: the inherent randomness in generating rollouts using vLLM as the backend inference engine, and potential variation in GPT-based evaluation. Additionally, methods involving search (RAG Workflow and On-Demand Search) may also be affected by the inherent randomness of external search tools, such as fluctuations in retrieved content over time or variations in result ranking. The results show that accuracy fluctu-

Metric (per question)	Qwen2.5-VL-7B (RAG)	MMSearch-R1-7B	Δ
Input tokens	2163.67	2030.11	-6.17%
Output tokens	16.44	161.74	+883.82%
Total tokens	2180.10	2191.85	+0.54%
Image Search Calls	1	0.73	-27.00%
Text Search Calls	1	0.60	-40.00%
Total Search Calls	2	1.33	-33.50%
Search Tool Latency	18.904s	12.207s	-35.42%
Generation Latency	0.903s	3.956s	+338.09%
End-to-End Latency	19.807s	16.163s	-18.39%

Table 9: Cost-Profiling Study

	Avg	FVQA_test	InfoSeek	MMSearch	SimpleVQA	LiveVQA
Num of samples	/	1800	2000	171	1013	3602
Accuracy on all samples (%)	54.63	58.44	55.1	53.8	57.45	48.4
Num of noisy samples	/	551	449	69	309	1412
Ratio of noisy samples (%)	32.62%	30.61%	22.45%	40.35%	30.50%	39.20%
Accuracy on noisy samples (%)	52.94	56.8	48.1	59.42	52.1	48.3

Table 10: Study on the impact of noisy search results.

Table 11: Ablation Study on Reward Modeling. "Acc (%)" denotes the accuracy evaluated by LLM-as-Judge, while "SR (%)" represents the search ratio, defined as the percentage of total search calls made relative to the maximum allowed search steps for each method. *EM indicates that the accuracy score used in the reward function is computed via exact string match (EM), while *4o denotes that correctness is judged by GPT-4o.

Model	Average		In-Domain				Out-of-Domain					
	Acc	SR	FVQA-test		InfoSeek		MMSearch		SimpleVQA		LiveVQA	
<i>Direct Answer</i>												
GPT4o	36.0	0	41.7	0	42.7	0	22.2	0	46.6	0	26.9	0
Gemini 2.5 Pro	36.4	0	37.2	0	37.0	0	26.9	0	53.4	0	27.7	0
Qwen2.5-VL-72B	26.6	0	27.1	0	28.0	0	15.7	0	42.2	0	20.1	0
Qwen2.5-VL-32B	25.0	0	24.7	0	25.8	0	15.7	0	40.1	0	18.7	0
Qwen2.5-VL-7B	21.9	0	20.3	0	20.1	0	12.8	0	38.4	0	17.8	0
<i>Prompt-based Search Workflow</i>												
Qwen2.5-VL-32B *ReAct	47.9	67.3	49.6	68.8	50.1	77.2	44.4	72.5	56.0	52.5	39.2	65.4
Qwen2.5-VL-7B *ReAct	43.1	51.9	46.4	55.0	43.8	61.0	40.4	62.0	50.6	38.1	34.2	43.4
<i>RAG Workflow</i>												
GPT4o	62.1	100	66.0	100	59.1	100	62.5	100	63.4	100	59.6	100
Gemini 2.5 Pro	61.8	100	66.1	100	56.7	100	62.5	100	65.9	100	57.8	100
Qwen2.5-VL-72B	59.6	100	62.2	100	59.4	100	59.6	100	61.0	100	56.0	100
Qwen2.5-VL-32B	55.1	100	57.0	100	56.8	100	57.9	100	54.5	100	49.6	100
Qwen2.5-VL-7B	51.6	100	52.9	100	53.7	100	52.2	100	51.6	100	48.0	100
<i>On-demand Search</i>												
MMSearch-R1-7B *EM	55.7	77.3	59.1	83.4	57.1	68.3	55.6	91.8	58.1	56.8	48.5	86.2
MMSearch-R1-7B *4o	59.5	82.6	62.1	88.9	59.1	81.8	61.9	92.7	59.8	72.6	54.7	76.9

ations remain small and comparable across both the RAG Workflow and On-Demand Search settings, which demonstrates consistent performance across repeated runs. The MMSearch dataset displays slightly higher variance in both accuracy and search ratio, which is expected due to its small evaluation size of only 171 examples. In contrast, all other datasets maintain low standard deviations, indicating stable behavior across trials. These observations confirm that our key findings are not sensitive to random factors and support the overall robustness and reliability of our experimental conclusions.

J Examples of FVQA Dataset

To better illustrate the characteristics of our dataset, we present representative examples from the FVQA dataset in Figure 9. QA pairs are designed to be knowledge-intensive and fact-oriented, covering a broad range of visual and textual knowledge types. We also present examples from the data construction process of FVQA-auto-vc in Figure 10.

K Case Studies

As shown in Figures 11–13, we present several representative case studies to illustrate how MMSearch-R1 performs on complex, real-world information-seeking VQA tasks. These examples highlight the model’s ability to reason about when and how to invoke search tools, generate effective queries, and synthesize the retrieved information to arrive at accurate answers. For ease of formatting and readability, we omit some of the retrieved search results in certain cases.

L Potential Risks

Our work focuses on improving the ability of multimodal models to access and external information through search. Although this capability has clear benefits for building more informative and adaptive assistants, it also introduces potential risks. For example, models that autonomously retrieve web content may surface outdated, biased, or misleading information. Additionally, over-reliance on real-time search may raise concerns around content verifiability, copyright, or inadvertent propagation of misinformation. To mitigate these potential risks, we recommend careful filtering of the retrieved content, attribution of information sources, and incorporating mechanisms that allow users to trace and verify the provenance of model-generated

responses. We also encourage future research on safer retrieval strategies, such as domain whitelisting and adaptive uncertainty calibration, especially in high-stakes applications.

M Ethical Considerations

All artifacts and datasets used in this work are consistent with their intended research use. Pre-trained models (e.g., vision-language backbones) and multimodal datasets (e.g., InfoSeek) were utilized strictly under their respective research licenses and only for non-commercial academic purposes. Our released artifacts, including the MMSearch-R1 agent, benchmark, and evaluation harness, are provided solely for research use, intended to support studies on multimodal reasoning and retrieval-based language agents.

Regarding data ethics, all datasets employed are publicly available research resources that have undergone prior filtering for personally identifying information (PII) and offensive or explicit content. We did not collect any new data involving human subjects. To further ensure data integrity, we performed additional automated keyword and metadata screening to confirm the absence of PII or sensitive material. All released examples and outputs have been carefully reviewed to comply with ethical research standards.

N Use of AI Assistants

In the preparation of this manuscript, we utilized AI assistants for two specific purposes. First, large language models were employed to aid in proof-reading and copy-editing. This involved correcting grammatical errors, improving sentence structure, and refining word choices to enhance the clarity and readability of the text. Second, an AI assistant was used to help generate a Python script for the visualization of our experimental results, specifically for plotting charts and figures like Figure 4. It is important to note that all content, including the final text and the logic of the analysis, was authored by the human researchers. The AI tools served solely as assistive instruments for language polishing and code generation.

Table 12: Statistical Evaluation: Mean \pm Std Dev over 5 Repetitions

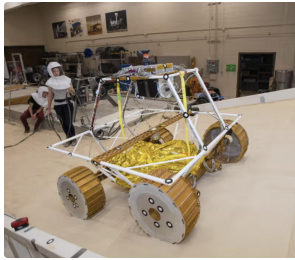
Model	Average		In Domain						Out of Domain			
	Acc	SR	FVQA_test		InfoSeek		MMSearch		SimpleVQA		LiveVQA	
	Acc	SR	Acc	SR	Acc	SR	Acc	SR	Acc	SR	Acc	SR
<i>Direct Answer</i>												
Qwen2.5-VL (7B)	21.9 ± 0.08	0 ± 0	20.4 ± 0.15	0 ± 0	20.0 ± 0.04	0 ± 0	13.2 ± 0.31	0 ± 0	38.4 ± 0.13	0 ± 0	17.8 ± 0.12	0 ± 0
<i>RAG Workflow</i>												
Qwen2.5-VL (7B)	51.6 ± 0.31	100 ± 0	52.4 ± 0.31	100 ± 0	53.6 ± 0.08	100 ± 0	52.1 ± 1.72	100 ± 0	52.1 ± 0.28	100 ± 0	47.9 ± 0.12	100 ± 0
<i>On-Demand Search</i>												
MMSearch-R1 (7B)	54.5 ± 0.31	66.6 ± 0.43	58.1 ± 0.29	67.3 ± 0.28	54.9 ± 0.14	61.6 ± 0	54.5 ± 1.39	88.1 ± 0.48	57.1 ± 0.27	42.6 ± 0	48.0 ± 0.20	73.8 ± 2.20

<p>Level-1 Knowledge Taxonomy: Arts</p>  <p>Q: What type of art style is depicted in this image? A: Fractal art.</p>  <p>Q: What painting technique is used in this artwork? A: Grisaille.</p>	<p>Level-1 Knowledge Taxonomy: Place</p>  <p>Q: What country does this place belong to? A: Vietnam.</p>  <p>Q: Where is the lake outflow to? A: Mincio.</p>
<p>Level-1 Knowledge Taxonomy: People</p>  <p>Q: Who designed this building? A: Helgo Zettervall.</p>  <p>Q: Who is the central figure addressing the group in this scene? A: Dronacharya.</p>	<p>Level-1 Knowledge Taxonomy: Industry</p>  <p>Q: What type of ammunition is shown in the image? A: .338 Federal.</p>  <p>Q: What is the brand of this vehicle? A: Land Rover.</p>
<p>Level-1 Knowledge Taxonomy: Science</p>  <p>Q: What is this colorful astronomical object called? A: Crab nebula.</p>  <p>Q: What is the scientific term for the organisms? A: Haemoproteus.</p>	<p>Level-1 Knowledge Taxonomy: Nature</p>  <p>Q: What is the name of this plant with yellow flowers? A: Bladderwort.</p>  <p>Q: What type of fish is shown in the image? A: Lizardfish.</p>
<p>Level-1 Knowledge Taxonomy: Entities</p>  <p>Q: What is the name of the company associated with this logo? A: Brainfeeder.</p>  <p>Q: What is this logo for? A: Google Duo</p>	<p>Level-1 Knowledge Taxonomy: Events</p>  <p>Q: What event is taking place at this race track? A: Dutch Grand Prix.</p>  <p>Q: What event does this map represent? A: 2020 Polish presidential election.</p>

Figure 9: Examples of FVQA dataset.



Figure 10: Examples of FVQA-auto-vc QA Generation.



Source: MMSearch

Question: In which day of 2024 was the lunar rover project in the picture announced to be canceled?

Answer: 07-17.

Direct Answer



GPT4o-20241120

July 1, 2024.



Qwen2.5-VL-72B

Not announced.

On-Demand Search



MMSearch-R1-7B

<reason>The image shows a lunar rover, but there is no information about its announcement or cancellation date. The image does not provide enough context to answer the user's question confidently.**</reason>**

<search></search>



Image:



Title: VIPER

Image:



Title: Virtual Intelligent Planetary Exploration Rover, VIPER Mobil... - NASA Science

Image:



Title: Scientists slam 'indefensible' axing of Nasa's \$450m Viper moon rover | Nasa | The Guardian



<reason>The image search results mention the "Viper" lunar rover project and a news article about the cancellation of the project. However, the image does not provide information about the exact date when the project was canceled. To answer the user's question, a text search is necessary to find the specific date of the cancellation.**</reason>**

<text_search>When was the NASA Viper moon rover project canceled in 2024?</text_search>



1. (link:<https://www.nasa.gov/news-release/nasa-ends-viper-project-continues-moon-exploration/>) NASA announced the cancellation of the VIPER (Volatiles Investigating Polar Exploration Rover) project on July 17, 2024. The decision followed a comprehensive internal review ...

2. (link:<https://www.nasa.gov/blogs/missions/2025/05/07/nasa-to-explore-additional-methods-to-send-viper-to-moon/>) The NASA VIPER (Volatiles Investigating Polar Exploration Rover) moon rover project was canceled in July 2024. Following this cancellation ...

3. (link:<https://spacenews.com/nasa-backtracks-on-viper-commercial-partnership/>) NASA canceled the VIPER lunar rover mission in July 2024. The decision was made due to ...

....



<reason>The search results indicate that the NASA VIPER lunar rover project was canceled on July 17, 2024. The information is consistent across multiple sources, suggesting that this is the correct date for the cancellation.**</reason>**

<answer>July 17</answer>

Figure 11: Case study 1 of MMSearch-R1.



Source: LiveVQA

Question: Based on the provided image, why was the concert programme moved to this location?

Answer: Funding cuts.

Direct Answer



GPT4o-20241120

The location offers better acoustics and seating.



Qwen2.5-VL-72B

Acoustics.




On-Demand Search



MMSearch-R1-7B

<reason>The image shows a well-designed auditorium with a stage set up for a performance. However, there is no information in the image about why the concert programme was moved to this location. The image does not provide any context or details about the event or the location's history. Therefore, I cannot confidently answer the question based on the image alone. **</reason>**
<search></search>



- Image:  Title: St John's Auditorium refurbished and officially reopened | St John's College, Oxford
- Image:  Title: Garden Quadrangle, St. John's College Oxford – Sound Space Vision
- Image:  Title: Institute of Theatre Consultants



<reason>The image search results suggest that the location in the image is St John's Auditorium in St John's College, Oxford. However, the image search results do not provide information about why the concert programme was moved to this location. To answer the user's question, a text search is necessary to find out the reason for the location change. **</reason>**
<text_search>Why was the concert programme moved to St John's Auditorium, St John's College, Oxford?</text_search>



1. (link:<https://www.artsprofessional.co.uk/news/newsreel/concert-programme-to-relocate-after-funding-cuts>) The concert programme, SJE Arts, was moved to St John's Auditorium at St John's College, Oxford, after its previous host, St Stephen's House theological college, withdrew funding support. St Stephen's House concluded it could no longer resource the programme...
2. (link:<https://www.sjc.ox.ac.uk/discover/news/st-johns-hosts-international-piano-series/>) The International Piano Series, previously based at St John the Evangelist Church, is moving to St John's College, Oxford, starting in autumn 2025. This relocation aims to enhance the series' reach and impact by utilizing the acoustically rich and intimate ...
3. (link:<https://bsky.app/profile/oxfordclarion.bsky.social/post/3lknzttvynk26>) The concert programme of Oxford's International Piano Series is moving to St John's College after its previous venue, SJE Arts at St John the Evangelist on Iffley Road, ends its concert programme ...



<reason>The search results indicate that the concert programme was moved to St John's Auditorium at St John's College, Oxford, after its previous host, St Stephen's House theological college, withdrew funding support. St John's College has now offered its auditorium as a new venue, citing its acoustic quality and potential to enhance the programme's reach and impact. **</reason>**
<answer>Funding cuts.</answer>

Figure 12: Case study 2 of MMSearch-R1.



Source: FVQA-test

Question: What historical event is depicted in this battle scene?

Answer: Battle of Flodden.

Direct Answer



GPT4o-20241120

Battle of Agincourt.



Qwen2.5-VL-72B

Battle of Agincourt.

On-Demand Search



MMSearch-R1-7B

<reason>The image depicts a battle scene with armored soldiers engaged in combat. The style of armor and the type of weaponry suggest it might be from a specific historical period. However, without additional context or identifiable markers, it's challenging to determine the exact historical event. The image does not provide enough specific details to identify the event confidently. **</reason>**
<search></search>



Image:  Title: Butchered at the Battle of Flodden - Warfare History Network


Image:  Title: IVÁN DUQUE - Stock Photos, Illustrations and Images - Album



Image:  Title: Reconstruction painting showing a clash between English and Scottish pikemen during the Battle of Flodden Field on the 9th of September 1513 (IC037/008) Archive Item - English Heritage(EH):Archive Collection | Historic England

Image:  Title: What was the composition of European medieval armies (military, middle ages, and history)? - Quora



<reason>The image search results suggest that the battle scene in the image is related to the "Battle of Flodden." The visual element from the search results is the title "Battle of Flodden." This indicates that the historical event depicted in the battle scene is likely the "Battle of Flodden," which took place on September 9, 1513, during the Scottish Wars of Independence. **</reason>**
<answer>Battle of Flodden.</answer>

Figure 13: Case study 3 of MMSearch-R1.