MATCH WHAT YOU KNOW: SEARCH AGENT ACTS TO KNOWLEDGE BOUNDARIES FOR KNOWLEDGE-BASED VISUAL QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge-Based Visual Question Answering (KBVQA) requires models to identify task-relevant entities in images and retrieve complementary information from external resources when their internal knowledge is insufficient. Existing retrieval-augmented generation methods rely on predefined retrieval pipelines, which may lead to either insufficient evidence acquisition or redundant retrieval. Although search agents can dynamically select tools, reinforcement learning often drives them towards a conservative full-pipeline retrieval workflow of "image search followed by text search". Consequently, even when the model already possesses the necessary entity-recognition or question-answering capabilities, it still tends to invoke external tools first, leading to underutilization of its existing capabilities and unnecessary tool calls. To address this issue, we propose **BoundarySA**, a multimodal search agent for KBVQA. Before reinforcement learning, we conduct tool-free behavioral probing to estimate the model's initial capabilities in three aspects: identifying task-relevant entities, answering questions when the correct entity is provided, and directly answering questions from the original image-question pair. During training, we use the probing results to construct capability-boundary supervision, which encourages the model to prioritize its existing capabilities and invoke external tools only when the corresponding capabilities are insufficient. BoundarySA therefore learns to expand its problem-solving ability through tool use while continuing to leverage its original capabilities. Experiments on MMHops, InfoSeek, and E-VQA demonstrate that BoundarySA maintains competitive answer performance while making more effective use of the model's existing capabilities and reducing reliance on the fixed workflow.

1 INTRODUCTION

Recent multimodal large language models (Chen et al., 2026b; He et al., 2025; Liu et al., 2026; Jin et al., 2025; Wu et al., 2026) have made remarkable progress in vision-language tasks. In knowledge-based visual question answering (KBVQA), models must identify long-tail entities and acquire factual knowledge beyond the image, which often can't be fully covered by their parametric knowledge. An effective KBVQA system therefore needs not only to identify question-relevant entities in the image, but also to determine whether the model already possesses the knowledge required to answer the question and whether evidence should be retrieved when its capabilities are insufficient.

Most existing KBVQA methods (Chen et al., 2026a) follow the retrieval-augmented generation (RAG) pipelines. ReAG (Compagnoni et al., 2025) improves the relevance of retrieved information through multi-stage retrieval and evidence filtering; QKVQA (Ye et al., 2026b) further reduces irrelevant context through question-guided fine-grained knowledge selection; and WikiSeeker (Zhu et al., 2026) improves answer generation by iteratively inspecting and refining retrieved results. Although these methods improve the quality and utilization of retrieved evidence, their tool-use processes are constrained by predefined pipelines and can't flexibly adapt to the capabilities that the model already exhibits on different instances. Consequently, they may perform unnecessary retrieval when the model can already identify the relevant entity or answer the question.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

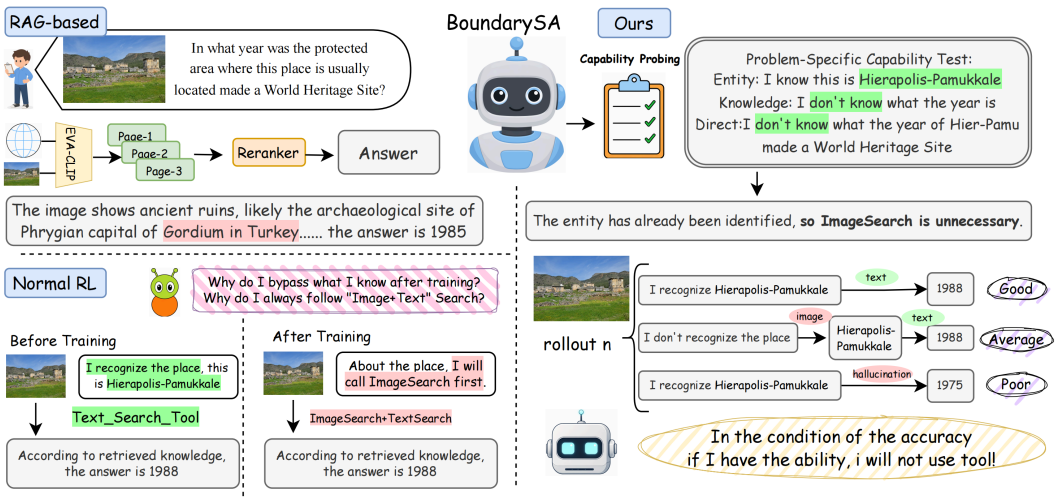


Figure 1: Comparison of RAG-based methods, standard RL, and our BoundarySA pipeline. By probing the model's task-specific capabilities, BoundarySA reduces redundant tool calls and selects more appropriate retrieval paths.

Inspired by tool-augmented agents, recent studies have begun to formulate KBVQA as a multi-turn decision-making process. MMhops-R1 (Zhang et al., 2025) turns KB-VQA into a multimodal search agent task, where the model dynamically chooses between image search, text search, and direct answering to perform multi-hop reasoning over images and Wikipedia. However, we find that after reinforcement learning (RL), search agents often gradually adopt a conservative fallback pattern: regardless of whether full retrieval is actually required, the model tends to invoke image search first, followed by one or more text searches. Consequently, capabilities are increasingly bypassed rather than effectively leveraged during problem solving. Although they can select tools, their training processes still struggle to align tool-use behavior with the model's pre-existing capabilities.

Motivated by these observations, we propose **BoundarySA**, a capability-boundary-aware multimodal search agent for KBVQA. Its core idea is to explicitly estimate the model's existing entity-recognition and question-answering capabilities. These capability estimates are then used to guide tool use during reinforcement learning, encouraging external tools to fill capability gaps rather than replace the model's original abilities. Specifically, BoundarySA consists of two key stages. First, before reinforcement learning, we perform multiple tool-free behavioral probes for each training instance to estimate whether the model can identify the **question-relevant entity**, whether it can **answer the question using its internal knowledge when the correct entity is provided**, and whether it can **directly generate the correct answer from the original image-question pair**; together, these three estimates define the model's initial capability boundary for that instance. Together, the three probing results are used to measure how well different tool-use paths align with the model's initial capability boundary. Second, during reinforcement learning, we use this capability boundary to encourage the model to prioritize solution paths that are consistent with its existing capabilities. As a result, when the model already possesses a particular capability, it relies on its internal capability rather than invoking external tools; when the corresponding capability is insufficient, the model can still invoke search tools to acquire the necessary evidence.

We train BoundarySA on MMHops (Zhang et al., 2025) and evaluate it on MMHops, InfoSeek (Chen et al., 2023), and E-VQA (Mensink et al., 2023). BoundarySA maintains competitive answer performance while better preserving the model's existing capabilities and reducing collapse towards an image-search-then-text-search workflow. Our contributions are summarized as follows:

- We identify two issues in RL-based KBVQA search agents: collapse towards fixed search chains and underutilization of pre-existing entity-question-related capabilities.
- We propose BoundarySA, which probes instance-level capability boundaries before training and uses them to guide tool use during reinforcement learning.

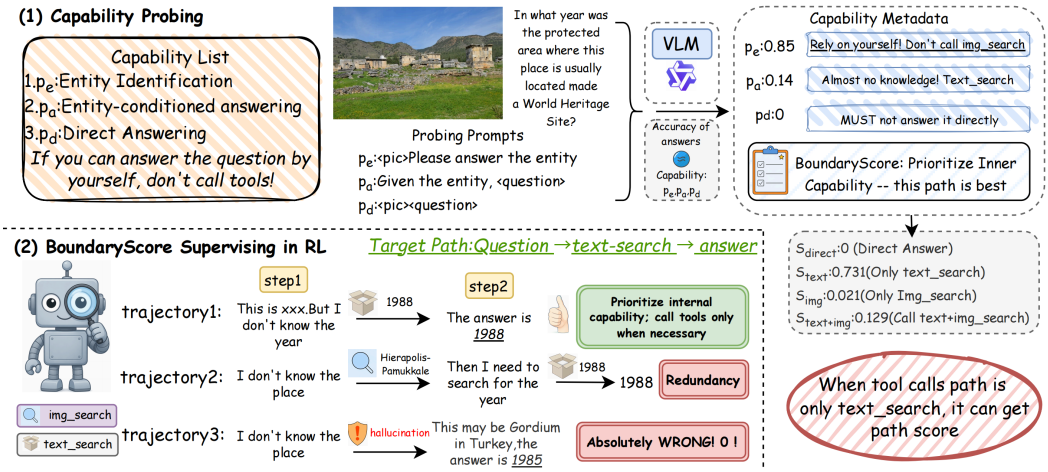


Figure 2: Overview of our method. The method rewards prioritizing the model's intrinsic capabilities over tool calls.

- Experiments on benchmarks show that BoundarySA maintains answer performance while reducing redundant tool calls and reliance on the fixed workflow.

2 Related Works

2.1 Knowledge-Based Visual Question Answering

Knowledge-based visual question answering (KBVQA) requires models to integrate visual content with external knowledge beyond images to answer questions that can't be resolved through visual information alone. Early benchmarks, such as OK-VQA (Marino et al., 2019) and A-OKVQA (Schwenk et al., 2022), primarily focus on open-domain commonsense knowledge. Subsequent benchmarks emphasize fine-grained entities, long-tail factual knowledge, and more complex reasoning processes. InfoSeek (Chen et al., 2023) focuses on information-seeking questions about specific visual entities; E-VQA (Mensink et al., 2023) introduces two-hop reasoning over external knowledge; and MMHops (Zhang et al., 2025) further extends reasoning chains and introduces dual-image inputs, requiring stronger model capabilities.

Many KBVQA studies have mainly focused on retrieval-augmented generation (RAG) methods. EchoSight (Yan & Xie, 2024) retrieves Wikipedia articles through visual search and reranks them using the image–question pair for KB-VQA. ReAG (Compagnoni et al., 2025) combines coarse-to-fine multimodal retrieval, critic-based passage filtering, and reinforcement learning to improve reasoning over retrieved evidence. RMCD (Kim et al., 2026) proposes a training-free relevance-aware multi-context contrastive decoding method that aggregates useful information across multiple retrieved contexts while suppressing irrelevant evidence. QKVQA (Ye et al., 2026b) employs question-guided knowledge representations and dynamic fine-grained text-chunk selection across multiple articles to reduce evidence-selection errors. Although these methods improve the quality and utilization of retrieved knowledge, they mainly focus on how to retrieve, filter, and exploit external evidence, while paying limited attention to whether retrieval remains necessary when the model already possesses the required capabilities.

2.2 Search-Based Agents

Compared with RAG methods that rely on fixed retrieval pipelines, search agents can adaptively decide whether to retrieve, which tool to invoke, and how to refine queries. MMSearch-R1 (Wu et al., 2025) extends this paradigm to multimodal settings to teach models when to invoke tools. LMM-Searcher (Du et al., 2026) targets long-horizon multimodal search by representing retrieved visual evidence as lightweight file references and loading the corresponding images only when needed. In

the KBVQA setting, MMhops-R1(Zhang et al., 2025) employs a composite reward to train model strategies according to question complexity. Although adaptive search improves KBVQA reasoning, dynamic tool selection does not guarantee alignment with the model's existing capabilities. Without explicitly modeling instance-level capability boundaries, current methods may still learn conservative full-retrieval policies that duplicate capabilities the model already possesses.

3 METHODS

3.1 OVERVIEW

KBVQA requires models to decide whether external tools are needed based on their existing capabilities. However, we observe that reinforcement learning often drives agents towards a conservative fallback policy in which models repeatedly follow an image-search-then-text-search workflow. As a result, models increasingly fail to leverage capabilities already available before training, such as entity recognition and direct answering, leading to unnecessary tool use. Motivated by these observations, BoundarySA consists of two stages: capability-boundary probing and boundary-aware reinforcement learning. Before training, we first perform tool-free behavioral probing to evaluate the model's initial capabilities in entity identification, entity-conditioned question answering, and direct answering, thereby characterizing its capability boundary for each instance. During training, we optimize not only answer correctness but also the alignment between tool-use behavior and the capability boundary, encouraging the model to prioritize its existing capabilities and invoke tools only when the corresponding capability is insufficient.

3.2 BOUNDARY SEARCH AGENT

3.2.1 ACTION SPACE FOR KBVQA

In contrast to RAG methods, we formulate KBVQA as a multi-turn sequential decision-making process. Formally, given an image I, a question q, and a multimodal large language model π_θ, at each reasoning step t, the model generates intermediate reasoning conditioned on the current state s_t and selects its action from the action space

$$\mathcal{A} = \{\texttt{ImageSearch}, \texttt{TextSearch}, \texttt{Answer}\}.$$

After a tool action is executed, the resulting observation is appended to the current context to construct the next state s_{t+1}. The `ImageSearch` tool uses the input image as the query and retrieves the top-k pages associated with relevant visual entities from the external knowledge source. The `TextSearch` retrieves the top-k pages relevant to the textual query. During retrieval, pages returned in previous steps are excluded to reduce redundant evidence. We perform evidence summarization for each retrieved page and return it to the model as the tool observation.

3.2.2 CAPABILITY PROBING

The model's ability to identify question-relevant entities and its knowledge of the corresponding entities directly determine whether external tools are required and which tool-use path should be adopted. To explicitly characterize the capabilities already exhibited by the model before reinforcement learning and use them to supervise subsequent tool-use behavior, we perform tool-free capability probing on the initial model. Specifically, for each training instance $x_i = (I_i, q_i, y_i, e_i)$, where I_i, q_i, y_i, and e_i denote the input image, question, ground-truth answer, and question-relevant entity, respectively, we detect the following three capabilities:

1. **Entity identification:** Given the original image and question, whether the model can identify the entity relevant to answering the question by itself.
2. **Entity-conditioned answering:** Given the question and the correct entity, whether the model can generate the correct answer using only its internal knowledge.
3. **Direct answering:** Given the original image and question, whether the model can directly generate the correct answer without invoking any external tool.

These three probing tasks correspond to three key capabilities in the KBVQA solution process: grounding the relevant entity from the image, accessing internal knowledge conditioned on the entity, and directly integrating visual information with internal knowledge to answer the question. Since our goal is to estimate the model's initial capabilities before external assistance, both image search and text search are disabled throughout the probing process.

We use the initial model π_0 to perform the three probing tasks. For each instance and each probing task, we independently sample m outputs and use a judge $J(\cdot)$ to determine whether each output is correct. For the i-th instance, the entity-identification, entity-conditioned answering, and direct-answering capabilities are estimated as

$$p_e^{(i)} = \frac{1}{m}\sum_{j=1}^{m} J\left(\hat{e}_{i,j}, e_i\right) p_a^{(i)} = \frac{1}{m}\sum_{j=1}^{m} J\left(\hat{y}_{i,j}^{e}, y_i\right) p_d^{(i)} = \frac{1}{m}\sum_{j=1}^{m} J\left(\hat{y}_{i,j}^{d}, y_i\right).$$

Here, $\hat{e}_{i,j}$ is the output of entity probe j, $\hat{y}_{i,j}^{e}$ is the answer generated with the correct entity, and $\hat{y}_{i,j}^{d}$ is the direct answer from the original image-question pair. J denotes the entity-matching or answer-correctness judge.

From Capability Probing to Path Compatibility Scores. Based on the estimated capabilities, we further construct capability-compatibility scores for four representative solution paths:

$$S_{\text{direct}}^{(i)} = \max\left(p_d^{(i)}, p_e^{(i)} p_a^{(i)}\right),\ S_{\text{image}}^{(i)} = g\left(p_d^{(i)}\right)\left(1 - p_e^{(i)}\right) p_a^{(i)},$$
$$S_{\text{text}}^{(i)} = g\left(p_d^{(i)}\right) p_e^{(i)}\left(1 - p_a^{(i)}\right),\ S_{\text{image+text}}^{(i)} = g\left(p_d^{(i)}\right)\left(1 - p_e^{(i)}\right)\left(1 - p_a^{(i)}\right).$$

The direct-answering capability suppresses unnecessary paths through the following gating function:

$$g(p_d) = \begin{cases} 1, & p_d \le \rho, \\ \dfrac{1 - p_d}{1 - \rho}, & p_d > \rho, \end{cases} \qquad \rho = 0.75.$$

These scores measure how well each tool-use path matches the model's initial capability boundary, rather than the probability that the model selects the corresponding path. Specifically, a high S_{direct} indicates that the model has the internal capabilities for tool-free answering and should therefore prioritize the direct path over unnecessary retrieval. A high S_{image} indicates that the model lacks entity-identification capability but can answer using its internal knowledge once the correct entity is available. A high S_{text} indicates that the model can identify the relevant entity but lacks the entity-related knowledge required to answer the question. A high $S_{\text{image+text}}$ indicates that the model lacks both entity-identification and entity-conditioned answering capabilities, and requires both image search and text search. Under this construction, the stronger the model's corresponding capability, the lower the score assigned to tool-use paths that compensate for that capability.

3.2.3 TRAINING

BoundarySA training has two stages: rejection-sampling supervised fine-tuning and RL.

1. Rejection-Sampling Supervised Fine-Tuning (SFT). Directly applying RL to the initial model is often unstable, so we first construct a small SFT dataset through rejection sampling.

For training instances, we sample N trajectories from the initial policy π_0. Each trajectory consists of model-generated reasoning, tool actions, environment observations, and a final answer. We retain only trajectories that simultaneously satisfy the following conditions:

$$\mathcal{V}_i = \left\{\tau \in \mathcal{T}_i \mid C_{\text{ans}}(\tau) = 1,\ C_{\text{fmt}}(\tau) = 1,\ C_{\text{tool}}(\tau) = 1\right\},$$

where C_{ans} indicates whether the final answer is correct, C_{fmt} indicates whether the trajectory follows the predefined interaction format, and C_{tool} indicates whether all tool actions can be successfully executed and return valid observations. Specifically, each reasoning segment and action must be enclosed within the corresponding structured tags, and every search query must be correctly parsed and executed by the environment such as `<answer></answer>`.

We then perform SFT using the standard autoregressive maximum-likelihood objective:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\sum_{\tau \in \mathcal{D}_{\mathrm{SFT}}} \sum_{t \in \mathcal{M}(\tau)} \log \pi_\theta \left(y_t \mid x_i, y_{<t} \right),$$

where $\mathcal{M}(\tau)$ denotes the positions of tokens generated by the model in trajectory τ. Search results returned by the environment are excluded from the loss. Excessive supervised fine-tuning may cause the model to imitate the search preferences present in the cold-start data and shift its pre-RL capability boundary. To mitigate this effect, we strictly limit both the size of the cold-start dataset and the number of supervised fine-tuning steps.

2. Reinforcement Learning. After cold-start training, we optimize BoundarySA using Group Sequence Policy Optimization (GSPO). For a trajectory τ, we map it to one of four tool-use paths according to the search actions it contains:

$$b(\tau) \in \mathcal{B} = \{\mathrm{direct}, \mathrm{image}, \mathrm{text}, \mathrm{image{+}text}\}.$$

Here, direct denotes answering without invoking any search tool, image and text denote using only image search or text search, respectively, and $\mathrm{image{+}text}$ denotes trajectories that invoke both types of search tools. Based on the four path compatibility scores above, we define the boundary-consistency reward as

$$r_{\mathrm{bound}}(\tau, x_i) = \mathbf{1}\left[S^{(i)}_{b(\tau)} = \max_{b \in \mathcal{B}} S^{(i)}_b \right].$$

The reward is 1 when the tool-use path adopted by the trajectory matches one of highest-scoring capability-compatible paths for the current instance. If multiple paths share the maximum score, all of them are regarded as boundary-consistent. The total trajectory reward is defined as

$$R(\tau, x_i) = r_{\mathrm{ans}}(\tau)\big(1 + \lambda r_{\mathrm{bound}}(\tau, x_i)\big) + \beta r_{\mathrm{format}}(\tau).$$

where r_{ans} is the final-answer correctness reward, r_{format} indicates whether the output follows the predefined reasoning and action format, and λ and β control the weights of the capability-boundary reward and format reward, respectively. This ensures that the capability-boundary reward is activated only when the final answer is correct. In other words, an incorrect trajectory does not receive additional reward when it follows a boundary-consistent tool-use path.

4 Experiments

4.1 Datasets and Evaluation Metrics

We train BoundarySA on the MMHops (Zhang et al., 2025) training set and evaluate its in-domain performance on the MMHops test set. To further assess cross-dataset generalization, we evaluate models on InfoSeek (Chen et al., 2023) and E-VQA (Mensink et al., 2023). We use normalized matching for InfoSeek, BERT Matching for E-VQA, and the official MMHops metric. Detailed dataset settings and descriptions are provided in the supplementary material.

4.2 Implementation Details

We initialize BoundarySA from Qwen3-VL-8B-Instruct (Bai et al., 2025) and implement it using LLaMA-Factory (Zheng et al., 2024) and VeRL (Sheng et al., 2025). For cold-start training, we use 1.5K MMHops instances and retain at most one valid trajectory per instance. We fine-tune for two epochs with a learning rate of 1×10^{-5}. Reinforcement learning is conducted on the remaining training instances for three epochs with a batch size of 128, a learning rate of 1×10^{-6}, maximum interaction turns of $T = 8$, and $\rho = 0.75$. We use EVA-CLIP (Sun et al., 2023) and BGE (Chen et al., 2025) for image and text retrieval, and retrieve the top three results for each search. Qwen3-VL-32B is used to summarize retrieved evidence before it is returned to the agent. Additional training, sampling, retrieval, and reward hyperparameters are provided in the supplementary material.

Method	Retriever	Bridging				Comparison
		String	Numerical	Time	Overall	
Vanilla MLLMs						
Qwen3-VL-2B	-	16.85	15.18	15.78	15.70	3.42
Qwen3-VL-8B	-	23.94	22.24	23.12	22.82	5.24
Search-Agent Models, Qwen3-VL-8B Base						
MMSearch-R1(Wu et al., 2025)	BGE+EVA-CLIP-8B	43.12	41.23	42.31	41.89	18.10
DeepEyesV2(Hong et al., 2026a)	BGE+EVA-CLIP-8B	42.93	44.82	43.68	44.15	18.97
mRAG and Agentic mRAG, Qwen3-VL-8B Base						
EchoSight(Yan & Xie, 2024)	EVA-CLIP-8B	30.62	27.86	29.72	28.88	9.81
CC-VQA(Hong et al., 2026b)	EVA-CLIP-8B	31.48	29.67	30.50	30.27	11.50
REAL(Ye et al., 2026a)	EVA-CLIP-8B	34.21	31.32	32.22	32.20	12.81
MMhops-R1	BGE+EVA-CLIP-8B	48.25	48.02	47.12	47.91	20.75
BoundarySA (Ours)	**BGE+EVA-CLIP-8B**	**50.82**	**53.18**	**51.86**	**52.36**	**22.98**

Table 1: Performance comparison on MMHops. All methods use Qwen3-VL-8B as the base model unless otherwise specified.

Reward	MMHops(Overall)
Ans	37.3
Ans+fmt	43.8
Ans+fmt+tool penalty	41.2
BoundarySA	**47.9**

Table 2: Ablation of different reward designs (Ans: answer,fmt: format).

Text	Image	MMHops(Overall)
2	2	41.4
3	**3**	**47.9**
4	4	46.2

Table 3: Performance under different image- and text-search budgets.

top-k	MMHops(Overall)
1	35.3
2	42.6
3	**47.9**
4	46.4

Table 4: Ablation on image and text retrieval top-K. (Overall:Full dataset Eval)

4.3 Main Results

Table 1 reports the main results on MMHops. Overall, BoundarySA achieves the best performance on MMHops, reaching **52.36%** and **22.98%** on bridging and comparison set. Compared with vanilla models, BoundarySA effectively supplements missing long-tail entity knowledge through image and text search. Compared with several fixed-pipeline RAG methods, BoundarySA can dynamically determine whether retrieval is necessary and which tool should be invoked according to the current reasoning state. Compared with existing search agents, BoundarySA also improves answer accuracy in both bridging and comparison subsets.

To further evaluate generalization, we evaluate the model on InfoSeek and E-VQA. As shown in Table 5, BoundarySA achieves **42.54%** and **45.62%** on the two cross-dataset evaluations respectively. These results suggest that BoundarySA does not merely learn a tool-use pattern specific to MMHops, but instead acquires a capability-aware search strategy that transfers across different entity distributions, knowledge-base scales, and reasoning depths.

4.4 Capability Boundaries and Tool-Use Behavior

To analyze how capability-boundary supervision affects tool-use behavior, we first examine the distribution of capability-compatible target paths over the training set. As shown in Figure 4(c), many instances can be solved through direct answering, image search only, or text search only. This observation shows that even on a challenging benchmark, applying the complete search chain uniformly is not always necessary.

Figure 3 presents the actual tool-path distributions of the baseline and BoundarySA during RL. As training progresses, the baseline rapidly increases its use of the $\mathrm{image}{+}\mathrm{text}$ path and gradually adopts it as a uniform fallback strategy across different types of questions. In contrast, the tool-path

Method	E-VQA		InfoSeek		
	S-Hop	All	Un-Q	Un-E	All
		Search-Agent Models			
MMSearch-R1	40.16	40.24	39.78	38.83	39.10
DeepEyesV2	39.58	38.97	41.16	40.31	40.55
		Retrieval-Augmented Models			
EchoSight	32.68	30.12	33.32	34.05	33.84
CC-VQA	40.23	39.67	39.94	39.12	39.35
REAL	43.37	42.76	41.38	40.86	41.01
MMhops-R1	40.35	39.74	40.47	39.87	40.04
BoundarySA	**45.24**	**45.62**	**43.43**	**42.18**	**42.54**

Table 5: Performance comparison on E-VQA and InfoSeek.

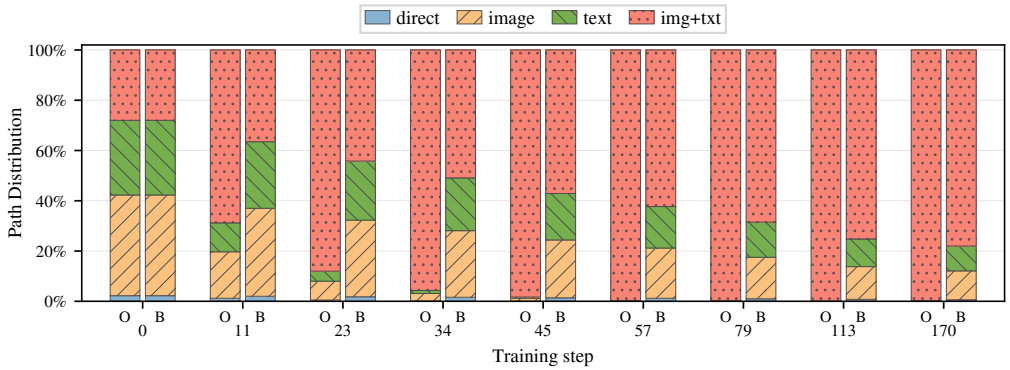


Figure 3: Answer-path analysis during training. We compare the baseline trained with answer and format rewards (O) and BoundarySA (B) across four paths: direct answering, image search, text search, and image-plus-text search.

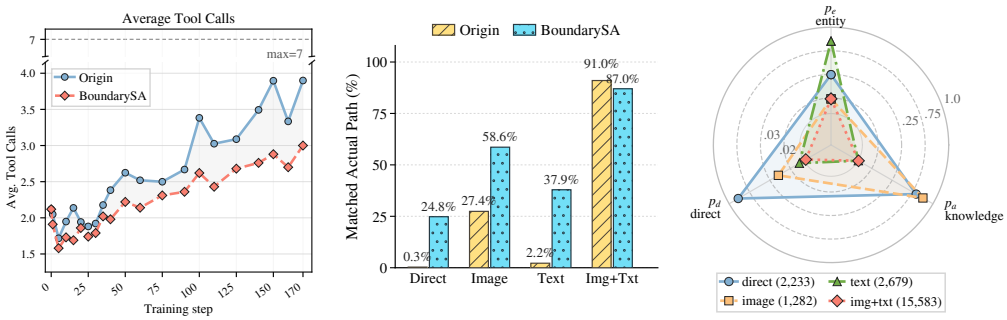
distribution of BoundarySA remains more consistent with the capability boundaries estimated before training. These results indicate that capability-boundary supervision mitigates the collapse towards a fixed tool chain during RL. Figure 4(b) shows that BoundarySA substantially improves alignment on the direct, image-only, and text-only paths. Origin achieves higher alignment on the image-plus-text path, largely because it disproportionately adopts the full-retrieval workflow. Figure 4(a) further compares the average number of tool calls during training. The baseline exhibits a rapid increase in tool usage, whereas BoundarySA slows this growth. The reduction mainly comes from retrieval steps that duplicate capabilities already available within the model.

4.5 Ablation Studies

In the ablation studies, "Overall" denotes performance on the full set, computed jointly over bridging and comparison questions. The detailed ablation analysis is as follows.

4.5.1 Reward

To evaluate the effectiveness of the capability-boundary reward, we compare three training objectives: answer-only reward, answer reward combined with a tool penalty, and the complete BoundarySA reward. For the tool-penalty baseline, we use the following commonly adopted tool cost: $r_{\text{tool}}(\tau) = -\frac{n_{\text{tool}}(\tau)}{N_{\max}}$, where $n_{\text{tool}}(\tau)$ denotes the number of tool calls in trajectory τ. $N_{\max}$ denotes the maximum allowed number of tool calls. As shown in Table 2, BoundarySA achieves better performance while all other designs perform worse than BoundarySA. In particular, our analysis suggests that the performance degradation caused by the tool penalty is attributable to the prevalence of multi-hop questions in the dataset. When successful trajectories are sparse, differences in tool-call counts dominate the reward variation, causing the policy to favor shorter trajectories with



(a) Average tool calls. (b) Boundary-path alignment ratio. (c) Capability scores.

Figure 4: Training dynamics and capability-boundary analysis. (a) Average number of tool calls during training. (b) Alignment ratio between actual tool-use paths and capability-compatible target paths at the final RL step. (c) Mean capability scores of training instances grouped by capability-compatible target path.

smaller penalties when additional retrieval is necessary, which reduces task success and drives the model towards insufficient search.

4.5.2 Tool Calls and Top-K

We further conduct ablations on the tool budget and retrieval top-K. As shown in Table 3 and Table 4, the best performance is achieved with a tool-budget configuration of $(3, 3)$ and $K = 3$. Insufficient search fails to retrieve enough useful evidence, whereas excessive search introduces irrelevant information and retrieval noise, placing additional burdens on evidence summarization and subsequent reasoning. Similarly, an appropriate top-K provides sufficient evidence while limiting context length, enabling the model to better focus on each retrieved item.

4.5.3 Hyperparameter Analysis

We further analyze the effects of the capability-probing sample size m, the direct-answering threshold ρ, and the reward weights λ and β on answer accuracy and tool-use behavior. Larger m improves estimation reliability with diminishing returns, while ρ balances reliance on intrinsic capability and external retrieval. The weights λ and β regulate capability-consistent tool use and format compliance, respectively, although overly large values may over-constrain the policy. Detailed results are provided in the Section C.0.3.

Capability-Boundary Shifts during Training. We do not assume that the model's capability boundary remains fixed throughout training. Instead, we formally show that, even when the boundary shifts, optimizing the policy with respect to the initial capability boundary is no worse than the original baseline in terms of trajectory discrimination and directional guidance under the stated assumptions. A proof of this formulation is provided in Section B.1.

5 Conclusion

To address the tendency of KBVQA agents to overlook their existing capabilities and over-rely on fixed tool chains, we propose BoundarySA. It characterizes the model's entity-recognition and question-answering boundaries through capability probing and uses them to guide tool invocation, allowing the model to prioritize its internal capabilities. Experiments on KBVQA benchmarks show that BoundarySA maintains competitive answer performance while effectively reducing redundant tool calls and reliance on the fixed image-search-then-text-search workflow.

6 AI USE STATEMENT

In this work, we used generative AI tools solely to polish the writing of the manuscript. We have reviewed all AI-assisted edits and take responsibility for the final content of this work, including all text, claims, and artifacts.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2025. URL https://arxiv.org/abs/2402.03216.

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

Zhuohong Chen, Zhengxian Wu, Zirui Liao, Shenao Jiang, Hangrui Xu, Yang Chen, Chaokui Su, Xiaoyu Liu, and Haoqian Wang. R3g: A reasoning-retrieval-reranking framework for vision-centric answer generation, 2026a. URL https://arxiv.org/abs/2602.00104.

Zhuohong Chen, Zhenxian Wu, Yunyao Yu, Hangrui Xu, Zirui Liao, Zhifang Liu, Xiangwen Deng, Pen Jiao, and Haoqian Wang. Learning to search: A decision-based agent for knowledge-based visual question answering. *CoRR*, abs/2604.07146, 2026b. doi: 10.48550/ARXIV.2604.07146. URL https://doi.org/10.48550/arXiv.2604.07146.

Alberto Compagnoni, Marco Morini, Sara Sarto, Federico Cocchi, Davide Caffagni, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Reag: Reasoning-augmented generation for knowledge-based visual question answering. *CoRR*, abs/2511.22715, 2025. doi: 10.48550/ARXIV.2511.22715. URL https://doi.org/10.48550/arXiv.2511.22715.

Yifan Du, Zikang Liu, Jinbiao Peng, Jie Wu, Junyi Li, Jinyang Li, Wayne Xin Zhao, and Ji-Rong Wen. Towards long-horizon agentic multimodal search, 2026. URL https://arxiv.org/abs/2604.12890.

Yicheng He, Chengsong Huang, Zongxia Li, Jiaxin Huang, and Yonghui Yang. Visplay: Self-evolving vision-language models from images, 2025. URL https://arxiv.org/abs/2511.15661.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model, 2026a. URL https://arxiv.org/abs/2511.05271.

Yuyang Hong, Jiaqi Gu, Yujin Lou, Lubin Fan, Qi Yang, Ying Wang, Kun Ding, Yue Wu, Shiming Xiang, and Jieping Ye. Cc-vqa: Conflict- and correlation-aware method for mitigating knowledge conflict in knowledge-based visual question answering, 2026b. URL https://arxiv.org/abs/2602.23952.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.09516.

Jongha Kim, Byungoh Ko, Jeehye Na, Jinsung Yoon, and Hyunwoo J. Kim. Relevance-aware multi-context contrastive decoding for retrieval-augmented visual question answering, 2026. URL https://arxiv.org/abs/2602.06050.

Ruiqi Liu, Xiaolei Lv, Gengsheng Li, Ximo Zhu, Zhiheng Wang, Zhengbo Zhang, Junkai Chen, Zhiheng Li, Bo Li, Jun Gao, and Shu Wu. Visual-advantage on-policy distillation for vision-language models, 2026. URL https://arxiv.org/abs/2605.21924.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge, 2019. URL https://arxiv.org/abs/1906.00067.

Thomas Mensink, Jasper Uijlings, Lluis Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araujo, and Vittorio Ferrari. Encyclopedic vqa: Visual questions about detailed properties of fine-grained categories, 2023. URL https://arxiv.org/abs/2306.09224.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge, 2022. URL https://arxiv.org/abs/2206.01718.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL http://dx.doi.org/10.1145/3689031.3696075.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale, 2023. URL https://arxiv.org/abs/2303.15389.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. Mmsearch-r1: Incentivizing lmms to search, 2025. URL https://arxiv.org/abs/2506.20670.

Zhengxian Wu, Hangrui Xu, Kai Shi, Zhuohong Chen, Yunyao Yu, Chuanrui Zhang, Zirui Liao, Jun Yang, Zhenyu Yang, Haonan Lu, and Haoqian Wang. Promsa:progressive multimodal search agents for knowledge-based visual question answering. *CoRR*, abs/2606.27974, 2026. doi: 10.48550/ARXIV.2606.27974. URL https://doi.org/10.48550/arXiv.2606.27974.

Yibin Yan and Weidi Xie. Echosight: Advancing visual-language models with wiki knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1538–1551. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.83. URL http://dx.doi.org/10.18653/v1/2024.findings-emnlp.83.

Kai Ye, Xianwei Mao, Sheng Zhou, Zirui Shao, Ye Mo, Liangliang Liu, Haikuan Huang, Bin Li, and Jiajun Bu. Real: Resolving knowledge conflicts in knowledge-intensive visual question answering via reasoning-pivot alignment, 2026a. URL https://arxiv.org/abs/2602.14065.

Wei Ye, Yixin Su, Yueguo Chen, Longxiang Gao, Jianjun Li, Ruixuan Li, and Rui Zhang. QKVQA: question-focused filtering for knowledge-based VQA. *CoRR*, abs/2601.13856, 2026b. doi: 10.48550/ARXIV.2601.13856. URL https://doi.org/10.48550/arXiv.2601.13856.

Tao Zhang, Ziqi Zhang, Zongyang Ma, Yuxin Chen, Bing Li, Chunfeng Yuan, Guangting Wang, Fengyun Rao, Ying Shan, and Weiming Hu. Mmhops-r1: Multimodal multi-hop reasoning, 2025. URL https://arxiv.org/abs/2512.13573.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models, 2024. URL https://arxiv.org/abs/2403.13372.

Yingjian Zhu, Xinming Wang, Kun Ding, Ying Wang, Bin Fan, and Shiming Xiang. Wikiseeker: Rethinking the role of vision-language models in knowledge-based visual question answering, 2026. URL https://arxiv.org/abs/2604.05818.

A Implementation Details

This section summarizes the key hyperparameters used for BoundarySA. Unless otherwise specified, we use Qwen3-VL-8B-Instruct as the base model and train the agent for three epochs with GSPO-style policy loss.

Training Configuration	
Base model	Qwen3-VL-8B-Instruct
RL algorithm / Learning rate	GSPO / 1×10^{-6}
Training epochs	3
Train batch size	128
Validation batch size	512
PPO mini-batch size	128
PPO micro-batch size / GPU	2
Clip ratio (low / high)	0.20 / 0.28
Entropy coefficient	0
KL loss	0

Table 6: Training configuration for BoundarySA.

Rollout and Agent Configuration	
Rollout samples per prompt	8
Max assistant turns	8
Max tool response length	8192
Inference backend	SGLang
Presence penalty	1.5
Repetition penalty	1.0

Table 7: Rollout and agent configuration.

Reward and Tool Settings	
Reward manager	`boundary-tool-gated`
Format reward β	0.5
BoundaryScore weight λ	0.2
Max tool calls	8
Judge evaluation	enabled

Table 8: Reward and tool settings.

System Configuration	
Vision tower / GPU configuration / inference parallelism	Qwen3-VL / 8 GPUs per node / TP$= 4$

Table 9: System configuration.

BoundarySA consists of following stages: tool-free capability probing, cold-start supervised fine-tuning, and boundary-aware reinforcement learning. The probing stage estimates the model's instance-level capabilities before reinforcement learning. The reinforcement-learning stage then uses these estimates to reward successful trajectories whose tool-use paths match the model's capability boundary.

For each training instance $x_i = (I_i, q_i, y_i, e_i)$, we estimate three capabilities. The entity-identification score $p_e^{(i)}$ measures whether the model can identify the question-relevant entity from the image and question. The entity-conditioned answering score $p_a^{(i)}$ measures whether the model can answer the question when the correct entity is given. The direct-answering score $p_d^{(i)}$ measures whether the model can answer directly from the original image-question pair without external tools.

Algorithm 1 BoundarySA Training

Require: $\mathcal{D}$, π_0, m, $\mathcal{B}$, λ, β
Ensure: Boundary-aware search agent π_θ
Initialize $\theta \leftarrow \theta_0$
for each instance $x_i = (I_i, q_i, y_i, e_i) \in \mathcal{D}$ **do**
  Probe entity outputs $\{\hat{e}_{i,j}\}_{j=1}^{m}$ without tools
  Probe entity-conditioned answers $\{\hat{y}^{e}_{i,j}\}_{j=1}^{m}$
  Probe direct answers $\{\hat{y}^{d}_{i,j}\}_{j=1}^{m}$
  Estimate $p_e^{(i)}$, $p_a^{(i)}$, and $p_d^{(i)}$
  Compute path scores $S_b^{(i)}$ for all $b \in \mathcal{B}$
end for
Rejection-sample valid and correct trajectories from π_0
Supervised fine-tune π_θ on retained trajectories
while RL training is not finished **do**
  for each sampled instance x_i **do**
    Sample $\{\tau_{i,g}\}_{g=1}^{G}$ from the current policy
    for each trajectory $\tau_{i,g}$ **do**
      Identify $b(\tau_{i,g})$ and compute r_{bound}
      Compute $R(\tau_{i,g}, x_i)$
    end for
    Update π_θ with GSPO
  end for
end while
return π_θ

The path set is

$$\mathcal{B} = \{\text{direct}, \text{image}, \text{text}, \text{image+text}\}\,.$$

For a trajectory τ, $b(\tau)$ is determined by whether it invokes image search, text search, both tools, or no external tool before producing the final answer. The boundary-consistency reward is

$$r_{\text{bound}}(\tau, x_i) = \mathbb{I}\left[S^{(i)}_{b(\tau)} = \max_{b \in \mathcal{B}} S^{(i)}_b\right],$$

where all of the highest-scoring paths are treated as boundary-consistent when ties occur.

B Reward-Level Analysis

B.1 Reward Preservation and Preference Refinement

We provide a reward-level analysis tailored to BoundarySA. The analysis characterizes how the proposed reward preserves and refines the trajectory preferences induced by the original objective.

Setup. Let τ denote a complete trajectory for an instance x_i. Before reinforcement learning, BoundarySA probes the model's initial capability profile $c_0^{(i)}$ and identifies a fixed set of capability-compatible target paths for x_i. We denote whether τ is consistent with this initial target-path set by

$$r^0_{\text{bound}}(\tau, x_i) = r_{\text{bound}}\left(\tau, x_i; c_0^{(i)}\right) \in \{0, 1\}.$$

Although the model's latent capability profile may evolve during training, the boundary label r^0_{bound} remains anchored to the initially probed capability profile.

The original search-agent reward is

$$R_{\text{orig}}(\tau, x_i) = r_{\text{ans}}(\tau, x_i) + \beta r_{\text{format}}(\tau),$$

where $r_{\text{ans}}(\tau, x_i) \in \{0, 1\}$ and $r_{\text{format}}(\tau) \in \{0, 1\}$. BoundarySA uses

$$R_{\text{B}}(\tau, x_i) = r_{\text{ans}}(\tau, x_i)\left(1 + \lambda r^0_{\text{bound}}(\tau, x_i)\right) + \beta r_{\text{format}}(\tau),$$

where $\lambda \geq 0$. Equivalently,

$$R_{\mathrm{B}}(\tau, x_i) = R_{\mathrm{orig}}(\tau, x_i) + \lambda r_{\mathrm{ans}}(\tau, x_i) r^0_{\mathrm{bound}}(\tau, x_i). \tag{1}$$

Property 1: Pointwise Reward Preservation

For every trajectory τ,

$$R_{\mathrm{B}}(\tau, x_i) \geq R_{\mathrm{orig}}(\tau, x_i).$$

Thus, BoundarySA never subtracts raw reward from the original objective.

Proof. From Equation (1),

$$R_{\mathrm{B}}(\tau, x_i) - R_{\mathrm{orig}}(\tau, x_i) = \lambda r_{\mathrm{ans}}(\tau, x_i) r^0_{\mathrm{bound}}(\tau, x_i).$$

Since $\lambda \geq 0$ and both reward indicators are binary,

$$\lambda r_{\mathrm{ans}}(\tau, x_i) r^0_{\mathrm{bound}}(\tau, x_i) \geq 0.$$

Therefore,

$$R_{\mathrm{B}}(\tau, x_i) \geq R_{\mathrm{orig}}(\tau, x_i).$$

More specifically,

$$R_{\mathrm{B}}(\tau, x_i) = \begin{cases} R_{\mathrm{orig}}(\tau, x_i), & r_{\mathrm{ans}}(\tau, x_i) = 0, \\ R_{\mathrm{orig}}(\tau, x_i), & r_{\mathrm{ans}}(\tau, x_i) = 1,\ r^0_{\mathrm{bound}}(\tau, x_i) = 0, \\ R_{\mathrm{orig}}(\tau, x_i) + \lambda, & r_{\mathrm{ans}}(\tau, x_i) = 1,\ r^0_{\mathrm{bound}}(\tau, x_i) = 1. \end{cases}$$

Consequently, a correct trajectory induced by an evolved capability but inconsistent with the initial boundary still retains the complete original reward. A correct trajectory consistent with the initial boundary receives an additional bonus. □

Preservation of correctness discrimination. Consider a correct trajectory τ_c and an incorrect trajectory τ_w with the same format reward:

$$r_{\mathrm{format}}(\tau_c) = r_{\mathrm{format}}(\tau_w).$$

Under the original objective,

$$R_{\mathrm{orig}}(\tau_c, x_i) - R_{\mathrm{orig}}(\tau_w, x_i) = 1.$$

Under BoundarySA,

$$R_{\mathrm{B}}(\tau_c, x_i) - R_{\mathrm{B}}(\tau_w, x_i) = 1 + \lambda r^0_{\mathrm{bound}}(\tau_c, x_i).$$

Because $r^0_{\mathrm{bound}}(\tau_c, x_i) \in \{0, 1\}$,

$$R_{\mathrm{B}}(\tau_c, x_i) - R_{\mathrm{B}}(\tau_w, x_i) \geq R_{\mathrm{orig}}(\tau_c, x_i) - R_{\mathrm{orig}}(\tau_w, x_i). \tag{2}$$

Therefore, BoundarySA never weakens the raw-reward preference for a correct trajectory over an incorrect trajectory. When the correct trajectory is also consistent with the initial capability boundary, the reward margin increases from 1 to $1 + \lambda$.

Property 2: Preservation of Correctness Ordering

For trajectories with the same format reward, BoundarySA preserves the baseline ordering between correct and incorrect trajectories. The corresponding raw-reward margin is either unchanged or enlarged.

Preference refinement among successful trajectories. Now consider two correct trajectories with the same format reward. Let τ^+ be consistent with the initial capability boundary and let τ^0 be inconsistent with it:

$$r_{\mathrm{ans}}(\tau^+, x_i) = r_{\mathrm{ans}}(\tau^0, x_i) = 1,$$
$$r^0_{\mathrm{bound}}(\tau^+, x_i) = 1, \qquad r^0_{\mathrm{bound}}(\tau^0, x_i) = 0.$$

The original objective assigns the same reward to both trajectories:

$$R_{\text{orig}}(\tau^+, x_i) = R_{\text{orig}}(\tau^0, x_i).$$

Hence,

$$R_{\text{orig}}(\tau^+, x_i) - R_{\text{orig}}(\tau^0, x_i) = 0.$$

In contrast, BoundarySA gives

$$R_{\text{B}}(\tau^+, x_i) - R_{\text{B}}(\tau^0, x_i) = \lambda.$$

Therefore, when $\lambda > 0$,

$$R_{\text{B}}(\tau^+, x_i) > R_{\text{B}}(\tau^0, x_i). \tag{3}$$

BoundarySA thus breaks the reward tie between two successful trajectories that the original objective cannot distinguish.

To express this relation more explicitly, define

$$\mathcal{T}_{\text{A}} = \left\{\tau : r_{\text{ans}}(\tau, x_i) = 1,\ r^0_{\text{bound}}(\tau, x_i) = 1\right\},$$
$$\mathcal{T}_{\text{E}} = \left\{\tau : r_{\text{ans}}(\tau, x_i) = 1,\ r^0_{\text{bound}}(\tau, x_i) = 0\right\},$$

and

$$\mathcal{T}_{\text{W}} = \left\{\tau : r_{\text{ans}}(\tau, x_i) = 0\right\}.$$

Here, $\mathcal{T}_{\text{E}}$ may include successful trajectories produced by capabilities acquired after the initial probing stage.

For trajectories with the same format reward, the original objective induces the ordering

$$\mathcal{T}_{\text{A}} = \mathcal{T}_{\text{E}} > \mathcal{T}_{\text{W}},$$

whereas BoundarySA, for $\lambda > 0$, induces

$$\mathcal{T}_{\text{A}} > \mathcal{T}_{\text{E}} > \mathcal{T}_{\text{W}}.$$

Therefore, for trajectories with the same format reward, BoundarySA preserves every strict correctness preference provided by the original reward while converting one of its ties into a strict preference.

Property 3: Successful-Path Preference Refinement

For trajectories with the same format reward, BoundarySA is no less discriminative than the original reward in terms of trajectory ordering. It becomes strictly more discriminative when successful trajectories differ in their consistency with the initial capability boundary.

Propagation to group-relative advantages. For a rollout group

$$\mathcal{G}_i = \{\tau_1, \ldots, \tau_K\},$$

consider the group-normalized advantage

$$A_k = \frac{R(\tau_k, x_i) - \overline{R}_i}{\sigma_i + \epsilon},$$

where

$$\overline{R}_i = \frac{1}{K}\sum_{j=1}^{K} R(\tau_j, x_i),$$

σ_i is the group reward standard deviation, and $\epsilon > 0$.

For any two trajectories τ_a and τ_b in the same group,

$$A(\tau_a) - A(\tau_b) = \frac{R(\tau_a, x_i) - R(\tau_b, x_i)}{\sigma_i + \epsilon}. \tag{4}$$

Because the denominator is strictly positive, group normalization preserves the sign of every within-group reward difference.

For the two successful trajectories τ^+ and τ^0 defined above, the original reward gives

$$A_{\mathrm{orig}}(\tau^+) - A_{\mathrm{orig}}(\tau^0) = 0.$$

BoundarySA instead gives

$$A_{\mathrm{B}}(\tau^+) - A_{\mathrm{B}}(\tau^0) = \frac{\lambda}{\sigma_{\mathrm{B},i} + \epsilon} > 0, \tag{5}$$

whenever $\lambda > 0$. Thus, when both types of successful trajectories appear in the same rollout group, the reward refinement is directly transferred to the relative learning weights used by the policy optimizer.

Group-level cases. The behavior of BoundarySA can be summarized by the composition of a rollout group.

- If no trajectory satisfies
$$r_{\mathrm{ans}}(\tau_k, x_i) r^0_{\mathrm{bound}}(\tau_k, x_i) = 1,$$
then
$$\mathbf{R}_{\mathrm{B}} = \mathbf{R}_{\mathrm{orig}},$$
and the group-relative advantages are identical.
- If every trajectory receives the same boundary increment, then BoundarySA applies a uniform reward shift. Such a shift does not alter the group mean-centered rewards, the reward standard deviation, or the relative advantages.
- If the group contains both boundary-consistent and boundary-inconsistent successful trajectories, BoundarySA introduces the strict preference in Equation (5), whereas the original objective assigns them equal relative preference.
- If a boundary-consistent successful trajectory and an incorrect trajectory coexist, BoundarySA preserves the original correctness ordering and enlarges their raw-reward margin by λ.

Overall Consequence

Under capability drift, BoundarySA either reproduces the trajectory preferences of the original objective or refines them with an additional preference for successful trajectories consistent with the initial capability profile. For comparisons with equal format reward, it is no less discriminative than the baseline at the raw-reward ordering level and, when distinguishable successful trajectories coexist, at the group-relative preference level.

Scope of the analysis. The above results establish pointwise reward preservation, preservation of the correct-versus-incorrect ordering, and additional discrimination among successful trajectories. They do not imply that the complete neural-network update vector is always aligned with the answer-only gradient, nor do they provide an unconditional guarantee that the final test accuracy is no lower than that of the baseline. Those outcomes additionally depend on rollout coverage, shared-parameter interactions, optimization dynamics, and generalization.

C Additional Experiments

C.0.1 Effect of Reinforcement Learning

To isolate the contribution of reinforcement learning, we compare the base model, the model after supervised fine-tuning (SFT), and the final model after boundary-aware reinforcement learning (RL) on MMHops, E-VQA, and InfoSeek. Table 10 shows a consistent improvement across the three training stages on all benchmarks.

Dataset	BaseLoop	Stage 1 SFT	Stage 2 (Final) RL
MMHops (Overall)	31.65	34.87	47.91
E-VQA (All)	32.26	33.01	45.62
InfoSeek (All)	34.82	35.14	42.54

Table 10: Performance (%) across training stages. MMHops Overall combines bridging and comparison questions; E-VQA and InfoSeek use their All metrics. Stage 1 denotes cold-start supervised fine-tuning (SFT), and Stage 2 denotes the final BoundarySA model after reinforcement learning (RL). (BaseLoop: tool-enabled agentic VLM before SFT and RL).

C.0.2 Tool-Use Efficiency Across Datasets

We further compare the average number of tool calls per evaluation instance across MMHops, E-VQA, and InfoSeek. Relative to MMhops-R1, BoundarySA uses fewer tool calls on all three benchmarks. About probing cost, the additional capability-probing stage introduces only limited computational overhead: in our experiments, its runtime accounts for less than 5% of the combined SFT and RL training time. Moreover, probing does not require initializing the VERL training framework or incurring the repeated latency associated with multi-turn tool interaction. In practice, the cumulative cost of tool calls—particularly the latency from external tool execution and waiting—substantially exceeds the cost of capability probing. Therefore, we primarily focus on tool-call efficiency when assessing the practical efficiency of BoundarySA.

Dataset	MMhops-R1	BoundarySA	Reduction
MMHops	3.9	3.1	20.5%
E-VQA	1.8	1.5	16.7%
InfoSeek	1.7	1.5	11.8%

Table 11: Average number of tool calls per evaluation instance. Reduction reports the relative decrease of BoundarySA compared with MMhops-R1.

C.0.3 Hyperparameter Analysis

We analyze the effects of the capability-probing sample size m, the direct-answering threshold ρ, and the reward weights λ and β on answer accuracy. Increasing m from 4 to 12 yields a substantial improvement, while the additional gain from $m = 16$ to $m = 24$ is only 0.1 points. The threshold ρ controls when the direct-answering score suppresses tool-use paths: under the gating function in the main paper, suppression begins only when $p_d > \rho$. Among the tested settings, $\rho = 0.75$, $\lambda = 0.2$, and $\beta = 0.5$ achieve the highest Overall score in their respective sweeps. Larger tested values of λ reduce performance, whereas the tested values of β yield smaller, non-monotonic differences. In all ablations, "Overall" denotes performance on the full MMHops set, combining bridging and comparison questions.

(a) Sample Size m

Value	Overall
4	41.6
8	46.7
12	47.9
16	47.9
24	48.0

(b) Threshold ρ

Value	Overall
0.50	42.7
0.65	46.1
0.75	47.9
0.85	47.6

(c) Weight λ

Value	Overall
0.1	46.4
0.2	47.9
0.5	45.3
1.0	40.8

(d) Weight β

Value	Overall
0.2	46.6
0.5	47.9
0.7	46.8
1.0	47.1

Table 12: Effects of (a) the capability-probing sample size m, (b) the direct-answering threshold ρ, (c) the boundary-reward weight λ, and (d) the format-reward weight β.

D DATASETS AND BENCHMARKS

We evaluate BoundarySA on three knowledge-intensive visual question answering benchmarks: MMHops (Zhang et al., 2025), Encyclopedic-VQA (E-VQA) (Mensink et al., 2023), and InfoSeek (Chen et al., 2023). These benchmarks cover multimodal multi-hop reasoning, fine-grained encyclopedic entity understanding, and visual information-seeking questions, respectively.

MMHops. MMHops is designed for multimodal multi-hop reasoning. Unlike conventional KB-VQA datasets where many questions can be answered after identifying a single entity and retrieving one supporting fact, MMHops requires models to construct longer reasoning chains over visual inputs and external knowledge. The benchmark contains bridging and comparison questions. Bridging questions require the model to move from a visual entity to intermediate knowledge and then to the final answer, while comparison questions require reasoning over multiple entities or attributes before producing the answer.

E-VQA. Encyclopedic-VQA is an entity-centric KBVQA benchmark that focuses on detailed properties of fine-grained visual entities. Questions in E-VQA often cannot be answered from the image alone because the required information is encyclopedic knowledge such as attributes, facts, categories, locations, or other entity-related properties. The benchmark is useful for evaluating whether a model can connect visual grounding with external factual knowledge.

InfoSeek. InfoSeek is a visual information-seeking benchmark. Its questions are grounded in visual entities but are designed so that common visual perception or commonsense knowledge alone is insufficient. To answer these questions, a model usually needs to recognize the entity in the image and then acquire fine-grained external knowledge about that entity. InfoSeek is especially useful for testing whether a method has learned a robust knowledge-seeking strategy rather than memorizing frequent entity-question associations.

E CASE STUDIES

Figure 5 presents two representative rollouts illustrating how BoundarySA can align tool use with the model's instance-specific capability boundary. In both illustrated examples, standard RL reaches the correct answer through image search followed by text search, whereas BoundarySA reaches the same answer with one tool call by using a capability identified during probing.

Leveraging pre-existing entity-recognition capability. In the first example, the question asks for the county town of the historic county containing the village shown in the image. Capability probing yields $p_e = 1.0$, $p_a = 0.0$, and $p_d = 0.0$, indicating that the model identifies the question-relevant entity as Reculver but cannot answer from its internal knowledge alone. The text-only path consequently has the highest boundary score. The illustrated standard-RL rollout invokes image search and then text search, whereas BoundarySA uses the recognized entity directly in a text search and answers Maidstone. Thus, for this example, BoundarySA reaches the same answer without the image-search call.

Leveraging pre-existing question-answering capability. In the second example, the question asks when the brand of the depicted vehicle was founded. The probing results, $p_e = 0.5$, $p_a = 1.0$, and $p_d = 0.4583$, indicate that the model is uncertain about the visual entity but can answer once the correct entity is provided. Under these scores, direct answering and the image-only path are tied as the highest-scoring paths; the illustrated BoundarySA rollout selects image-only. The standard-RL rollout uses image search followed by text search, whereas BoundarySA uses image search to identify the vehicle and then answers 1967 using its internal knowledge. This rollout therefore reaches the correct answer without the additional text-search call.

Together, these examples illustrate that BoundarySA is not optimized solely to minimize tool calls. Instead, its boundary reward can favor different paths according to the capabilities identified for an instance.

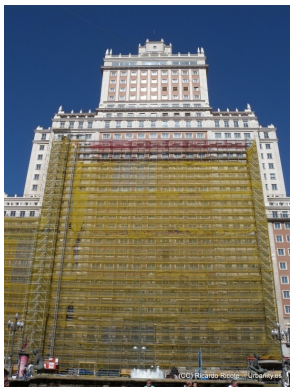


Figure 5: Case studies comparing standard RL and BoundarySA. In the first illustrated rollout, BoundarySA omits image search after the model identifies the visual entity during probing. In the second, it omits text search after image search identifies the entity.

E.1 Bad Case Analysis

Figure 6 illustrates a limitation of the four-category path reward. The boundary estimate assigns the highest score to the image+text route, but both the image+text and image+text+text rollouts receive the same path category despite differing in their number of tool calls. Hence, path-category supervision alone does not distinguish a shorter successful rollout from a longer successful rollout that invokes the same tool types.

Bad Case: Within-Path Length Ambiguity



Question. What is the main venue for classical music concerts in the city or region where this building is located?
Answer. Auditorio Nacional de Música.

Trajectory	Calls	Tools
Short correct	2	I → T
Longer correct	3	I → T → T

Path	Dir.	Img.	Txt.	Img.+Txt.
Score	0.076	0.007	0.076	**0.840**

Diagnosis. The boundary score favors image+text, but it does not distinguish I→T from I→T→T. Both trajectories are therefore assigned to the same path category despite their different lengths.

Figure 6: Bad-case analysis on a training example from MMHops. The example illustrates that the path-category reward does not distinguish successful rollouts with different numbers of calls when they use the same tool types.