V-CUES: IMPROVING MULTIMODAL SEARCH AGENTS BY CENTERING VISUAL-CUE UTILIZATION VIA EXPERIENCE AND SKILLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Search Agents (MSAs) have emerged as a promising paradigm for solving real-world tasks by combining visual understanding with external tools. While recent works have improved the mechanics of tool use and information retrieval, agents still struggle to effectively utilize visual cues during task execution. As a result, agents remain susceptible to Visual-Cue Omission and Visual-Cue Misuse, which can lead to irrelevant retrieval and task failure. To address these limitations, we propose V-CUES, a framework that bootstraps reusable knowledge centered on visual-cue utilization. V-CUES extracts recurring patterns from historical trajectories to capture how visual cues should be utilized under different input contexts, while dynamically refining them through case-level experiences accumulated from new interactions. During execution, V-CUES contextualizes accumulated knowledge for each newly encountered task, providing task-specific supervision on how to incorporate visual cues throughout the search process. Evaluation across four knowledge-intensive benchmarks and three basic models shows that V-CUES consistently improves performance over the previous state-of-the-art methods, demonstrating the effectiveness of explicitly centering visual cues for multimodal search agents.

1 INTRODUCTION

Recent advances in Multimodal Large Language Models (MLLMs) have improved the ability to solve a wide range of visual reasoning tasks (OpenAI et al., 2024; Team et al., 2024). However, many real-world visual tasks require up-to-date or domain-specific knowledge that may not be available in models' static parametric knowledge. Multimodal Search Agents (MSAs) address this limitation by combining visual reasoning with external tools (Tao et al., 2026; Fu et al., 2025). They shift the solution paradigm from static generation toward dynamic interaction in open environments.

In general, MSAs iteratively obtain the missing information needed for the task and analyze it together with the input context. Recent works have followed two main directions. The first optimizes policies for multi-step tool interaction through reinforcement learning, enabling on-demand search (Wu et al., 2026a), long-horizon tool use (Lai et al., 2026), heterogeneous tool invocation (Lu et al., 2026), and parallel retrieval (Li et al., 2026). The second abstracts trajectories into reusable knowledge, where recent methods (Jiang et al., 2026; Xiong et al., 2026) extract workflows, tool templates, and execution experience to enable more reliable and flexible tool use. Despite their different techniques, these approaches primarily focus on how agents obtain external information through tools.

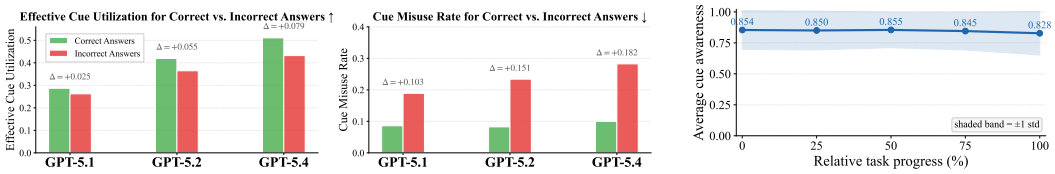
However, better tool use alone is not sufficient for effective multimodal search agents. Search agents also need to effectively understand and utilize visual cues throughout task execution. Existing works primarily strengthen tool use and overlook how visual cues should be incorporated and what roles they should play during task execution. Consequently, agents still exhibit two types of failures illustrated in Figure 1: (a) *Visual-Cue Omission*, where they omit crucial visual cues, leading to irrelevant retrieval. (b) *Visual-Cue Misuse*, where they assign inappropriate roles to visual cues, leading to unsupported conclusions. These failures indicate that agents struggle to utilize visual cues appropriately. Recent studies similarly show that multimodal models often bypass visual cues in favor of language priors or simple text matching (Guan et al., 2024; Huang et al., 2026).

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: **Visual-cue utilization in multimodal search agents.** (a) *Visual-Cue Omission*: Omitting the event-branding cues leads agents to retrieve a different match involving the same players. (b) *Visual-Cue Misuse*: Identity cues are incorrectly treated as conclusive evidence of the outcome rather than as search anchors. (c) *Successful Execution*: Jointly using the player, opponent, and event cues enables agents to identify the correct match and retrieve evidence for the correct winner.

To further diagnose this problem, we define Effective Cue Utilization (EU) as the proportion of task-relevant visual cues that are appropriately used during information obtaining, and Cue Misuse Rate (CM) as the proportion of utilized visual cues that are assigned inappropriate roles (Details definitions are in Appendix A.1.4). As shown in Figure 2a, cases with correct answers consistently exhibit higher EU, and cases with incorrect answers exhibit higher CM. This result shows that effective visual-cue utilization is closely associated with task success. We further examine whether these failures arise because agents fail to recognize relevant visual cues. As shown in Figure 2b, when explicitly queried during task execution, agents can consistently identify most task-relevant visual cues. This shows that the main bottleneck is not visual-cue awareness. Instead, agents can recognize relevant visual cues but fail to utilize them appropriately during task execution. Therefore, it reveals a clear gap between visual-cue awareness and utilization for multimodal search agents.



(a) **Impact of visual-cue utilization.** (b) **Visual-Cue awareness.**

Figure 2: **Statistical analysis for visual-cue.** (a) *Visual-cue utilization*: Cases with correct answers exhibit higher Effective Cue Utilization (EU), while cases with incorrect answers exhibit higher Cue Misuse Rates (CM). (b) *Visual-cue awareness*: Agents can identify most task-relevant visual cues when explicitly queried during execution, despite often failing to utilize them effectively.

To address this challenge, we propose **V-CUES** (**V**isual-**C**ue **U**tilization via **E**xperience and **S**kills), a framework that bootstraps reusable knowledge centered on visual-cue utilization to supervise multimodal search agents. V-CUES learns recurring patterns from successful trajectories, capturing which visual cues are relevant and what roles they play under different input contexts. It also accumulates case-level experiences from the target task distribution, to capture specific visual conditions that may not be covered by general patterns. For each task, V-CUES adapts the accumulated skills and experience to the new input image-text pair, providing task-specific supervision for subsequent execution. New interactions are further used to expand case-level experiences and refine recurring patterns, enabling continuous adaptation without updating model parameters.

Extensive evaluations across four knowledge-intensive benchmarks show that V-CUES generally outperforms the previous state-of-the-art methods. And V-CUES improves the overall **Avg@2 by 1.73 to 6.08 points** over the tool-only methods across diverse models. Furthermore, analysis shows that these improvements are accompanied by more effective visual-cue utilization during execution, supporting the effectiveness of explicitly centering visual cues in multimodal search agents.

In summary, our main contributions are as follows:

- We empirically identify visual-cue utilization as a critical bottleneck in multimodal search agents. Through diagnostic analyses, we reveal that the omission and misuse of task-relevant visual cues during execution are important causes of task failure.
- We propose V-CUES, a framework leveraging reusable knowledge centered on visual-cue utilization to supervise multimodal search agents. By adapting skills and experiences to the specific task, V-CUES helps ground execution in effective visual-cue utilization.
- Extensive experiments demonstrate that V-CUES substantially improves overall performance across four benchmarks under three backbone models. Our method generally outperforms strong baselines, establishing a more visually-grounded execution paradigm.

2 Related Work

2.1 Multimodal Search Agents

To solve real-world visual tasks, models are required to retrieve evidence rather than relying solely on static parametric knowledge (Jiang et al., 2024; Mei et al., 2025; Compagnoni et al., 2026). Consequently, by integrating with image processing and search tools, MLLMs are transformed into agentic systems for dynamic interaction in open environments (Chen et al., 2026; Geng et al., 2026). Recent studies employ reinforcement learning to scale interaction depth (Jiao et al., 2026; Lai et al., 2026), induce adaptive tool use (Wu et al., 2026a;b; Hong et al., 2026), enforce structured tool use (Narayan et al., 2025; Lu et al., 2026), and enable parallel retrieval (Li et al., 2026).

Concurrently, to sustain extended search interactions, other frameworks explicitly decouple retrieval architectures (Yu et al., 2026), optimize system scalability and context efficiency (Chu et al., 2026; Du et al., 2026), and synthesize multi-hop training data (Zhang et al., 2026; PENG et al., 2026). However, these methods focus on obtaining information, leaving a critical gap: they do not explicitly optimize how visual cues guide information acquisition and evidence integration. Even with refined execution mechanics, if agents omit task-relevant visual cues or assign them inappropriate roles, they will retrieve irrelevant information and fail the task.

2.2 Experience and Skill Bootstrapping

Beyond parametric training, distilling historical trajectories into reusable knowledge has emerged as an effective paradigm for bootstrapping agent capabilities (Hu et al., 2025; Hatalis et al., 2025). Recent methods typically abstract raw execution traces into high-level procedural skills (Wang et al., 2024; Ni et al., 2026) and context-specific experiences (Tang et al., 2025; Cai et al., 2025; Cao et al., 2026). This bootstrapping paradigm has naturally extended to multi-modal agents. Recent frameworks extract visually conditioned tool workflows alongside tactical execution experiences (Jiang et al., 2026), and consolidate multi-modal tool templates through prioritized sampling and semantic clustering (Xiong et al., 2026).

However, they primarily construct reusable knowledge around execution mechanics by mapping visual observations to specific tool actions. While they effectively capture how to invoke tools based on visual states, they do not explicitly represent how task-relevant visual cues guide information acquisition and evidence integration. Consequently, agents remain susceptible to omitting or misusing visual cues during execution. Our method addresses these limitations by bootstrapping reusable experience and skills centered on visual-cue utilization, capturing and contextualizing the task-specific roles of visual cues to guide multimodal search agents.

3 Method

3.1 problem formulation

Given an image set V and a question Q, we represent a multimodal search task as $x = (V, Q)$. To solve the task, agents identify what remains unresolved, use tools to acquire the missing information, and assess how the observations should contribute to the answer. This process produces an execution

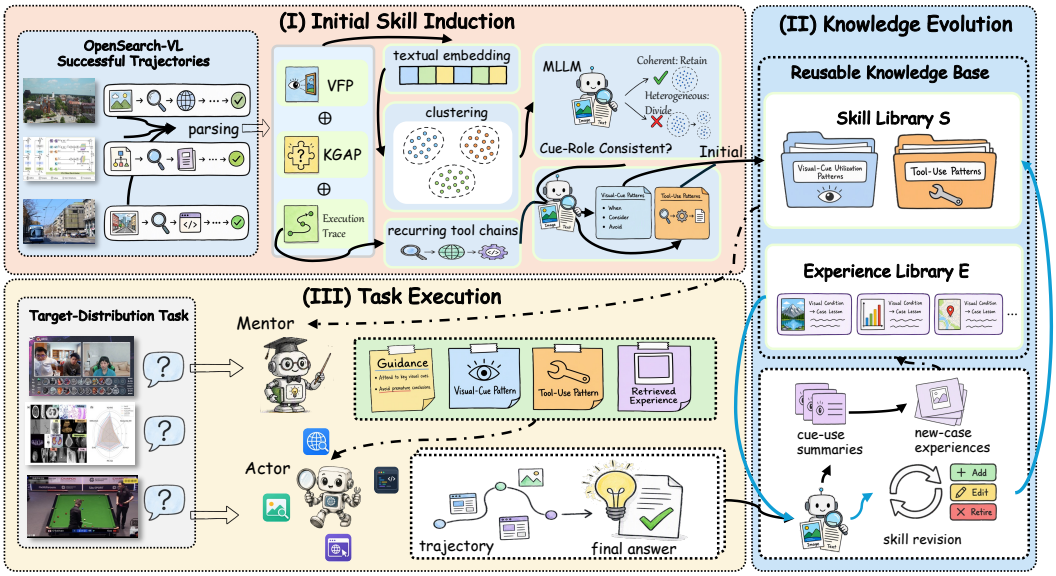


Figure 3: **Overview of our proposed V-CUES.** *(I) Initial Skill Induction*: V-CUES induces reusable skills from successful multi-domain expert trajectories, with visual-cue utilization patterns validated in original input contexts. *(II) Knowledge Evolution*: Interactions from the target task distribution are distilled into case-level experiences and used to refine recurring skills. *(III) Task Execution*: For each newly encountered task, the Mentor selects relevant skills and experience to supervise the Actor to obtain information and integrate evidence in subsequent execution.

trajectory τ_x and yields an answer $\hat{y}_x$. Let $\mathcal{C}_x$ denote the set of task-relevant visual cues in V. During execution, these cues help specify what information remains to be obtained, ground tool actions as search anchors, constraints, or disambiguation signals, and relate them back to the image–question context during evidence integration. Effective cue utilization therefore requires the agent to draw on relevant cues when needed and use them in roles appropriate to the current image–question context.

To achieve this, we introduce a reusable knowledge base $\mathcal{KB}$ constructed for the target task distribution $\mathcal{D}_b$. Conditioned on this $\mathcal{KB}$, the trajectory and answer generated by the frozen agent follow the conditional distribution $p_\theta(\tau_x, \hat{y}_x \mid x, \mathcal{KB})$. Let $y^\star$ denote the reference answer and $J(\hat{y}_x, y^\star)$ be the correctness evaluation metric. Our objective is to optimize $\mathcal{KB}$ to maximize the agent's expected answer correctness: $\max_{\mathcal{KB}} \mathbb{E}_{(x,y^\star)\sim\mathcal{D}_b}\left[\mathbb{E}_{(\tau_x,\hat{y}_x)\sim p_\theta(\cdot|x,\mathcal{KB})}\left[J(\hat{y}_x, y^\star)\right]\right]$.

3.2 Framework Overview

As illustrated in Figure 3, V-CUES bootstraps a reusable knowledge base through three phases. First, it abstracts successful multi-domain expert trajectories into a skill library $\mathcal{S}_0 = (\mathcal{S}^{\mathcal{V}}, \mathcal{S}^{\mathcal{T}})$. The visual-cue utilization patterns $\mathcal{S}^{\mathcal{V}}$ capture relevant visual cues, their task roles, and common misinterpretations across image–question contexts, while the tool-use patterns $\mathcal{S}^{\mathcal{T}}$ summarize recurring patterns of tool use during execution. Second, V-CUES distills new target-distribution interactions into an experience library $\mathcal{E}$ containing case-level lessons tied to concrete visual conditions. Recurring lessons across these experiences are periodically consolidated to revise the skill library, yielding the evolving knowledge base $\mathcal{KB} = (\mathcal{S}, \mathcal{E})$. Finally, for a current task, we employ two specialized MLLMs: the *Mentor* selects relevant patterns in the skill library and visually applicable experiences in the experience library based on the image–question context, and provides them with guidance for the *Actor* to obtain information and integrate evidence.

3.3 Initial Skill Induction

V-CUES initializes the skill library $\mathcal{S}_0$ from successful expert trajectories in the multi-domain OpenSearch-VL dataset (Chen et al., 2026). These trajectories connect visual cues and execution with correct answers, providing reliable positive examples for knowledge initialization.

Trajectory parsing and clustering. From each source trajectory, we extract three textual signals. The Visual First-Pass (VFP) records visual cues, while the Knowledge Gap (KGAP) describes information that remains unresolved. We complement them with the execution trace. Their concatenation is embedded and clustered to discover candidate regularities in visual-cue utilization. In parallel, we extract recurring tool-call chains from the execution traces to serve as candidate tool-use patterns.

Multimodal context validation. Since clustering relies only on textual trajectory representations, it may group samples with similar descriptions or tool behaviors even when their visual cues play different roles in the corresponding image–question contexts. To resolve this, V-CUES validates each candidate cluster within its original multimodal context. Concretely, a knowledge MLLM examines representative samples, each presented with its image, question, VFP, and KGAP, and assesses whether the visual cues serve consistent task roles. Coherent clusters are retained, whereas heterogeneous clusters are further divided according to the identified subgroups.

Pattern induction and Skill construction. For each cluster, an MLLM examines its representative image–question cases and abstracts their shared cue–role relationship into a visual-cue utilization pattern with `When`, `Consider`, and `Avoid` fields, which respectively describe its applicable context, recommended cue use, and common misuse. Tool-chain candidates are verbalized as tool-use patterns that summarize recurring execution regularities. After structural filtering, redundancy consolidation, and language review, the two types of patterns form the initial skill library $\mathcal{S}_0$.

3.4 Knowledge Evolution

The initial $\mathcal{S}_0$ captures patterns from successful multi-domain trajectories, but may not fully cover the visual conditions encountered in a target task distribution. V-CUES therefore accumulates case-level experiences from target-distribution interactions and uses recurring lessons to revise $\mathcal{S}$. Prompts are provided in Appendix A.5.1.

Case experience distillation. For each target task, V-CUES generates multiple search rollouts using the current $\mathcal{KB}$. After each rollout, an outcome evaluator assesses the final answer against the reference answer and returns a correctness verdict. Let $\mathcal{R} = \{\mathcal{R}^{(r)}\}_{r=1}^{R}$ denote the rollout records, where $\mathcal{R}^{(r)} = (P^{(r)}, \tau^{(r)}, \hat{y}^{(r)}, v^{(r)})$ contains the complete *Actor* prompt, trajectory, final answer, and correctness verdict of the r-th rollout. A knowledge MLLM $\mathcal{M}_{\mathrm{K}}$ first derives a visual-cue utilization summary $\mathcal{H}_{\mathrm{cue}}$ which records the selection, interpretation, and operationalization of visual cues throughout the trajectory from each rollout and then compares these summaries together with the original image to distill new case experiences:

$$\mathcal{H}_{\mathrm{cue}} = \left\{ \mathcal{M}_{\mathrm{K}} \left(Q, y^{\star}, \mathcal{R}^{(r)} \right) \right\}_{r=1}^{R}, \qquad \Delta\mathcal{E} = \mathcal{M}_{\mathrm{K}} \left(V, Q, y^{\star}, \mathcal{H}_{\mathrm{cue}} \right). \tag{1}$$

Each experience is represented as $e = (c_e, l_e)$, where c_e describes the visual condition under which the experience applies and l_e captures a transferable lesson about how visual cues should be used in that condition and what common misinterpretations to avoid in that condition.

Experience accumulation and Skill revision. The distilled experiences are first added to $\mathcal{E}$, yielding $\mathcal{E}' = \mathcal{E} \cup \Delta\mathcal{E}$. At each revision step, V-CUES selects a bounded subset $\widetilde{\mathcal{E}}$ from $\mathcal{E}'$, prioritizing experiences frequently selected by the *Mentor* and using recency to break ties. The knowledge MLLM $\mathcal{M}_{\mathrm{K}}$ then consolidates recurring lessons in $\widetilde{\mathcal{E}}$ to revise the current $\mathcal{S}$:

$$\mathcal{S}' = \mathrm{Revise}_{\mathcal{M}_{\mathrm{K}}} \left(\mathcal{S}, \widetilde{\mathcal{E}} \right). \tag{2}$$

Lessons concerning visual conditions, cue roles and cue misuse are incorporated into visual-cue utilization patterns, whereas execution-side lessons are incorporated into tool-use patterns. A new pattern must be supported by experiences from multiple distinct tasks; the revision also refines or retires existing patterns. The resulting knowledge base $\mathcal{KB}' = (\mathcal{S}', \mathcal{E}')$ is used for subsequent tasks.

3.5 Task Execution

Given a task $x = (V, Q)$ and $\mathcal{KB} = (\mathcal{S}, \mathcal{E})$, V-CUES adapts $\mathcal{KB}$ according to the image–question context to support task execution. The same execution procedure is used to generate target-

Table 1: **Comparison with previous state-of-the-art methods on four knowledge-intensive benchmarks**. The metrics evaluated are Average@2 and Pass@2 (%). The "Overall" denotes the arithmetic mean across the four benchmarks. The best results are highlighted in bold.

Base Model	Method	MMSearch-Plus		VDR-Bench		LiveVQA		FVQA		Overall	
		Avg@2	Pass@2	Avg@2	Pass@2	Avg@2	Pass@2	Avg@2	Pass@2	Avg@2	Pass@2
GPT-5.2	Direct Answer	2.84	4.74	7.50	10.50	50.75	58.50	34.00	40.00	23.77	28.43
	w/ Tools	23.70	30.81	24.00	32.50	62.75	72.00	71.25	78.00	45.42	53.33
	XSKILL	25.36	32.70	25.75	33.00	62.50	72.50	71.00	76.50	46.15	53.68
	V-CUES (Ours)	**31.04**	**37.91**	**27.75**	**38.50**	**67.75**	**73.50**	**75.25**	**80.00**	**50.45**	**57.48**
GPT-5.4	Direct Answer	6.16	8.53	9.00	12.00	55.25	62.00	47.50	50.50	29.48	33.26
	w/ Tools	28.91	38.39	23.00	31.50	65.25	73.50	68.50	74.50	46.41	54.47
	XSKILL	30.57	39.34	24.50	32.00	67.00	75.50	66.50	72.50	47.14	54.84
	V-CUES (Ours)	**37.20**	**47.87**	**27.50**	**36.00**	**71.50**	**77.00**	**73.75**	**77.50**	**52.49**	**59.59**
Gemini-3-Flash	Direct Answer	22.04	25.59	11.00	13.50	59.25	68.50	58.50	62.50	37.70	42.52
	w/ Tools	49.29	54.03	28.00	34.50	**73.75**	**83.50**	73.50	79.00	56.13	62.76
	XSKILL	48.58	56.87	28.50	34.00	72.75	78.50	75.25	**81.00**	56.27	62.59
	V-CUES (Ours)	**51.18**	**59.72**	**29.75**	**36.00**	73.25	82.00	**77.25**	**81.00**	**57.86**	**64.68**

distribution rollouts during knowledge evolution and to answer test questions. Prompts are provided in Appendix A.5.2.

Experience retrieval. Each experience $e = (c_e, l_e)$ consists of a visual condition c_e and a case-level lesson l_e. We maintain separate embedding indices for the two fields. Given the current question Q with embedding $\mathbf{v}_Q$, V-CUES independently retrieves the top-K experiences from the condition and lesson indices:

$$\mathcal{R}_Q^c = \mathrm{TopK}_{e\in\mathcal{E}} \cos(\mathbf{v}_Q, \mathbf{v}_{c_e}), \qquad \mathcal{R}_Q^l = \mathrm{TopK}_{e\in\mathcal{E}} \cos(\mathbf{v}_Q, \mathbf{v}_{l_e}). \tag{3}$$

The two result lists are merged, retaining the higher similarity score for experiences retrieved from both indices. The merged results are then reranked, and the top-K entries form the candidate set $\mathcal{E}_Q^{\mathrm{cand}}$. This retrieval accounts for both the applicability conditions and the lessons of prior experiences; the *Mentor* subsequently assesses each candidate using the current image.

Task adaptation. The *Mentor* jointly considers the current image and question, $\mathcal{S}$ and $\mathcal{E}_Q^{\mathrm{cand}}$. It selects visual-cue utilization patterns and tool-use patterns relevant to the current task, and filters the candidate experiences by verifying whether their specific c_e are concretely grounded in the current image. Based on the selected knowledge, the *Mentor* produces *guidance*: a task-specific interpretation that relates reusable knowledge to the current case by highlighting task-relevant visual cues and their possible task roles. Formally,

$$\left(G_x, \mathcal{S}_x^{\mathcal{V}}, \mathcal{S}_x^{\mathcal{T}}, \mathcal{E}_x\right) = \mathrm{Mentor}\left(V, Q, \mathcal{S}, \mathcal{E}_Q^{\mathrm{cand}}\right), \tag{4}$$

where G_x denotes the guidance, and $\mathcal{S}_x^{\mathcal{V}}$, $\mathcal{S}_x^{\mathcal{T}}$, and $\mathcal{E}_x$ denote the selected visual-cue utilization patterns, tool-use patterns, and case experiences, respectively.

Actor execution. The selected knowledge and guidance are provided to the *Actor* as task-specific knowledge support. Visual-cue utilization patterns and case experiences inform how visual cues may be used under the current image–question context, while tool-use patterns provide complementary execution knowledge. The *Actor* then combines this knowledge to obtain information, integrate evidence, and generate answer:

$$(\tau_x, \hat{y}_x) = \mathrm{Actor}\left(V, Q; G_x, \mathcal{S}_x^{\mathcal{V}}, \mathcal{S}_x^{\mathcal{T}}, \mathcal{E}_x\right). \tag{5}$$

The *Mentor–Actor* design separates knowledge adaptation from task execution, providing task-specific visual cue knowledge for subsequent information acquisition and evidence integration.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Benchmarks. We evaluate our method on four knowledge-intensive VQA benchmarks: MMSearch-Plus (Tao et al., 2026), VDR-Bench (Huang et al., 2026), LiveVQA (Fu et al., 2025),

Table 2: Ablation study of performance (%) on MMSearch-Plus. **-Exp**: removes the Experience Library; **-Skill**: removes the Skill Library; **-S&E**: removes knowledge base (*Mentor-only*); and **-Mentor**: removes the *Mentor* (knowledge-only).

Model	Metric	Settings				
		Full	**-Exp**	**-Skill**	**-S&E**	**-Mentor**
GPT-5.2	Avg@2	**31.0**	30.6	28.2	24.2	22.8
	Pass@2	**37.9**	37.0	34.1	31.3	30.3
GPT-5.4	Avg@2	**37.2**	37.0	34.1	31.5	35.3
	Pass@2	**47.9**	46.5	42.7	39.3	42.7

Table 3: Performance (%) of pattern-only variants on MMSearch-Plus. To isolate their impact, each variant exclusively utilizes either Tool-Use Patterns ($\mathcal{S}^{\mathcal{T}}$) or Visual-Cue Utilization Patterns ($\mathcal{S}^{\mathcal{V}}$), omitting experiences.

Model	Pattern	Avg@2	Pass@2
GPT-5.2	$\mathcal{S}^{\mathcal{T}}$	28.0	34.6
	$\mathcal{S}^{\mathcal{V}}$	**28.7**	**36.0**
GPT-5.4	$\mathcal{S}^{\mathcal{T}}$	32.9	42.2
	$\mathcal{S}^{\mathcal{V}}$	**34.8**	**44.1**

and FVQA (Wang et al., 2018). Together, they cover visual understanding, web retrieval, and multi-hop reasoning, requiring agents to accurately interpret visual content and use it to acquire external evidence for answering the question. Following the operation of XSKILL (Jiang et al., 2026), we randomly sample 100 tasks from each benchmark as the target-distribution adaptation set for experience accumulation and skill revision, and use the remaining tasks for evaluation.

Tool Sets and Baselines Following the recent tool setting (Jiang et al., 2026), we empower our agent with four types of tools to handle tasks as follows: (1) **code interpreter**: runs stateful Python scripts to facilitate programmatic image manipulation, mathematical computations, and complex data analysis; (2) **web search**: issues text queries to external search engines to harvest online knowledge snippets and URLs; (3) **image search**: discovers related visual entities and their web context by executing either text queries or reverse image retrieval; and (4) **visit**: accesses target URLs to parse and extract the main textual content from designated webpages. We compare our method against three baselines: *Direct Answer*, which answers questions without tool use; *w/ Tools*, a vanilla multimodal search agent equipped with the tool sets; and *XSKILL* (Jiang et al., 2026), a method that improves multimodal agents through experience and skills.

Evaluation metrics. We evaluate end-task performance using the success rate under two rollout-level aggregation protocols: Average@2 and Pass@2. Let $s_i^{(r)} \in \{0, 1\}$ denote whether the r-th rollout for task i is judged successful. Average@2 computes the mean success rate over two rollouts, whereas Pass@2 considers a task successful if either rollout succeeds.

Implementation details. We conduct evaluation on four benchmarks using three base models: GPT-5.2 (OpenAI, 2025c), GPT-5.4 (OpenAI, 2026a), and Gemini-3-Flash (DeepMind, 2025). For each backbone, the *Mentor* and the *Actor* use the same model but different generation parameters. Specifically, the *Mentor* uses temperature $T = 0.2$ and top-$p = 0.9$, while the *Actor* uses temperature $T = 0.6$ and top-$p = 1.0$. Text embeddings for experience retrieval are computed by text-embedding-3-small (OpenAI, 2024). We set the experience retrieval depth to $K = 3$. During target- distribution adaptation, we use two rollouts per task and revise the skill library every 25 tasks.

4.2 Main Results

Table 1 presents the end-task performance across four benchmarks. Our method surpasses both *w/ Tools* and *XSKILL*, achieving overall absolute gains of 1.73 to 6.08 points in Avg@2 across different backbone models. These results demonstrate that leveraging reusable knowledge provides a substantial performance boost beyond mere tool access. Furthermore, the improvements are particularly evident on MMSearch-Plus and VDR-Bench, where successful answering heavily depends on grounding obtaining information in task-relevant visual cues. For instance, under the GPT-5.4 backbone, our method achieves an Avg@2 of 37.20% on MMSearch-Plus, outperforming *w/ Tools* and *XSKILL* by 8.29 and 6.63 points, respectively. These substantial accuracy boosts demonstrate that explicit guidance for visual-cue utilization is highly beneficial for multimodal search agents.

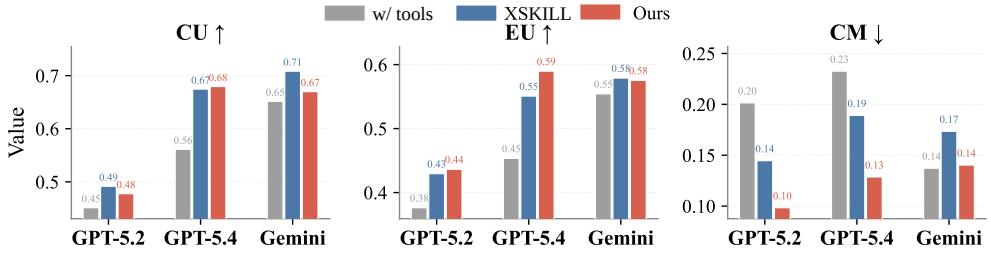
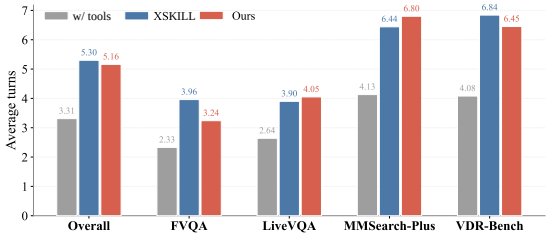


Figure 4: **Cue-level experimental results on MMSearch-Plus.** Comparison of three methods across different backbone models on Cue Utilization (CU), Effective Cue Utilization (EU), and Cue Misuse Rate (CM). Our method consistently achieves the highest EU and the lowest CM across all models, demonstrating that it effectively mitigates visual-cue omission and prevents cue misuse, ensuring accurate visual understanding.

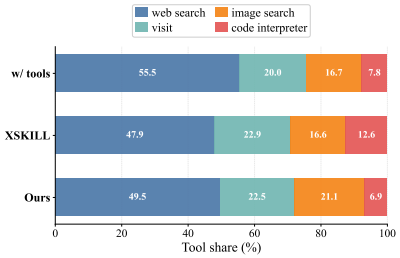
4.3 Component Analysis

We evaluate the individual components in Table 2. Removing the skill library causes a larger performance drop than removing the experience library. Under GPT-5.4, removing skills drops the Avg@2 to 34.1%, which is 2.9 points lower than the 37.0% when removing experiences. This shows that skills provide the primary performance boost, while case experiences provide additional case-level lessons for specific visual conditions. Furthermore, the *Mentor* is essential for applying this knowledge. Removing the *Mentor* decreases the Avg@2 to 22.8% under GPT-5.2 and 35.3% under GPT-5.4. This consistent decline shows that simply providing raw knowledge is insufficient; the system relies on the Mentor's task-specific adaptation and guidance to be fully effective.

Table 3 isolates the different pattern types in the skill library. The results show that visual-cue patterns consistently outperform tool-use patterns across both backbone models. This highlights that visual-cue guidance acts as a higher-level cognitive foundation than execution rules. By mitigating visual-cue omission and misuse, it ensures accurate visual grounding for subsequent execution, directly translating into larger performance gains.



(a) Average trajectory turns across benchmarks under GPT-5.4 for three different settings.



(b) Average Tool usage distribution across four benchmarks under GPT-5.4.

Figure 5: **Trajectory and tool use analysis.** (a) While both *XSKILL* and our method generate longer trajectories than w/ tools, our approach achieves higher accuracy with slightly fewer steps than *XSKILL*, indicating a more visually grounded and targeted execution process. (b) Compared to the baselines, our method allocates a larger share to image search and visit. This shift demonstrates that properly utilizing visual cues makes the subsequent information acquisition more effective.

4.4 Cue-Utilization Analysis

Figure 4 investigates the mechanism behind our performance gains by evaluating how the agent actually uses visual cues. Across all three base models, our method consistently achieves the highest Effective Cue Utilization and the lowest Cue Misuse Rate. Specifically, under GPT-5.4, our approach increases the EU score to 0.59 and reduces the CM to 0.13, clearly outperforming *XSKILL*, which scores 0.55 in EU and 0.19 in CM. Similar results are observed under GPT-5.2 and Gemini-3-Flash. These cue-level improvements indicates that our method successfully guides the agent to utilize visual cues effectively. By explicitly reducing visual-cue omission and misuse, our approach

ensures that the agent's execution is grounded in accurate visual understanding, which directly leads to the higher final accuracy seen in the main results.

4.5 Trajectory and Tool Use Analysis

We further analyze how V-CUES influences execution process of the agent. Figure 5a reports the average number of trajectory turns across benchmarks. Compared to the vanilla baseline, V-CUES consistently executes longer interaction trajectories, elevating the overall average from 3.31 to 5.16 turns. This indicates that reusable knowledge effectively encourages more exhaustive information acquisition and evidence integration.

Table 4: Tool usage distribution on MMSearch-Plus (%). Across models, V-CUES consistently increases the proportion of image search and visit compared to the baselines, demonstrating robust cross-model stability.

Model	Method	Tool Usage (%)			
		Web Search	Visit	Image Search	Code Interpreter
GPT-5.4	w/ tools	57.8	14.6	13.5	14.1
	XSKILL	47.0	18.2	14.9	19.9
	Ours	**50.5**	**19.6**	**18.8**	**11.2**
GPT-5.2	w/ tools	27.3	16.3	23.5	33.0
	XSKILL	28.8	14.7	18.7	37.8
	Ours	**38.4**	**20.0**	**25.8**	**15.8**

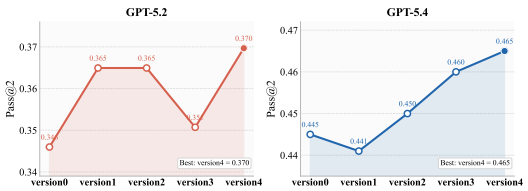


Figure 6: Performance generally improves across skill revisions, supporting adaptation to the target task distribution.

However, longer trajectories alone do not guarantee success. The *XSKILL* framework generates even longer trajectories, reaching an overall average of 5.30 turns, yet it falls short in final accuracy. This contrast supports that our advantage lies not merely in taking more steps, but in ensuring that each execution step is guided by accurate visual comprehension and information analysis.

To understand these qualitative differences, we examine the tool usage distribution. Figure 5b illustrates the overall tool usage across benchmarks. Compared with the *w/ tools* and *XSKILL*, our method shifts the global distribution toward image search. Table 4 further details this trend on the visually demanding MMSearch-Plus benchmark. Under GPT-5.4, our method allocates 18.8% of its calls to image search and 19.6% to the visit tool. This similar pattern under the GPT-5.2 backbone confirms that the behavioral shift is driven by our method. This demonstrates our core mechanism: properly utilizing visual cues provides a reliable basis for the agent, making subsequent information acquisition and evidence integration significantly more targeted and effective.

4.6 Evolution Analysis

To validate the continuous adaptation capability of our framework, we track the agent's performance across successive revisions of the skill library on MMSearch-Plus. Figure 6 illustrates the Pass@2 scores as the skill library evolves from its initial state (version0) to the final state (version4).

The results demonstrate a clear upward trajectory in accuracy as the reusable patterns mature. Under the GPT-5.4 backbone, the performance climbs from an initial Pass@2 of 44.5% to a peak of 46.5% at version4. A similar overall improvement emerges under GPT-5.2. Despite minor fluctuations during the intermediate revision steps, the model ultimately achieves its highest score of 37.0% at the final version, outperforming the initial 34.6%. This consistent growth validates our skill revision mechanism. It shows that by consolidating insights from ongoing interactions, our method generates increasingly effective visual-cue guidance that adapts to the target distribution.

5 Conclusion

In this paper, we identify a critical limitation in multimodal search agents: their inability to leverage visual cues during task execution. While existing works primarily optimize the mechanics of tool use and execution, our diagnostic analyses show that such optimizations offer limited benefits when agents omit crucial visual cues or assign them to inappropriate roles. To address this limitation, we introduce V-CUES, a framework that shifts the bootstrapping paradigm from execution mechanics to cognitive visual grounding. V-CUES provides agents with targeted cognitive support, enabling effective visual-cue utilization. Extensive experiments across four agent benchmarks and three base models demonstrate that our V-CUES consistently outperforms previous state-of-the-art methods.

AI USE STATEMENT

In this work, we used generative AI tools to assist in refining research hypotheses, providing feedback on the methodology and experimental design, implementing and debugging parts of the proposed method, supporting cue-level qualitative analysis, interpreting experimental results, and assisting with translation. We did not use generative AI tools to generate synthetic datasets or to clean or reformat the benchmark data. The research problem, method design, experimental protocol, analysis, and conclusions were determined and validated by the authors. We take full responsibility for the content of this paper, including all text, claims, code, and experimental artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We provide details of our method in the appendix. We additionally provide the project code to ensure reproducibility of the presented results. It can be accessed at the following repository: https://anonymous.4open.science/r/V-CUES-D852/

REFERENCES

Zhicheng Cai, Xinyuan Guo, Yu Pei, Jiangtao Feng, Jinsong Su, Jiangjie Chen, Ya-Qin Zhang, Wei-Ying Ma, Mingxuan Wang, and Hao Zhou. Flex: Continuous agent evolution via forward learning from experience, 2025. URL https://arxiv.org/abs/2511.06449.

Zouying Cao, Jiaji Deng, Li Yu, Weikang Zhou, Zhaoyang Liu, Bolin Ding, and Hai Zhao. Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16803–16822, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.829. URL https://aclanthology.org/2026.findings-acl.829/.

Shuang Chen, Kaituo Feng, Hangting Chen, Wenxuan Huang, Dasen Dai, Quanxin Shou, Yunlong Lin, Xiangyu Yue, Shenghua Gao, and Tianyu Pang. Opensearch-vl: An open recipe for frontier multimodal search agents, 2026. URL https://arxiv.org/abs/2605.05185.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, Shengchao Hu, Dongdong Kuang, Ming Liu, Bing Qin, and Xing Yu. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents, 2026. URL https://arxiv.org/abs/2602.14234.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11901–11911, June 2026.

Google DeepMind. Gemini 3 flash: frontier intelligence built for speed. https://blog.google/products-and-platforms/products/gemini/gemini-3-flash/, 2025.

Google DeepMind. Gemini 3.1 pro: A smarter model for your most complex tasks. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/, 2026.

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.

Mingyang Fu, Yuyang Peng, Dongping Chen, Zetong Zhou, Benlin Liu, Yao Wan, Zhou Zhao, Philip S Yu, and Ranjay Krishna. Seeking and updating with live visual knowledge. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38,

Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-2513. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6c7c9811d06b41b320b69abf37234f84-Paper-Datasets_and_Benchmarks_Track.pdf.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Kuan Li, Yida Zhao, Huifeng Yin, Yong Jiang, Pengjun Xie, Fei Huang, Huaxiu Yao, Yi R. Fung, and Jingren Zhou. Webwatcher: Breaking new frontiers of vision-language deep research agent. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 61240–61271, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/63ca1cca58101545e937302be7050098-Paper-Conference.pdf.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385, 2024. doi: 10.1109/CVPR52733.2024.01363.

Kostas Hatalis, Despina Christou, and Vyshnavi Kondapalli. Review of case-based reasoning for llm agents: Theoretical foundations, architectural components, and cognitive integration, 2025. URL https://arxiv.org/abs/2504.06943.

Jack Hong, Chenxiao Zhao, ChengLIn Zhu, Weiheng Lu, Guohai Xu, and XingYu. Deepeyesv2: Toward agentic multimodal model. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 114851–114872, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/badbef7d421640fb038399e9d7271b1c-Paper-Conference.pdf.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, Zhenrong Cheng, Xuanbo Fan, Jiaxin Guo, Xinlei Yu, Zhenhong Zhou, Zewen Hu, Jiahao Huo, Junhao Wang, Yuwei Niu, Yu Wang, Zhenfei Yin, Xiaobin Hu, Yue Liao, Qiankun Li, Kun Wang, Wangchunshu Zhou, Yixin Liu, Dawei Cheng, Qi Zhang, Tao Gui, Shirui Pan, Yan Zhang, Philip Torr, Zhicheng Dou, Ji-Rong Wen, Xuanjing Huang, Yu-Gang Jiang, and Shuicheng Yan. Memory in the age of ai agents, 2025. URL https://arxiv.org/abs/2512.13564.

Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, Zehui Chen, Yao Hu, Philip Torr, Feng Zhao, and Wanli Ouyang. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=2Cgy0P2v8s.

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. Mmsearch: Benchmarking the potential of large models as multi-modal search engines, 2024. URL https://arxiv.org/abs/2409.12959.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R. Fung. XSkill: Continual learning from experience and skills in multimodal agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=AjP1yvCyoG.

Zhengbo Jiao, Yiming Cheng, Yilei Jiang, Kaituo Feng, Rui Huang, Tianyi Jiang, Juanxi Tian, Jiapeng li, Qunzhong Wang, Tailai Chen, Qianshan Wei, Chuan Xiao, Shanyu Rong, Yangfu Li, Yanhan Zhou, Yunpu Ma, Yifan Zhang, and Xiangyu Yue. Searcheyes: Towards frontier multimodal deep search intelligence via search world simulation, 2026. URL https://arxiv.org/abs/2607.05943.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. In

C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 76722–76746, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7c40c5050bd029a3ea7ff8b01412f735-Paper-Conference.pdf.

Guankai Li, Jiabin Chen, Yi Xu, Xichen Zhang, and Yuan Lu. Hypereyes: Dual-grained efficiency-aware reinforcement learning for parallel multimodal search agents, 2026. URL https://arxiv.org/abs/2605.07177.

Meng Lu, Ran Xu, Yi Fang, Wenxuan Zhang, Yue Yu, Gaurav Srivastava, Yuchen Zhuang, Mohamed Elhoseiny, Charles Fleming, Carl Yang, Zhengzhong Tu, Yang Xie, Guanghua Xiao, Di Jin, Wenqi Shi, and Xuan Wang. Scaling agentic reinforcement learning for tool-integrated reasoning in vlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26518–26529, June 2026.

Lang Mei, Siyu Mo, Zhihan Yang, and Chong Chen. A survey of multimodal retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2504.08748.

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

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2skill: Distill trajectory-local lessons into transferable agent skills, 2026. URL https://arxiv.org/abs/2603.25158.

OpenAI. New embedding models and api updates. https://openai.com/index/new-embedding-models-and-api-updates/, 2024.

OpenAI. Introducing gpt-4.1. https://openai.com/index/gpt-4-1/, 2025a.

OpenAI. A smarter, more conversational chatgpt. https://openai.com/index/gpt-5-1/, 2025b.

OpenAI. Introducing gpt-5.2. https://openai.com/index/introducing-gpt-5-2/, 2025c.

OpenAI. Introducing gpt-5.4. https://openai.com/index/introducing-gpt-5-4/, 2026a.

OpenAI. Gpt-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, 2026b.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, et al. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

XIANGYU PENG, Can Qin, An Yan, Xinyi Yang, Ran Xu, Zeyuan Chen, and Chien-Sheng Wu. MTA-agent: An open recipe for multimodal deep search agents. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=6rtp2PaElk.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, Ge Zhang, Jiaheng Liu, Xingyao Wang, Sirui Hong, Chenglin Wu, Hao Cheng, Chi Wang, and Wangchunshu Zhou. Agent kb: Leveraging cross-domain experience for agentic problem solving, 2025. URL https://arxiv.org/abs/2507.06229.

Xijia Tao, Teng Yihua, Xinxing Su, Xinyu Fu, Jihao Wu, Chaofan Tao, Ziru Liu, Haoli Bai, Rui Liu, and Lingpeng Kong. Mmsearch-plus: Benchmarking provenance-aware search for multimodal browsing agents. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 70385–70422, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7291251c42879a93141d1340a96c6a22-Paper-Conference.pdf.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Fvqa: Fact-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40 (10):2413–2427, 2018. doi: 10.1109/TPAMI.2017.2754246.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory, 2024. URL https://arxiv.org/abs/2409.07429.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. MMSearch-r1: Incentivizing LMMs to search. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2456–2487, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.114. URL https://aclanthology.org/2026.acl-long.114/.

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, 2026b. URL https://arxiv.org/abs/2606.27974.

Feng Xiong, Zengbin Wang, Yong Wang, Xuecai Hu, Jinghan He, Liang Lin, Yuan Liu, and Xiangxiang Chu. Ace-skill: Bootstrapping multimodal agents with prioritized and clustered evolution, 2026. URL https://arxiv.org/abs/2605.08887.

Xiaohan Yu, Chao Feng, Lang Mei, and Chong Chen. M^{3}searcher: Modular multimodal information seeking agency with retrieval-oriented reasoning, 2026. URL https://arxiv.org/abs/2601.09278.

Ruiyang Zhang, Qianguo Sun, Chao Song, Yiyan Qi, and Zhedong Zheng. Vsearcher: Long-horizon multimodal search agent via reinforcement learning, 2026. URL https://arxiv.org/abs/2603.02795.

A Appendix

A.1 Method Details

A.1.1 Initial Skill Induction

We initialize the skill library from successful expert trajectories in the multi-domain OpenSearch-VL dataset because learning visual-cue utilization purely through trial-and-error is highly inefficient and prone to failures. Expert trajectories provide verified examples of how task-relevant visual cues correctly support information acquisition to reach the final answer. By extracting initial patterns from these reliable positive examples, we establish a foundational understanding of cue-role relationships, ensuring that the agent has a correct starting point before accumulating specific case-level lessons in the target distribution.

Table 5: Source-data settings for initial skill induction.

Parameter	**Value**
Source data	`OpenSearch-VL-SFT-36K`
Number of domains	7
Sampled trajectories per domain	500
Sample seed	0

Tables 5 and 6 summarize the main implementation settings. After clustering, each candidate cluster is reviewed with aligned centroid exemplars, shown as paired `(image, question, VFP,`

Table 6: Model and construction settings for initial skill induction.

Parameter	Value
Embedding model	`text-embedding-3-small`
Clustering method	K-means with $K = 15$
Clustering seed	0
MLLM for multimodal validation	`gemini-3.1-pro` (DeepMind, 2026)
Validation max tokens	4000
Centroid exemplars per cluster	5
MLLM for pattern authoring	`gemini-3.1-pro`
Authoring max tokens	4000
Output artifacts	`SKILL.md` and `evidence.jsonl`

`KGAP`) records, and the validator additionally inspect periphery images from the same cluster to test whether the inferred cue-utilization relation remains coherent away from the centroid. If a cluster is judged heterogeneous, the validator proposes 2–3 subgroups over image indices.

A.1.2 KNOWLEDGE EVOLUTION

Knowledge evolution operates over benchmark-local rollout data. In the current implementation, each rollout is first summarized into a structured cue-utilization diagnosis conditioned on the question, ground-truth answer, full turn-by-turn trajectory, final answer, and rollout-level verdict. These rollout summaries are then used to extract reusable lessons, which are accumulated into a benchmark-local experience library. When the library reaches its capacity, pruning is applied only at refresh boundaries: retained entries are re-ranked by mentor retrieval hits first and by recency (`sample_index`) second, and only the top-ranked items are kept. The retained library is then passed to a patch-based skill reviser, which consolidates benchmark-local evidence back into the current skill. New patterns are allowed only when the supporting experiences cover at least two distinct source question ids; otherwise the reviser is encouraged to amend existing patterns. Existing patterns are retired only when new evidence directly contradicts them.

Table 7: Main settings for knowledge evolution.

Parameter	Value
Rollouts per training sample	2
Refresh cadence	every 25 samples
Experience retrieval top-K	3
MLLM for rollout summarization	`gemini-3.1-pro`
MLLM for rollout lesson extraction	`gemini-3.1-pro`
Rollout-summary max tokens	4000
Lesson-extraction max tokens	6000
MLLM for skill revision	`gemini-3.1-pro`
Skill-revision max tokens	16000
Experience library cap	120
Minimum support for a new pattern	2 distinct source question ids

A.1.3 TASK EXECUTION

During evaluation, the agent interacts with the environment under a fixed execution budget. This budget constrains trajectory length, rollout-level generation, visual input usage, and external retrieval frequency, while actor decoding parameters are kept constant across runs. Table 8 lists the execution-stage settings used in our experiments.

Table 8: Execution-stage settings used during evaluation.

Parameter	Value
Maximum interaction turns	20
Max tokens per rollout	8192
Temperature	0.6
Top-p	1.0
Maximum images per sample	100
Image search call limit	5
Web search call limit	7

A.1.4 VISUAL-CUE UTILIZATION ANALYSIS

We use GPT-5.6-Terra (OpenAI, 2026b) as the evaluator to analyze visual-cue utilization during execution. Specifically, we measure whether task-relevant visual cues are utilized, whether they are utilized effectively, and whether they are misused. Let $\mathcal{C}$ denote the set of all reference visual cues, and let $\mathcal{C}_{\mathrm{UA}}$, $\mathcal{C}_{\mathrm{UI}}$, and $\mathcal{C}_{\mathrm{NU}}$ denote the subsets labeled as `used_appropriately`, `used_inappropriately`, and `not_used`, respectively. We compute

$$\mathrm{CU} = \frac{|\mathcal{C}_{\mathrm{UA}}| + |\mathcal{C}_{\mathrm{UI}}|}{|\mathcal{C}|}, \tag{6}$$

$$\mathrm{EU} = \frac{|\mathcal{C}_{\mathrm{UA}}|}{|\mathcal{C}|}, \tag{7}$$

$$\mathrm{CM} = \frac{|\mathcal{C}_{\mathrm{UI}}|}{|\mathcal{C}_{\mathrm{UA}}| + |\mathcal{C}_{\mathrm{UI}}|}. \tag{8}$$

The annotation protocol is implemented as a three-stage evaluation prompt. First, the evaluator identifies candidate visual cues from the image conditioned on the question. Second, the evaluator filters these candidates into a reference cue atlas and annotates, for each retained cue, its reasonable role in answering the question as well as its possible misleading use. Third, given this reference atlas and the full agent trajectory, the evaluator audits how each cue is used during execution, assigning one of three statuses: `used_appropriately`, `used_inappropriately`, or `not_used`. The reported CU, EU, and CM metrics are then aggregated programmatically from these cue-level usage annotations. The full prompt templates are provided in Appendix A.5.3.

A.2 EXPERIMENTAL DETAILS

A.2.1 EVALUATION DETAILS

Evaluation datasets. We evaluate all methods on four knowledge-intensive VQA benchmarks:

MMSearch-Plus is a 311-task benchmark enforcing multimodal understanding by requiring the extraction and propagation of fine-grained visual cues through iterative image–text retrieval and cross-validation under retrieval noise, designed to challenge multimodal browsing agents.

VDR-Bench comprises 2,000 multi-hop VQA instances across 10 diverse domains (e.g., Game, Sports, and Sci & Tech), designed to evaluate multimodal search agents in realistic settings. It was constructed to address two critical flaws in existing benchmarks. First, it enforces visual search-centric evaluation to prevent models from "leaking" answers via cross-textual cues or language priors. Second, it eliminates overly idealized retrieval scenarios (e.g., exact whole-image matching), thereby offering a rigorous testbed for authentic visual and textual information acquisition.

LIVEVQA is a pioneering large-scale benchmark dedicated to visual knowledge seeking and updating, comprising 107,143 samples across 12 diverse data categories. Designed to quantify the limitations of MLLMs in processing dynamic visual information, the dataset curates real-world content from prominent international news, media platforms, and academic literature. Furthermore, the benchmark evaluates reasoning capabilities hierarchically: Level 1 targets fundamental visual

entity recognition, while Level 2 demands complex, multi-hop cross-modality inferential reasoning across seven distinct semantic dimensions.

FVQA contains 5,826 questions derived from 2,190 images, designed to push visual reasoning beyond direct image perception. Unlike standard VQA tasks, FVQA strictly requires agents to retrieve external common-sense knowledge. Crucially, each instance is paired with a structural supporting-fact triplet, effectively challenging models to ground observed visual cues with external evidence before formulating an answer.

Following the VDR-Bench project, we use 300-sample subsets of VDR-Bench, LIVEVQA, and FVQA, along with the entire MMSearch-Plus dataset during the experiments.

We randomly sample 100 tasks from each dataset as the target-distribution adaptation set for experience accumulation and skill revision, and use the remaining tasks for evaluation, they are completely disjoint. All datasets are partitioned with random seed 42 to ensure reproducibility, and this dataset partition is kept fixed across all experiments.

Evaluation protocol. Following the official evaluation protocol of VDR-Bench, we use GPT-4o as an LLM judge to determine whether each predicted answer is correct with respect to the reference answer. The judge receives the question, reference answer, and model prediction, and produces a binary correctness verdict using the official VDR-Bench evaluation prompt. We apply the same judging procedure to all methods and benchmarks to ensure consistent evaluation. Unless otherwise specified, all evaluated backbone models use their default reasoning mode, without manually increasing or decreasing the model-specific reasoning budget. The decoding and reasoning configurations are held fixed across methods using the same backbone.

A.2.2 Additional Experimental Results

Cross-Backbone Knowledge Transfer. We examine whether the knowledge accumulated by V-CUES can be reused by a backbone different from the one that generated the adaptation trajectories. We consider two transfer settings on MMSearch-Plus. First, we construct $\mathcal{KB}_{\text{Gemini}}$ using Gemini-3-Flash and transfer the frozen knowledge base to GPT-5.4. Second, we construct $\mathcal{KB}_{5.4}$ using GPT-5.4 and transfer it separately to GPT-5.1 (OpenAI, 2025b) and GPT-4.1 (OpenAI, 2025a). In all transfer settings, the target backbone serves as both the *Mentor* and the *Actor*, while the transferred skill and experience libraries remain unchanged. We perform no additional target-backbone rollouts, experience accumulation, or skill revision before evaluation. All models use the same tool set, retrieval configuration, and rollout protocol.

Table 9: Cross-backbone knowledge transfer on MMSearch-Plus. Each transferred knowledge base is constructed using the source backbone and then frozen.

Target Backbone	Method	KB Source	Avg@2	Pass@2
GPT-5.4	w/ Tools	–	28.9	38.4
	V-CUES (Transferred)	Gemini-3-Flash	35.5	46.4
	V-CUES (Native)	GPT-5.4	37.2	47.9
GPT-4.1	w/ Tools	–	14.0	19.9
	V-CUES (Transferred)	GPT-5.4	21.6	28.0
GPT-5.1	w/ Tools	–	14.7	19.9
	V-CUES (Transferred)	GPT-5.4	22.0	25.6

As shown in Table 9, the transferred knowledge bases consistently improve all target backbones over their tool-only baselines. These results indicate that the knowledge induced by V-CUES is transferable across the evaluated backbones, while target-specific adaptation can provide additional benefits.

Comparison with Generic Visual-Cue Prompting. We examine whether the gains of V-CUES can be reproduced by a simple, fixed instruction for visual-cue utilization. Specifically, we append the following sentence to the prompt of the tool-only agent: *"Before using tools, identify the visual*

cues relevant to the question, determine how each cue may support the task, and use them to guide subsequent information acquisition and evidence integration." The same instruction is used for every task under both GPT-5.2 and GPT-5.4. This baseline uses neither the skill and experience libraries nor the task-specific knowledge adaptation of V-CUES, while all other tool and execution settings remain unchanged.

Table 10: Comparison with a fixed visual-cue instruction on MMSearch-Plus. The prompt-only baseline adds the same one-sentence instruction to every task, without reusable knowledge or task-specific knowledge adaptation.

Backbone	Method	Avg@2	Pass@2
GPT-5.2	w/ Tools	23.7	30.8
	w/ Generic Cue Prompt	24.2	31.3
	V-CUES	**31.0**	**37.9**
GPT-5.4	w/ Tools	28.9	38.4
	w/ Generic Cue Prompt	30.6	39.8
	V-CUES	**37.2**	**47.9**

As shown in Table 10, the fixed instruction provides only modest gains over the tool-only baseline and remains substantially below V-CUES on both backbones. Thus, simply reminding the agent to identify and use visual cues does not recover the full improvement of V-CUES, supporting the value of reusable visual-cue utilization knowledge and its task-specific adaptation.

Test-Time Token Consumption. We report the average total token consumption per task during test-time execution on MMSearch-Plus under GPT-5.2.

Table 11: Average total token consumption per task during test-time execution.

	w/ Tools	XSKILL	V-CUES
Avg. Tokens	22.2k	30.2k	31.0k

As shown in Table 11, knowledge-augmented execution increases the average token consumption from 22.2k for the tool-only baseline to 30.2k for XSKILL and 31.0k for V-CUES. Compared with XSKILL, V-CUES requires only 0.8k additional tokens (2.6%) while achieving higher accuracy.

A.3 Human–Evaluator Agreement

To validate the reliability of the LLM-based cue annotations, we sample 50 execution trajectories containing 335 reference visual cues. Given the same reference-cue atlas, a human expert and the LLM evaluator independently assign exactly one of three statuses to each cue: *Used Appropriately*, if the cue meaningfully enters the execution process and contributes positively, such as by forming a hypothesis, narrowing candidates, guiding tool use, filtering retrieval results, or verifying a candidate; *Used Inappropriately*, if the cue enters the execution process but is misinterpreted, over-interpreted, or incorrectly linked, thereby contributing to an incorrect retrieval, verification, intermediate judgment, or answer; and *Not Used*, if the cue does not meaningfully enter the reasoning or answer path.

Table 12: Cue-level agreement between the human expert and the LLM evaluator over 50 sampled execution trajectories.

Human \ LLM	Not Used	Used Appropriately	Used Inappropriately	Total
Not Used	147	17	3	167
Appropriate	2	121	1	124
Misused	1	18	25	44
Total	150	156	29	335

Table 12 presents the resulting confusion matrix. The LLM evaluator achieves an overall agreement of 87.5% with the human expert and an unweighted Cohen's κ of 0.79, indicating substantial agreement beyond chance.

A.4 Case Study

We include two representative cases where V-CUES succeeds while the same base model with tools alone fails.

A.4.1 Case1: Resolving an occluded opponent from visual cues



Question: "Who is the opponent of the player shown in the match?"

Ground Truth: *Jack Lisowski*

Figure 7: Original image.

Injected Mentor Guidance (V-CUES)

Guidance: ...broadcast graphics as the main anchor..., ...the lower-third scoreboard clearly names one player as Stephen Maguire..., ...specific televised match instance rather than a general rival..., ...the event/round context from the BBC and BetVictor snooker broadcast could help disambiguate..., ...any external match record should be verified against the exact broadcast context rather than a generic Stephen Maguire result.

V-CUES ✓ **Success**

Step 1: The visible player is *Stephen Maguire*, and the frame also shows *BBC*, *BetVictor*, and a scoreboard pattern around *2–3*. Since the opponent's name is occluded, the trajectory starts with a broad broadcast-context search.

```
> web_search(BBC BetVictor Stephen Maguire 2 3 opponent ...)
```

Observation: The first results are noisy and do not yet identify the event.

Step 2: The trajectory then narrows toward BBC tournament coverage that explicitly mentions Stephen Maguire and the BetVictor context.

```
> web_search(site:bbc.com/sport snooker Stephen Maguire BetVictor
...)
```

Observation: These results suggest BBC Welsh Open coverage as a plausible event context.

Step 3: The next step is to check a BBC Welsh Open page directly and see whether the match structure fits the screenshot.

```
> visit(BBC Welsh Open results page)
```

Observation: The page is for *2026*, not the target event, but it confirms that quarter-finals are *best-of-9*, which matches the screenshot cue.

Step 4: With the best-of-9 hypothesis now grounded, the trajectory narrows to *2025 Welsh Open* and directly searches for Stephen Maguire's quarter-final opponent.

```
> web_search(2025 Welsh Open Stephen Maguire quarter-final opponent
BBC)
```

Observation: The search results surface the *2025 Welsh Open* summary page as the key next source.

Step 5: Once the event hypothesis is specific enough, the trajectory verifies Maguire's opponent on the tournament summary page itself.

```
> visit(2025 Welsh Open summary page)
```

Observation: The page explicitly states that in the quarter-finals *Stephen Maguire beat Jack Lisowski 5–3*.

Step 6: Before answering, the retrieved tournament fact is checked against the visible frame state: a *best-of-9 quarter-final* ending *5–3* is compatible with the screenshot's intermediate *2–3* scoreline.

```
> [Answer Generation]
```

Observation: The final answer is grounded by a full chain from broadcast cues to event identification and opponent recovery.

Outcome [Correct]: *The best-of-9 quarter-final with Maguire at 3 and opponent at 2 fits the visible scoreboard exactly. Therefore the opponent shown in this match is Jack Lisowski.*

w/ Tools **✗ Failure**

Step 1: The baseline begins from the same broadcast cues, but its first retrieval path latches onto a speculative opponent lead rather than establishing the event structure.

```
> web_search(BBC Stephen Maguire 3-2 best of 9 opponent ...)
```

Observation: The early results are noisy and suggest a possible *Joe Perry* lead.

Step 2-4: *[Follows the speculative Joe Perry lead, visits a Facebook page, then a BBC Masters 2020 results page, eventually concluding that the Joe Perry hypothesis is incorrect and abandoning it.]*

Step 5: After the event-level lead collapses, the baseline switches to reverse image retrieval instead of returning to the stronger broadcast-context reasoning path.

```
> image_search(reverse, original image)
```

Observation: The reverse search returns a noisy *Barry Hawkins* YouTube short among the results.

Step 6: The Barry Hawkins hit is then treated as the strongest remaining lead.

```
> visit(Barry Hawkins YouTube short)
```

Observation: The page still does not verify that Hawkins is the opponent in this screenshot.

Step 7: The model answers from that noisy hit without a final consistency check against the scoreboard and tournament-structure cues.

```
> [Answer Generation]
```

Observation: It outputs *Barry Hawkins*, which is inconsistent with the event-level evidence that would have led to *Jack Lisowski*.

Outcome [Incorrect]: *The reverse image hint plus the likely appearance of the hidden left-side headshot could fit Barry Hawkins. I'll answer with that likely opponent.*

Takeaway: This example demonstrates the importance of systematically grounding search queries in all available visual context. V-CUES correctly operationalizes the broadcast cues (*BBC + BetVictor + 2(9)3*) into a robust event-level verification strategy. In contrast, the baseline omitted the broadcast context after failing to directly verify the match instance, ultimately being misled by a noisy reverse-image search result.

A.4.2 Case 2: Resolving the correct event location from visual cues



Figure 8: Original image.

Question: "Given that the match did not happen in Northern Ireland. Where was the match held?"

Ground Truth: *Saudi Arabia*

Injected Mentor Guidance (V-CUES)

Guidance: ...One thing to notice is that this is a snooker broadcast frame with useful hard anchors: the player names, the score strip, and venue-related branding around the table..., ...treat the visible names and tournament signage as identifiers for the exact match instance..., ...separate that from the question's location ask, since the frame itself does not plainly print the host city or country..., ...Worth checking whether the surrounding sponsor/backdrop text identifies the competition or series rather than the venue, and keeping in mind that the on-screen score is an in-progress state, not the sought location.

V-CUES ✓ **Success**

Step 1: The image shows a snooker match with the broadcast overlay identifying *Ronnie O'Sullivan* and *Lei Peifan*, plus an in-progress score state. Since the venue text is not directly visible, the trajectory treats the pairing as the main retrieval anchor and starts by searching for the exact match identity rather than guessing from broadcast style alone.

```
> web_search("Ronnie O'Sullivan" "Lei Peifan" 3-0 snooker tournament location)
```

Observation: The first results are noisy and do not yet uniquely determine which meeting is shown.

Step 2: The first search is too noisy, so the trajectory next searches more directly for the player pairing itself.

```
> web_search("Ronnie O'Sullivan" "Lei Peifan" snooker)
```

Observation: These results surface a plausible lead pointing to the *Saudi Arabia Masters*.

Step 3: With that lead in hand, and given the question's explicit note that the match was not in Northern Ireland, the trajectory checks a page that directly ties the two players to the event.

```
> visit(TNT Sports match page)
```

Observation: The page explicitly lists *R. O'Sullivan vs P. Lei — Saudi Arabia Masters* and gives the location as *Riyadh*.

Step 4: With the exact match page in hand, the answer is completed from the verified event instance rather than from background visual plausibility.

```
> [Answer Generation]
```

Observation: The full chain now closes from visible pairing cue to specific tournament page to location.

Outcome [Correct]: *The pairing is verified on a match page for the Saudi Arabia Masters, which lists the event location as Riyadh. Therefore, the match was held in Riyadh, Saudi Arabia.*

w/ Tools (Baseline) ✗ **Failure**

Step 1: The baseline begins from the same visible pairing cue, but immediately mixes it with broader visual impressions such as Chinese-looking broadcast text and table branding.

```
> web_search(Ronnie O'Sullivan Lei Peifan venue snooker match Strachan Chinese characters ...)
```

Observation: The first results include the known *Northern Ireland Open 2019* meeting, which the question already rules out.

Step 2: Instead of converting the exclusion condition into a direct match-level verification step, the trajectory opens an only loosely related BBC page and then continues with queries that already build in a *China* hypothesis.

```
> visit(BBC page); web_search("Ronnie O'Sullivan" "Lei Peifan" snooker venue China)
```

Observation: The retrieved pages do not verify a concrete venue for the shown match, but the subsequent search is already biased toward a China-based answer.

Step 3: The baseline then keeps following this hypothesis with another search tied to the *International Championship*, rather than checking a page that directly resolves which event the Ronnie–Lei pairing belongs to.

```
> web_search(site:snookerhq.com Ronnie O'Sullivan Lei Peifan International Championship venue)
```

Observation: This search still does not produce a page that ties the exact pairing to the inferred Chinese venue.

Step 4: Even without an event page that closes the chain from player pairing to tournament to location, the trajectory accepts the visually plausible hypothesis and answers with a specific Chinese host city.

```
> [Answer Generation]
```

Observation: The final answer, *Daqing, China*, is not supported by a direct match-level verification for the shown event instance.

Outcome [Incorrect]: *The baseline answers Daqing, China by following a plausible China-event hypothesis, but never verifies that the pictured Ronnie O'Sullivan vs Lei Peifan match actually belongs to that event.*

Takeaway: This example shows that success depends on treating the visible player pairing as an event-disambiguation anchor rather than collapsing early to a familiar broadcast context. V-CUES combines the pairing cue with the question's explicit exclusion (*not Northern Ireland*) and continues until a page explicitly ties *Ronnie O'Sullivan vs Lei Peifan* to the *Saudi Arabia Masters* in *Riyadh*. In contrast, *w/ Tools* shifts from the visible pairing to a visually plausible China-event hypothesis and completes the answer without closing the event-level verification chain.

A.5 Prompt Templates

A.5.1 Knowledge Evolution Prompts

Prompt for Rollout Cue-Utilization Summarization

You are analyzing one rollout of a multimodal search agent on a benchmark question. Your job is to observe how the agent *utilized visual cues from the image to drive its search*, not to score its tool use. Ground truth is provided so you can locate the exact cue-utilization step that failed or succeeded — do not reveal or copy the answer in your summary.

Case Question: {`QUESTION`}
Ground truth answer (for your reference only, do not copy): {`GROUND_TRUTH`}
Rollout trajectory (turn-by-turn): {`TRAJECTORY`}
Final answer the agent gave: {`FINAL_ANSWER`}
Judge verdict on this rollout: {`VERDICT`}

Task Write a short, structured observation covering the eight items below. Focus on visual-cue utilization: how the agent selected, interpreted, and operationalized visual information from the image to drive its search, and how that utilization held up (or failed) as evidence came in. Each item should be 1–2 sentences. Skip an item only if the trajectory genuinely offers no information about it (say `"N/A"` — do not fabricate).

1. `question_referent` — What object / region / information carrier in the image did the agent bind the question to at turn 0? Did that binding match what the question is actually asking about?
2. `requested_fact` — Did the agent keep the final requested fact visible throughout, or did it drift to a related-but-different fact?
3. `visual_cue_to_query` — Which visible cues did the agent lift into search queries? Which diagnostic cues were left on the table?
4. `instance_specificity` — Did the agent distinguish the specific depicted instance from the broader family, or collapse them?
5. `evidence_reshapes_reading` — When retrieval results came in, did they cause the agent to revise its interpretation of the image, or did the agent stay locked on the original reading?
6. `evidence_role` — For the decisive piece of evidence, was it treated as a clue, corroborative evidence, or a direct answer? Was that role warranted?
7. `skill_influence` — If a mentor advisory or nudge cards were injected, how did they visibly shape (or fail to shape) the cue-utilization path? `"N/A"` if none.
8. `failure_side` — If the rollout got the wrong answer, was the miss primarily on the cue-utilization side, or was cue utilization fine and the failure came from execution / retrieval quality / an obscure fact? Say `"success"` if the answer was correct.

Output shape (JSON only, no commentary)

```
{
   "question_referent": "...",
   "requested_fact": "...",
   "visual_cue_to_query": "...",
   "instance_specificity": "...",
   "evidence_reshapes_reading": "...",
   "evidence_role": "...",
   "skill_influence": "...",
   "failure_side": "..."
}
```

Prompt for Experience Distillation

You are extracting *reusable visual-cue utilization lessons* from one benchmark case (image + question + rollouts). A cue-utilization lesson has two inseparable parts:

1. `visual_case_cue`: a language description of the image condition that makes this lesson applicable. It must be concrete enough that a downstream mentor, looking at a different image, could decide whether the same condition holds. Do not name specific entities from the current image; describe visible properties instead.
2. `lesson`: the visual-cue utilization heuristic itself — a soft advisory about how to select, interpret, or operationalize the image's visual information for search, and what failure mode to be alert to. Do not tell the agent which tool to call, or in what order.

Case Question: {`QUESTION`}
Ground truth (for your reference only, do not copy into the lesson): {`GROUND_TRUTH`}
Per-rollout cue-utilization observations: {`ROLLOUT_SUMMARIES`}

Extraction rules (hard)

1. Zero or more lessons is fine. Output an empty list if the case does not yield a reusable pattern.
2. `visual_case_cue` is required and non-empty. If you cannot describe a repeatable image condition without naming this image's specific entities, drop the lesson.
3. Reject generic advice such as "verify externally", "search more", "cross-check sources", "be careful with dates", "don't rely on memory", or "prefer official sources".
4. Never copy the ground-truth answer, proper nouns from the image, exact search queries, or the benchmark id in either field.
5. This is not a fix recipe. Do not write "the correct move was to search X"; write the observation that would have made a next-time agent slow down or read the image differently.
6. Do not force the lesson into referent / carrier / evidence buckets; describe the mechanism plainly.
7. Limit `visual_case_cue` to 32 words and `lesson` to 64 words. Use the soft template "When [case configuration], consider [cue-utilization lesson], because [brief consequence/boundary]" only as a guide, not a schema.
8. Include boundary conditions inline when the case demonstrates one.
9. It is valid to extract a lesson from a successful rollout too.

Output shape (JSON only)

```
{
   "lessons": [
    {
      "visual_case_cue": "<image condition, <=32 words,
no specific entity names>",
      "lesson": "<cue-utilization lesson, <=64 words,
soft advisory>",
      "supports": ["<'rollout_0' | 'rollout_1' |
'both'>"],
      "analysis_tags": ["<optional free-form tag(s)>" ]
    }
  ]
}
```

If the case yields no reusable cue-utilization lesson, return {`"lessons": []`}.

Prompt for Skill Revision

You are consolidating a benchmark-local Experience library into a compact patch on the current advisory skill `S`. `S` is derived from an authored baseline `S_0` by applying prior refresh patches (0 patches on the first refresh — in that case `S` is exactly `S_0`). You will see `S` rendered in full below; treat `S` as the state to edit. Your ops become the next patch appended to the log; you may edit or retire cards that prior patches added. You will receive:

- The current skill body `S` (verbatim), containing D / U / T sections. Each card is `### U-XX --- <name>` or `### T-XX --- <name>` followed by a yaml block, `**when**` / `**consider**` / `**avoid**`, and italic trailers.
- A ranked list of Experience entries mined on this benchmark's own train split. Each has `visual_case_cue`, `lesson`, `source_qids`, and a usage `hits` count.

Task Produce a small ops list that improves `S` for this benchmark.

Consolidation is mandatory

1. Do not produce one card per E entry.
2. Group E entries whose `visual_case_cue` and `lesson` describe the same underlying cue-utilization pattern. Consolidate a group into one new card or one edit on an existing card.
3. A new card requires the union of `source_qids` across supporting E entries to contain at least `MIN_QIDS` distinct qids.
4. Do not re-consolidate patterns that prior patches already folded unless new evidence strictly extends the coverage.

Route each group to U or T

- **U**: what visual information to carry forward and why — which cues matter, what task role they play, what they cannot establish, and what misinterpretation to avoid.
- **T**: how to operationally acquire evidence — which tool first, when to refine a query, when to visit a source, when to isolate a region, or fallback strategies.

Card shape (must match S exactly) For every `add_U` / `add_T` op, `card_md` must contain the full markdown card with header, yaml block, `when`/`consider`/`avoid`, `consolidation_note`, and `evidence`. For `edit_U` / `edit_T`, `new_card_md` must be the full replacement card, preserving the original header and yaml block while amending the prose fields.
Card numbering Number new cards sequentially starting from `NEXT_U_ID` and `NEXT_T_ID`; numbering is monotonic across batches.
Retire Retire a card only when a consolidated group directly contradicts it. Do not retire for topical overlap alone.

Output shape (JSON only, no code fence, no commentary)

```
{
   "consolidation_summary":  "<3--6 line prose describing
groups, routing, and drops>",
   "ops":  [
    {"op":  "add_U", "card_id":  "U-<new_id>", "card_md":
"<full markdown for the new U card>"},
    {"op":  "add_T", "card_id":  "T-<new_id>", "card_md":
"<full markdown for the new T card>"},
    {"op":  "edit_U", "card_id":  "U-<existing_id>",
"new_card_md":  "<full replacement of that U card>",
"reason":  "<one line>"},
    {"op":  "edit_T", "card_id":  "T-<existing_id>",
"new_card_md":  "<full replacement of that T card>",
"reason":  "<one line>"},
    {"op":  "retire", "card_id":  "U-<existing_id>",
"reason":  "<one line naming the contradicting E ids>"}
  ]
}
```

Emit only ops you can defend. Empty `ops` (with only a `consolidation_summary`) is a valid, honest output.

A.5.2 TASK EXECUTION PROMPTS

Prompt for the Mentor Agent

You are the mentor for a multimodal search agent (the actor).
You see the image, the question, a library of nudge cards (U for visual-cue utilization, T for tool-use hints){{E_HAS_SUFFIX}}. You do one thing:

- produce an advisory reading of the case, and pick any genuinely relevant cards{{E_PICK_SUFFIX}}.

You do NOT solve the case. You do NOT hand the actor an answer, a plan of tool calls, or a "commit/abstain" verdict — the actor owns those.

Library (verbatim)
Read the library as a set of advisory lenses before examining the case. Its perspectives may help reveal distinctions that would otherwise be overlooked, but the concrete image and question remain primary. Do not force the case into a card or invent a utilization gap merely because a card describes one.
{{SKILL_MD}}
{{E_SECTION}}

Output shape (JSON only)

```
{
   "advisory_reading":  "<2--5 sentences of
soft-imperative reflection, preferably under 180
words>",
   "picked_U_ids":  ["U-XX", ...],
   "picked_T_ids":  ["T-XX", ...]{{E_OUTPUT_FIELD}}
}
```

Case
Question: {{QUESTION}}
(The image is attached.)

Respond with the JSON only, no code fence, no commentary.

Prompt for the Actor Agent

Below, the mentor has looked at the same image and question and left you an advisory reading plus a small set of nudge cards it thinks are relevant. This material is perspective, not directive:

- The mentor's advisory reading and picked cards are suggestions to consider, not commands. When your tool observations conflict with the mentor's reading, trust the observations.
- The cards describe soft heuristics ("when...", "consider...", "avoid..."). They do not tell you exactly which tool to call or what to answer.
- You still decide when to search, when to visit, and when to answer.

Mentor advisory reading
<MENTOR_ADVISORY>

Visual-cue utilization patterns to keep in mind
<PICKED_U_CARDS>

Tool-use hints to keep in mind
<PICKED_T_CARDS>

Analogous prior experiences
<PICKED_E_ENTRIES>

You are a visual reasoning agent. Your goal is to answer questions about images.

 # AVAILABLE TOOLS:
You have access to the following tools:

1. **web_search**: Search the web for information, facts, or current events
2. **image_search**: Search for related images using text query or reverse image search
3. **visit**: Visit a webpage and extract its main content
4. **code_interpreter**: Execute Python code for image processing, analysis, and calculations

 # INSTRUCTIONS:

1. **Analyze:** Carefully observe the image and the user's question.
2. **Think:** Explain your step-by-step reasoning process.
3. **Use Tools:** Call appropriate tool to gather information and help answer the question.
4. **Iterate as needed:** Continue reasoning and using tools in next turns until you are confident in your findings.
5. **Answer:** Once confident, provide the final answer inside `<answer>...</answer>` tags!

 # IMPORTANT:

- Always explain your detailed reasoning process before using any tool.
- You can ONLY call one tool at one turn! Do not call multiple tools in one turn!
- You MUST provide your final answer using complete `<answer>...</answer>` tags!

A.5.3 Visual-Cue Utilization Analysis Prompts

Prompt for Candidate Visual Cue Identification

You are an expert visual analyst. Given an image and a question, identify all visual cues in the image that are potentially relevant to answering the question.

Rules

- Only report cues that are genuinely visible in the image.
- Organize by semantic units; do not split one visual object into many micro-details.
- For each cue, describe what is visible and why it might relate to the question.
- Do not attempt to answer the question.
- For multi-image inputs, include one `GLOBAL_IMAGE` entry per image (`GLOBAL_IMAGE_1`, `GLOBAL_IMAGE_2`, etc.). For single-image inputs, use `GLOBAL_IMAGE`.
- `GLOBAL_IMAGE` represents the full image as a single visual entity suitable for reverse image search.

Output valid JSON

```
{
   "candidate_cues":  [
    {
      "cue_id":  "C1",
      "description":  "what is visible",
      "location":  "where in the image",
      "potential_relevance":  "why this might help answer
the question"
    },
    {
```

```
      "cue_id": "GLOBAL_IMAGE",
      "description": "the full image as a single visual
representation",
      "location": "entire image",
      "potential_relevance": "could be used for
reverse image search to identify the depicted
scene/event/object"
    }
  ],
   "atlas_quality": "high | medium | low",
   "atlas_quality_reason": "brief reason for quality
assessment"
}
```

Atlas quality reflects whether the visible content can be reliably described, not whether the image directly contains the final answer. Most knowledge-intensive tasks require external retrieval precisely because the answer is not in the image; that does not make the atlas low quality. Set `atlas_quality` to `"low"` only when severe image degradation prevents reliable cue identification.

Prompt for Reference Visual Cue Filtering and Role Annotation

You are an expert visual analyst reviewing candidate visual cues for task relevance.

Given

- the original image and question,
- previously identified candidate cues,
- and the reference answer.

Your job is to filter out irrelevant cues and annotate what each relevant cue can reasonably support.

Important rules

- Do not assign importance labels such as critical, useful, or weak. Every cue that survives filtering is simply relevant to the task.
- Be precise about what each cue supports versus what it does not support. Avoid hindsight: judge by what a competent observer would infer without the reference answer.
- `GLOBAL_IMAGE` is always preserved: it represents the full image usable for reverse image search.
- Remove any candidate cue that is genuinely irrelevant to the question.
- The reference answer is used only to clarify the requested target and review the roles of cues already identified earlier. Do not infer or add visual details solely because they are consistent with the reference answer.
- You may only preserve a candidate cue, remove it as irrelevant, or refine its role description. If you must add a cue that was previously missed, mark it accordingly and justify its visibility.

Output valid JSON

```
{
   "question_target": "what information the question
asks for",
   "visual_cues": [
    {
      "cue_id": "C1",
      "description": "what is visible",
      "location": "where in the image",
      "reasonable_role": "how this cue could legitimately
help answer the question",
      "misleading_use": "how this cue could be
misinterpreted leading to wrong answers"
    },
    {
      "cue_id": "GLOBAL_IMAGE",
```

```
      "description": "the complete original image",
      "location": "entire image",
      "reasonable_role": "reverse-image-search input to
locate the depicted instance",
      "misleading_use": ""
    }
  ],
   "removed_cues": [
    {"cue_id": "C4", "reason": "not relevant to the
question"}
  ],
   "remaining_information": "what still needs to be
found externally after using visual cues"
}
```

Prompt for Trajectory-Level Visual Cue Usage Audit

You are an expert visual-information analyst. You will audit how an AI search agent used (or did not use) the available visual cues from a reference atlas during its trajectory.

IMPORTANT
You are auditing visual-cue usage, NOT evaluating answer correctness. Do not infer that the trajectory failed or succeeded. Do not judge the final answer quality.

Core principle
A cue counts as USED if it meaningfully entered the agent's reasoning process and influenced an intermediate judgment, hypothesis, retrieval strategy, candidate selection, verification step, or final answer construction.
Do NOT restrict "used" to explicit search queries. A cue may be used through direct visual reasoning, candidate narrowing, result filtering, or answer justification without ever being turned into a query string.

Cue Status Rules
Each reference cue from the atlas gets exactly one status:

- `"used_appropriately"`: the cue entered the reasoning chain and contributed positively. This includes, but is not limited to, helping the agent form a correct hypothesis, narrow candidates, choose a suitable tool/strategy, filter noisy search results, verify a retrieved candidate, or justify an answer in a way that is faithful to the cue.
- `"used_inappropriately"`: the cue entered the reasoning chain, but it was misread, overinterpreted, incorrectly linked to an entity/event, or otherwise pushed the agent toward a wrong intermediate judgment, wrong retrieval direction, wrong verification, or wrong answer construction.
- `"not_used"`: the cue did not meaningfully enter the reasoning chain. Mere existence in the image is not enough. A fleeting mention or generic scene description does NOT count as used unless the cue is subsequently incorporated into the agent's reasoning or answer path.

Important clarifications

- If the agent explicitly invokes a cue when forming a hypothesis, narrowing candidates, interpreting search results, or supporting an answer, that generally counts as USED even if no tool call follows.
- If a cue is mentioned and the later reasoning continues along that same interpretation without rejecting it, treat the cue as having influenced the process.
- Do NOT mark a cue as `"used_inappropriately"` merely because it did not become a query or explicit verification step. Lack of external operationalization alone is NOT misuse.
- Cropping, OCR, or image search counts as engagement with a cue, but tool use alone does not determine whether the cue was used appropriately or inappropriately.
- Using the full original image for reverse image search means `GLOBAL_IMAGE` is used appropriately as a retrieval starting point unless the image is clearly used to support a wrong candidate or wrong interpretation.
- Multiple cues may jointly support the same inference. Do not require one cue to be the sole decisive cause.

Output Format
Output ONLY valid JSON (no markdown fences, no commentary):

```
{
   "cue_usage":  [
    {
      "cue_id":  "C1",
      "status":  "used_appropriately |
used_inappropriately | not_used",
      "first_use_turn":  null,
      "usage_description":  "how the agent engaged with
this cue",
      "evidence":  "specific trajectory turns/actions"
    }
  ],
   "cue_usage_note":  "1-2 sentence summary of overall
cue engagement pattern"
}
```

Do NOT compute cue counts or utilization metrics — those will be calculated programmatically.