APPENDIX

In the appendix, we provide our LLM usage statement, discuss related works and mention the limitations and broader impact of our work. Additionally, we focus on its implementation and provide extensive details about the prompts used for dataset curation and the evaluation. Furthermore, we expand on the results presented in the main paper, providing general VQA experiments and some additional analysis. In the end, we provide visual examples of the proposed dataset.

Table of Contents

A LLM USAGE STATEMENT

We use large language model (LLM) as a supportive tool in this work. It was employed to assist in debugging portions of the code, generating and refining visualization figures, and improving the clarity of the manuscript through proofreading, grammar checking, and polishing the overall writing style. The LLM's role was limited to these supportive tasks, while all substantive research ideas, methodological decisions, analyses, interpretations, and final code implementations were developed and validated independently by us.

B RELATED WORKS

B.1 MULTIMODAL LLMS

Multimodal large language models (MLLMs) combine visual encoders with powerful text-based large language models, enabling them to process and reason over both textual and visual inputs. Recent models such as GPT-4o Hurst et al. (2024), Gemini Team et al. (2023), Qwen2.5-VL Bai et al. (2025), InternVL Chen et al. (2024), LLaVA series Li et al. (2024a); Lin et al. (2023); Liu et al. (2023b;a; 2024a), Phi Series Abdin et al. (2024), Mantis Series Jiang et al. (2024a), OVIS series Lu et al. (2024b); Wang et al. (2025), VILA series Lin et al. (2024); Nath et al. (2025), Gemma series Team et al. (2024a;b) have demonstrated strong capabilities in visual perception, grounding, and multimodal reasoning, achieving remarkable progress in tasks like visual question answering, captioning, and multimodal dialogue. These advances highlight their potential as core components in real-world applications such as digital assistants, education, and information access. Despite these strengths, MLLMs face fundamental limitations in addressing knowledge-intensive or information-seeking queries Mensink et al. (2023); Chen et al. (2023); Cheng et al. (2025); Li et al. (2024c). Their training relies on static corpora, which inevitably leads to outdated knowledge as the real world evolves. Furthermore, the breadth of open-world knowledge follows a long-tail distribution, and it is infeasible to cover every rare or emerging fact within a fixed training dataset Mensink et al. (2023). This makes MLLMs struggle with long-tail knowledge and information that requires up-to-date context.

B.2 RAG-BASED SEARCH

The RAG paradigm as the name suggests retrieves external information from a fixed knowledge corpora using vector search and augments it into the model context to generate factually grounded responses. Early contributions in this space include REALM Guu et al. (2020), which introduced retrieval-augmented pretraining by jointly optimizing a dense retriever with a language model to enable knowledge-intensive tasks. RAG Lewis et al. (2020) further advanced this paradigm by integrating a generative seq2seq model with neural retrieval, demonstrating strong gains in open-domain question answering. Recent efforts have extended retrieval augmentation to multimodal settings. REVEAL Hu et al. (2023) presented a retrieval-augmented visual-language pretraining framework, in which the memory, encoder, retriever and generator are all pre-trained end-to-end on a massive amount of data. VisRAG Yu et al. (2024) proposed a vision-language model based RAG pipeline that directly embeds documents as images for retrieval, avoiding information loss from text parsing. This strategy enables the joint filtering of retrieved documents, retaining only the most relevant and accurate references. RoRA-VLM Qi et al. (2024) introduced a two-stage retrieval process with image-anchored textual-query expansion to synergistically combine the visual and textual information in the query and retrieve the most relevant multimodal knowledge snippets. Moreover, they improve the robustness of retrieval-augmented vision-language model by injecting adversarial noise in the training process. RaR Liu et al. (2024e) proposed a retrieving-and-ranking augmented multimodal framework tailored for visual recognition, highlighting the role of retrieval quality in multimodal perception tasks. Recently, MMKB-RAG Ling et al. (2025) proposed a novel multi-modal RAG framework that leverages the inherent knowledge boundaries of models to dynamically generate semantic tags for the retrieval process. Despite these advances, RAG methods rely on static corpora, and work with an unrealistic assumption that all information can be captured within a fixed dataset. In real-world scenarios, web information is dynamic and constantly evolving, and the complexity of retrieval remains high. These factors pose significant challenges for adopting RAG in real-world, open-ended VQA.

B.3 Prompt-based search agents

The prompt-based search agents act as plug-and-play modules that can be integrated with existing multimodal LLMs without additional finetuning. In this setup, the MLLM functions as an agent, incorporating web-retrieved information into its responses. For example, VSA Zhang et al. (2024) enables any vision-language model to operate as a multimodal automatic search engine. Its pipeline follows three steps: (1) visual content formulation, where the model identifies the object of interest; (2) web-knowledge search, where it generates multiple sub-questions and queries the web; and (3) summarization, where it consolidates the retrieved information before answering the user's query. Similarly, MM-Search Jiang et al. (2024b) introduces the MMSearch-Engine, a pipeline that augments large multimodal models with search capabilities through requerying, reranking, and summarization. OmniSearch Li et al. (2024c) further advances this idea by proposing a self-adaptive planning agent for multimodal retrieval. It dynamically decomposes complex questions into sequential sub-questions and selects retrieval actions accordingly. At each step, the planner evaluates prior retrieval feedback (via a solver) to decide whether to refine the query, switch retrieval mode (e.g., text, image, web), or generate new sub-questions. This flexible, feedback-driven process replaces rigid heuristics with a query-planning loop, better suited for dynamic, multi-hop, and multimodal VQA scenarios. However, across these approaches, the base model itself is not trained to engage effectively with web-retrieved information and external search tools, leaving it less capable of handling the noisy and complex nature of such real-world web information.

B.4 Web-search equipped MLLMs

Recent work focuses on R1-optimization of MLLMs to equip web-search capabilities in MLLMs. This trend follows from the success of reasoning models such as OpenAI o1,o3 and DeepSeek-R1. DeepResearcher Zheng et al. (2025) uses a multi-agent browsing architecture and the GRPO algorithm to learn to navigate, extract, and filter information from arbitrary web pages under realistic constraints (e.g., API limits, network latency, anti-crawling). R1-Searcher Song et al. (2025) presents a two-stage outcome-based reinforcement learning framework that allows LLMs to autonomously invoke external search systems during reasoning for knowledge-intensive tasks. In stage one, the model is rewarded for learning to trigger retrieval (without regard to answer correctness), and in stage two it is further trained to integrate retrieved evidence to maximize answer accuracy. Search-R1 Jin et al. (2025) incorporates retrieved-token masking, which prevents the RL objective from directly optimizing over retrieved content, stabilizing training when mixing generated and retrieved tokens. However, all these works are restricted to text search and are unable to perform an image search, which limits their applicability in mulitmodal knowledge-intensive question answering. MMSearch-R1 Wu et al. (2025) is the only prior work that performs multimodal retrieval, but it has notable limitations. First, although the model can autonomously decide which tool to use, it is constrained to a single invocation per tool, which limits its ability to revise decisions through self-reflection and self-correction. Second, in knowledge-intensive VQA tasks, it is essential to precisely identify the visual entity in the image that the question refers to. However, in real-world settings, background clutter and the presence of irrelevant visual entities often introduce noise into the retrieval process. This noise can hinder accurate localization of the target entity, leading to suboptimal retrieval and reduced effectiveness of image search in practice. To address these limitations, we propose DeepMMSearch-R1, which performs image search using relevant image crops and can iteratively refine its text search queries to better navigate noisy real-world web information.

C Datasets

C.1 InfoSeek

InfoSeek Chen et al. (2023) is a large-scale knowledge-intensive visual question answering dataset designed for information-seeking tasks. It consists of 8,900 human-written question–answer pairs over 806 entities and 527 entity types, as well as 1.35 million automatically generated QA triplets covering 11,481 entities across 2,739 entity types. The dataset is split into UNSEEN ENTITY and UNSEEN QUESTION partitions to test generalization. InfoSeek is widely used for evaluating multimodal models in knowledge retrieval and reasoning beyond surface-level recognition.

C.2 FVQA

FVQA Wu et al. (2025) is a multimodal search VQA dataset constructed to enable evaluation and training of models that must decide when and how to perform external searches in a knowledge-intensive setting. The FVQA training split

(FVQA-train) comprises around 6,000 image–question–answer samples ("FVQA-auto-vc") focused on visual knowledge, plus 7,000 text-knowledge examples drawn from InfoSeek ("FVQA-auto-txt"), and an additional 800 manually annotated "FVQA-manual-train" samples. The test split (FVQA-test) is manually curated for higher quality and diverse knowledge demands.

C.3 ENCYCLOPEDIC VQA

Encyclopedic VQA Mensink et al. (2023) is a large-scale visual question answering dataset that focuses on visual questions about detailed properties of fine-grained object categories and specific instances. It comprises 221,000 unique question–answer pairs, each associated with up to 5 different images, yielding a total of around 1,000,000 (1 M) image-question-answer instances. The dataset is backed by a controlled knowledge base derived from Wikipedia, where each QA is linked to supporting evidence from Wikipedia articles.

C.4 SIMPLEVQA

SimpleVQA Cheng et al. (2025) is a multimodal benchmark created to evaluate the factuality of MLLMs in answering short, natural-language visual questions. It contains 2,025 high-precision image–question–answer pairs, spanning 9 task categories (e.g. Object Identification & Recognition, Time & Event, Person & Emotion, Location & Building, Text Processing, Quantity & Position, Art & Culture, Object Attributes) and 9 topic domains. SimpleVQA's design ensures coverage across domains, concise and clear answer formats, and suitability for automated evaluation (e.g. via LLM-as-judge). It is intended to challenge MLLMs' abilities to ground answers in factual knowledge rather than hallucinate, and is often used to probe the knowledge boundaries of vision-language models.

C.5 DYNVQA

DynVQA Li et al. (2024c) is a benchmark dataset constructed to assess multimodal retrieval-augmented generation (mRAG) systems on dynamic visual question answering tasks that require adaptive retrieval strategies. It contains 1,452 questions spanning 9 domains, evenly split across English (715) and Chinese (737) items. The questions are categorized into three dynamic types: (1) those with rapidly changing answers (385 questions, $\sim$26.5%), (2) multimodal-knowledge questions requiring non-textual evidence (178 questions, $\sim$12.3%), and (3) multi-hop questions requiring multi-step reasoning (112 questions, $\sim$7.7%). Across all questions, 59.6% require external visual knowledge beyond what is directly in the image, and 26.7% require more than two reasoning hops. DynVQA is designed with temporal dynamism, and some answers may change over time. Therefore, the dataset is periodically updated to maintain answer correctness.

C.6 OKVQA

OKVQA Marino et al. (2019) is a knowledge-based visual question answering dataset in which the visual content alone is insufficient to answer questions—models must draw on external knowledge. It comprises 14,055 open-ended question–answer (QA) pairs associated with 14,031 images. Each QA is annotated with 5 ground truth answers per question. To reduce dataset bias, frequently repeated answers were pruned, such that questions whose most common answer appeared more than 5 times were removed. The dataset covers a diverse set of 10 knowledge categories (e.g., Vehicles & Transportation; Cooking & Food; Science & Technology) determined via crowd annotations. Baseline VQA models that perform well on standard VQA benchmarks show significant performance drops on OKVQA, highlighting the difficulty of knowledge retrieval and reasoning in this setup.

C.7 A-OKVQA

A-OKVQA (Augmented OK-VQA) is a crowdsourced visual question answering benchmark designed to require commonsense and world knowledge beyond simple fact lookup. It comprises approximately 24,903 question–answer–rationale triplets spread across 17.1K training, 1.1K validation, and 6.7K test splits. Each question is accompanied by both multiple-choice (MC) options and direct-answer (DA) alternatives, along with a rationale (one explanatory sentence) for the train split. To ensure diversity, A-OKVQA filters out trivial or overly repetitive questions, enforces quality control via crowd annotation, and clusters similar images to discourage repetitive phrasing. Compared

to OKVQA, A-OKVQA contains about twice as many questions and adds rationale annotations to support explainable reasoning.

D Implementation Details

We use the LLaMA-Factory framework to perform supervised finetuning. Our base model is Qwen2.5-VL-7B-Instruct, which we finetune using LoRA with a rank of 8 applied across all target modules. Training is performed for 3 epochs with a learning rate of 1e-4, following a cosine scheduler with a warmup ratio of 0.1. We enable bf16 mixed precision for computational efficiency. The model is trained using 1 node with 8 Nvidia H100 GPUs, with per-device batch size is set to 1, with gradient accumulation of 1 step, resulting in a global batch size of 8. Since the VQA dataset consists of multi-turn conversations, we apply input masking during training to ensure that the model is optimized only on the generated outputs. For online RL optimization, we adopt the GRPO algorithm implemented in the veRL framework. The reward model is GPT-4o, which evaluates generated responses and provides feedback for optimization, with λ_{fmt} set to 0.1. We apply a KL-penalty with a coefficient of 0.001 and the clip ratio is set to 0.2. Training is performed for 20 epochs on 4 nodes each with 8 Nvidia H100 GPUs. We use a batch size of 256 with a mini-batch size of 64, and set the rollout number to 8 per iteration. A warmup phase of 45 steps is applied to stabilize learning rates, which are initialized at 2e-6. The image search/cropped image search tool can be called once while the text-search tool can be called multiple times, with total tool calls restricted to 10 per rollout. The maximum response length is set to 8192 tokens We again mask the input tokens to ensure that optimization focuses only on the generated outputs.

E PROMPTS

E.1 SFT DATASET GENERATION

Initial Prompt

You are an expert visual assistant. Your task is to answer a user's question based on the provided image.
Step 1: Analyze the Image
Carefully examine the image and the user's question: `{question}`. Identify all recognizable entities, objects, text, and other visual clues.
Step 2: Plan Your Action
Based on your analysis, you must perform one of the following actions. You must include your thinking process inside a `<reason>...</reason>` block before choosing an action.

- **Action 1: Answer Directly**
  If you can confidently identify the visual element and have the internal knowledge regarding the facts sufficient to answer the question, provide a direct, concise answer inside `<answer>...</answer>` tag.
  Example: `<answer>The construction of Eiffel Tower was finished on March 31, 1889.</answer>`
- **Action 2: Use Image Search**
  If you are not sure about the visual element and need to identify the visual element in the image, you can use one of the following image search methods.
  - **Cropped search (Preferred for specific questions):** Use this if the question is clearly about a specific visual element such as an object, person, animal, plant, aircraft, etc., or if the background is irrelevant. Describe the visual element concisely inside the `<img_search>...</img_search>` tags.
    Example:
    `<img_search>the face of the person on the left</img_search>`
    `<img_search>the red logo on the baseball cap</img_search>`
  - **Whole image search:** Only use this if the question is about the entire scene in general, its location, or the overall context. Output only: `<img_search><img></img_search>`.
    Note: Do not output `<img_search><img></img></img_search>`.
- **Action 3: Use Text Search**
  If you can identify the visual element confidently but need more specific information to answer the question, invoke the text search tool. Generate a focused query and output it as `<text_search>your search query</text_search>`.

Remember, search results will be provided to you in subsequent turn. You can analyze the search results and decide your next action. You can perform image search only once, but have the option to perform multiple text searches to gather relevant information. All search results will be placed inside `<information>...</information>`.
Here is the image and question: `<image>{question}`

After Image Search

You have received information from an image search. Your goal is to use this new information to answer the original question: `{question}`.
Step 1: Analyze the Results
Review the provided information within the `<information>...</information>` block. Synthesize what you've learned about the visual element in question.
Step 2: Plan Your Next Action
Include your thinking process inside a `<reason>...</reason>` block. Then, choose one of the following actions:

- **Action 1: Answer Directly**
  If the image search results have helped you identify the visual element and you can confidently answer the question with your internal knowledge, provide the final, concise answer inside an `<answer>...</answer>` tag.
- **Action 2: Use Text Search**
  If the image search results have helped you identify the visual element but you need more specific details to answer the question, invoke the text search tool. Formulate a precise query based on the image search results and output it as `<text_search>your search query</text_search>`. You can use the text search tool multiple times in subsequent turns if needed.

After Text Search

You have received results from a text search. Your goal is to analyze this new information and decide the next best step to answer the original question: `{question}`.
Step 1: Analyze the Results
Review the new information provided in the `<information>...</information>` block. Compare it against the information you already have and what is still needed to answer the question.
Step 2: Plan Your Next Action
Include your thinking process inside a `<reason>...</reason>` block. Then, choose one of the following actions:

- **Action 1: Answer Directly**
  If you have now gathered all the necessary information, provide the final, concise answer inside an `<answer>...</answer>` tag.
- **Action 2: Search Again**
  If the results are helpful but still insufficient, perform another text search. Create a new, more specific, or modified query to find the missing facts. Output the new query as `<text_search>your refined search query</text_search>`.
- **Action 3: Give Up**
  If you have exhausted your search attempts and believe the answer cannot be found from the provided information, conclude by outputting `<answer>Unable to answer due to lack of relevant information</answer>`.

E.2 WebSearch Equipped MLLMs evaluation Prompt

The evaluation prompt is same as SFT data generation prompts as detailed in Section E.1.

E.3 RAG WORKFLOW PROMPT

Initial Prompt

You are a helpful assistant designed to answer questions about images using external knowledge. You are given a question accompanied by an image that cannot be answered without external knowledge.
You are provided with a question, the corresponding image, and a text summary from a reverse image search that identifies the main visual subject. Based on all this information, your task is to formulate a single, effective query for a search engine (e.g., Google) to find the specific facts needed to answer the question.
Question: `{question}`
Reverse Image Search Information: `{information}`
Provide only the text query you will use for the search, in the format `<text_search>your query</text_search>`.

Final Answer Prompt

You have now received the results from your text search. Your goal is to analyze the text search results to provide a final concise answer to the original question based on the image provided.
Original Question: `{question}`
Text Search Results: `{information}`
Follow the following process:

1. Briefly explain your reasoning process by analyzing the facts from the search results that are relevant to the question. Enclose this reasoning inside `<reason>your reason</reason>` tags.
2. Provide the final, direct answer to the question between `<answer>` and `</answer>` tags. If the information is insufficient, respond **ONLY** with:
   `<answer>Unable to answer due to lack of relevant information.</answer>`

E.4 PROMPT-BASED SEARCH AGENT PROMPT

The prompt-based search agent prompts are same as SFT data generation prompts as detailed in Section E.1.

E.5 LLM-AS-JUDGE PROMPT

LLM-as-judge Prompt

You are an impartial judge evaluating a model's answer for a visual question answering task. Your task is to determine if the **Predicted Answer** is correct by comparing it against the **Ground-Truth Answer(s)**.
IMPORTANT INSTRUCTION: The **Ground-Truth Answer(s)** field may contain alternate correct answers. The predicted answer should be considered **CORRECT** if it is semantically equivalent to **at least ONE** of the provided ground-truth answers.
Please respond with only `[CORRECT]` if the prediction is correct, and `[INCORRECT]` otherwise.

— Evaluation Details —
Question: `{question}`
Ground-Truth Answer(s): `{references_for_prompt}`
Predicted Answer: `{candidate}`

E.6 GPT AS REWARD MODEL PROMPT

GPT-4o as reward model prompt

You are a strict evaluation judge for short-answer matching. Given a model's final answer and a list of gold answers, decide if the model's answer matches **ANY** gold answer.
Rules:

1. **Semantic Equivalence:** Consider synonyms, paraphrases, and common aliases as valid matches.
   Example: `"NYC"` ≈ `"New York City"`.
2. **Ignore Trivial Differences:** Do not penalize differences in articles, punctuation, word order, or casing.
   Example: `"The Pacific Ocean"` ≈ `"pacific ocean"`.
3. **At Least One Match:** If the model's answer aligns with ANY gold answer based on the rules, set `match=true`. Otherwise, `match=false`.
4. **Numerical Flexibility:** For answers involving numbers, an answer is a MATCH if it meets any of these criteria:
   (a) **Range Inclusion:** The model provides a range that contains the gold answer.
   Example: Model: `"20 to 24"`, Gold: `["21"]`.
   (b) **Reasonable Rounding:** The model's answer is a reasonably rounded version of the gold answer.
   Example: Model: `"176"`, Gold: `["176.124"]`.
   (c) **Unit Conversion:** The model's answer is equivalent but in a different unit.
   Example: Model: `"3 km"`, Gold: `["3000 m"]`.
5. **Substantive Difference:** If the meaning, entity, or value differs in a way not covered by the rules above, it is NOT a match.
   Example: `"Jupiter"` ≠ `"Mars"`.
   Example: `"5.2"` ≠ `"52"`.
   Example: Model: `"10-15"`, Gold: `["16"]` → **NO MATCH**.

Output Format:

```
MATCH: true/false
REASON: A concise explanation focusing only on why the answer matches or
does not match.
```

E.7 LLM SUMMARIZER PROMPT

Image Search Summarization Prompt

Based on the following text extracted from the title and description of the retrieved images obtained from a Google Lens search, concisely describe the primary visual content (such as faces, objects, locations, events, logos, or text) of the original image in four to five sentences.
Extracted Text:

```
{formatted_results}
```

Text Search Summarization Prompt

Review all the provided text references to find the most relevant information to answer the question. Analyze the relevant facts from these references into a single, concise summary of 10–12 sentences that answers the question.
Question: `{original_question}`

References:
`{references_text}`

F Additional Content and Results

F.1 Introduction Figure

Figure F.1: An image of a boat race.

F.2 SFT MODEL CROPPING EXAMPLE

We present examples where the SFT model performs unnecessary cropping in Figure F.2. RL training with GRPO corrects this issue, making tool usage more efficient. The RL-optimized model performs cropping only when required.

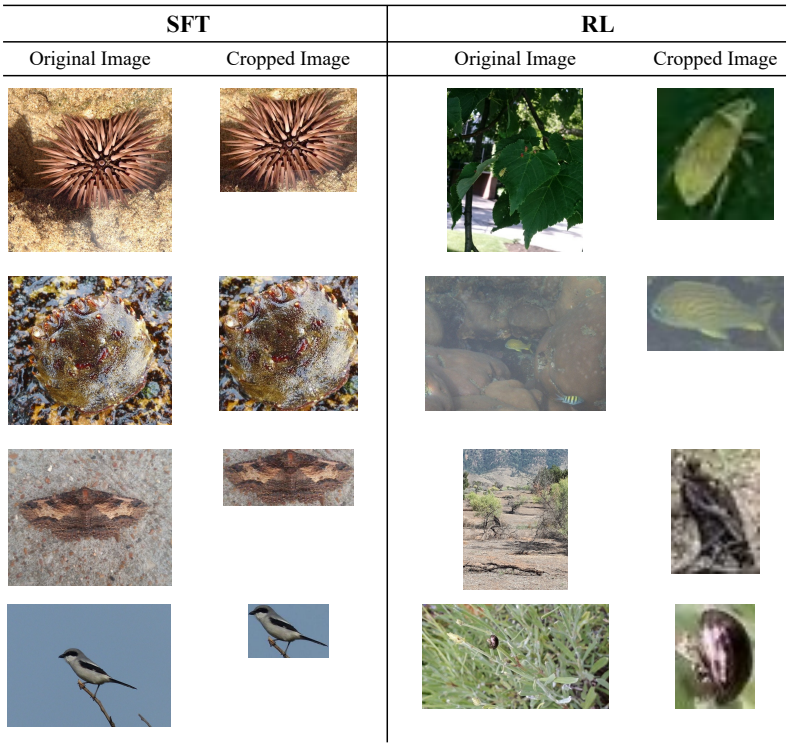

Figure F.2: Croppd image search usage: SFT model vs RL optimized model

F.3 PERFORMANCE ON GENERAL VQA

To quantify the impact of SFT + RL training on the proposed model's general VQA and reasoning ability, we benchmark DeepMMSearchR1-7B (RL) on a range of benchmarks, including OCRBench Liu et al. (2024d), MMVet Yu et al. (2023), AI2D Kembhavi et al. (2016), MathVista Lu et al. (2024a), MMBench Liu et al. (2024c), DocVQA Mathew et al. (2021), and InfoVQA Mathew et al. (2022). We observe that the model maintains its overall performance while achieving improvements on MathVista and MMVet. These results suggest that the proposed model effectively learns to interact with web-search tools while preserving its general visual understanding and reasoning capabilities.

Models	OCRBench	MMVet	AI2D	MathVista MINI	MMBench	DocVQA	InfoVQA
Qwen2.5-VL-7B-Instruct	88.30	68.30	83.74	68.20	83.84	94.97	82.58
DeepMMSearch-R1-7B (RL)	87.60	69.81	82.57	66.80	83.76	94.63	81.63

Table F.1: Comparison of model performance on GeneralVQA.

G DeepMMSearchVQA Samples

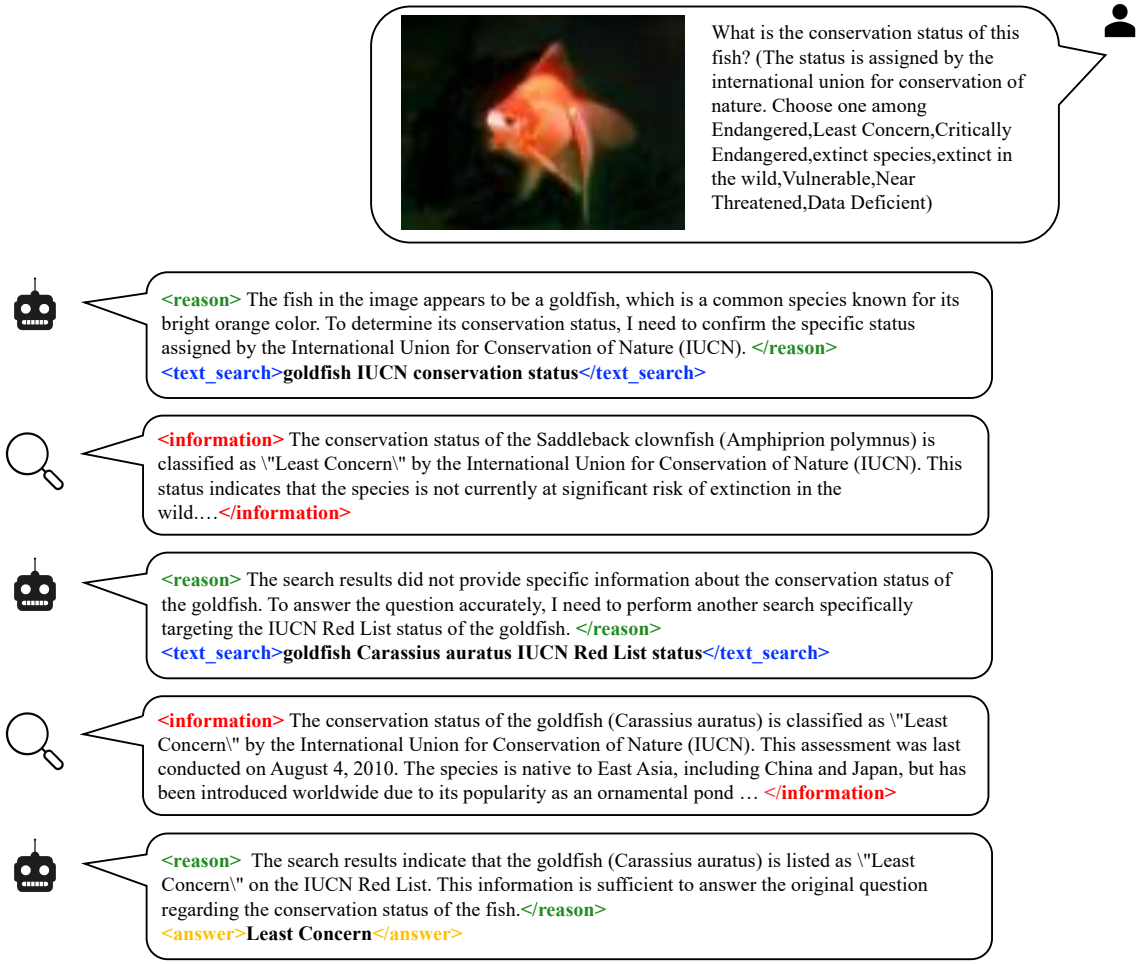

Figure G.1: A sample in DeepMMSearchVQA

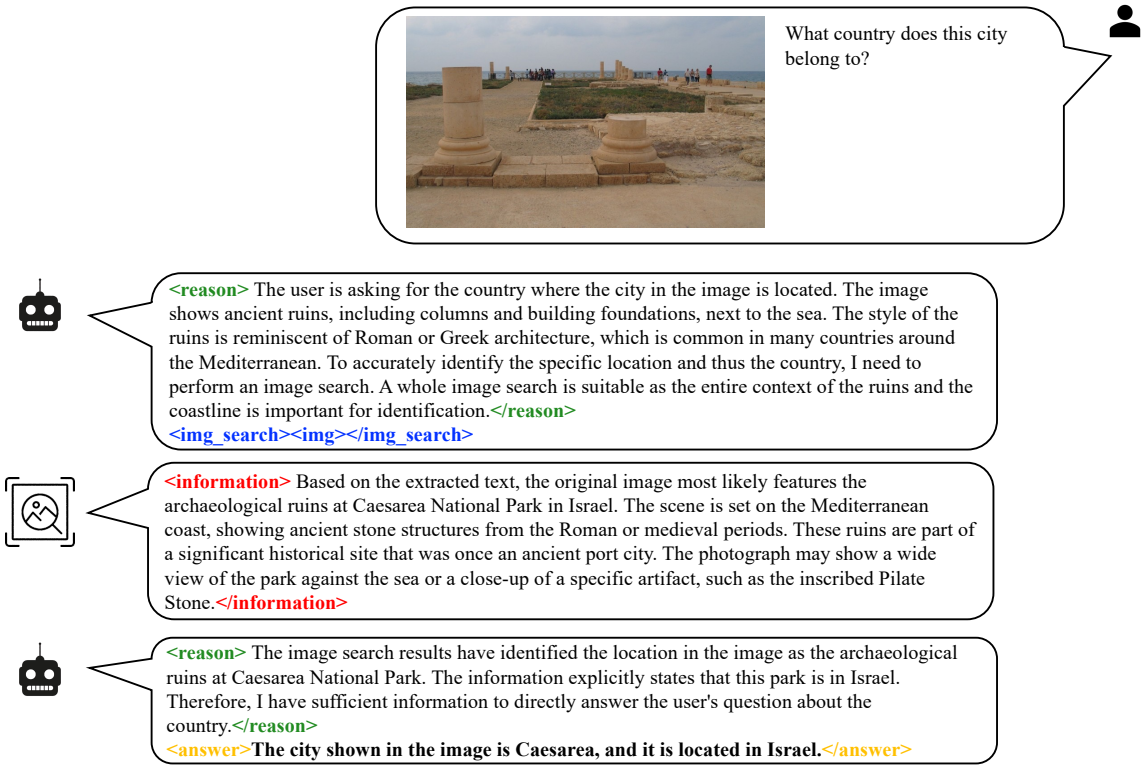

Figure G.2: A sample in DeepMMSearchVQA

H Limitation and Broader Impact

H.1 Limitation

Despite the demonstrated benefits, our work faced several limitations. First, reliance on multiple search tools inherently increases susceptibility to errors arising from tool failures, latency, API query limits, or scraping blocks, all of which can disrupt the reasoning process. Secondly, since web-search tools are dynamic and continuously updated, retrieval outcomes may vary over time, introducing variability in both training and evaluation.

A major bottleneck we encountered during training was performing online GRPO with live web-search tools, which posed challenges in terms of stability and reliability. To mitigate this, we had to implement extensive fail-checks, retries, and safeguards to ensure robust information extraction from the web tools. These limitations highlight that integrating real-time web-search retrieval is not a trivial task and requires substantial efforts.

H.2 Broader Impact

Our work has the potential to significantly advance multimodal information-seeking systems by enabling multimodal LLMs to dynamically retrieve and reason over real-world knowledge. This opens up promising applications in education, digital assistants, research support, and accessibility tools, where timely and accurate information retrieval is essential.

However, reliance on web-search also introduces risks, including amplification of misinformation, propagation of biased or low-quality sources, and challenges related to copyright when models retrieve or summarize content from proprietary sources. Furthermore, depending on poorly reliable web data can undermine factual accuracy and erode user trust. To address these concerns, future work should prioritize incorporating mechanisms for source attribution and quality filtering of retrieved content. We emphasize the importance of responsible deployment, with safeguards to ensure factual reliability, and equitable access, so that such systems can be harnessed in a safe and beneficial manner.