

# DORA: A Dual-Objective Reinforcement Learning Framework for Effective and Efficient Multimodal Agentic Search

Guangming Qin<sup>1\*</sup>, Yuhao Deng<sup>1</sup>, Yukun Zhao<sup>2</sup>, Zhenyang Li<sup>3</sup>,  
Junfeng Wang<sup>3</sup>, Dawei Yin<sup>3</sup>, Ye Yuan<sup>1†</sup>, Guoren Wang<sup>1</sup>,  
Yizhou Yan<sup>4</sup>, Chengliang Chai<sup>1†</sup>, Lei Cao<sup>5</sup>

<sup>1</sup>Beijing Institute of Technology, <sup>2</sup>Shandong University,

<sup>3</sup>Baidu Inc, <sup>4</sup>Meta, <sup>5</sup>University of Arizona

Correspondence: {qgm527, yuan-ye, ccl}@bit.edu.cn

## Abstract

The most recent research uses reinforcement learning (RL) to post-train Multi-modal Large Language Models (MLLMs) such that these models are able to iteratively call search engines to dynamically access external knowledge when handling complex Visual Question Answering (VQA) tasks. However, existing methods face two major limitations in effectiveness and efficiency: *i*) For effectiveness, the objective of these methods, which only considers the correctness of the generated final response, overlooks the quality of intermediate search results, thus leading to suboptimal search strategies. *ii*) For efficiency, existing methods often unnecessarily invoke search calls during reasoning, making the inference inefficient. To address these issues, we propose DORA, a customized dual-objective reinforcement learning framework to improve the search strategies of MLLMs, enhancing their search quality yet minimizing search frequency. The key ideas include (1) a reward function that promotes correct reasoning trajectories with fewer search calls; and (2) a dual-level optimization objective that jointly optimizes search quality and answer correctness. Extensive experiments on 3 real-world datasets demonstrate that DORA outperforms state-of-the-art methods, achieving up to 8.4% higher accuracy while reducing the number of search calls by 9.7%.

## 1 Introduction

Visual Question Answering (VQA) has become a central benchmark task to assess the reasoning, grounding, and multimodal understanding capabilities of MLLMs. To further enhance the reasoning ability, the most recent research (Wu et al., 2025) integrates MLLMs with external search engines. By iteratively retrieving external knowledge in real time, these models are able to access up-to-date

\*Work done while interning at Baidu.

†Ye Yuan and Chengliang Chai are corresponding authors.

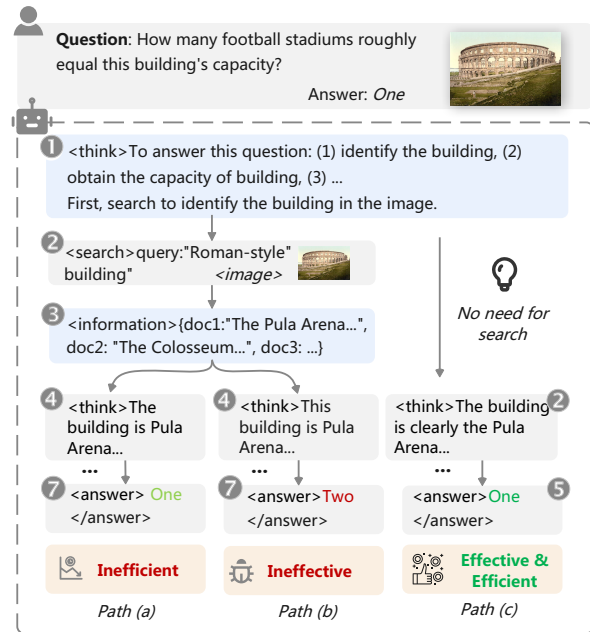


Figure 1: Examples of three reasoning paths for the same query: (a) a correct answer using 7 steps; (b) accurate retrieval but an incorrect final answer; and (c) a correct answer with 5 steps.

information during question answering, thus overcoming the inherent issue that their knowledge acquired through pre-training is limited. The typical pipeline follows a three-stage process—thinking, searching and answering—as illustrated in Figure 1. Specifically, the model first analyzes the input query, decomposes it, and decides whether external knowledge is required (thinking). Then, the model calls a search engine to retrieve relevant information (searching) and iterates this process until it gathers sufficient information. Finally, the model integrates this information with its own internal knowledge to produce an answer (answering). This search-augmented design has shown its advantage on knowledge-intensive VQA tasks over traditional approaches such as Retrieval Augmented Generation (RAG) (Lin et al., 2023; Caffagni et al., 2024; Cocchi et al., 2025).

To improve the performance of this new paradigm, recent work (Wu et al., 2025) has leveraged reinforcement learning (RL) with Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to post-train MLLMs. Unlike supervised approaches that rely on curated annotations, RL enables models to acquire search and reasoning strategies through feedback-driven exploration, offering better scalability and adaptability. For example, Search-R1 (Jin et al., 2025) employs PPO/GRPO to post-train LLMs to interact with web search, while MMSearch-R1 (Wu et al., 2025) extends this approach to multimodal retrieval, incorporating penalties to discourage excessive tool usage.

Despite these advances, existing RL-based agentic methods still suffer from two fundamental limitations. First, they primarily optimize the correctness of the final answers while overlooking the quality of intermediate retrieval results. This often produces suboptimal learning signals. For example, as shown in Figure 1, *Path(b)* produces an incorrect final answer despite correct retrieval. Based on their current reward functions, the entire reasoning path will be penalized, rather than solely penalizing the final answering process, preventing the model from learning from positive retrieval signals.

Second, current methods do not adequately address search efficiency. Excessive search calls increase the inference cost and token consumption, and risk introducing noisy information. As shown in Figure 1, *Path(a)* and *Path(c)* both yield correct answers, yet *Path(a)* solves the problem with more search calls than *Path(c)*. Although both succeed, the *Path(c)* is clearly more efficient. Current frameworks generally do not differentiate such cases, often rewarding both pipelines equally. This tends to limit search efficiency (Wu et al., 2025).

**Our proposal.** To overcome the limitations mentioned above, in this paper, we propose DORA, a customized dual-objective reinforcement learning framework to improve the performance of MLLMs in knowledge-intensive visual question answering (KI-VQA), focusing on training MLLMs to better interact with search engines. Specifically, we introduce two innovative ideas to achieve this goal: a search efficiency reward function and a search quality driven dual optimization objective.

(1) **Search efficiency reward.** The group sampling mechanism of GRPO requires a model to sample multiple reasoning paths for a specific problem. Our reward strategy assigns larger rewards to correct paths with fewer search calls, thereby

reducing search frequency, yet preserving answer accuracy. To achieve this, we design a group-wise efficiency reward that ranks correct paths by their search efficiency within group, assigning higher rewards to those achieving accurate answers with fewer search calls. This provides a fine-grained supervision signal that encourages models to balance correctness and efficiency during reasoning.

(2) **Dual optimization objective.** The dual optimization objective includes *i*). the *search objective*, which focuses on optimizing search quality and is applied only to the search process; and *ii*). the *answer objective*, which targets answer accuracy and is applied throughout the reasoning path. This design enables the model to learn a more effective search policy, which in turn leads to more accurate answers.

**Contributions.** Our main contributions include:

(1) We point out two key limitations in effectiveness and efficiency for current RL-based agentic search methods: they mainly optimize the final answer accuracy while overlooking the quality of intermediate retrievals, and they fail to encourage efficient search strategies.

(2) We introduce a novel RL framework that addresses these limitations through a group-wise efficiency reward and a dual optimization objective, enabling MLLMs to learn to *retrieve precise knowledge*, and *do so efficiently*.

(3) Extensive experiments on multiple multimodal VQA benchmarks demonstrate that DORA significantly improves retrieval quality, reduces redundant search usage, and achieves up to 7.8% improvements in answer accuracy while cutting search frequency by 9.7% compared to baselines.

## 2 Preliminary

*Group Relative Policy Optimization* (GRPO) for LLMs (Shao et al., 2024; DeepSeek-AI et al., 2025) is based on the policy gradient framework, where the main idea is to improve the policy by guiding LLMs to generate responses with higher advantages. The advantages are produced by an advantage function which, given an input question, compares a group of responses that the model samples and assigns an advantage score to each response.

Specifically, given an input  $q$ , a group of  $G$  candidate responses  $\{o_i\}_{i=1}^G$  are sampled from policy  $\pi_{\text{old}}$ . Each response  $o_i$  (corresponding to a reasoning path) is assigned a reward  $r(q, o_i)$ . GRPO computes the token-level advantage for  $o_i$  by nor-

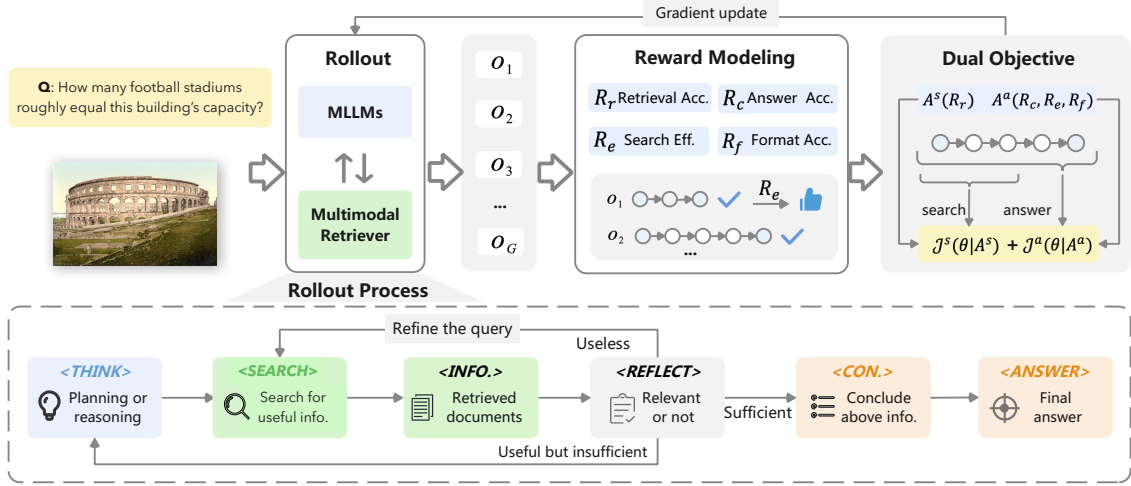


Figure 2: Overview of DORA framework (Top: the training pipeline integrated with our dual-level objective and efficiency reward. Bottom: a detailed view of the rollout process of multimodal retrieval).

malizing the rewards within the sampled group:

$$A_{i,t} = \frac{r(q, o_i) - \text{mean}\{r(q, o_j)\}_{j=1}^G}{\text{std}\{r(q, o_j)\}_{j=1}^G} \quad (1)$$

where  $t$  represents the  $t$ -th token of response  $o_i$ .

The optimization goal of GRPO is to maximize the expected advantage-weighted log probability of responses generated by LLMs, which could be formulated as follows:

$$\mathcal{J}(\theta) = \mathbb{E} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \left( \rho_{i,t} A_{i,t}, \right. \right. \\ \left. \left. \text{clip}(\rho_{i,t}, 1 - \varepsilon, 1 + \varepsilon) A_{i,t} \right) - \beta \mathbb{D}_{\text{KL}} [\pi_{\theta} \parallel \pi_{\text{ref}}] \right] \quad (2)$$

where  $\rho_{i,t}$  represents the ratio of current to old policy (*i.e.*,  $\rho_{i,t} = \frac{\pi_{\theta}(o_{i,t}|q, o_{i,<t})}{\pi_{\text{old}}(o_{i,t}|q, o_{i,<t})}$ ). The function  $\text{clip}(\cdot)$  constrains this ratio to avoid overly large updates and  $\varepsilon$  is a hyperparameter that represents the tolerance to policy deviation.  $\mathbb{D}_{\text{KL}} [\pi_{\theta} \parallel \pi_{\text{ref}}]$  represents the KL divergence loss that prevents overly aggressive updates of the policy from a well-behaved reference policy  $\pi_{\text{ref}}$ .  $\beta > 0$  controls the strength of the divergence.

### 3 The Proposed DORA Approach

Figure 2 presents the overview of DORA, which is based on an MLLM equipped with a multimodal retriever, highlighting the rollout process and the optimization pipeline. Given an input query, the model follows a structured rollout process (*i.e.* reasoning path) — `<think>`, `<search>`, `<information>`, `<reflect>`, `<conclude>`, and `<answer>`. Specifically, the `<think>` and `<search>` stages guide the model to analyze the initial question and issue effective text-or image-based retrieval queries

to a multimodal retriever (e.g., LamRA (Liu et al., 2025), UniIR (Wei et al., 2023)) with a unified interface for multimodal inputs. The `<information>` stage integrates the retrieved information into the reasoning path. See Appendix A for details of multimodal retrieval.

In particular, we further incorporate a `<reflect>` stage (Sec. 3.3) to evaluate the retrieved information. If it is useless, the model will refine the query and perform another search. If useful information is found, but not sufficient to answer the question, the model will summarize the key evidence and continue subsequent retrieval and reasoning. If it is sufficient to answer, the model proceeds to `<conclude>` to aggregate information across the entire reasoning path, and finally invokes `<answer>` to produce the final response.

To enhance the effectiveness and efficiency of these reasoning steps simultaneously, DORA first designs a comprehensive reward modeling strategy to guide the model to produce correct and efficient reasoning responses. The reward strategy explicitly evaluates both the search phase (Sec. 3.1.1) and the answer phase (Sec. 3.1.2). Building upon these multi-dimensional rewards, DORA further formulates dual optimization objective (Sec. 3.2) that integrate retrieval supervision with answering optimization, achieving end-to-end improvement in both search efficiency and answer accuracy.

#### 3.1 Rewards

Here, we introduce multiple types of reward that supervise different aspects of the reasoning pipeline.

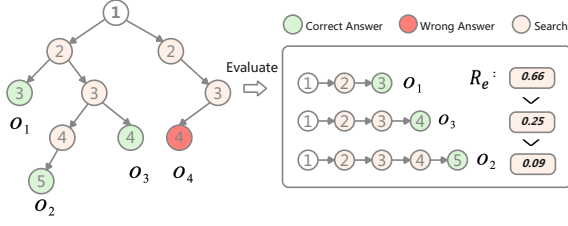


Figure 3: Design of efficiency reward.

### 3.1.1 Reward in the Search Phase

In the search phase, given a response  $o_i$ , a model executes search calls and retrieves a set of documents  $\mathcal{D}(o_i)$ . We define the retrieval reward to measure the quality of these retrieved documents as follows :

$$R_r(o_i) = r_h \cdot \mathbb{1}[\mathcal{D}(o_i) \cap \mathcal{D}^* \neq \emptyset] \quad (3)$$

where  $\mathcal{D}^*$  represents the set of golden documents.  $\mathbb{1}[\cdot]$  represents an indicator function, which yields 1 when its condition is met and 0 otherwise.  $r_h > 0$  is a constant. This reward ensures that any response retrieving at least one correct document receives positive reinforcement, regardless of the final answer. In contrast to prior methods that let the correctness of the final answer dictate the supervision of retrieval, our design allows the model to benefit from positive retrieval behavior even if reasoning later fails. This retrieval reward thus provides a direct training signal for search quality, enabling the model to develop more accurate and robust retrieval strategies.

### 3.1.2 Reward in the Answer Phase

The reward in the answer phase consists of three components as follows.

**Answer correctness reward.** To evaluate whether a response  $o_i$  produces the correct answer, we define an answer correctness reward function  $R_c(\cdot)$ . Given the final output  $O_i$  and the ground-truth answer  $O_i^*$ , the reward is defined as

$$R_c(o_i) = \phi(O_i, O_i^*) \quad (4)$$

where  $\phi(\cdot)$  measures the correctness of the response, e.g., using exact match or a large language model as a judge.

**Efficiency reward.** To reduce the number of redundant search calls that do not help answer questions, we design an efficiency reward that provides fine-grained supervision within each GRPO sampled group. In Figure 3, this reward evaluates all correct responses in each group and allocates rewards proportional to their search efficiency, allowing the

model to gradually develop a preference for concise and correct search behaviors.

Specifically, for each query, the model generates a group of candidate responses  $\mathcal{G} = \{o_1, \dots, o_G\}$ . Each may involve multiple search and reasoning steps. Let  $T(o_i)$  denote the number of search calls used by response  $o_i$ .

Among all correct responses ( $R_c(o_i) = 1$ ), we assign an *efficiency score* inversely proportional to the number of search calls:

$$S(o_i) = -\alpha \cdot T(o_i) \text{ s.t. } R_c(o_i) = 1 \quad (5)$$

where  $\alpha > 0$  is a scaling factor that prefers fewer search calls. The efficiency reward is then computed using group-wise softmax normalization:

$$R_e(o_i, \mathcal{G}) = \frac{\exp(S(o_i))}{\sum_{o_j \in \mathcal{G}, R_c(o_j)=1} \exp(S(o_j))} \quad (6)$$

This formulation provides a smooth and informative signal that reflects the relative efficiency among all correct responses. For example, as shown in Figure 3, response  $o_1$  that achieves a successful answer with the fewer search calls receives a higher reward, while response  $o_2$ , which is correct but less efficient, obtains a non-zero but smaller reward. This encourages the model to balance correctness and efficiency, progressively shaping a reasoning policy that identifies necessary knowledge with minimal tool usage.

**Format reward.** In addition to correctness and efficiency, it is crucial that the response adheres to a valid structural format. Similar to existing RL frameworks for tool-augmented methods (Wu et al., 2025; Zheng et al., 2025), we incorporate a format reward  $R_f(o_i)$  that verifies whether the generated response adheres to the predefined token order (*i.e.* `<think>`  $\rightarrow$  `<search>`  $\rightarrow$  `<information>`  $\rightarrow$  `<reflect>`  $\rightarrow$  `<conclude>`  $\rightarrow$  `<answer>`). Any structural violations, such as missing `<reflect>` after retrieval or improperly formatted search queries, are unable to pass the format check. Formally, the format reward is defined as:

$$R_f(o_i) = \frac{N_{\text{pass}}(o_i)}{N_{\text{total}}} \quad (7)$$

where  $N_{\text{pass}}$  represents the number of passing format checks and  $N_{\text{total}}$  denotes the total number of such checks.

## 3.2 Dual Optimization Objective

To integrate the two levels of reward, we adopt dual objective under the GRPO framework. Each sampled response  $o_i$  is assigned both a search reward

and an answer reward. The search reward of  $o_i$  captures the retrieval accuracy, which is defined as:

$$r_s(q, o_i) = R_r(o_i) \quad (8)$$

The answer reward of  $o_i$  evaluates the retrieval efficiency, final correctness and format validity, which is defined as:

$$r_a(q, o_i) = R_c(o_i) + R_f(o_i) + R_e(o_i, \mathcal{G}) \quad (9)$$

Following GRPO, both rewards are normalized within the group to construct token-level advantages  $A_{i,t}^s$  and  $A_{i,t}^a$ :

$$A_{i,t}^s = \frac{r_s(q, o_i) - \text{mean}\{r_s(q, o_j)\}_{j=1}^G}{\text{std}\{r_s(q, o_j)\}_{j=1}^G} \quad (10)$$

$$A_{i,t}^a = \frac{r_a(q, o_i) - \text{mean}\{r_a(q, o_j)\}_{j=1}^G}{\text{std}\{r_a(q, o_j)\}_{j=1}^G} \quad (11)$$

Based on search and answer advantages, we further build the search objective and the answer objective. We define  $\mathcal{I}(o_i, t)$  as a token mark function.  $\mathcal{I}(o_i, t) = 1$  indicates that the  $t$ -th token of  $o_i$  is a search token and an answering token otherwise. Therefore, the search objective can be defined as follows:

$$\mathcal{J}^s(\theta) = \mathbb{E} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t \in \mathcal{T}_r(o_i)} \min \left( \rho_{i,t} A_{i,t}^s, \text{clip}(\rho_{i,t}, 1 - \varepsilon, 1 + \varepsilon) A_{i,t}^s \right) - \beta_s \mathbb{D}_{\text{KL}} [\pi_\theta \| \pi_{\text{ref}}] \right] \quad (12)$$

where  $\mathcal{T}_r(o_i) = \{t | \mathcal{I}(o_i, t) = 1\}$  and  $\beta_s > 0$  controls the strength of the KL divergence.

The answer objective is defined as follows:

$$\mathcal{J}^a(\theta) = \mathbb{E} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \left( \rho_{i,t} A_{i,t}^a, \text{clip}(\rho_{i,t}, 1 - \varepsilon, 1 + \varepsilon) A_{i,t}^a \right) - \beta \mathbb{D}_{\text{KL}} [\pi_\theta \| \pi_{\text{ref}}] \right] \quad (13)$$

To enable an end-to-end optimization of both search and answer, We combine the search objective and answer objective as dual optimization objective. To balance the strength of search optimization and answer optimization, we further employ a dynamic weight annealing strategy that gradually shifts the optimization focus from the search objective to the answer objective as training progresses. Specifically, the dual objective at training step  $\tau$  is defined as:

$$\mathcal{J}_{\text{total}}(\tau; \theta) = \alpha_\tau \cdot \mathcal{J}^s(\theta) + \beta_\tau \cdot \mathcal{J}^a(\theta) \quad (14)$$

where  $\alpha_\tau$  and  $\beta_\tau$  are time-dependent weights controlling the relative importance of the two objectives.

### 3.3 Reflection

After obtaining the retrieved information, previous rollouts (Jin et al., 2025; Wu et al., 2025) usually incorporate the text into the reasoning chain without active evaluation. However, given that not all information is relevant to the question, we introduce a <reflect> block after each retrieval to explicitly assess the relevance of the information, summarize key points, or identify retrieval errors and propose a refined query.

Existing MLLMs (Shinn et al., 2023; Wang et al., 2025) typically apply reflection mechanisms after the reasoning processes have been completed, particularly for correcting or verifying results. Instead, DORA integrates reflection immediately after retrieving information, aiming to filter out irrelevant information and focus on useful content. Concretely, we design a dedicated prompt to guide the reflection behavior, as shown below:

#### Prompt snippet for the <reflect> stage

```
[Reflection Stage]
After each <information> block, you MUST
include one <reflect> block.

Assess whether the retrieved content is
relevant to the query in the previous
<search> block.

- If irrelevant, refine the query and
search again.
- If partially relevant but insufficient
for the original question, summarize
key points and continue reasoning.
- If sufficiently relevant and
informative, move to <conclude> for
integration and final answering.
```

This design mitigates the negative impact of noisy data and allows the reflection process to be a target of optimization through reinforcement learning. Therefore, it enables the model to concurrently fine-tune the retrieval accuracy and subsequent answering.

Specifically, the rollout may undergo several iterations until enough evidence is gathered. The iterations could be formulated as:

<search> → <information> → <reflect>

Then, it generates a <conclude> token, synthesizing the reasoning and collected information, followed by a brief final response in <answer>.

## 4 Experiments

Our experiments focus on answering the following three key questions in both effectiveness and efficiency:

- *Q1: Method Effectiveness.* How does our framework enhance retrieval and QA performance on comprehensive multi-modal benchmarks compared to various existing works?
- *Q2: Search Efficiency.* Does the model learn to use fewer search calls while maintaining strong performance?
- *Q3: Ablation Studies.* How do the dual optimization objective, efficiency reward and reflection respectively affect the performance?

### 4.1 Experiment Setup

**Datasets.** We use three knowledge-based VQA datasets: Enc-VQA (Mensink et al., 2023), InfoSeek (Chen et al., 2023), and SK-VQA (Su et al., 2024). Following prior work (Wu et al., 2025; Cocchi et al., 2025), we curate subsets to emphasize multi-hop reasoning while controlling training cost. Specifically, for Enc-VQA, we directly use its 2-hop subset (11.5k samples) to emphasize multi-step reasoning; for InfoSeek and SK-VQA, we employ filtering based on question types (e.g., How, What) and image domains (e.g., animals, landmarks). This process yields three curated subsets containing approximately 11,500, 15,000, and 15,000 samples, respectively, which are then used to train our model and baseline methods.

**Baselines.** We compare with three categories: (1) Inference without retrieval: Qwen2.5-VL (Bai et al., 2025), Intern3VL (Zhu et al., 2025), GPT-4o (OpenAI et al., 2024) and Gemini-2.5-Pro (Comanici et al., 2025). (2) Retrieval-augmented generation methods: RA-VQAv2 (Lin et al., 2023), WikiLLaVA (Caffagni et al., 2024), EchoSight (Yan and Xie, 2024) and ReflectiVA (Cocchi et al., 2025). RA-VQAv2 performs direct retrieval using the image-text input, while WikiLLaVA, EchoSight, and ReflectiVA implement a two-stage retrieval approach, initially obtaining candidate documents based on the image and subsequently re-ranking or using LLM-based selection to choose top paragraphs as model context. (3) Search engine as tool: Search-R1 (Jin et al., 2025) and MMSearch-R1 (Wu et al., 2025). For inference-without-retrieval baselines, we prompt them to think first

and then give a concise answer. For RAG and search-as-tool baselines, to ensure a fair comparison, we replace their original retrievers with the same one used in our approach (*i.e.*, LamRA), and ensure all models access a consistent knowledge base organized in chunk-level units. For Search-R1, we adapt it to the multimodal setting by replacing its base model with Qwen2.5-VL (7B).

**Evaluation settings.** For effectiveness, we use exact match (EM) (Jin et al., 2025), which is illustrated in Section 3.1, as the evaluation metric for answer accuracy evaluation across all the datasets. For efficiency, we measure the average number of search calls for each method. For retrieval accuracy evaluation, we use the recall@5 as the evaluation metric. To be specific, given retrieved passages set  $D_{\text{top5}}(o_i)$  and gold information set  $D^*$ , we compute  $\text{Recall@5} = \frac{1}{N} \sum_{i=1}^N \mathbb{1} [D_{\text{top5}}(o_i) \cap D^* \neq \emptyset]$ , where  $N$  is the total number of queries.

**Implementation details.** We train Qwen2.5-VL-Instruct (3B,7B) (Bai et al., 2025) using the verl (Sheng et al., 2025) framework. Each training step includes 128 queries with 6 rollouts. Models are allowed up to 5 search calls per query, retrieving 3 passages per call, aligned with the approach in (Jin et al., 2025). We adopt the competitive retriever LamRA (Liu et al., 2025) as the default tool, ensuring that retrieval does not become a performance bottleneck of DORA. We employ a knowledge base curated through Enc-VQA, Infoseek and SK-VQA. To ensure fairness, all retrieval-based models use the same retriever (LamRA) and an identical knowledge base. We start with  $\alpha_0$  set to 0.7 and  $\beta_0$  assigned a value of 0.3, subsequently modifying them dynamically. We analyze the sensitivity of these hyper-parameters in Appendix C.2.

### 4.2 Effectiveness

Table 1 reports results on three benchmarks. Our 7B model outperforms all baselines across all datasets. DORA improves the accuracy by +6.9 - 10.2% compared to MMSearch-R1 because it only rewards the correctness of the final answer, but DORA explicitly rewards the quality of intermediate retrieval and penalizes redundant searches. DORA also consistently exceeds all RAG-based and inference without retrieval methods. For example, DORA gains +55.11% for Qwen2.5-VL-3B and +45.47% improvement for Qwen2.5-VL-7B. This is because our dual objective leads to more reliable retrieval and better answering.

Method	Model Size	Enc-VQA-test	InfoSeek-test	SK-VQA-test	Average
<i>Inference without Retrieval</i>					
Qwen2.5-VL (Bai et al., 2025)	3B	7.41	12.25	18.79	12.81
Qwen2.5-VL	7B	12.79	16.43	23.15	17.46
Qwen2.5-VL	72B	18.92	24.16	36.76	26.61
InternVL3 (Zhu et al., 2025)	8B	13.01	20.09	29.57	20.89
GPT-4o (OpenAI et al., 2024)	-	19.03	30.69	35.96	28.56
Gemini2.5-Pro (Comanici et al., 2025)	-	19.14	31.86	34.94	28.64
<i>RAG Methods</i>					
RA-VQAv2 (Lin et al., 2023)	3B	13.14	29.31	32.18	24.87
Wiki-LLaVA (Caffagni et al., 2024)	7B	14.95	35.18	34.96	28.36
EchoSight (Yan and Xie, 2024)	7B	16.23	34.58	37.06	29.29
ReflectiVA (Cocchi et al., 2025)	7B	22.30	42.16	38.12	34.19
<i>Search Engine as Tool</i>					
Search-R1 (Jin et al., 2025)	7B	23.04	41.82	36.90	33.92
MMSearch-R1 (Wu et al., 2025)	7B	23.19	41.65	37.18	34.00
<b>DORA-small</b>	3B	20.08	40.71	38.06	32.95
<b>DORA</b>	7B	<b>25.01</b>	<b>44.53</b>	<b>40.98</b>	<b>36.84</b>

Table 1: Performance comparison of all methods. The accuracy scores (%) are computed based on exact match.

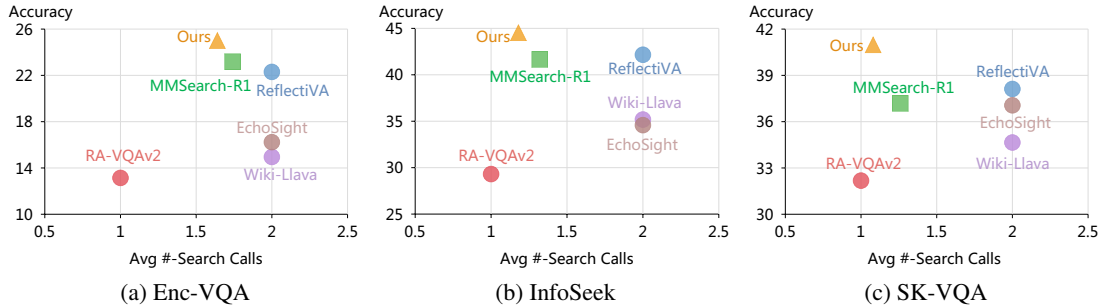


Figure 4: Search efficiency of different models across three benchmarks.

Method	Enc-VQA Recall@5	InfoSeek Recall@5	SK-VQA Recall@5
UniIR	17.06	46.97	47.85
LamRA	19.96	55.45	52.08
ReflectiVA	29.98	63.10	58.95
MMSearch-R1	31.82	62.98	58.18
<b>DORA</b>	<b>35.47</b>	<b>65.13</b>	<b>60.65</b>

Table 2: Retrieval performance comparison (Recall@5).

Furthermore, to demonstrate the effectiveness of our search optimization, we compare our method with several advanced multimodal retrieval models. As shown in Table 2, DORA consistently outperforms strong retrieval models (e.g. UniIR (Wei et al., 2023), LamRA) and retrieval strategies in ReflectiVA and MMSearch-R1. The reason is that the retrieval reward directly supervises search quality, teaching the model to dynamically issue precise queries. In contrast, RAG methods rely on static retrieval and MMSearch-R1 lacks explicit retrieval

supervision, leading to lower search quality. To further illustrate the behavior of DORA, we provide case studies in Appendix E.

Method / Type	Enc-VQA	InfoSeek	SK-VQA
<b>Effective</b>			
MMSearch-R1	1.23	0.90	0.81
DORA	<b>1.28</b>	<b>0.94</b>	<b>0.88</b>
<b>Irrelevant</b>			
MMSearch-R1	0.51	0.42	0.45
DORA	<b>0.36</b>	<b>0.24</b>	<b>0.20</b>
<b>Insufficient</b>			
MMSearch-R1	1.10	0.56	0.52
DORA	<b>1.03</b>	<b>0.51</b>	<b>0.49</b>

Table 3: Distribution of search call types. Average number of search calls per query for each search type is used as metric.

### 4.3 Search Efficiency

Figure 4 illustrates the trade-off between accuracy (y-axis) and search calls (x-axis). DORA

achieves higher accuracy with fewer search calls compared to both RAG-based and search-as-tool baselines. For example, a RAG-based method like RA-VQAv2 uses a single retrieval, resulting in low search frequency but noticeably lower accuracy due to limited external knowledge. Other RAG methods (Wiki-LLaVA, EchoSight, ReflectiVA) adopt the two-stage retrieval pipeline, which improves accuracy but incurs more redundant calls. In contrast, our efficiency reward learns to invoke search only when necessary, achieving a better balance between efficiency and accuracy. On the other hand, search engine as tool methods like MMSearch-R1 often rely heavily on search tools. DORA reduces the frequency of use of search tools while maintaining similar levels of accuracy, highlighting the success of our reward design. Moreover, we also provide a computational cost analysis in the Appendix C.3.

In addition, to better understand the sources of efficiency improvements of DORA over the baselines, namely which types of search calls are reduced, we group search calls into three types: *effective* (the retrieved information can answer the retrieval query), *irrelevant* (the retrieved information is unrelated to the retrieval query), and *insufficient* (the retrieved information is relevant but inadequate for final answering), and report the average number of search calls per query for each category. As shown in Table 3, DORA consistently reduces *irrelevant* search calls compared to MMSearch-R1 (e.g., from 0.51 to 0.36 on Enc-VQA), while preserving and slightly increasing *effective* ones (e.g., from 1.23 to 1.28 on Enc-VQA). At the same time, *insufficient* search calls are also moderately reduced. Overall, the improvement in search efficiency mainly comes from the reduction of irrelevant and insufficient searches rather than a uniform decrease in search frequency.

#### 4.4 Ablation Studies

We conduct ablation studies to examine how each component influences QA accuracy and search efficiency, as summarized in Table 4.

**Efficiency reward.** Removing the efficiency reward barely impacts accuracy, but increases search calls by 59% on average relative to DORA (full). This is because the model no longer distinguishes efficient from inefficient successful paths, resulting in overuse of retrieval.

**Dual optimization objective.** Eliminating the dual optimization objective has a negligible effect on search calls, but decreases the accuracy of the QA

Method	Enc-VQA	InfoSeek	SK-VQA
<i>Accuracy</i>			
DORA (full)	25.01	44.53	40.98
-w/o DualObj	22.08	42.14	38.27
-w/o Reflection	23.77	42.86	39.15
-w/o EffReward	25.16	45.02	41.28
<i>Avg #-Search Calls</i>			
DORA (full)	1.64	1.18	1.08
-w/o DualObj	1.68	1.16	1.18
-w/o Reflection	1.62	1.15	1.04
-w/o EffReward	2.24	2.02	1.94

Table 4: Impact of each component on accuracy and average search calls. DualObj, Reflection, and EffReward refer to the dual-level objective, reflection mechanism, and efficiency reward, respectively.

by approximately 7.5%. This is because without separating search and answer objectives, the model cannot align retrieval quality with answer correctness, leading to suboptimal search usage.

**Reflection.** Removal of the reflection mechanism leads to a 3% – 4% drop in accuracy and slightly fewer search calls, as noisy passages are not filtered and the model is less likely to search when faced with uninformative content. Additional ablation studies can be found in Appendix C.1, C.4

## 5 Related Work

Recently, reinforcement learning has become a promising method for optimizing LLMs to employ search tools for acquiring external knowledge. For instance, Search-R1 (Jin et al., 2025) shows that reinforcement learning with a reward based solely on answer correctness helps LLMs understand when to search and how to incorporate multi-turn evidence. Similarly, DeepRetrieval (Jiang et al., 2025b) demonstrates that employing retrieval metrics as rewards enables effective query generation without costly supervision, allowing LLMs to efficiently find and utilize information. MMSearch-R1 (Wu et al., 2025) integrates reinforcement learning with *multimodal retrieval*. It utilizes GRPO to minimize redundant search calls while improving accuracy with search-penalized reward. However, their optimization strategy has limitations. First, they focus on ensuring the correctness of the final answer, while overlooking the quality of search. Second, they improve search efficiency by simply rewarding responses that avoid invoking search calls, inadequately tackling complex queries that need multiple retrieval steps. Unlike these works, DORA uses fine-granularity reward signals

to distinguish retrieval and answering and employs flexible strategies to boost retrieval efficiency. We introduce more related work in Appendix B.

## 6 Conclusion

We proposed DORA, a Dual-Objective Reinforcement Learning framework for multimodal agentic search. DORA jointly optimizes retrieval quality and reasoning efficiency through a search efficiency reward and a dual optimization objective. A built-in reflection mechanism further ensures retrieval relevance. Experiments on Enc-VQA, InfoSeek, and SK-VQA show that DORA significantly improves answer accuracy, retrieval quality, and search efficiency over state-of-the-art baselines.

## Limitations

DORA requires some effort in tuning hyperparameter and reward scores to properly balance the local and global objectives. In addition, the current framework focuses on text-image multimodal reasoning, and extending it to more complex modalities such as video or interactive multi-agent environments is left for future work.

## Acknowledgments

Chengliang Chai is supported by the NSFC (U25B2019, 62472031), the National Key Research and Development Program of China (2024YFC3308200), Beijing Nova Program, and Huawei. Ye Yuan is supported by the NSFC (Grant Nos. 62225203, 62532001), Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (Grant No. JYB2025XDXM108), Beijing Natural Science Foundation (Grant No. L241010). Guoren Wang is supported by the NSFC (62427808, U2001211), and the Liaoning Revitalization Talents Program (XLYC2204005). Lei Cao is supported by the NSF (DBI-2327954) and Amazon Research Awards.

## References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuezhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. [Qwen2.5-vl technical report](#). *Preprint*, arXiv:2502.13923.

Davide Caffagni, Federico Cocchi, Nicholas Moratelli, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and

Rita Cucchiara. 2024. Wiki-llava: Hierarchical retrieval-augmented generation for multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1818–1826.

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?](#) *Preprint*, arXiv:2302.11713.

Federico Cocchi, Nicholas Moratelli, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2025. Augmenting multimodal llms with self-reflective tokens for knowledge-based visual question answering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9199–9209.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.

Yihao Ding, Kaixuan Ren, Jiabin Huang, Siwen Luo, and Soyeon Caren Han. 2024. [Pdf-mvqa: A dataset for multimodal information retrieval in pdf-based visual question answering](#). *Preprint*, arXiv:2404.12720.

Negar Foroutan, Angelika Romanou, Matin Ansari-pour, Julian Martin Eisenschlos, Karl Aberer, and Rémi Le-bret. 2025. [Wikimixqa: A multimodal benchmark for question answering over tables and charts](#). *Preprint*, arXiv:2506.15594.

Dongze Hao, Qunbo Wang, Longteng Guo, Jie Jiang, and Jing Liu. 2024. Self-bootstrapped visual-language model for knowledge selection and question answering. *arXiv preprint arXiv:2404.13947*.

Ziniu Hu, Ahmet Iscen, Chen Sun, Kai-Wei Chang, Yizhou Sun, David Ross, Cordelia Schmid, and Alireza Fathi. 2023. Avis: Autonomous visual information seeking with large language model agent. *Advances in Neural Information Processing Systems*, 36:867–878.

Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, and 1 others. 2025a. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*.

- Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, Peng Gao, Yu Liu, Chunyuan Li, and Hongsheng Li. 2024. [Mmsearch: Benchmarking the potential of large models as multi-modal search engines](#). *Preprint*, arXiv:2409.12959.
- Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025b. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223*.
- 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*.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. [Let’s verify step by step](#). *Preprint*, arXiv:2305.20050.
- Weizhe Lin, Jinghong Chen, Jingbiao Mei, Alexandru Coca, and Bill Byrne. 2023. [Fine-grained late-interaction multi-modal retrieval for retrieval augmented visual question answering](#). *Preprint*, arXiv:2309.17133.
- Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. 2022. Revive: Regional visual representation matters in knowledge-based visual question answering. *Advances in neural information processing systems*, 35:10560–10571.
- Yikun Liu, Yajie Zhang, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiangchao Yao, Yanfeng Wang, and Weidi Xie. 2025. Lamra: Large multimodal model as your advanced retrieval assistant. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4015–4025.
- Thomas Mensink, Jasper Uijlings, Lluís Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araujo, and Vittorio Ferrari. 2023. [Encyclopedic vqa: Visual questions about detailed properties of fine-grained categories](#). *Preprint*, arXiv:2306.09224.
- OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. [Hybridflow: A flexible and efficient rlhf framework](#). In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys ’25, page 1279–1297. ACM.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.
- Xin Su, Man Luo, Kris W Pan, Tien Pei Chou, Vasudev Lal, and Phillip Howard. 2024. Sk-vqa: Synthetic knowledge generation at scale for training context-augmented multimodal llms. *arXiv preprint arXiv:2406.19593*.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.
- Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhui Chen. 2025. VI-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*.
- Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhui Chen. 2023. [Uniir: Training and benchmarking universal multimodal information retrievers](#). *Preprint*, arXiv:2311.17136.
- Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. 2021. [Multi-modal answer validation for knowledge-based vqa](#). *Preprint*, arXiv:2103.12248.
- Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. 2025. Mmsearch-r1: Incentivizing llms to search. *arXiv preprint arXiv:2506.20670*.
- Hongshen Xu, Zihan Wang, Zichen Zhu, Lei Pan, Xingyu Chen, Shuai Fan, Lu Chen, and Kai Yu. 2025. Alignment for efficient tool calling of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17787–17803.
- Yibin Yan and Weidi Xie. 2024. [Echosight: Advancing visual-language models with wiki knowledge](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, page 1538–1551. Association for Computational Linguistics.
- Zhixin Zhang, Yiyuan Zhang, Xiaohan Ding, and Xiangyu Yue. 2024. [Vision search assistant: Empower vision-language models as multimodal search engines](#). *Preprint*, arXiv:2410.21220.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. 2025. *Deepeyes: Incentivizing "thinking with images" via reinforcement learning*. *Preprint*, arXiv:2505.14362.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, and 32 others. 2025. *InternV3: Exploring advanced training and test-time recipes for open-source multimodal models*. *Preprint*, arXiv:2504.10479.

## A Multimodal Retrieval

We elaborate here on how the multimodal retrieval in this paper is conducted. Our framework incorporates a popular multimodal retrieval model (LamRA (Liu et al., 2025)), which is capable of jointly processing textual and visual information. The concrete search process is conducted as follows.

During the rollout, whenever the agent model identifies missing external knowledge, it actively issues a `<search>...</search>` call. Inside this tag, the model generates a JSON-formatted retrieval query that contains two fields:

- "query": a textual question or keyword describing what should be retrieved.
- "with\_image": a flag ("yes" or "no") indicating whether the input image should be included in the retrieval request.

If "with\_image" = "yes", the multimodal retriever receives both the raw input image and the textual query. If "with\_image" = "no", it receives only the text query.

The multimodal retriever then encodes the input (including both the images and text) into a unified embedding. This embedding is matched against a pre-indexed, chunk-level Wikipedia-derived knowledge base, and the top- $k$  most similar chunks ( $k = 3$  as default in our paper) are returned. These retrieved chunks are then wrapped inside an `<information>...</information>` block and appended immediately after the `<search>` call. The agent model then enters the `<reflect>` phase, where it evaluates the relevance of the retrieved passages, summarizes useful evidence, or decides whether additional searches are needed. This iterative cycle—search, information integration, reflection—continues until the model determines that

sufficient knowledge has been gathered to produce a final answer.

## B Related Work on RAG

Given the limited domain-specific understanding of LLMs, retrieval-augmented generation (RAG) has been introduced to fetch external information to aid in response generation. Early studies (Wu et al., 2021; Lin et al., 2022, 2023; Hao et al., 2024) typically followed a static one-shot paradigm, retrieving evidence once and concatenating it with the input. These methods cannot adapt retrieval to task complexity and struggle with multi-hop reasoning. To address these limitations, more recent dynamic approaches (Jiang et al., 2024; Hu et al., 2023; Cocchi et al., 2025; Zhang et al., 2024) now support multi-step, decision-driven retrieval. MMSearch (Jiang et al., 2024) introduces a multimodal search pipeline and benchmark, emphasizing query reformulation, reranking, and summarization. AVIS (Hu et al., 2023) views retrieval as an agent-driven task with an LLM managing tool use via tree searches, while ReflectiVA (Cocchi et al., 2025) uses reflective tokens to decide when and what to retrieve. However, these methods still rely on fixed retrieval pipelines or curated datasets, limiting scalability and robustness. In contrast, our DORA dynamically and adaptively manages search calls, refining the retrieval process to be more responsive and efficient in handling complex queries.

## C Additional Experiments

### C.1 Analysis of Reward Signals

We have conducted an ablation study by varying the coefficients of the retrieval reward ( $r_r$ ), answer correctness reward ( $r_c$ ), efficiency reward ( $r_e$ ), and format reward ( $r_f$ ). Table 5 presents the performance across three benchmarks (Enc-VQA, InfoSeek, and SK-VQA) under different hyperparameter configurations. The setting ( $r_r = 1.0, r_c = 1.0, r_e = 0.5, r_f = 0.5$ ) is our default configuration.

Key findings and analysis:

**Varying  $r_r$ .** We first compare the default setting to a configuration without retrieval reward (i.e.,  $r_r = 0$ , row 1). We then observe that removing the retrieval reward leads to a clear decline in performance across all datasets. For instance, accuracy decreases from 44.53% to 41.81% on InfoSeek. This observation validates that relying solely on the final answer is insufficient and that the retrieval

reward provides the necessary signal to guide the model toward a high-quality answer.

**Varying  $r_c$ .** We also compare the default setting against a configuration lacking the answer correctness reward ( $r_c = 0$ , row 2). Setting  $r_c = 0$  causes a significant performance drop, with the accuracy falling from 25.01% to 14.23% in Enc-VQA. This confirms that while other rewards optimize the process, the answer correctness reward ( $r_c$ ) serves as the foundational learning signal.

**Varying  $r_e$ .** We first compare the default setting against configurations with removed ( $r_e = 0$ , row 3) or strengthened ( $r_e = 1.0$ , row 4) efficiency constraints. When the efficiency reward is removed ( $r_e = 0$ ), the model achieves high accuracy (e.g., 25.16% on Enc-VQA) but incurs a significant increase in the number of search calls (2.24 on Enc-VQA). In contrast, when the efficiency penalty is increased to  $r_e = 1.0$ , the search calls drop to their lowest point (1.46 on Enc-VQA), but this results in poor accuracy. Therefore, a moderate penalty ( $r_e = 0.5$ ) is essential to achieve a better trade-off between resource usage and performance.

**Varying  $r_f$ .** We compare the default setting against the configuration with removed ( $r_f = 0$  row 5) and increased ( $r_f = 1.0$  row 6). We observed a slight drop in performance and an increase in search calls compared to the default. This is because removing the format reward causes the model to produce invalid outputs (e.g., incorrect search-call formats), which makes reasoning more error-prone and reduces performance. Moreover, without this constraint, the model attempts more searches because it no longer risks being penalized for malformed search calls.

Note that while this default setting is robust, fine-grained tuning of these hyperparameters by users for specific downstream tasks could potentially yield even better results.

## C.2 Hyperparameter Analysis

We further examine two key hyperparameters in our method: the GRPO clipping threshold ( $\epsilon$ ) and the time-dependent dual-objective weights ( $\alpha_\tau, \beta_\tau$ ), where our experiments focus specifically on the effect of their initial values.

Key findings and analysis:

- We conduct an ablation study with  $\epsilon$  values of 0.1, 0.2, and 0.3 to investigate its impact. We observe that  $\epsilon = 0.2$  produces the highest accuracy, while

the settings  $\epsilon = 0.1$  and  $\epsilon = 0.3$  lead to performance degradation (1.6% and 0.9% on average, respectively). This is expected: the purpose of  $\epsilon$  is to control the magnitude of model updates. A relatively large value ( $\epsilon = 0.3$ ) results in aggressive model updates, leading to insufficient stability and suboptimal performance. In contrast, a relatively small value ( $\epsilon = 0.1$ ) causes overly conservative updates, lacking in exploration, which also degrades performance.  $\epsilon = 0.2$  balances stability and exploration, thereby achieving superior results.

- To investigate the impact of  $\alpha_\tau$  and  $\beta_\tau$  on our model’s performance, we fix their sum to 1 and adjust their ratio for testing. We find that the model achieves the best performance when  $\alpha_\tau = 0.7$  and  $\beta_\tau = 0.3$ . Further increasing the proportion of  $\alpha_\tau$  (0.8) while decreasing that of  $\beta_\tau$  (0.2) leads to a performance drop of 0.8% on average. As  $\alpha_\tau$  continues to decrease and  $\beta_\tau$  increases accordingly (from 0.3 to higher values), the model’s performance consistently degrades, falling on average from 36.84% to 35.57%. This aligns with our expectations: in our work,  $\alpha_\tau$  decreases over time while  $\beta_\tau$  increases. Here, we set their initial values—a relatively large initial  $\alpha_\tau$  ensures that the search objective dominates the early training phase, allowing the model to fully learn effective retrieval strategies. As training progresses,  $\beta_\tau$  gradually increases, allowing the model to build on the well-learned retrieval capabilities to further optimize answer generation. This naturally leads to superior results. However, an excessively large initial  $\alpha_\tau$  (e.g.,  $\alpha_\tau = 0.8$ ) causes the model to delay its focus on answer accuracy during training. Consequently, despite achieving high-quality retrieval, the model fails to generate correct answers.

## C.3 Computational Cost

Here, we provide a computational cost comparison against the two most relevant agentic retrieval baselines: **Search-R1** and **MMSearch-R1**. Following recent work on efficient tool calling (Xu et al., 2025; Jiang et al., 2025a), we report: (i) latency per query measured using VLLM on  $8 \times A800$  GPUs, (ii) the average number of search calls (Avg-#Search calls), and (iii) a Utility metric defined in (Xu et al., 2025) as  $\text{Utility} = \text{Accuracy} - \alpha \cdot \text{Avg-#Search calls}$ , which balances answer quality and retrieval cost. A higher utility indicates better

Reward Setting				Accuracy			Avg.# Search Calls		
$r_r$	$r_c$	$r_e$	$r_f$	Enc-VQA	InfoSeek	SK-VQA	Enc-VQA	InfoSeek	SK-VQA
0	1.0	0.5	0.5	23.06	41.81	36.98	1.66	1.23	1.18
1.0	0	0.5	0.5	14.23	23.98	21.99	1.87	1.24	1.20
1.0	1.0	0	0.5	25.16	45.02	41.28	2.24	2.02	1.94
1.0	1.0	1.0	0.5	24.31	43.10	38.99	1.46	1.06	0.95
1.0	1.0	0.5	0	24.02	42.89	38.90	1.78	1.29	1.21
1.0	1.0	0.5	1.0	23.75	42.13	38.72	1.67	1.10	1.02
1.0	1.0	0.5	0.5	25.01	44.53	40.98	1.64	1.18	1.08

Table 5: Performance and average number of search calls under different reward-signal combinations ( $r_r, r_c, r_e, r_f$ ).

Hyperparameter	Enc-VQA	InfoSeek	SK-VQA
$\epsilon = 0.1$	24.76	44.10	39.87
$\epsilon = 0.2$	<b>25.01</b>	<b>44.53</b>	<b>40.98</b>
$\epsilon = 0.3$	24.64	44.24	40.63
$\alpha_\tau = 0.8, \beta_\tau = 0.2$	24.76	44.13	40.64
$\alpha_\tau = 0.7, \beta_\tau = 0.3$	<b>25.01</b>	<b>44.53</b>	<b>40.98</b>
$\alpha_\tau = 0.6, \beta_\tau = 0.4$	24.36	44.08	40.25
$\alpha_\tau = 0.5, \beta_\tau = 0.5$	24.12	44.02	40.26
$\alpha_\tau = 0.4, \beta_\tau = 0.6$	24.08	44.03	40.02
$\alpha_\tau = 0.3, \beta_\tau = 0.7$	23.98	43.46	39.28

Table 6: Performance under different hyperparameter settings.

performance. Following (Xu et al., 2025), we set  $\alpha = 0.4$ . Note that we do not use the FLOPs metric commonly adopted in LLM evaluation, because it only reflects the intrinsic computation of a single LLM pass and cannot capture the real cost of search-augmented agents, which is dominated by retrieval calls, latency, and context expansion.

Method	Latency (s)	Avg-#Search	Utility
Search-R1	30.17	2.16	33.06
MMSearch-R1	26.38	1.44	33.42
<b>Ours (DORA)</b>	<b>24.56</b>	<b>1.30</b>	<b>36.32</b>

Table 7: Computational cost comparison with Search-R1 and MMSearch-R1. Results are averaged over Enc-VQA, InfoSeek, and SK-VQA.

**Analysis.** As shown in Table 7, DORA achieves the lowest latency, the fewest search calls, and the highest Utility among all agentic baselines. **Search-R1** does not optimize retrieval efficiency, so it tends to search more, leading to higher cost. **MMSearch-R1** incorporates search penalties, but its efficiency modeling is coarse-grained, as it penalizes only whether a search occurs without considering how many times the model searches, and it does not explicitly supervise retrieval quality, resulting in reducing unnecessary searches only to a

limited extent. In contrast, DORA combines dual-objective training with a fine-grained efficiency reward, enabling a more precise and efficient search.

#### C.4 Normalization Method

Here, we conduct additional experiments comparing our softmax-based normalization in Equation (6) with alternative smoothing strategies, including linear normalization and temperature-adjusted softmax (TA Softmax, temperature is set to 2). As shown in Table 8, different normalization methods produce marginal performance variations. Softmax performs slightly better on Enc-VQA, while linear or temperature-adjusted variants bring negligible fluctuations on InfoSeek and SK-VQA.

Norm. Method	Enc-VQA	InfoSeek	SK-VQA
Softmax	<b>25.01</b>	44.53	40.98
Linear	24.89	<b>44.62</b>	40.74
TA Softmax	24.78	44.57	<b>41.06</b>

Table 8: Performance under different normalization method.

We believe these small differences occur because the retrieval reward is only one part of the dual-objective optimization, whereas the learned policy is influenced by multiple reward signals. Therefore, changing the normalization of retrieval reward has limited influence on training, leading to the modest and non-systematic variations observed.

#### C.5 Training Dynamics

To provide an analysis of training dynamics, we report the average retrieval reward and the answer reward in multiple training steps (see Table 9).

Across all three datasets, both retrieval and answer rewards exhibit clear learning progress and eventual convergence. Retrieval reward rises rapidly during the early stages (steps 10–100), indicating that the agent quickly learns to issue more

Step	Enc-VQA		InfoSeek		SK-VQA	
	RetReward	AnsReward	RetReward	AnsReward	RetReward	AnsReward
10	0.21	0.03	0.31	0.10	0.29	0.09
50	0.28	0.06	0.54	0.18	0.56	0.17
100	0.32	0.18	0.63	0.27	0.58	0.29
150	0.34	0.23	0.63	0.39	0.60	0.39
200	0.35	0.26	0.64	0.43	0.61	0.42
250	0.34	0.27	0.65	0.45	0.61	0.43
300	0.35	0.27	0.65	0.45	0.60	0.42

Table 9: reward trends across all three datasets.

effective search queries. After roughly 150–200 steps, the retrieval reward curves flatten, showing that the retrieval policy has largely converged and remains stable thereafter.

Answer reward increases more slowly but continues to improve after retrieval reward stabilizes, reflecting the intended training behavior: once a reliable retrieval policy is established, the model can further refine its reasoning and answer generation with higher-quality documents. By step 200–250, the answer reward also reaches a plateau across all datasets, suggesting that the overall policy has converged.

In the revised version, we will provide full training curves—covering response length, valid search rate, and other essential training signals—to offer a more comprehensive picture of the RL dynamics.

## D Discussion and Future Work

**Comparison with text-only agentic search method.** Compared with text-only agentic search, multimodal settings require the model to perform both image-based and text-based retrieval. This naturally results in more search calls. For example, the model may first retrieve information to identify the image content, and then issue additional text queries based on that result. As the number of search steps grows, the likelihood of receiving partially relevant or noisy results also increases. In practice, minor uncertainties in visual understanding often lead to multiple follow-up searches, and some of these can introduce irrelevant information that negatively affects the subsequent reasoning process. These challenges make multimodal retrieval more error-prone and highlight the need for explicitly optimizing search quality. To address this, we introduce a dedicated dual-objective that provides direct supervision for search processing, encouraging the model to produce clearer queries, avoid redundant retrieval, and filter out low-quality

results. This helps the model maintain more accurate retrieval and more stable reasoning in complex multimodal tasks.

**Annotations for the retrieval reward.** We emphasize that, in retrieval-based VQA datasets, retrieval annotations can be obtained with relatively low effort, for two main reasons:

(i) **All datasets used in our experiments already provide annotations of document retrieval.** As shown in our paper, Enc-VQA, InfoSeek, and SK-VQA annotate the gold documents that contain the information needed to answer each question. Our method directly reuses these annotations as training signals: during the rollout, when the model retrieves the gold document, we assign a retrieval reward of 1 to the retrieval process.

(ii) **It is not expensive to obtain retrieval annotations when creating these data sets, thus not a bottleneck when creating more benchmarks.** For the above VQA datasets, these annotations of document retrieval are always naturally labeled together with the labeling of final answers. To be specific, the dataset construction begins with a small set of related documents. Images are extracted from one or more of the above documents, and LLMs are leveraged to generate multiple questions, each of which is relevant to the image(s) and the document paragraph. Then, these questions are combined to produce the final question via LLMs. In this way, the final answer of the final question can be deduced based on these related documents, which naturally correspond to the annotations of document retrieval. Moreover, many other retrieval-based VQA datasets also naturally provide gold annotations required for our retrieval reward. For example, WikiMixQA (Foroutan et al., 2025) focuses on structured elements such as ta-

bles and charts in Wikipedia page to generate questions. PDF-MVQA (Ding et al., 2024) is built from multi-page PDF documents, where questions are constructed from specific paragraphs, figures and tables.

**Finer-grained advantage assignment.** In most rollouts, some search calls are inevitably ineffective, and a finer-grained advantage assignment would allow the model to learn only the effective searches while suppressing the ineffective ones. This idea has great potential to further improve performance. However, the implementation may encounter the following two challenges:

**(i) Difficulty of credit assignment.** Accurately determining the contribution of a single search step to the final answer is non-trivial without dense human supervision (Lightman et al., 2023). An "irrelevant-looking" search might implicitly aid in reasoning by eliminating incorrect paths.

**(ii) Risk of reward noise.** Assigning advantages per step without a perfect verifier introduces noise. As shown in comparisons between Outcome-based and Process-based supervision (Uesato et al., 2022), noisy step-level rewards can sometimes destabilize training compared to coarser but ground-truth-based outcome rewards.

We plan to tackle these challenges and explore the full realization of the idea in future work.

## E Case Studies

### E.1 Qualitative Analysis

We provide four representative case studies below, grouped into two types: Case Type I (Fig 6, 7) illustrates how DORA improves effectiveness through better retrieval and reasoning, while Case Type II (Fig 8, 9) shows how DORA achieves higher efficiency by avoiding unnecessary search calls.

**Analysis.** With representative case studies, we find that DORA improves both retrieval quality and reasoning trajectories primarily due to the following four mechanisms, which together produce more accurate and efficient multi-step reasoning.

**(i) Better problem decomposition** Through direct supervision on retrieval accuracy (Sec. 3.1.1), DORA learns to decompose complex questions into simple sub-questions: 1) identifying the core object, 2) retrieving intermediate attributes (e.g., construction year, location), 3) performing targeted factual lookup. For example, the question decomposition in *Case Type I (1)* is: "First, identify what the building is; then, obtain the year of its construction; and finally, determine which Nobel Prize winner was born in that year." This behavior emerges naturally from the reward design rather than any prompting heuristics.

**(ii) DORA generates more precise search queries.** The dual-objective optimization encourages the model to produce:

- more specific entity descriptors, such as "A Mexican-style building with a white facade accented by gray stone decorative lines" in *Case Type I (1)*,
- better-disambiguated terms aligned with the evolving reasoning state, such as "The start year of construction for Casa del Lago" in *Case Type I (1)*.

**(iii) Retrieval and reasoning mutually reinforce each other.** DORA produces clearer intermediate reasoning, which in turn leads to more accurate and well-scoped search queries. Conversely, the retrieved information often guides the model toward more correct reasoning steps. For instance, in *Case Type I (2)*, Retrieval results confirm the structure in the figure is "Tardebige Locks", enabling the model to more explicitly infer that the canal is located in "Worcestershire". This further leads to a more targeted subsequent retrieval query: "Worcestershire recusants religion".

**(iv) DORA learns to skip unnecessary searches when internal knowledge is sufficient.** Through the efficiency reward, DORA learns to recognize when the current reasoning state already contains enough information to proceed without external retrieval. In such cases, the model strategically skips search calls that would be redundant, leading to more concise search usage. For example, in *Case Type II (1)*, DORA skips the retrieval step

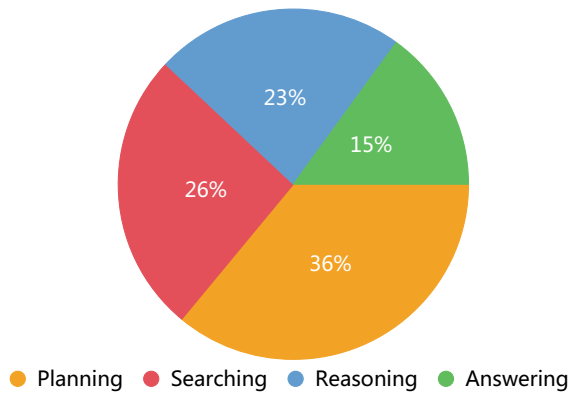


Figure 5: Distribution of error stages.

for recognizing the "McKinley National Memorial", as it can directly identify it.

## E.2 Statistical Analysis

In this section, we conduct a statistical analysis of error cases to better understand the failure modes of our DORA framework. We randomly sample a subset of incorrect predictions and categorize each case into one of four stages—planning, searching, reasoning, and answering—based on the primary source of failure. The categorization is performed using Qwen2.5-VL (72B), which examines the intermediate reasoning traces, retrieval results, and final outputs to identify the dominant failure stage. The resulting error distribution is summarized in Figure 5.

As shown in Figure 5, errors are not evenly distributed across different stages of the agentic pipeline. Specifically, planning errors account for 36% of all failures, followed by search errors (26%), reasoning errors (23%), and answering errors (15%).

**Planning errors** constitute the largest portion, indicating that incorrect or incomplete problem decomposition remains a primary source of failure. In these cases, the model often formulates sub-goals that miss key constraints of the original question, which subsequently propagates errors to later stages.

**Searching errors** mainly arise from suboptimal query formulation or insufficient retrieval results, where the model issues reasonable plans but fails to retrieve the necessary evidence to support downstream reasoning.

**Reasoning errors** occur when relevant evidence is successfully retrieved but is incorrectly interpreted, grounded, or combined across multiple steps, reflecting challenges in multi-hop and cross-modal reasoning.

**Answering errors**, which account for the small-

est fraction, typically correspond to cases where sufficient evidence is available, but the final answer generation fails due to incorrect selection, aggregation, or formatting.

Overall, this distribution suggests that the earlier stages of the agentic pipeline—particularly planning and search—are more error-prone and have a larger impact on overall performance, highlighting the importance of improving decision-making and retrieval strategies in multimodal agentic search.

### Case Type I (1): Input



#### Question.

"Who was born the year construction of this building began and later won a nobel prize?"

#### Ground Truth.

"Luis Federico Leloir"

### Case Type I (1): MMSearch-R1 Output (Baseline)

```
<reason> The image shows a white building in Mexican style and does not provide enough information to tell "Who was born the year construction of this building began and later won a nobel prize?" .</reason>
<search> <img> </search>
<information> Title: "Casa del Lago", Title2: "Casa del Lago UNAM", Title3: "Universidad de Guanajuato"</information>
<reason> The image search results mention the Casa del Lago and Universidad de Guanajuato. However, the image does not provide information about who was born when the Casa del Lago began construction and later won a Nobel Prize. To answer the question, a text search is necessary to find the religion of the recusants. </reason>
<text_search> Who was born when the Casa del Lago began construction and later won a Nobel Prize?</text_search>
<information> 1.Casa del Lago was instituted in 1959 as the first off-campus Cultural Center of the National Autonomous University of Mexico..., 2.The campus hosted the main events of the 1968 Summer Olympics,[16] and was the birthplace of the student movement of 1968.[17] All Mexican Nobel laureates have been alumni of UNAM..., 3.Octavio Paz Lozano[a] (March 31, 1914 – April 19, 1998) was a Mexican poet and diplomat. For his body of work, he was awarded the 1977 Jerusalem Prize, the 1981 Miguel de Cervantes Prize, the 1982 Neustadt International Prize for Literature, and the 1990 Nobel Prize in Literature... </information>
<reason> The search results show that Octavio Paz Lozano is a Nobel Winner.</reason>
<answer>Octavio Paz Lozano</answer>
```

### Case Type I (1): DORA Output (Ours)

```
<think> The image shows a white building in Mexican style. To answer the above question, the first step is to identify what the building is, then obtain the year of its construction, and finally determine which Nobel Prize winner was born in that year. To identify this building, we need to call a retrieval tool</think>
<search> "query": "A Mexican-style building with a white facade accented by gray stone decorative lines", "with_image": "yes" </search>
<information> <doc id="0">"Casa del Lago.Casa del Lago was instituted in 1959 as the first off-campus Cultural Center of the National Autonomous University of Mexico..."</doc> <doc id="1">"Casa del Lago.As in the beginning, half a century ago, Casa del Lago, opens its doors to young..."</doc> <doc id="2">"Universidad de Guanajuato. The university traces its history back to the educational institute called the Hospice of the Holy Trinity, which was established on October 1..."</doc></information>
<reflect> According to the retrieved documents, this building should be Casa del Lago</reflect>
<think> We already know the name of this building, and we can now obtain the year of its construction </think>
<search> "query": "The start year of construction for Casa del Lago", "with_image": "no" </search>
<information> <doc id="0">"Casa del Lago.The construction of the building began in 1906, but it was inaugurated as the headquarters of the first Automobile Club in Mexico..."</doc> <doc id="1">"Casa del Lago. It revived, from a contemporary perspective, what initially was known as Poetry Out Loud, just a few years after the inauguration of this university forum in Chapultepec, through a festival that seeks to reclaim this tradition that called thinkers, intellectuals and artist of the time..."</doc> <doc id="2">"Casa del Lago.As in the beginning, half a century ago, Casa del Lago, opens its doors to young..." </doc> </information>
<reflect> According to the retrieved documents, this building began construction in 1906</reflect>
<think> After obtaining the construction year of the building, the next step is to find the Nobel Prize winner born in 1906</think>
<search>"query": "Nobel Prize winner born in 1906", "with_image": "no"</search>
<information> <doc id="0">" 1906. September 6 – Luis Federico Leloir, French-born Argentine chemist, Nobel Prize laureate (d. 1987) September 8 – Andrei Kirilenko, Soviet politician (d. 1990)... </doc> <doc id="1">"Santiago Ramón y Cajal.He and Camillo Golgi received the Nobel Prize in Physiology or Medicine in 1906..."</doc> <doc id="2">"..."</doc> </information>
<conclude> It is found from the content of the article that Luis Federico Leloir was born on September 9, 1906 and won the Nobel Prize in 1987</conclude>
<answer>Luis Federico Leloir</answer>
```

### Case Type I (1): Analysis

**Result:**

MMSearch-R1: 2 retrievals, wrong answer.

DORA: 3 retrievals, correct answer.

**Reason:** Compared to MMSearch-R1, DORA achieves clearer problem decomposition: "The first step is to identify what the building...". By addressing these sub-problems, the model yields the correct answer.

Figure 6: Case Type I (1)

### Case Type I (2): Input



**Question.**

"What religion were many of the recusants in the county where this canal is located?"

**Ground Truth.**

"Catholic"

### Case Type I (2): MMSearch-R1 Output (Baseline)

```
<reason> The image shows a canal. The image does not provide enough context to tell religion of the recusants in the county.</reason>
<search> <img> </search>
<information> Title: "Worcester and Birmingham Canal", Title2: "Tardebigge Locks", Title3: "Leeds and Liverpool Canal"</information>
<reason> The image search results mention the Worcester and Birmingham Canal and Tardebigge Locks. However, the image does not provide information about religion of the recusants in the county where this canal is located. To answer the user's question, a text search is necessary to find the religion of the recusants. </reason>
<text_search> What religion were many of the recusants in the county where Worcester and Birmingham Canal located?</text_search>
<information> 1.The Worcester and Birmingham Canal. Those reporting no religion or declining to state an allegiance make up 32.3 per cent. The next largest religious group, Muslims, makes up 2.9 per cent..., 2.The Worcester and Birmingham Canal is a canal linking Birmingham and Worcester in England. It starts in Worcester, as an 'offshoot' of the River Severn (just after the river lock)... 3.The Worcester and Birmingham Canal.The parliamentary bill permitting its construction was passed in 1791 as the Worcester and Birmingham... </information>
<reason> The search results did not explicitly indicate the religion of the recruits in Worcester, only providing the largest proportion of non religious groups</reason>
<answer>no religion</answer>
```

### Case Type I (2): DORA Output (Ours)

```
<think>The image shows a system of canal locks. To answer the question "What religion were many of the recusants in the county where this canal is located?", we must first identify the canal in the image, then determine the county where it is located, and finally retrieve the religion commonly associated with recusants in that county. Therefore, we first need to identify the canal using a retrieval call. </think>
<search> "query": "locks over canal UK long flight", "with_image": "yes" </search>
<information> <doc id="0"> "Tardebigge Locks. Tardebigge Locks is a flight of 30 locks on the Worcester and Birmingham Canal, near Bromsgrove in Worcestershire..." </doc> <doc id="1"> "Worcester and Birmingham Canal. This canal runs from Birmingham to Worcester and includes the Tardebigge flight..." </doc> <doc id="2"> "Worcestershire waterways. Notable features include the Tardebigge Locks on the Worcester and Birmingham Canal..." </doc> </information>
<reflect> The retrieved documents indicate that the locks in the image match Tardebigge Locks, located on the Worcester and Birmingham Canal, in Worcestershire. </reflect>
<think> Since the canal is located in Worcestershire, we now need to identify the religion of many recusants in Worcestershire. We perform another retrieval step. </think>
<search> "query": "Worcestershire recusants religion", "with_image": "no" </search>
<information> <doc id="0"> "Worcestershire Recusancy. Many recusants in Worcestershire remained Catholic following the English Reformation..." </doc> <doc id="1"> "Recusants in the West Midlands. The majority were Catholic families resisting the Anglican state church..." </doc> <doc id="2"> "English recusancy. Recusants were typically Roman Catholics who refused to attend Anglican services..." </doc> </information>
<reflect> The retrieved documents consistently indicate that recusants in Worcestershire were predominantly "Catholic". </reflect>
<conclude> The canal in the image is Tardebigge Locks on the Worcester and Birmingham Canal, located in Worcestershire, where many recusants were Catholic. </conclude>
<answer>Catholic</answer>
```

### Case Type I (2): Analysis

**Result:**

MMSearch-R1: 2 retrieval, wrong answer.

DORA: 2 retrievals, correct answer.

**Reason:** Compared to MMSearch-R1, DORA generates more precise queries (e.g., "Worcestershire recusants religion"), enabling the retrieval of accurate information ("Catholic") and thus leading to correct answer.

Figure 7: Case Type I (2)

### Case Type II (1): Input



**Question.**

"Who was the sultan of the ottoman empire when construction of this memorial began?"

**Ground Truth.**

"Abdul Hamid II"

### Case Type II (1): MMSearch-R1 Output (Baseline)

```
<reason> The image shows a memorial structure. The image does not provide sufficient information to know the
sultan of the ottoman empire when construction of this memorial began </reason>
<search> <img> </search>
<information> Title1: " McKinley National Memorial ", Title2: " National McKinley Birthplace Memorial ",
Title3: " Arlington National Cemetery "</information>
<reason> The image search results reference the McKinley National Memorial, but they lack information about the
Sultan of the Ottoman Empire at the time construction of the National McKinley Birthplace Memorial commenced.
To answer this question, a text-based search is required to first determine the start year of the memorial's construction.
</reason> <text_search> When did the construction of the McKinley National Memorial begin?</text_search>
<information> 1. McKinley National Memorial. Construction of the memorial began on June 6, 1905, when
Magonigle removed the first shovel of soil from the site... 2. The McKinley National Memorial in Canton, Ohio,
United States, is the final resting place of William McKinley, who served as the 25th president of the United States
from 1897 to his assassination in 1901... 3. McKinley National Memorial. In June 1903 contributions reached
$500,000, and the Association invited people to submit design ideas for the proposed memorial... </information>
<reason> The search results indicate the start year of the memorial's construction is 1905. However, the search
results do not provide the information of the Sultan of the Ottoman Empire. To answer this question, a text-based
search is required to further determine the Sultan of the Ottoman Empire in 1905.</reason> <text_search> Who
was the Sultan of the Ottoman Empire in 1905?</text_search>
<information> 1. Abdul Hamid II. Reigin: 31 August 1876 –27 April 1909 (32 years, 239 days). Son of Abdulmejid
I and Tirimüjgan Kadın (later became the adoptive son of Rahime Perestu Sultan)... 2. 1905. May 22 – Abdul
Hamid II, the Sultan of the Ottoman Empire establishes the Ullah millet for the Aromanians of the empire. For this
reason, the Aromanian National Day... 3. Constitutionalism was established during the reign Abdul Hamid II, who
thus became the empire's last absolute ruler and its reluctant first constitutional monarch... </information>
<reason> The search results indicate the the Sultan of the Ottoman Empire in 1905 is Abdul Hamid II. </reason>
<answer> Abdul Hamid II </answer>
```

### Case Type II (1): DORA Output (Ours)

```
<think> The image clearly shows ' McKinley National Memorial '. To solve the problem, we should first search for
the construction time of McKinley National Memorial, and then obtain who the Sultan was at the construction time.
</think>
<search> "query": "When was McKinley National Memorial constructed", "with_image": "no" </search>
<information> <doc id="0"> "McKinley National Memorial. Construction of the memorial began in 1905 and
it was dedicated in 1907..." </doc> <doc id="1"> "William McKinley Memorial. By October 10, 1901, the
Association issued a public appeal for $600,000 in contributions for the construction project... " </doc> <doc
id="2">...</doc> </information>
<reflect> From the documents, we know the construction began in 1905 </reflect>
<think> Next, we will investigate who the Sultan was in 1905. </think>
<search> "query": " Sultan in 1905 of Ottoman Empire ", "with_image": "no" </search>
<information> <doc id="0">1905. May 22 – Abdul Hamid II, the Sultan of the Ottoman Empire establishes the
Ullah millet for the Aromanians of the empire...</doc> <doc id="1"> "Abdul Hamid II. Sultan of the Ottoman
Empire from 1876 to 1909..."</doc> <doc id="2">... </doc> </information> <reflect> From the above
documents, In 1905, Abdul Hamid II was the Sultan of the Ottoman Empire. </reflect>
<conclude> The building in the image was constructed in 1905, when Abdul Hamid II was the Sultan of the Ottoman
Empire. </conclude>
<answer>Abdul Hamid II </answer>
```

### Case Type II (1): Analysis

#### Result:

MMSearch-R1: 3 retrievals, correct answer.

DORA: 2 retrievals, correct answer.

**Reason:** Compared to MMSearch-R1, DORA directly identifies the building is "William McKinley Memorial" and skip the step of identifying it through retrieval, resulting in fewer search calls.

Figure 8: Case Type II (1)

### Case Type II (2): Input



**Question.**

"What is the country of origin of this animal?"

**Ground Truth.**

"Hungary"

### Case Type II (2): MMSearch-R1 Output (Baseline)

```
<reason> The image shows a brown dog and does not provide sufficient context to tell the country of origin of this animal.</reason>
<search> <img> </search>
<information> Title1: " Magyar Vizsla ", Title2: " Vizsla ", Title3: " Wirehaired Vizsla "</information>
<reason> The image search results mention the Magyar Vizsla, but they lack information about the native country of this dog. To answer this question, a text-based search is required to determine the country of origin of the Magyar Vizsla. </reason> <text_search> What is the country of origin of the Magyar Vizsla?</text_search>
<information> 1. Magyar Vizsla is a dog breed from Hungary and belongs to the Fédération Cynologique Internationale (FCI) group 7 (Pointing Dogs), the Canadian Kennel Club (CKC) group 1 (Sporting group)... 2. A careful search of Hungary and a poll of Hungarian sportsmen revealed only about a dozen Vizslas of the true type still alive in the country... 3. The Vizsla started arriving in the United States at the close of World War II. As interest in and devotion to the breed began to increase,... </information>
<reason> The search results indicate the country of origin of this dog is Hungary.</reason>
<answer> Hungary </answer>
```

### Case Type II (2): DORA Output (Ours)

```
<think> The image shows a brown dog. To solve the problem, we should first obtain the breed of this dog, and then determine the country of origin of this dog. </think>
<search> "query": "A tan short-haired dog with naturally drooping ears", "with_image": "yes" </search>
<information> <doc id="0"> "The Magyar Vizsla is a short-coated hunting dog of distinguished appearance and bearing. Robust, but rather lightly built..." </doc> <doc id="1"> "The body structure of a Vizsla is very similar in appearance to a Weimaraner and a Redbone Coonhound, though the Vizsla is typically leaner. . ." </doc> <doc id="2"> . . . </doc> </information>
<reflect> From the documents, we know that this dog's breed is the Magyar Vizsla </reflect>
<think> Next, we will investigate the country of origin of Magyar Vizsla. Magyar Vizsla also known as Hungarian Vizsla or Hungarian Pointer is from Hungary. </think>
<conclude> The country of origin of Magyar Vizsla in the image is Hungary. </conclude>
<answer> Hungary </answer>
```

### Case Type II (2): Analysis

**Result:**

MMSearch-R1: 2 retrievals, correct answer.

DORA: 1 retrieval, correct answer.

**Reason:** Compared to MMSearch-R1, DORA directly inferred the dog's country of origin ("Hungary"), thereby avoiding an unnecessary search call.

Figure 9: Case Type II (2)