SEARCHEYES: TOWARDS FRONTIER MULTIMODAL DEEP SEARCH INTELLIGENCE VIA SEARCH WORLD SIMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal search agents must reason over long horizons, yet existing pipelines design training data, search environments, and rewards in isolation. This fragmentation discards synthesis metadata, relies on irreproducible external engines, and leaves RL with sparse trajectory-level supervision. We introduce **SearchEyes**, a knowledge-graph-based *simulated search world* that unifies these components. **Perception-Knowledge Chains (PKC)** sample constrained multi-hop paths over visual and knowledge entities while retaining hop-level metadata that defines both a self-contained environment and reward anchors. **Hop-Anchored Policy Optimization (HaPO)** uses these anchors for step-level credit assignment without a learned process reward model. Across six multimodal knowledge-intensive benchmarks, SearchEyes achieves state-of-the-art performance among open-source multimodal search agents; SearchEyes-27B improves over the strongest open-source baseline by 6.2 points on average.

1 INTRODUCTION

Reinforcement learning (RL) now underpins autonomous agents that reason, use tools, and act over long horizons Team (2026a); AI (2026); Team (2026b); DeepSeek-AI (2026). Search agents extend this paradigm by iteratively retrieving and synthesizing external evidence. Text systems such as Search-R1 Jin et al. (2025), ASearcher Gao et al. (2025b), and REDSearcher Chu et al. (2026b) demonstrate long-horizon reasoning over retrieved evidence, while Vision-DeepResearch Huang et al. (2026b), OpenSearch-VL Chen et al. (2026a), MMSearch-R1 Wu et al. (2025b), and VSearcher Zhang et al. (2026a) extend this capability to multimodal search. Despite this progress, training long-horizon multimodal search agents remains difficult because data synthesis, environments, and rewards are typically designed in isolation.

This separation creates three linked failures. Graph-based synthesis can produce multi-hop questions Chen et al. (2026a); Huang et al. (2026b); Lu et al. (2025), but discards path structure, entity IDs, and relation types once question–answer pairs are generated. Without this structure, training environments fall back on external search engines Jin et al. (2025); Chu et al. (2026b); Li et al. (2025c), which are costly, nondeterministic, and difficult to reproduce. RL then receives only final-answer, trajectory-level rewards Shao et al. (2024); Yu et al. (2025), whose sparsity grows rapidly with the number of reasoning hops Feng et al. (2025); Dong et al. (2025); Cui et al. (2025).

SearchEyes addresses this disconnect by using a knowledge graph as a persistent substrate across all three stages. **Perception-Knowledge Chains (PKC)** sample constrained multi-hop paths over visual and knowledge entities and retain their structural metadata. The same graph instantiates a self-contained search environment, while **Hop-Anchored Policy Optimization (HaPO)** turns the retained entity sequence into step-level credit anchors, blending hop-level and trajectory-level advantages without a separate process reward model. Thus, metadata produced during synthesis becomes supervision for downstream agent training rather than a discarded artifact.

Figure 1 illustrates the overall framework. Across six multimodal knowledge-intensive benchmarks, SearchEyes achieves state-of-the-art performance among open-source multimodal search agents; SearchEyes-27B improves over the strongest open-source baseline by 6.2 points on average.

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.

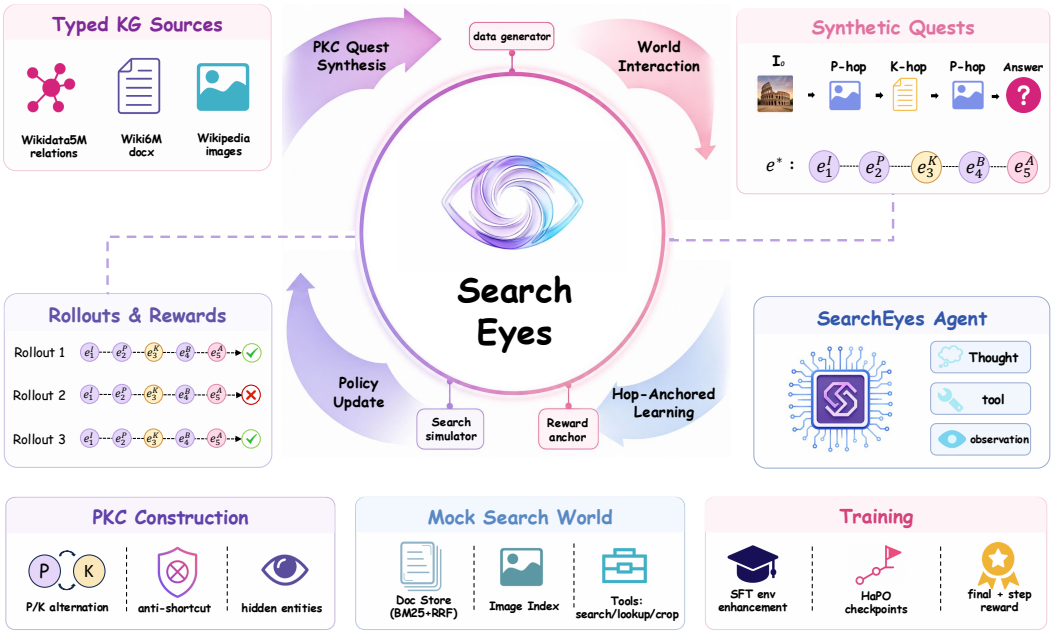


Figure 1: **Overview of SearchEyes.** A knowledge graph serves as the unified backbone for three tightly-coupled stages: PKC question synthesis (left), a self-contained search world for agent interaction (center), and hop-anchored reward signals for policy optimization (right). By retaining structural metadata across all stages, SearchEyes addresses the data–environment–reward disconnect that limits existing pipelines.

Our contributions are summarized as follows:

1. **Unified Search World Simulation.** We introduce SearchEyes, which constructs a knowledge graph linking Wikidata relations with entity descriptions and images. The graph supports data synthesis, environment simulation, and RL training within a unified simulated search world.
2. **Perception-Knowledge Chain Synthesis.** We propose PKC, a KG-driven pipeline that samples constrained multi-hop paths over the visual–knowledge intersection graph, with multi-level anti-shortcut filtering to produce high-quality training data. KG metadata is retained throughout synthesis to support downstream step-level training.
3. **Agentic Post-Training Pipeline.** We design a complete post-training pipeline comprising SFT with retrieval-boosted trajectory generation and observation denoising, followed by HaPO (Hop-Anchored Policy Optimization), which reuses PKC's hop-level gold entity IDs for step-level credit assignment without a separately trained process reward model.
4. **Empirical Results.** Experiments on six multimodal knowledge-intensive benchmarks show that SearchEyes achieves state-of-the-art performance among open-source multimodal search agents; SearchEyes-27B improves over the strongest open-source baseline by 6.2 points on average.

2 Method

2.1 Overview

Given an image I_0 and question q, a multimodal agent interacts with a search environment $\mathcal{E}$ to retrieve evidence and produce an answer. We model this process as an episodic MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R)$ whose transition kernel $\mathcal{P}$ is induced by the agent–environment interaction. At turn t, the state $s_t = (I_0, q, z_1, a_1, o_1, \ldots, z_{t-1}, a_{t-1}, o_{t-1})$ contains the full interaction history. A ReAct-style policy $\pi_\theta(y_t \mid s_t)$ Yao et al. (2023) generates $y_t = (z_t, a_t)$, where z_t is a reasoning span and a_t is either a structured tool call or a terminal answer. A tool call returns $o_t = \mathcal{E}(a_t)$; a terminal answer

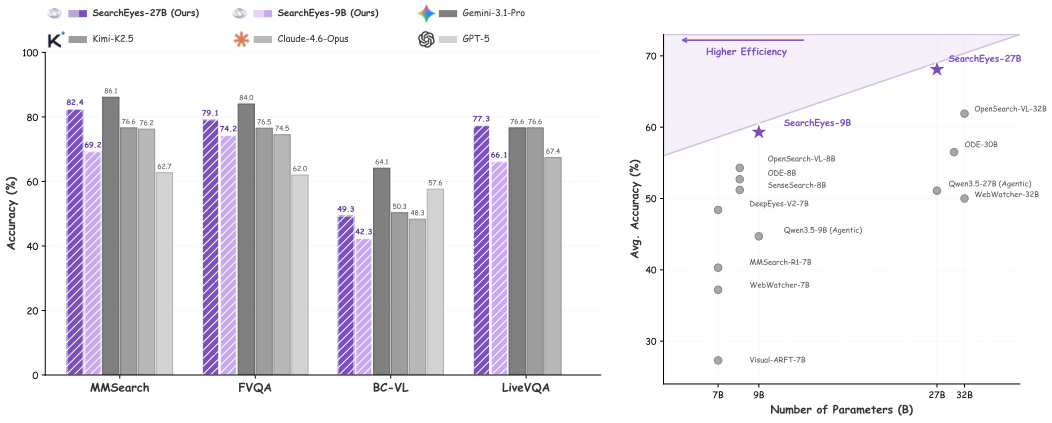


Figure 2: **Comparison of model scale versus average accuracy on multimodal knowledge-intensive benchmarks.** *Left:* SearchEyes-27B (purple, hatched) achieves 82.4 on MMSearch and 79.1 on FVQA, competitive with Gemini-3.1-Pro while using only a 27B open-source backbone. *Right:* Parameter efficiency of open-source multimodal search agents. SearchEyes-9B nearly matches 30B-scale baselines (e.g., OpenSearch-VL-30B at 59.8%) with 3.3× fewer parameters, demonstrating the parameter efficiency enabled by PKC synthesis and HaPO training.

ends the trajectory $\tau = (z_1, a_1, o_1, \ldots, z_T, a_T)$ and receives binary reward $R(\tau) \in \{0, 1\}$. We optimize $J(\theta) = \mathbb{E}[R(\tau)]$ while masking environment observations from the loss. SearchEyes uses one knowledge graph $\mathcal{G}$ to define the search world, synthesize the questions and trajectories used for training, and retain the entity anchors used by HaPO.

2.2 Knowledge Graph and Search Environment

Knowledge graph construction. We combine Wikidata5M relation triples Wang et al. (2021), Wiki6M entity text Hu et al. (2023), and Wikipedia images into $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, where $\mathcal{R} \subseteq \mathcal{V} \times \mathcal{P} \times \mathcal{V}$. Retained entities occur in both text sources, with images attached where available; $\mathcal{V}_v$ denotes entities with an indexed image and $\mathcal{V}_k = \mathcal{V} \setminus \mathcal{V}_v$ the remaining knowledge-only entities. For PKC sampling, a curated predicate subset $\mathcal{P}_{\text{PKC}}$ is mapped by φ to four semantic domains $\mathcal{T} = \{\text{PERSON}, \text{WORK}, \text{ORG}, \text{GEO}\}$. Appendix B.1 reports construction details.

Self-contained search environment. The entity documents $\mathcal{D} = \{\text{doc}(v) \mid v \in \mathcal{V}\}$ form a deterministic training search world without external APIs. Dense retrieval ranks entity documents by embedding similarity, whereas BM25 supplies a complementary lexical ranking. Text search fuses the two rankings using reciprocal rank fusion (RRF):

$$\text{score}_{\text{RRF}}(q, v) = \sum_{r \in \{\text{dense, BM25}\}} \frac{1}{\kappa + \text{rank}_r(q, v)}, \tag{1}$$

where κ is a smoothing constant. The five tools provide text search, visual search, source lookup, summarization, and lightweight computation. Canonical retrieval records expose a title, QID, snippet, and optional indexed image. Visual search accepts a region from I_0 or a previously exposed image, so downstream visual checkpoints use only interaction-visible evidence. We define $\text{entities}(o)$ as the explicit result IDs in o, excluding unlinked free-text mentions; these IDs align observations with $\mathbf{e}^* = (v_1, \ldots, v_{K+1})$. Appendix B.4 gives the backends and schemas.

2.3 Perception-Knowledge Chain Synthesis

This section describes the pipeline from the knowledge graph $\mathcal{G}$ to supervised fine-tuning data (see Figure 3 for an overview). The pipeline proceeds in three stages: constrained path sampling on $\mathcal{G}$, information-concealed question generation, and trajectory synthesis in the search environment.

Constrained Path Sampling. From $\mathcal{G}$, we sample multi-hop chains via a constrained random walk. Starting from an anchor entity $v_1 \in \mathcal{V}_v$ that possesses at least two outgoing edges, the walk produces

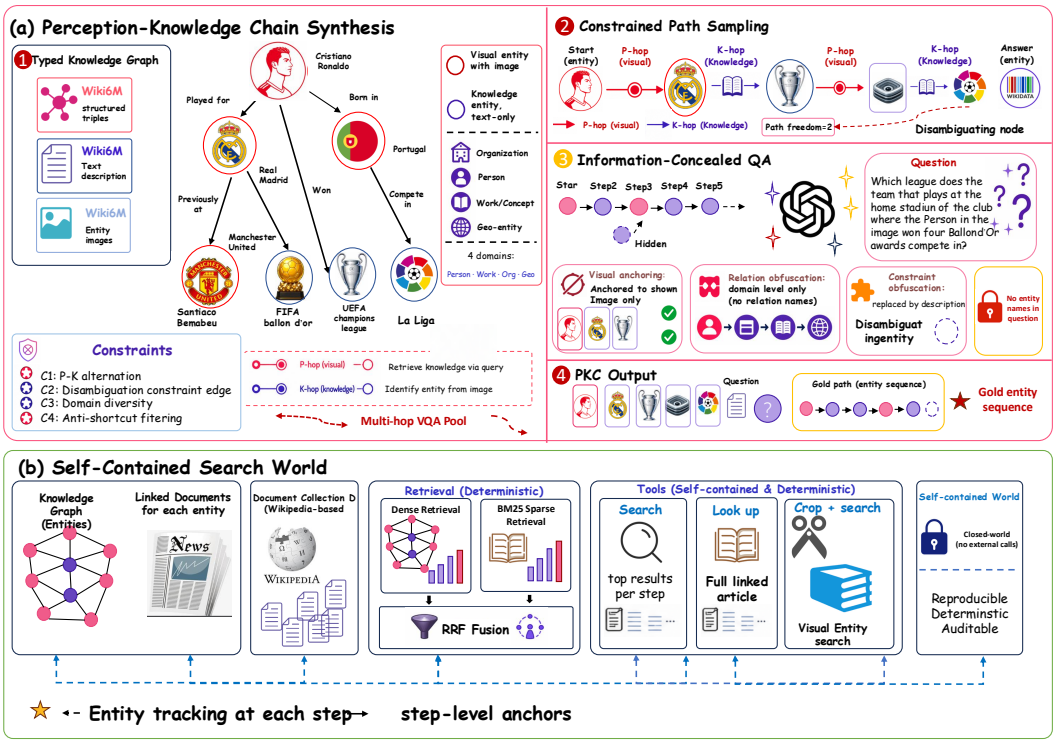


Figure 3: **Method architecture.** (a) Perception-Knowledge Chain (PKC) synthesis: constrained path sampling produces multi-hop questions with interleaved perception checkpoints and a disambiguation edge. The first P label grounds the input image; a downstream P label marks an image-backed entity whose visual attachment can enter the interaction after retrieval. These synthesis labels specify evidence availability, not a one-to-one pattern of rollout tool calls; consecutive K labels remain permitted. Information concealment prevents entity-name leakage. (b) The same graph defines a deterministic search world with hybrid BM25+dense retrieval fused by RRF.

a chain of K graph hops:

$$\mathbf{p} = \left(v_1 \xrightarrow{r_1} v_2 \xrightarrow{r_2} \cdots \xrightarrow{r_K} v_{K+1}\right), \tag{2}$$

where each $r_k \in \mathcal{P}_{\mathrm{PKC}}$ and $(v_k, r_k, v_{k+1}) \in \mathcal{R}$ for $k \in \{1, \ldots, K\}$. Thus, a K-hop chain contains K transitions and $K+1$ entities, including the visual anchor and answer. We impose four constraints:

(C1) Multi-perception interleaving. Each entity position stores a synthesis label $h_j \in \{\mathrm{P}, \mathrm{K}\}$: h_1 labels the input anchor v_1, and h_j for $j \geq 2$ labels the entity v_j reached after transition r_{j-1}. The labels satisfy

$$h_1 = \mathrm{P}, \qquad h_j = \mathrm{P} \implies \left(h_{j-1} = \mathrm{K} \wedge v_j \in \mathcal{V}_v\right) \ (2 \leq j \leq K+1), \qquad \sum_{j=1}^{K+1} \mathbf{1}[h_j = \mathrm{P}] \geq 2. \tag{3}$$

The first P is observed as I_0; a downstream P image becomes addressable only after v_j is retrieved or opened. Thus P positions are separated by K, consecutive K labels remain allowed, and chains without a downstream P are discarded. The labels constrain evidence availability, not the policy's exact tool-call sequence.

(C2) Disambiguation constraint edge. The final transition must be one-to-few, creating sibling candidates $\mathrm{Sib}(v_{K+1}) = \{u \neq v_{K+1} \mid (v_K, r_K, u) \in \mathcal{R}\}$. We attach a constraint edge (v_{K+1}, r_c, v_c) unique among these siblings:

$$(v_{K+1}, r_c, v_c) \in \mathcal{R} \quad \wedge \quad \forall\, u \in \mathrm{Sib}(v_{K+1}) : (u, r_c, v_c) \notin \mathcal{R}. \tag{4}$$

This target-specific property requires a second evidence branch to identify v_{K+1}.

(C3) Semantic domain diversity. Let $\mathbf{d} = \big(\varphi(r_1), \ldots, \varphi(r_K)\big) \in \mathcal{T}^K$ be the domain sequence of $\mathbf{p}$, where each $\varphi(r_k)$ maps predicate r_k to one of the four semantic domains in $\mathcal{T}$. The path must satisfy both local diversity and global coverage:

$$\varphi(r_k) \neq \varphi(r_{k+1}) \;\; \forall\, k \in \{1, \ldots, K-1\}, \qquad |\{\varphi(r_k)\}_{k=1}^{K}| \geq \min(3, K). \tag{5}$$

These constraints avoid same-domain chains while ensuring heterogeneous coverage.

(C4) Anti-shortcut filtering. We apply complementary structural and behavioral checks. A bounded graph search rejects a candidate when its answer is reachable from the anchor in fewer than K transitions; string checks remove leaked names, duplicate entity sequences, and repeated anchor–answer tasks. Finally, restricted verification rejects questions answerable without search, without the image, or from the image alone. These filters operate during data construction and never supply information to the trained policy; Appendix B.1 provides the complete procedure.

Each sampled chain retains the gold entity sequence $\mathbf{e}^* = (v_1, v_2, \ldots, v_{K+1})$ as metadata. Its length is exactly $K+1$, comprising the visual anchor followed by one target entity per graph hop, and it serves as step-level supervision in §2.4.

Information-Concealed Question Generation. The generator hides the anchor name, KG predicate labels, path entities, and exact constraint entity. At turn zero, I_0 is the sole anchor representation; downstream P images can enter context only through retrieved or opened records. Relations and the disambiguating property are rendered as natural-language descriptions. We reject leaked, trivially guessable, or visually unsupported questions and require an indexed image at every P position.

Trajectory Synthesis. For each synthesized question, an expert policy samples multiple trajectories in a teacher-only environment with privileged retrieval and observation denoising. We retain correct trajectories, replay their tool calls in the unmodified search world, and replace the teacher-facing observations with the raw replayed results. A replay is exported only when every call remains executable and the evidence used by later reasoning is present in its preceding raw history. Consequently, the student never observes the retrieval boost or denoised text used to obtain successful demonstrations. Appendix B.1 gives the sampling, acceptance, and context-consistency checks. The student is trained by causal language modeling over its thoughts, actions, and answer, with environment observations masked from the loss. Let $\mathbf{x} = (x_1, \ldots, x_L)$ denote the full token sequence of an exported sample, and let $M_{\text{gen}}(\ell) \in \{0, 1\}$ indicate whether token x_ℓ belongs to agent-generated spans (i.e., within some z_t, a_t, or the final answer). The SFT objective is:

$$\mathcal{L}_{\text{SFT}} = -\sum_{\ell=1}^{L} M_{\text{gen}}(\ell) \cdot \log \pi_\theta(x_\ell \mid x_{<\ell}), \tag{6}$$

where observation tokens (o_t spans) are provided as context but excluded from the loss. Each sample additionally retains $\mathbf{e}^*$ as metadata for step-level credit assignment in §2.4.

2.4 Hop-Anchored Policy Optimization

Long-horizon search makes binary final-answer rewards sparse, while GRPO Shao et al. (2024) assigns one trajectory-level advantage to every generated token. **HaPO** (**H**op-**A**nchored **P**olicy **O**ptimization) instead reuses the PKC entity sequence $\mathbf{e}^*$ for step-level credit without a learned process reward model (Figure 4). For each question, we sample a group of G trajectories $\{\tau^{(i)}\}_{i=1}^{G}$ with binary outcomes $R^{(i)} \in \{0, 1\}$. The same outcome reward is used at both levels; HaPO changes only how rollouts are grouped and how their relative advantages are assigned across generated tokens.

Trajectory-level advantage. The trajectory-level advantage is:

$$\hat{A}_{\text{ep}}^{(i)} = \frac{R^{(i)} - \bar{R}}{\sigma_R + \epsilon}, \qquad \bar{R} = \frac{1}{G}\sum_{j=1}^{G} R^{(j)}, \qquad \sigma_R = \text{std}\big(\{R^{(j)}\}_{j=1}^{G}\big). \tag{7}$$

Hop-anchored step-level advantage. Exact state matching is brittle because different queries can expose the same evidence through different rankings or snippets. We instead anchor $\tau^{(i)}$ at gold entity $v_k \in \mathbf{e}^*$ when its QID first appears in a structured retrieval result at step $t_k^{(i)}$. The first occurrence

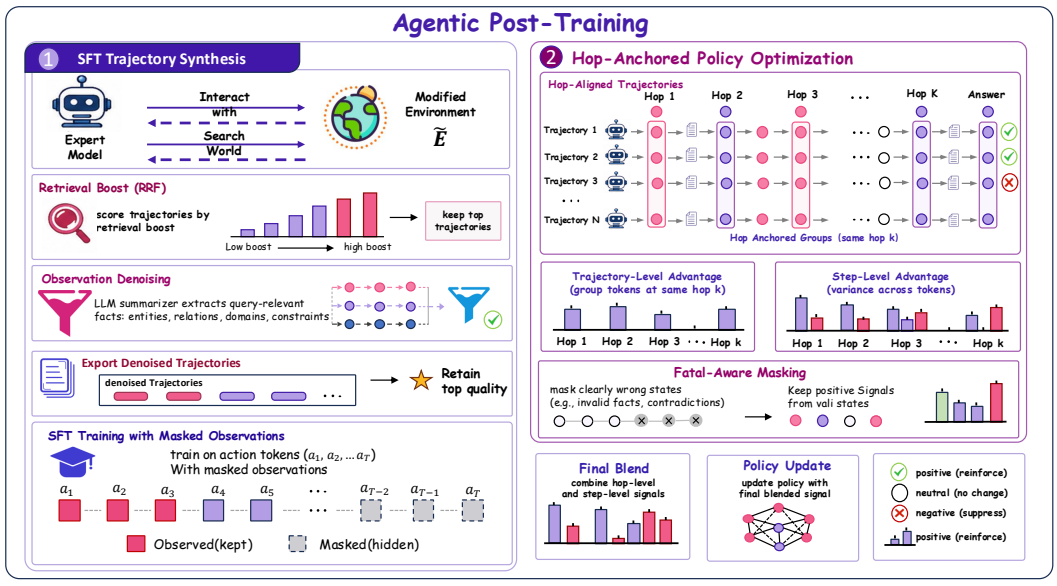


Figure 4: **Agentic post-training pipeline.** Stage 1 (SFT): an expert model generates trajectories in a retrieval-boosted environment; final-answer correctness determines acceptance, and LLM-denoised observations guide the generator. Before export, accepted tool calls are replayed in the unmodified environment and the resulting unboosted raw observations are used for masked-observation supervised training. Stage 2 (HaPO): online rollouts are grouped by shared hop anchors to compute step-level advantages; fatal-aware masking suppresses degenerate suffixes while one-sided clamping retains positive signals from valid prefixes.

marks the earliest point at which the policy could use that evidence. It therefore aligns semantically comparable rollout prefixes without assuming identical queries, observations, or reasoning text.

For each gold entity v_k, we form a hop anchor group:

$$\mathcal{H}_k = \{(i, t_k^{(i)}) \mid v_k \in \text{entities}(o^{(i)}_{t_k^{(i)}}),\ t_k^{(i)} = \min\{t : v_k \in \text{entities}(o_t^{(i)})\}\}, \tag{8}$$

For groups with $|\mathcal{H}_k| \geq 2$ and non-zero outcome variance, we compute

$$\hat{A}^{(i,t_k^{(i)})}_{\text{hop}} = \frac{R^{(i)} - \bar{R}_k}{\sigma_{R_k} + \epsilon}, \tag{9}$$

where $\bar{R}_k$ and σ_{R_k} are computed within $\mathcal{H}_k$. At token x_ℓ, $\hat{A}^{(i,\ell)}_{\text{hop}}$ is the value of the most recent valid anchor, or 0 before any such anchor; it is replaced when a later valid anchor appears. A rollout with no valid group therefore falls back to trajectory-level credit. Groups compare continuations that exposed the same entity, not merely the same turn index.

Combined advantage. The final per-token advantage interpolates between the two levels:

$$\hat{A}^{(i,\ell)}_{\text{final}} = \alpha \cdot \hat{A}^{(i)}_{\text{ep}} + (1 - \alpha) \cdot \hat{A}^{(i,\ell)}_{\text{hop}}, \tag{10}$$

where $\alpha \in [0, 1]$ interpolates between trajectory-only and hop-only credit.

Fatal-aware masking. If M consecutive tool errors begin at t_f, we mask the suffix $t \geq t_f$. Following fatal-aware GRPO Chen et al. (2026a), the valid prefix uses one-sided clamping $\hat{A}_{\text{final}} \leftarrow \max(\hat{A}_{\text{final}}, 0)$ so actions preceding the failure are not penalized.

Policy optimization. Following SAPO Gao et al. (2025a), HaPO replaces the standard hard-clipped surrogate with a smooth sigmoid gate applied to the combined advantage. The gating function uses asymmetric temperatures γ^+ and γ^- (with $\gamma^- > \gamma^+$):

$$g(r_\ell, \hat{A}) = \begin{cases} \sigma\big(\gamma^+(r_\ell - 1)\big) \cdot \frac{4}{\gamma^+}, & \text{if } \hat{A} > 0, \\ \sigma\big(\gamma^-(r_\ell - 1)\big) \cdot \frac{4}{\gamma^-}, & \text{if } \hat{A} \leq 0, \end{cases} \tag{11}$$

Table 1: **Performance on multimodal knowledge-intensive QA and web-search benchmarks.** All benchmark columns report Pass@1 answer accuracy (%) under the unified VDR-Bench evaluation setting. **Bold** and underline mark the best and second-best in each column. BC-VL denotes BrowseComp-VL. Avg. is computed over all six benchmarks; – denotes unavailable scores.

Model	SimpleVQA	VDR	MMSearch	LiveVQA	BC-VL	FVQA	Avg.
Direct Reasoning							
GPT-4o OpenAI (2024)	51.7	1.7	18.7	28.1	5.5	48.0	25.6
GPT-5 (Direct) OpenAI (2025)	61.6	9.8	35.1	44.4	48.6	54.4	42.3
Gemini-2.5-Flash Comanici et al. (2025b)	57.9	6.2	30.4	51.0	37.1	47.7	38.4
Gemini-2.5-Pro (Direct) Comanici et al. (2025a)	63.0	8.0	39.8	60.3	43.1	60.7	45.8
Claude-4-Sonnet (Direct) Anthropic (2025b)	50.9	2.0	18.7	38.5	29.3	35.3	29.1
Claude-3.7-Sonnet Anthropic (2025a)	42.7	4.6	21.1	38.0	32.3	36.7	29.2
Qwen3-VL-8B (Direct) Bai et al. (2025)	47.1	2.8	11.7	23.1	24.1	24.2	22.2
Qwen3-VL-30B-A3B (Direct) Bai et al. (2025)	53.2	3.8	18.7	42.7	29.6	34.7	30.5
Qwen3-VL-32B Bai et al. (2025)	58.0	4.1	19.8	45.5	30.8	34.1	32.1
Agentic Workflow							
GPT-5 (Agentic) OpenAI (2025)	67.3	17.6	62.7	–	57.6	62.0	–
Gemini-2.5-Pro (Agentic) Comanici et al. (2025a)	74.3	10.0	55.7	–	42.3	65.0	–
Gemini-3.1-Pro Google DeepMind (2026)	–	–	**86.1**	76.6	**64.1**	**84.0**	–
Claude-4-Sonnet (Agentic) Anthropic (2025b)	–	13.6	67.2	–	48.6	69.0	–
Claude-4.6-Opus Anthropic (2026)	–	–	76.2	67.4	48.3	74.5	–
Kimi-K2.5 Team (2026a)	–	–	76.6	76.6	50.3	76.5	–
Multimodal DeepResearch Models							
Visual-ARFT-7B Liu et al. (2025)	42.4	3.3	34.5	25.4	16.5	41.7	27.3
MMSearch-R1-7B Wu et al. (2025b)	57.4	2.9	53.8	48.4	20.9	58.4	40.3
DeepEyes-V2-7B Hong et al. (2025)	59.4	7.8	63.7	–	–	60.6	–
WebWatcher-7B Geng et al. (2025)	54.3	10.3	49.1	51.2	21.2	–	–
Qwen3-VL-8B (Agentic) Bai et al. (2025)	52.0	17.0	37.4	50.6	27.9	58.7	40.6
SenseNova-MARS-8B Chng et al. (2025)	61.7	19.4	67.4	56.2	35.1	67.1	51.2
ODE-8B Huang et al. (2026a)	70.3	20.4	66.0	–	41.9	64.7	–
OpenSearch-VL-8B Chen et al. (2026a)	71.6	20.8	64.5	59.6	37.6	71.5	54.3
Qwen3-VL-30B-A3B (Agentic) Bai et al. (2025)	55.1	20.2	44.2	62.0	34.1	63.0	46.4
ODE-30B Huang et al. (2026a)	71.0	26.4	69.7	–	46.1	69.3	–
OpenSearch-VL-30B Chen et al. (2026a)	74.9	33.5	68.7	67.4	41.1	73.2	59.8
WebWatcher-32B Geng et al. (2025)	59.0	–	55.3	58.7	27.0	–	–
OpenSearch-VL-32B Chen et al. (2026a)	76.2	33.8	72.3	70.5	43.8	74.7	61.9
SearchEyes (Ours)							
Qwen3.5-9B (Agentic) Team (2026b)	56.8	19.3	41.6	55.4	31.7	63.2	44.7
SearchEyes-9B (Ours)	75.4	28.3	69.2	66.1	42.3	74.2	59.3
Qwen3.5-27B (Agentic) Team (2026b)	64.3	25.8	59.1	50.6	39.4	67.3	51.1
SearchEyes-27B (Ours)	**80.9**	**39.4**	82.4	**77.3**	49.3	79.1	**68.1**

where $\sigma(\cdot)$ is the sigmoid function and $\gamma^- > \gamma^+$ more strongly damps negative-advantage updates. Let $r_\ell^{(i)} = \pi_\theta(x_\ell \mid x_{<\ell})/\pi_{\text{old}}(x_\ell \mid x_{<\ell})$. For fixed SFT reference π_{ref}, define $\Delta_\ell^{(i)} = \log \pi_{\text{ref}} - \log \pi_\theta$ on token x_ℓ and use $d_{\text{KL}}^{(i,\ell)} = \exp(\Delta_\ell^{(i)}) - \Delta_\ell^{(i)} - 1$. As in SFT, $M_{\text{gen}}^{(i)}(\ell)$ masks observations and $Z_i = \sum_\ell M_{\text{gen}}^{(i)}(\ell)$ normalizes each rollout:

$$\mathcal{L}_{\text{HaPO}} = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{Z_i}\sum_{\ell=1}^{|\tau^{(i)}|} M_{\text{gen}}^{(i)}(\ell)\Big[-g\big(r_\ell^{(i)}, \hat{A}_{\text{final}}^{(i,\ell)}\big)\hat{A}_{\text{final}}^{(i,\ell)} + \lambda d_{\text{KL}}^{(i,\ell)}\Big], \tag{12}$$

Here, λ weights the KL term; per-trajectory normalization gives every rollout equal aggregate weight, and $\epsilon > 0$ stabilizes group-relative estimates under low outcome variance.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

We instantiate SearchEyes-9B and SearchEyes-27B from Qwen3.5-9B and Qwen3.5-27B Team (2026b) using training data synthesized by our PKC pipeline; ablations use the 9B variant unless stated otherwise. We evaluate all six benchmarks in Table 1 under the official VDR-Bench configuration and report Pass@1 answer accuracy. Data construction, optimization, agent tools, inference, and scoring details are consolidated in Appendix B.

3.2 MAIN RESULTS

Overall performance. SearchEyes-27B reaches 68.1 average accuracy, surpassing the strongest open model with complete results, OpenSearch-VL-32B, by 6.2 points. It leads prior open models in every benchmark column: the gains over OpenSearch-VL-32B range from 4.4 points on FVQA to 10.1 points on MMSearch, rather than arising from a single favorable dataset. The 27B model also obtains 77.3 on LiveVQA, slightly above the proprietary Gemini-3.1-Pro and Kimi-K2.5 agents reported under the same benchmark setting, while remaining an open 27B model.

Parameter efficiency. SearchEyes-9B averages 59.3, improving on OpenSearch-VL-8B by 5.0 points and outperforming it on all six benchmarks. Despite using fewer than one-third as many parameters, it comes within 0.5 points of OpenSearch-VL-30B and approaches the 32B baseline. The 27B configuration further raises the SearchEyes average by 8.8 points, with the largest absolute improvements on MMSearch, LiveVQA, and VDR. Figure 2 visualizes this scaling trend.

Controlled comparison. Tool access alone is not responsible for these gains. With the same five-tool interface and evaluation loop, SearchEyes raises its matched Qwen3.5 agentic base from 44.7 to 59.3 at 9B and from 51.1 to 68.1 at 27B. The improvements are positive on every benchmark and are especially large for the 27B model on LiveVQA ($+26.7$) and MMSearch ($+23.3$), indicating that structured search training transfers across both live-information and multimodal-retrieval tasks.

3.3 ABLATION STUDY

We report Qwen3.5-9B ablations on four benchmarks: SimpleVQA, VDR, MMSearch, and FVQA.

PKC data quality. Removing any component lowers the four-benchmark average, showing that PKC's constraints are complementary rather than cosmetic. Anti-shortcut filtering has the largest effect (-7.7), followed by retrieval-boosted teacher generation (-4.7) and multi-P interleaving (-4.2). Cross-modal grounding contributes 3.3 points, while the disambiguation edge and information concealment each contribute 2.5 points; even the smaller domain-diversity term adds 1.8 points. Full PKC also exceeds free-form synthesis, Wikipedia path sampling Chen et al. (2026a), and HopChain-style augmentation Wang et al. (2026b) by 12.6, 8.3, and 6.4 points, respectively. Together, these comparisons separate the value of graph-constrained synthesis from simply increasing the number of generated trajectories.

HaPO credit assignment. All optimizer baselines share the SFT checkpoint, RL question pool, search world, rollout budget, and update count; only the optimization algorithm changes. HaPO improves over GRPO, GiGPO, and DAPO by 4.0, 3.3, and 3.7 points. Replacing entity-anchor groups with groups defined by turn index loses 2.7 points relative to full HaPO, and episode-only credit loses 2.8 points; conversely, hop-only credit remains stronger than episode-only but is 0.7 points below their interpolation. Smooth gating, fatal-aware masking, one-sided clamping, and observation masking each provide additional gains. Together, they complement semantic credit assignment with stable long-horizon optimization.

Qualitative analysis. Appendix D traces a representative successful rollout. The agent first grounds the unnamed museum through visual search, then retrieves its architect and the shared prize before using the concealed opera-house clue to resolve the final candidate. The main-chain QIDs become visible in order, while the disambiguating entity appears only in the final observation. This example illustrates how PKC structure can align semantically equivalent progress across rollouts without requiring tool calls to mirror graph hops one-for-one.

Table 2: **PKC ablation.** Each structural constraint contributes to final performance. We report per-benchmark accuracy and the average over four benchmarks.

Configuration	SVQA	VDR	MMS	FVQA	Avg.
Full PKC pipeline	75.4	28.3	69.2	74.2	**61.8**
w/o multi-P interleaving constraint	72.1	22.6	64.8	70.8	57.6
w/o disambiguation constraint edge	73.8	25.1	66.4	72.0	59.3
w/o anti-shortcut filtering	69.5	20.8	59.7	66.3	54.1
w/o domain diversity balancing	74.2	26.0	67.1	72.5	60.0
w/o cross-modal grounding filter	72.6	24.7	65.8	71.0	58.5
w/o information concealment	73.0	25.4	66.9	71.8	59.3
w/o retrieval boost (β=1)	71.8	23.2	63.5	69.7	57.1
Free-form LLM synthesis (no graph)	65.2	16.4	52.3	62.8	49.2
Wikipedia path sampling Chen et al. (2026a)	68.7	19.2	58.6	67.4	53.5
HopChain-style augmentation Wang et al. (2026b)	70.1	21.5	61.0	68.9	55.4

Table 3: **HaPO ablation.** Per-benchmark accuracy and Δ relative to standard GRPO.

Configuration	SVQA	VDR	MMS	FVQA	Avg.	Δ
Full HaPO (ours)	75.4	28.3	69.2	74.2	**61.8**	**+4.0**
w/ turn-index grouping (no entity anchors)	73.1	25.8	66.0	71.5	59.1	+1.3
w/o smooth asymmetric gating	74.2	27.0	67.8	72.8	60.5	+2.7
w/o fatal-aware masking	73.8	26.5	67.2	72.2	59.9	+2.1
w/o one-sided advantage clamping	74.5	27.3	68.0	73.0	60.7	+2.9
w/o observation token masking	74.0	26.8	67.5	72.5	60.2	+2.4
episode-only (α=1)	73.4	25.5	65.8	71.2	59.0	+1.2
w/ fixed α=0 (hop-only)	74.8	27.5	68.5	73.5	61.1	+3.3
Standard GRPO Shao et al. (2024)	72.0	24.5	64.3	70.2	57.8	0.0
GiGPO Feng et al. (2025)	72.8	25.2	65.1	70.8	58.5	+0.7
DAPO Yu et al. (2025)	72.3	24.8	64.8	70.5	58.1	+0.3

4 Related Work

Recent multimodal deep-research agents learn long-horizon visual search through tool-augmented SFT and RL Wu et al. (2025b); Geng et al. (2025); Chen et al. (2026a), but generally treat the training corpus, retrieval environment, and optimization signal as separate components. In parallel, environment-scaling and task-synthesis methods construct interactive worlds or multi-hop questions Dong et al. (2026); Zhang et al. (2026b); Jiao et al. (2026), while agentic RL algorithms improve optimization stability or introduce finer-grained credit assignment Shao et al. (2024); Gao et al. (2025a); Feng et al. (2025). SearchEyes connects these directions by preserving structure across stages: the entity path retained during PKC synthesis also instantiates the simulated search world and defines HaPO's hop anchors, rather than being discarded after question generation. Appendix A provides a detailed comparison with each line of work.

5 Conclusion

We introduced **SearchEyes**, a unified framework for training multi-hop visual search agents. PKC constructs structurally verified training paths and preserves their metadata; HaPO reuses that metadata as semantic anchors for step-level credit assignment. SearchEyes achieves state-of-the-art results among open multimodal search agents on six knowledge-intensive benchmarks. Beyond the absolute gains, the central result is architectural: the same entity path governs question synthesis, the simulated search world, and HaPO's hop anchors. This reuse supplies auditable training supervision without a learned process reward model or a training-time external search engine. The consistent improvements across both model scales suggest that preserving structure across stages is a practical route to stronger and more reproducible multimodal search agents.

AI USE STATEMENT

Generative AI tools were used to assist with migrating the manuscript's LaTeX source to the ICLR 2027 template, improving language and presentation, checking internal consistency, and verifying bibliographic metadata and the resulting build. The authors reviewed all suggested changes, made the scientific and editorial decisions, and take responsibility for the final content.

REFERENCES

Pranjal Aggarwal, Graham Neubig, and Sean Welleck. Gym-anything: Turn any software into an agent environment. *arXiv preprint arXiv:2604.06126*, 2026.

Zhipu AI. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Anthropic. The claude model family: Claude 3.7 sonnet. *Anthropic Model Card*, 2025a. URL https://docs.anthropic.com/en/docs/about-claude/models.

Anthropic. Claude 4 sonnet system card. *Anthropic Technical Report*, 2025b. URL https://www.anthropic.com/research/claude-4-sonnet-system-card.

Anthropic. Claude opus 4.6 system card. *Anthropic Technical Report*, 2026. URL https://www-cdn.anthropic.com/14e4fb01875d2a69f646fa5e574dea2b1c0ff7b5.pdf.

Shuai Bai, Yuxuan Cai, Rui Chen, Keqin Chen, Xingyu Chen, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Keer Lu, Hao Liang, Wentao Zhang, and Weipeng Chen. Research: Learning to reason with search for llms via reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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. *arXiv preprint arXiv:2605.05185*, 2026a.

Zhaorun Chen, Zhuokai Zhao, Kai Zhang, Bo Liu, and Qi Qi. Scaling agent learning via experience synthesis. *arXiv preprint arXiv:2511.03773*, 2026b. ICLR 2026.

Xiao Cheng, Wei Zhang, Shuai Zhang, Jie Yang, Xinyu Guan, Xingyu Wu, et al. Simplevqa: Multimodal factuality evaluation for multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4637–4646, 2025.

Yong Xien Chng, Tao Hu, Wenwen Tong, Xueheng Li, Jiandong Chen, Haojia Yu, Jiefan Lu, Hewei Guo, Hanming Deng, Chengjun Xie, Gao Huang, Dahua Lin, and Lewei Lu. Sensenova-mars: Empowering multimodal agentic reasoning and search via reinforcement learning. *arXiv preprint arXiv:2512.24330*, 2025.

Meng Chu, Xuan Billy Zhang, Kevin Qinghong Lin, Lingdong Kong, and Jize Zhang. Agentic world modeling: Foundations, capabilities, laws, and beyond. *arXiv preprint arXiv:2604.22748*, 2026a.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, and Shengchao Hu. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*, 2026b.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Panupong Pasupat, Noveen Sachdeva, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025a.

Gheorghe Comanici et al. Gemini 2.5 flash: A fast yet capable multimodal model. *Google DeepMind Technical Report*, 2025b.

Ganqu Cui, Lifan Yuan, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025. Online implicit PRM without step-level annotations.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence. `https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro`, 2026. Technical Report.

Guanting Dong, Junting Lu, Junjie Huang, Wanjun Zhong, and Longxiang Liu. Agent-world: Scaling real-world environment synthesis for evolving general agent intelligence. *arXiv preprint arXiv:2604.18292*, 2026.

Guanting Dong, Hangyu Mao, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025. RUC-NLPIR; entropy-guided agentic rollout + step-wise credit.

Kaixuan Fan, Kaituo Feng, et al. Exploring reasoning reward model for agents. *arXiv preprint arXiv:2601.22154*, 2026. S-Lab NTU / Shanghai AI Lab (MMLab); multi-faceted reasoning reward.

Lang Feng, Zhenghai Xue, et al. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025. NeurIPS 2025.

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. *arXiv preprint arXiv:2504.05288*, 2025.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025a. Qwen Team, Alibaba.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b. arXiv:2508.07976.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025.

Google DeepMind. Gemini 3.1 pro technical report. *Google DeepMind Technical Report*, 2026.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

Hexiang Hu, Yi Luan, Yang Chen, et al. Open-domain visual entity recognition: Towards recognizing millions of wikipedia entities. In *ICCV*, 2023.

Shijue Huang, Hangyu Guo, Guanting Dong, Chenxin Li, Junting Lu, Xinyu Geng, Zhaochen Su, Zhenyu Li, Shuang Chen, Hongru Wang, and Yi R. Fung. Towards on-policy data evolution for visual-native multimodal deep search agents. *arXiv preprint arXiv:2605.10832*, 2026a.

Wenxuan Huang, Yu Zeng, Qiuchen Wang, and Zhen Fang. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*, 2026b.

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. *arXiv preprint arXiv:2409.12959*, 2024.

Zhengbo Jiao, Shaobo Wang, Zifan Zhang, Xuan Ren, and Wei Wang. Agentic proposing: Enhancing large language model reasoning via compositional skill synthesis. *arXiv preprint arXiv:2602.03279*, 2026. ICML 2026.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Guankai Li, Jiabin Chen, Yi Xu, Xichen Zhang, and Yuan Lu. Hypereyes: Dual-grained efficiency-aware reinforcement learning for parallel multimodal search agents. *arXiv preprint arXiv:2605.07177*, 2026a.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, and Yida Zhao. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. *arXiv preprint arXiv:2509.13305*, 2025a.

Wanli Li, Bince Qu, Bo Pan, Jianyu Zhang, and Zheng Liu. Literesearcher: A scalable agentic rl training framework for deep research agent. *arXiv preprint arXiv:2604.17931*, 2026b.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025b.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*, 2025c. NeurIPS 2025.

Ziyu Liu, Yuhang Zang, Yushan Zou, Zijian Liang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual agentic reinforcement fine-tuning. *arXiv preprint arXiv:2505.14246*, 2025.

OpenAI. Gpt-4o system card. *OpenAI Technical Report*, 2024. URL https://cdn.openai.com/gpt-4o-system-card.pdf.

OpenAI. Gpt-5 system card. *OpenAI Technical Report*, 2025. URL https://openai.com/index/gpt-5-system-card/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. NeurIPS 2024; introduces GRPO.

Kimi Team. Kimi k2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026a. Moonshot AI.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, 2026b. Alibaba Cloud, Technical Report.

Dunwei Tu, Hongyan Hao, Hansi Yang, Yihao Chen, and Yi-Kai Zhang. Scaleenv: Scaling environment synthesis from scratch for generalist interactive tool-use agent training. *arXiv preprint arXiv:2602.06820*, 2026. ICML 2026.

Bowen Wang, Dunjie Lu, Junli Wang, Tianyi Bai, and Shixuan Liu. Cua-gym: Scaling verifiable training environments and tasks for computer-use agents. *arXiv preprint arXiv:2605.25624*, 2026a.

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*, 40:2413–2427, 2017.

Shaobo Wang, Zhengbo Jiao, Zifan Zhang, Yilang Peng, and Xu Ze. Socratic-zero: Bootstrapping reasoning via data-free agent co-evolution. *arXiv preprint arXiv:2509.24726*, 2025.

Shenzhi Wang, Shixuan Liu, Jing Zhou, Chang Gao, and Xiong-Hui Chen. Hopchain: Multi-hop data synthesis for generalizable vision-language reasoning. *arXiv preprint arXiv:2603.17024*, 2026b.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, and Liwen Zhang. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*, 2025a. NeurIPS 2025.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. Mmsearch-r1: Incentivizing lmms to search. *arXiv preprint arXiv:2506.20670*, 2025b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2023.

Qiying Yu, Zheng Zhang, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. NeurIPS 2025; ByteDance.

Yu Zeng, Wenxuan Huang, Zhen Fang, Shuang Chen, Yufan Shen, Yishuo Cai, Xiaoman Wang, Zhenfei Yin, Lin Chen, Zehui Chen, Shiting Huang, Yiming Zhao, Xu Tang, Yao Hu, Philip Torr, Wanli Ouyang, and Shaosheng Cao. Vision-deepresearch benchmark: Rethinking visual and textual search for multimodal large language models. *arXiv preprint arXiv:2602.02185*, 2026.

Ruiyang Zhang, Qianguo Sun, Chao Song, Yiyan Qi, and Zhedong Zheng. Vsearcher: Long-horizon multimodal search agent via reinforcement learning. *arXiv preprint arXiv:2603.02795*, 2026a.

Xichen Zhang, Ziyi He, Yinghao Zhu, Sitong Wu, and Shaozuo Yu. Searchgym: Bootstrapping real-world search agents via cost-effective and high-fidelity environment simulation. *arXiv preprint arXiv:2601.14615*, 2026b.

Rui Lu et al. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn RL. *arXiv preprint arXiv:2509.10446*, 2025.

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

Zefang Zong, Dingwei Chen, et al. AT^{2}PO: Agentic turn-based policy optimization via tree search. *arXiv preprint arXiv:2601.04767*, 2026. ACL 2026; turn-level tree exploration + credit assignment.

A Related Work

Deep Research Agents. Augmenting LLMs with search tools and training them via reinforcement learning has become a dominant paradigm for building autonomous research agents. On the text side, Search-o1 Li et al. (2025b) first integrated agentic search into reasoning chains by dynamically invoking search engines during inference. Search-R1 Jin et al. (2025) trained LLMs end-to-end with outcome-based RL to autonomously decide when and what to search. ReSearch Chen et al. (2025) explored RL-driven interleaved search and reasoning, while REDSearcher Chu et al. (2026b) achieved cost-efficient long-horizon agent training through graph-structured task synthesis and mid-training. In the multimodal domain, MMSearch-R1 Wu et al. (2025b) introduced the first end-to-end RL framework for multimodal search. WebWatcher Geng et al. (2025) reformulated text QA as visual QA via reverse image search. DeepEyes Zheng et al. (2025) incentivized "thinking with images" by treating the model's native visual grounding as a tool and training end-to-end with RL without cold-start SFT. Vision-DeepResearch Huang et al. (2026b) enabled MLLMs to perform tens of reasoning steps with hundreds of search engine interactions through cold-start SFT and RL training. VSearcher Zhang et al. (2026a) transformed static multimodal models into long-horizon tool-calling search agents. ASearcher Gao et al. (2025b) scaled the interaction horizon beyond 100 turns through fully asynchronous RL. OpenSearch-VL Chen et al. (2026a) provided a fully open-source recipe combining Wikipedia-based multi-hop data synthesis with fatal-aware GRPO. SearchEyes complements this line by retaining synthesis structure and reusing it across the environment and RL objective.

RL Environment Scaling. The scarcity of high-quality interactive environments has become a central bottleneck for agentic RL training Chu et al. (2026a). In the general tool-use setting, ScaleEnv Tu et al. (2026), DreamGym Chen et al. (2026b), and Agent-World Dong et al. (2026) address this bottleneck through diverse environment synthesis, world-model-based experience generation, and self-evolving co-evolution of environments and policies, respectively. CUA-Gym Wang et al. (2026a) and Gym-Anything Aggarwal et al. (2026) further automate environment construction for computer-use and arbitrary software scenarios. For search agents, the challenge extends to matched multi-hop reasoning data. WebSailor-V2 Li et al. (2025a) pioneered KG-based search data synthesis on a cyclic knowledge graph with a paired simulated environment. WebDancer Wu et al. (2025a) proposed scalable QA synthesis with two-stage SFT-then-RL training. HopChain Wang et al. (2026b) synthesized multi-hop vision-language data for RLVR. On the environment side, SearchGym Zhang et al. (2026b) and LiteResearcher Li et al. (2026b) constructed high-fidelity search simulators to reduce real-engine costs. Agentic Proposing Jiao et al. (2026) and Socratic-Zero Wang et al. (2025) explored agent-driven problem synthesis via goal-directed decision processes and multi-agent co-evolution. SearchEyes differs by coupling task generation and environment construction through the same knowledge graph.

Agentic Reinforcement Learning. Agentic RL has become the standard paradigm for training strong agents, with recent flagship models such as Kimi K2.5 Team (2026a), GLM-5 AI (2026), Qwen3.5 Team (2026b), and DeepSeek-V4 DeepSeek-AI (2026) all adopting it as a core post-training component. The algorithmic foundation rests on GRPO Shao et al. (2024), which eliminates the value model via group-relative advantage estimation. DAPO Yu et al. (2025) scaled GRPO to industrial-level training, while SAPO Gao et al. (2025a) replaced hard clipping with a smooth sigmoid gate for improved long-sequence stability. However, these methods assign a uniform trajectory-level advantage to all tokens, causing credit assignment to degrade in multi-turn agent tasks. To address this, GiGPO Feng et al. (2025) introduced nested group structures for step-level credit assignment. ARPO Dong et al. (2025) combined entropy-guided agentic rollouts with step-level reward allocation for multi-turn web agents. AT^{2}PO Zong et al. (2026) unified turn-level tree search with policy optimization through entropy-guided expansion and turn-wise credit. An alternative line tackles credit assignment through denser reward signals. PRIME Cui et al. (2025) pioneered implicit process rewards that can be estimated online without step-level annotation. More recently, Agent-RRM Fan et al. (2026) extended this idea to agent trajectories with multi-dimensional structured critiques. HaPO instead derives step-level supervision directly from entity anchors retained during PKC synthesis, avoiding a separate process reward model. This ties each credit boundary to an observable event in the search world—the first retrieval of a retained entity—rather than to a free-form textual step boundary.

B EXPERIMENTAL SETUP

This section consolidates the complete experimental configuration used to construct the training data, optimize both model scales, expose tools to the agent, and evaluate all systems. The main text reports only the model family, the fact that our data are synthesized by PKC, and adherence to the official benchmark configuration.

B.1 PKC DATA CONSTRUCTION

Knowledge Graph Statistics. The graph is built from Wikidata5M Wang et al. (2021) (4.6M entities, 822 relations, 20.6M triples) intersected with Wiki6M/OVEN Hu et al. (2023) (6.06M entity descriptions, 2.03M images). After intersection and filtering, the knowledge graph contains 1.2M entities (of which 340K are visual entities with images), 5.8M relation triples, and 822 raw predicate types. PKC path sampling operates on a curated relation subset, with each retained predicate assigned to one of four semantic domains.

Path Sampling Parameters. We sample chains with $K \in \{3, 4, 5\}$ graph hops (relation transitions), yielding $K+1$ entities per main chain. During path sampling, we exclude meta-level predicates from candidate transitions. Each of the $K+1$ entity positions receives a P or K label. The anchor v_1 is P; a downstream entity v_j can be P only when v_j has an indexed image and the preceding entity position is K. Consecutive K labels are allowed, and each chain contains at least two P positions and covers at least $\min(3, K)$ semantic domains. Eligible downstream positions are labeled P during sampling. If a completed chain still has only the anchor P, we promote an image-supported position whose predecessor is K; the chain is discarded when no such position exists. At export, I_0 supplies the anchor image, whereas each downstream P image remains attached to its entity record and becomes visible only after that record is retrieved or opened.

Anti-Shortcut Verification. We first apply deterministic graph and string checks. For a chain of length K, a bounded breadth-first search from the anchor rejects the candidate if its answer is reachable in fewer than K directed transitions; direct anchor–answer adjacency is therefore rejected as a special case. We additionally remove answer leakage, overly popular or underspecified answers, repeated anchor–answer tasks, duplicate entity sequences, missing intermediate descriptions, and missing anchor images. We then run DeepSeek-V4-Pro-Preview at zero temperature under three restricted conditions: image and question without search, question without image or search, and image without the composed question context. Responses are compared with the gold answer after case, punctuation, and article normalization; a candidate answered correctly in any restricted condition is discarded. The verifier is used only for data filtering and is not part of the trained SearchEyes policy.

Question Generation. We use Claude Opus 4.6 to convert each sampled path into one question candidate. The anchor is presented only through its image, with its name and aliases withheld. The generator paraphrases relation semantics rather than exposing KG predicate labels and describes the disambiguation constraint relationally rather than naming its entity. We reject any question containing a path entity name or alias, as well as trivially guessable questions. Every retained P checkpoint must have a corresponding image in the visual index. After deduplication and filtering, the scale-specific pools contain 22K questions for SearchEyes-9B (10K for SFT trajectory distillation and 12K for RL rollouts) and 23K questions for SearchEyes-27B (8K for SFT and 15K for RL). The two corpora are synthesized independently rather than shared across model scales.

Evaluation-set Decontamination. Before trajectory synthesis, we de-duplicate each scale-specific training pool against the released fixed evaluation subsets at both the question and image levels. Any matched example is removed from the SFT and RL pools. This evaluation-set check is applied in addition to the within-pool path and anchor–answer deduplication used during PKC synthesis.

Trajectory Synthesis. During teacher rollout, gold-path entities receive a retrieval multiplier $\beta = 3.0$:

$$\widetilde{\text{score}}(q', v) = \begin{cases} \beta \, \text{score}_{\text{RRF}}(q', v), & v \in \mathbf{e}^*, \\ \text{score}_{\text{RRF}}(q', v), & \text{otherwise}, \end{cases} \tag{13}$$

where q' is the current retrieval query. This boost changes only the observations available to the offline generator; it does not modify the generator policy or decoding rule. For each scale-specific SFT pool, we independently sample multiple trajectories per question from a DeepSeek-V4-Pro-

Preview generator policy and retain candidates with correct final answers before the remaining quality checks. Independent sampling allows different accepted trajectories for the same question to vary in query wording, tool-call order, and intermediate evidence selection; we preserve this multiplicity as trajectory-level augmentation rather than collapsing each question to one demonstration. Observation denoising uses a separate DeepSeek-V4-Pro-Preview call to extract query-relevant facts from retrieved documents. These teacher models are used only for offline data construction and are not available to SearchEyes during RL or benchmark inference. The LLM-denoised, retrieval-boosted observations are used only by the generator. After a trajectory is accepted, its tool calls are replayed against the ordinary search environment with $\beta = 1$, and these unboosted raw observations are stored in the exported sequence. Exact rankings and snippets may drift after replay. We retain the trajectory only if every call remains executable and the evidence referenced by each later reasoning/action span is present in the preceding unboosted interaction history. The final scale-specific SFT datasets contain approximately 30K accepted expert trajectories for SearchEyes-9B and 28K for SearchEyes-27B. The two trajectory corpora are sampled independently, further avoiding identical demonstration sets across model scales while retaining the same acceptance criteria.

B.2 Training Configuration

SFT Stage. We train SearchEyes-9B and SearchEyes-27B from Qwen3.5-9B and Qwen3.5-27B, respectively. Each variant uses a separately constructed corpus generated by the same PKC pipeline; trajectories are not shared across model scales. The two variants follow the same optimization recipe, with differences confined to the base checkpoint and distributed sharding configuration. We apply full-parameter fine-tuning (including the multimodal backbone) in bfloat16 with ZeRO-3 and gradient checkpointing. Learning rate: 2×10^{-5} with cosine schedule and 100-step warmup. Batch size: 32 (effective, with gradient accumulation). Max sequence length: 8192 tokens. Training: 3 epochs over the scale-specific trajectory datasets (30K for SearchEyes-9B and 28K for SearchEyes-27B) on 8 H20 GPUs. Observation tokens are masked from the loss (M_{gen} mask).

HaPO Stage. The two model scales share the following HaPO hyperparameters and are trained on 64 H20 GPUs; scale-specific model parallelism is used for parameter placement while preserving the same effective optimization batch. The outcome reward is strictly binary: a final answer receives 1 if it matches any accepted gold alias after lowercasing and normalization of articles, punctuation, and whitespace, and 0 otherwise; no partial-match reward is used. HaPO changes how this outcome is assigned to generated tokens, not the reward scale itself. Gold entity membership is used only by the trainer to form anchor groups. The policy receives the same canonical search-result records as in ordinary interaction and is never told which retrieved QID matches the retained gold path. Anchor groups with fewer than two members or zero outcome variance contribute no hop-level signal; the trajectory-level component remains available through the mixed advantage. Group size: $G = 8$ rollouts per question. Mixing ratio: $\alpha = 0.3$ (favoring hop-anchored signals). KL penalty: $\lambda = 0.02$. Gate temperatures: $\gamma^+ = 1.0$, $\gamma^- = 1.05$ (asymmetric; negative-advantage updates decay faster). Fatal step threshold: $M = 3$ consecutive tool errors. Learning rate: 5×10^{-7} with linear warmup (50 steps) and constant schedule. Training: 200 update steps with online rollouts over the scale-specific RL pools (12K questions for SearchEyes-9B and 15K for SearchEyes-27B). Retrieval top-k: 5 documents per search call.

Self-Contained Search-World Implementation. The RL search world indexes each Wiki6M entity by its title and summary. Sparse retrieval uses BM25 with $k_1 = 1.5$ and $b = 0.75$. Dense text and visual retrieval use Qwen3-VL-Embedding-2B to encode text and images into 2,048-dimensional, ℓ_2-normalized vectors. Entity text and image embeddings are precomputed, stored as float16 NumPy arrays, and memory-mapped at runtime; query embeddings are compared with the corresponding index by normalized dot product (cosine similarity), using chunked NumPy computation or GPU matrix multiplication. For text search, the dense and BM25 candidate lists are fused by reciprocal-rank fusion with $\kappa = 60$, after which the top five entities are returned. Visual search applies the same encoder to the requested image crop and retrieves from the image-embedding index, while `lookup` performs an exact Wikidata-QID lookup. All indices are fixed throughout RL, so identical tool calls receive deterministic results.

Table 4: **SearchEyes tool interface.** Function signatures remain fixed; only the retrieval backend changes between the simulated training world and external-benchmark evaluation.

Capability	Argument(s)	Training backend	Evaluation backend
		Retrieval and evidence access	
Text search	query	Hybrid BM25+dense retrieval over the fixed PKC knowledge base	Serper web search with canonicalized result records
Visual search	image-region reference	Nearest-neighbor retrieval over indexed entity images	Google Lens with the same result-record fields
Source lookup	source ID	Exact access to the corresponding QID entity document	Open the web source returned by retrieval
		Auxiliary analysis	
Summarization	text, query	Query-focused evidence extraction	Same extractor and argument structure
Computation	code	Sandboxed Python for lightweight computation and string processing	Same sandbox and safety restrictions

B.3 Agent System Prompt

The SearchEyes agent uses the same ReAct-style instruction hierarchy during inference and RL rollouts. Function schemas are supplied through the native tool interface summarized in Table 4; the prompt itself specifies the agent's role, search discipline, and termination rule.

Shared Agent Instructions

Role and objective. Act as a visual research assistant. Given an image and a complex multi-hop question, gather sufficient evidence from the available search and analysis tools before answering.

Search policy.

1. Inspect the image and form an initial entity hypothesis.
2. Use visual or text search to ground that hypothesis, then open the returned sources when additional evidence is needed.
3. Follow the relationships in the question step by step, verifying every transition before proceeding.
4. Summarize long evidence or use lightweight computation only when it supports the current reasoning step.
5. Stop only after the complete chain has been traced and the answer is supported by the retrieved observations.

Answer protocol. Return a short, precise entity name enclosed by the literal `<answer>` and `</answer>` delimiters; do not place explanatory prose inside the delimiters.

During RL training, the same instruction text and function signatures are passed to the `verl-agent` multi-turn rollout engine. Calls and observations use its native `<tool_call>` and `<tool_response>` wrappers, while the policy retains the same explicit Thought/Action interaction pattern used at evaluation time.

B.4 Tool Specifications

Table 4 summarizes the fixed agent–tool interface and makes the training-to-evaluation backend substitution explicit. Search and source observations use a canonical record containing a title, a source identifier (an entity ID internally or a URL externally), a text snippet, and an optional addressable image attachment. The visual-search function accepts a region from the input image or from any such attachment already present in the interaction history. Benchmark inference therefore changes the retrieval corpus, but not the action space or observation structure learned during training.

B.5 Official Benchmark Evaluation Protocol

Released Evaluation Sets. We use the fixed evaluation suite distributed with the official VDR-Bench release Zeng et al. (2026): 300 examples from SimpleVQA, the 500-example VDR-Bench `testmini` split, the 171-example visual subset of MMSearch, the 300-example WebWatcher subset of LiveVQA, 199 Level-1 plus 200 Level-2 examples from BrowseComp-VL, and 300 examples from FVQA Cheng et al. (2025); Jiang et al. (2024); Fu et al. (2025); Geng et al. (2025); Wang et al. (2017). The "VDR" column in Table 1 denotes VDR-Bench `testmini`; "BC-VL" denotes BrowseComp-VL.

Official Scoring Rule. We follow VDR-Bench exactly. Each model produces one response per example. We extract the final answer from the last reasoning step and use **Qwen3-VL-30B-A3B-Instruct** as the judge, with the judge-prompt template adopted by VDR-Bench from the Tongyi DeepResearch report Zeng et al. (2026). The judge receives the question, reference answer, and final model response and returns a binary correctness verdict. A correct verdict receives 1; an incorrect, missing, failed, or unparseable response receives 0. Per-benchmark Pass@1 answer accuracy is the mean verdict multiplied by 100.

Aggregation. BrowseComp-VL is reported as the unweighted average of its Level-1 and Level-2 accuracies, following the official convention. The overall Avg. in Table 1 is the unweighted mean of the six benchmark scores and is shown only for models with all six results available.

Our Inference Configuration. We deploy SearchEyes using vLLM with tensor parallelism across 8 H20 GPUs. The agent operates as a multi-turn ReAct loop: at each turn, it generates a thought and a tool call, receives the tool observation, and continues until producing a final `<answer>` tag or reaching the 50-turn limit. Inference uses temperature $T{=}0.6$ and top-$p{=}0.95$. Unlike the self-contained search world used for training, benchmark inference uses external APIs: `text_search` uses the Serper web search API, and `visual_search` uses Google Lens. These rollout choices define our agent configuration but do not alter the official VDR-Bench scoring rule above.

Result Sources. All values in Table 1 are reported under the official VDR-Bench benchmark setting: the released evaluation subsets, Pass@1 convention, and benchmark-level aggregation described above. We run SearchEyes and the matched Qwen3.5 agentic bases with the official evaluator. Results for previously published systems are the values reported in their original papers and VDR-Bench result tables, whose authors likewise identify them as following the official VDR-Bench setting Chen et al. (2026a); Huang et al. (2026a); Li et al. (2026a); we do not reimplement or rescore those systems. Unavailable entries are marked with –. Thus, Table 1 compares systems under the same released benchmark and reporting standard while retaining each published baseline's original inference implementation.

C Additional Ablations and Analysis

C.1 Clarifying the Main-text Ablations

This subsection defines the controls for the two main-text ablations. Tables 2 and 3 use the full SearchEyes-9B configuration: Qwen3.5-9B initialization, 10K SFT questions yielding 30K accepted trajectories, a 12K-question RL pool, $G{=}8$ rollouts, and 200 online updates. Every row is evaluated once on the same fixed SimpleVQA, VDR, MMSearch, and FVQA subsets with the common external-tool agent and official scoring protocol in Appendix B.5.

PKC controls and interpretation. The PKC rows hold model initialization, accepted-data counts, SFT schedule, RL pool size, rollout decoding, HaPO hyperparameters, and tool interface fixed. For each structural row, we regenerate a size-matched 10K/12K question pair and 30K accepted SFT trajectories after removing only the named constraint; we do not obtain a row by deleting examples from the full-PKC pool. Removing multi-P interleaving permits chains with only the anchor P; removing the disambiguation edge restores a linear answer path; removing domain balancing permits semantically repetitive chains. The anti-shortcut row bypasses both shorter-path and restricted-answerability checks, while the cross-modal row drops visual-grounding validation at designated P positions. Information concealment is removed only during question realization. The $\beta{=}1$ row instead keeps the full questions and changes only teacher retrieval during trajectory synthesis; replay, SFT, RL, and the student interface remain unchanged. The three synthesis baselines are in-

Table 5: **Effect of the RL environment and training stages.** The 1K local and live rows change only the rollout backend and use the same questions and optimization budget; the final row reports the complete local-world HaPO run.

Training stage	VDR	MMS	FVQA	Avg.
Qwen3.5-9B agentic base	19.3	41.6	63.2	41.4
+ PKC SFT only	24.7	58.2	69.8	50.9
+ 1K live-world RL	26.1	63.8	71.6	53.8
+ 1K local-world RL	26.6	65.1	72.3	54.7
+ Full local-world HaPO	**28.3**	**69.2**	**74.2**	**57.2**

dependently constructed but matched to PKC in accepted question and trajectory counts. Because structural variants require regenerated corpora, these rows are matched system-level removal tests rather than paired counterfactuals for individual questions. We therefore attribute the anti-shortcut drop to the complete filter bundle and do not assign it to any single constituent check.

Interpreting multi-P. Removing multi-P lowers SimpleVQA, VDR, MMSearch, and FVQA by 3.3, 5.7, 4.4, and 3.4 points, respectively; the larger reductions on VDR and MMSearch are consistent with their heavier multi-step retrieval demands. A P label specifies the availability of image-grounded evidence during synthesis, not a mandatory visual-search action at the corresponding graph position. The anchor P is supplied by the input image; a downstream P becomes visible only after its image-backed entity record is retrieved or opened, after which the policy may inspect that attachment, issue another visual search, or proceed with sufficient textual evidence. Thus, the ablation measures the value of downstream image-backed checkpoints rather than enforcing a fixed number or order of visual calls. Appendix D shows the latter case: downstream evidence is sufficient after textual retrieval, so the successful policy need not mirror every synthesis label with a separate invocation.

HaPO controls and interpretation. All HaPO rows start from the identical full-PKC SFT checkpoint and use the same ordered 12K RL pool, local search-world snapshot, $G{=}8$, 200 updates, binary reward, rollout decoding, learning rate, and tool interface. GRPO, GiGPO, and DAPO replace only the optimization algorithm; shared training and evaluation settings are unchanged. The turn-index control forms groups at the same numbered interaction step, with the same minimum-size and non-zero-variance rules as entity-anchor groups. Episode-only and hop-only set α to 1 and 0. Removing smooth gating uses the hard-clipped surrogate while retaining the combined advantage; removing fatal masking leaves the detected suffix in the loss, whereas removing clamping still masks that suffix but preserves signed prefix advantages. Removing observation masking includes observation tokens in the loss without changing the context shown to the policy. These controls isolate semantic grouping from the accompanying long-horizon optimization components.

Common protocol for the 1K diagnostics. Sections C.2–C.4 are separate matched diagnostics. Unless a subsection states otherwise, every row starts from the same full-PKC SearchEyes-9B SFT checkpoint, uses one fixed 1K-question subset of the 12K RL pool, samples eight rollouts per question, and follows the same 200-update HaPO schedule and hyperparameters as the full run. The named factor is the only intervention within a table. All accuracy columns use the same fixed VDR, MMSearch, and FVQA subsets and the official scoring protocol.

C.2 Simulated versus Live Search Worlds

Following SearchGym's separation of simulated, local, and live search environments Zhang et al. (2026b), we compare a fixed 1K-question continuation using either SearchEyes' local world or live external APIs. Question order, rollout sampling, optimizer, and update count are identical; only the retrieval backend changes.

Across the three benchmarks, SFT raises the base average from 41.4 to 50.9, about 60% of the full 15.8-point gain; full HaPO supplies the remaining 6.3 points. Under the matched 1K continuation, local-world RL exceeds live-world RL by 0.9 points without extra questions or updates. Headline evaluation still uses external search.

Table 6: **Retrieval fidelity and reproducibility.** Results use the same balanced set of 300 held-out PKC requests (100 each at 3, 4, and 5 hops) in the local and live environments.

Environment	Text R@5	Visual R@5	Coverage	Repeat@5	Reward agree.
Live web APIs	81.2	76.5	84.7	79.8	88.5
SearchEyes local world	**88.6**	**84.3**	**92.1**	**100.0**	**100.0**

Table 7: **Retrieval fidelity by path length.** Each stratum contains 100 held-out PKC requests. Coverage requires all main-chain entities to be retrieved before answering.

	Live web APIs			SearchEyes local world		
Hops	Text R@5	Visual R@5	Coverage	Text R@5	Visual R@5	Coverage
3	84.6	80.2	88.9	**91.2**	**87.5**	**94.8**
4	81.1	76.8	84.4	**88.9**	**84.6**	**92.0**
5	77.9	72.5	80.8	**85.7**	**80.8**	**89.5**

Table 8: **Per-rollout interaction and cost profile.** Latency is the mean tool-response latency; cost counts external search APIs only.

Environment	Turns	Text	Visual	Lookup	Latency	API cost
Live web APIs	24.1	12.0	4.0	6.0	2.10s	$0.072
SearchEyes local world	20.8	10.2	3.5	5.2	0.18s	$0.000

On held-out PKC tasks with known next entities, text/visual Recall@5 measures whether the corresponding call returns the gold entity in its top five. Coverage requires every main-chain entity before answering; Repeat@5 is top-five Jaccard agreement over five identical requests; reward agreement is the fraction of replays with unchanged binary outcome. Fixed local indices are deterministic, whereas live results may vary with indexing, ranking, availability, and time.

The local index improves text Recall@5, visual Recall@5, and coverage by 7.4, 7.8, and 7.4 points. Its identical repeated calls and replays also remove the ranking and reward variation observed live, making updates reproducible from the stored questions, checkpoint, and index snapshot.

Depth boundary. We further stratify the same held-out requests by gold path length. Table 7 shows that retrieval becomes harder for both backends as the chain grows, while the local world retains higher recall and complete-chain coverage at every depth.

From three to five hops, coverage falls from 88.9 to 80.8 live and from 94.8 to 89.5 locally. At five hops, the local backend retains gains of 7.8 points in text Recall@5, 8.3 in visual Recall@5, and 8.7 in coverage. Thus chain depth remains a measured boundary: deterministic retrieval reduces, but does not eliminate, compounding loss. Table 5 shows that the advantage transfers to accuracy under the same 1K update budget.

Cost is reported per rollout, not multiplied by the eight RL samples. A representative live rollout makes 12 text and 4 visual searches; at $\$0.001$ and $\$0.015$ per call, its external-search cost is $\$0.072$. Source fetches and local computation are unbilled; the local world has no marginal API charge after index construction.

The local world reduces mean tool latency from 2.10s to 0.18s and removes the $\$0.072$ marginal charge per rollout; its shorter trajectories also show fewer repeated reformulations. These figures exclude shared policy inference and one-time index construction/storage, so they quantify marginal interaction cost rather than a complete training bill.

C.3 Tool Selection

We evaluate the five SearchEyes tools with matched leave-one-tool-out continuations. Every row starts from the same five-tool SFT checkpoint; during both the 1K RL continuation and evaluation, the prompt and call validator expose exactly the tool schema named in that row. Thus, these

Table 9: **Tool-selection ablation.** All rows use a matched 1K RL budget. Coverage is complete main-chain evidence coverage; Calls is the mean number of tool calls per rollout; Speed is rollout throughput relative to the five-tool interface.

Tool configuration	VDR	MMS	FVQA	Avg.	Cov.	Calls	Speed
Full five-tool interface	**26.6**	**65.1**	**72.3**	**54.7**	91.4	19.1	1.00×
w/o text search	21.7	53.8	63.1	46.2	70.8	14.7	1.07×
w/o visual search	20.9	56.7	65.4	47.7	73.0	15.8	1.05×
w/o source lookup	24.8	61.2	69.3	51.8	82.6	15.1	1.08×
w/o summarization	25.9	63.8	71.1	53.6	87.9	18.2	1.02×
w/o computation	26.3	64.7	72.0	54.3	90.6	19.0	1.00×
+ OCR / layout parsing	26.5	64.8	72.1	54.5	**91.6**	20.6	0.91×
+ all extra visual tools	25.8	63.5	72.1	53.8	90.5	23.8	0.76×

runs isolate tool availability after SFT rather than retraining a separate SFT checkpoint for each inventory. OpenSearch-VL Chen et al. (2026a) additionally exposes OCR/layout parsing, perspective correction, super-resolution, and sharpening. Its explicit crop action is already covered by the region argument of our visual search. We test OCR/layout parsing alone and the complete extra visual-preprocessing suite under the same rollout and update budget.

The leave-one-out comparisons separate necessary retrieval capabilities from auxiliary analysis. Removing text search lowers average accuracy by 8.5 points and coverage by 20.6 points; removing visual search lowers them by 7.0 and 18.4 points. Source lookup has a smaller but still substantial effect, while summarization and computation contribute 1.1 and 0.4 average points. Adding OCR changes accuracy by only -0.2 while adding 1.5 calls and reducing throughput by 9%.

C.4 Training-path Length

We also vary the hop composition of the fixed-size 1K continuation pool while keeping the SFT checkpoint and optimization budget unchanged. The single-length rows contain only the named value of K; the mixed row draws equal numbers of 3-, 4-, and 5-hop questions (with the remainder assigned uniformly). Three-hop paths are easier but undertrain long-horizon planning, whereas five-hop-only paths increase interaction difficulty and slightly reduce FVQA accuracy. The 3–5-hop mixture is used by SearchEyes.

Table 10: **Hop-length ablation.** All settings use 1K PKC questions and the same SearchEyes-9B optimization budget.

Training paths	VDR	MMS	FVQA	Avg.
3-hop only	25.2	62.1	71.9	53.1
4-hop only	26.0	63.8	72.0	53.9
5-hop only	25.5	64.2	71.1	53.6
Mixed 3–5 hops	**26.6**	**65.1**	**72.3**	**54.7**

The mixed composition improves the average by 0.8 points over the strongest single-length condition. Three-hop-only training is strongest on FVQA but undertrains longer evidence chains, while five-hop-only training improves MMSearch relative to shorter paths at the cost of lower VDR and FVQA accuracy. Because question count, update count, and the SFT checkpoint are fixed, this diagnostic isolates the distribution of path lengths in the continuation pool; it does not introduce a staged curriculum. Mixing three to five hops retains easier trajectories while exposing the policy to longer planning horizons.

D Case Study: Multi-hop Agent Trajectory

We present a representative successful SearchEyes trajectory grounded in a valid three-hop PKC chain with an additional disambiguation constraint. Each box shows the agent's reasoning

(*Thought*), tool invocation (*Action*), and retrieved evidence (*Observation*). For readability, observations are abridged to the ranked results used by the subsequent reasoning; entity IDs retain the format exposed by the search world. Rollout tool calls need not correspond one-for-one to graph hops.

Phase I: Visual grounding and bridge discovery.

Case Study: Four-step Perception–Knowledge Rollout

Question: *The image shows a titanium-clad museum beside a river. Its architect received a major international architecture prize. Among the other recipients of that prize, identify the architect whose notable works include an opera house in the southern Chinese city historically known as Canton.*
Gold Answer: Zaha Hadid **Rollout Steps:** 4
Gold Entity Sequence: Guggenheim Museum Bilbao (`Q179199`) → Frank Gehry (`Q180374`) → Pritzker Architecture Prize (`Q133160`) → Zaha Hadid (`Q47780`)
Constraint: Zaha Hadid $\xrightarrow{\text{notable work}}$ Guangzhou Opera House (`Q3354553`)

Step 1: Visual Identification

Thought: The curved titanium facade suggests a Guggenheim museum, but I need to identify the specific building before following the architect clue.
Action: `visual_search(region="center")`
Observation: `[1] Guggenheim Museum Bilbao (ID: Q179199)` — museum of modern and contemporary art in Bilbao, designed by Frank Gehry. `[2] Solomon R. Guggenheim Museum (ID: Q201469)` — art museum in New York, designed by Frank Lloyd Wright.

Step 2: Knowledge Retrieval

Thought: The top result matches the titanium building and river setting. I will verify its architect using the identified museum name.
Action: `text_search(query="Guggenheim Museum Bilbao architect")`
Observation: `[1] Guggenheim Museum Bilbao (ID: Q179199)` — the museum building was designed by Canadian-American architect Frank Gehry. `[2] Frank Gehry (ID: Q180374)` — architect whose works include the Guggenheim Museum Bilbao.

Phase II: Prize and constraint resolution.

Step 3: Prize Identification

Thought: The architect is Frank Gehry. I now need the major international architecture prize connecting him to the requested recipient.
Action: `text_search(query="Frank Gehry major international architecture prize")`
Observation: `[1] Frank Gehry (ID: Q180374)` — received the Pritzker Architecture Prize in 1989. `[2] Pritzker Architecture Prize (ID: Q133160)` — international architecture prize with multiple architect laureates.

Step 4: Constraint Resolution

Thought: The shared prize is the Pritzker. The answer is another laureate, uniquely constrained by an opera house in the Chinese city historically called Canton.
Action: `text_search(query="Pritzker recipient architect opera house Canton")`
Observation: `[1] Zaha Hadid (ID: Q47780)` — Pritzker laureate whose notable works include Guangzhou Opera House. `[2] Guangzhou Opera House (ID: Q3354553)` — opera house in Guangzhou, China, designed by Zaha Hadid.

Final Answer: Zaha Hadid ✓

Analysis. The trajectory grounds the museum visually, retrieves its architect and prize, then uses the concealed opera-house clue to select Zaha Hadid. The four main-chain QIDs appear in order, while the separate constraint entity appears only in the final observation, matching the evidence structure used to form hop anchors.