SEARCH ENVIRONMENT AGENT FOR EFFICIENT DEEP SEARCH REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training deep search agents with reinforcement learning (RL) on the real web is costly, motivating cheaper simulated environments. Yet we observe *reward–transfer decoupling*: training reward in the local environment improves without corresponding real-web gains. Even factually correct feedback can omit evidence obtainable by effective queries or over-assist underspecified ones, distorting which search behaviors RL reinforces. We introduce **Search Environment Agent (SEA)**, a memory-augmented environment manager that calibrates action-dependent access to evidence. Guided by *Oracle Memory*, its internal record of real-tool experience, SEA selects and reranks locally supported evidence and reserves real execution for predicted decision-relevant discrepancies that cannot be repaired locally. Paired local and real feedback supervise response selection and routing, with counterfactual policy continuations informing routing labels. Verified real observations update the memory and webpage index. Evaluated exclusively with real-web tools without SEA or Oracle Memory, SEA-RL-35B surpasses reported specialized-agent baselines across six benchmarks. In controlled 4B comparisons, SEA outperforms Cached-Web RL on all four benchmarks and Real-Web RL on three, using approximately 4.8% of the latter's measured training API expenditure. Diagnostics further show improved evidence recall alongside reduced over-assistance.

1 INTRODUCTION

Reinforcement learning (RL) enables deep search agents to learn evidence acquisition through repeated search, browsing, and reasoning. Scaling this training on the live web, however, requires costly tool calls and lengthy interactions with external services. Local retrieval and simulated tools offer a less expensive source of experience (Jin et al., 2025). Their value ultimately depends on what that experience teaches: cheaper rollouts are useful only insofar as they produce better search behavior at deployment. The goal is therefore not to minimize interaction cost in isolation, but to maximize real-web capability under a limited training budget.

Existing approaches make substantial progress toward affordable search training. Search-R1 (Jin et al., 2025) uses local Wikipedia retrieval, while LiteResearcher (Li et al., 2026) provides search and browsing over cached webpages. SearchGym (Zhang et al., 2026) constructs grounded, solvable tasks with aligned document corpora, and ZeroSearch (Sun et al., 2025) learns a generative search simulator with a document-quality curriculum. Yet making the right evidence available is different from making it available in response to the right action. A training environment must also preserve how the policy's search choices affect its access to evidence.

We observe *reward–transfer decoupling*: training reward in the local environment continues to improve without corresponding gains in real-web performance. Our analysis identifies a mechanism beyond factual errors and missing webpages. A simulated environment can *under-support* an effective query by withholding evidence that real search would retrieve. Conversely, it can *over-assist* an underspecified query by supplying decisive evidence recovered from a past successful interaction. Even when every returned fact is correct, these errors can erase or reverse differences in evidence acquisition between actions. RL may then discourage productive search strategies or reinforce queries that succeed only because the simulator compensates for their deficiencies. A simulator can thus be factually correct while teaching the wrong search behavior.

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.

The modeling objective is to preserve these differences in action effectiveness, not to reproduce every detail of a live-web response. Value-aware model learning and value equivalence formalize a related principle: model accuracy should reflect the information needed for downstream value computation, rather than indiscriminate reconstruction of the environment (Farahmand et al., 2017; Grimm et al., 2020). For deep search, we focus on feedback errors that could change which behaviors RL reinforces. This also determines where real interaction is worth its cost: discrepancies that alter evidence acquisition or subsequent search decisions warrant correction; harmless differences in presentation need not.

In this paper, we introduce **Search Environment Agent (SEA)**, a learnable, memory-augmented environment manager for low-cost deep search RL. SEA maintains *Oracle Memory*, an internal record of prior actions and their verified real-tool responses, to inform both environment routing and local feedback calibration. For each policy action, SEA uses the interaction history and relevant memory records to assess whether candidates from a local webpage index, supplemented by action-compatible memory evidence, can support action-consistent feedback. Crucially, SEA accounts for possible local correction before deciding to escalate: result reranking and source-excerpt selection can surface supported clues and remove unwarranted evidence. SEA reserves real execution for predicted decision-relevant discrepancies that cannot be resolved locally; otherwise, it constructs a grounded response through evidence selection and rendering. The resulting real observations support the current rollout and update Oracle Memory, while newly acquired webpage content expands the index. By calibrating what each action reveals, SEA aims to reduce feedback errors that reinforce ineffective queries or discourage productive search behavior.

We fine-tune SEA's shared parameters on routing and response-selection tasks using paired local and real feedback for the same history and action. The supervision targets whether a feedback discrepancy changes subsequent evidence acquisition and whether it can be repaired locally, rather than whether the two responses are textually similar. For routing annotation, we compare policy continuations under local and real feedback, using the same policy checkpoint and real tools for all subsequent calls in both branches. Real execution is labeled necessary only when a decision-relevant mismatch cannot be corrected using the local evidence frozen before the real call. Paired real responses additionally supervise the selection and ordering of locally supported search results, without supplying new content to the local target. The paired real references and continuations are used only for annotation; SEA conditions on the history, action, and pre-existing local evidence. SEA is trained before policy RL and remains parameter-frozen during RL, while verified real interactions continue to update its memory and the local index.

We evaluate SEA-trained policies at the 4B and 35B scales on BrowseComp (Wei et al., 2025), DeepSearchQA (Gupta et al., 2026), xBench-DeepSearch (Chen et al., 2025), GAIA (Mialon et al., 2023), WideSearch-EN (Wong et al., 2025), and AutoResearchBench-Deep (Xiong et al., 2026), using real-web tools exclusively and without SEA or Oracle Memory at evaluation. SEA-RL-35B surpasses the reported specialized-agent baselines across all six benchmarks, reaching 74.0 on BrowseComp and 80.6 on DeepSearchQA. In controlled 4B comparisons, SEA outperforms Cached-Web RL on all four evaluated benchmarks and achieves higher scores than Real-Web RL on three, while reducing measured training API expenditure by approximately 95%. Further analyses show improved evidence recall alongside reduced over-assistance, as well as better alignment between training reward and real-web performance over the observed training horizon. Together, these results support selective feedback calibration as a cost-effective approach to learning transferable search behavior.

Our contributions are threefold: (1) We characterize *reward–transfer decoupling* in deep search RL, showing how under-support and over-assistance can misrepresent action effectiveness despite factually correct feedback. (2) We introduce **SEA**, a memory-augmented environment manager that learns action-compatible evidence reuse and local feedback calibration from paired local–real feedback, reserving real execution for predicted decision-relevant discrepancies that cannot be corrected locally. (3) We demonstrate improved real-web transfer at substantially lower training API cost, with feedback diagnostics and component ablations supporting the complementary roles of selective execution and local calibration.

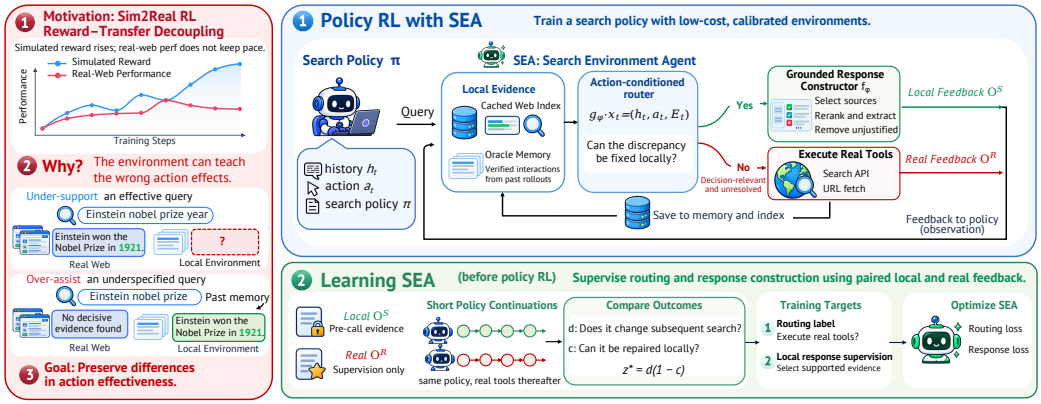


Figure 1: **Overview of Search Environment Agent (SEA). Left:** Under-support and over-assistance can decouple training reward from real-web performance. **Top right:** SEA constructs action-consistent feedback from cached webpages and Oracle Memory, invoking real tools only for predicted decision-relevant discrepancies that cannot be corrected locally. **Bottom right:** Paired local and real responses supervise routing through counterfactual continuations and response selection over pre-existing local evidence.

2 METHOD

In this section, we present Search Environment Agent (SEA), a memory-augmented environment manager trained to calibrate decision-relevant discrepancies between local and real feedback, reserving real execution for mismatches that cannot be corrected with available evidence.

2.1 PROBLEM SETUP

We study *deep search*, where an agent answers complex questions through long-horizon web search, webpage browsing, and evidence synthesis across multiple sources. The policy we train acts as a deep search agent, deciding what to search, which pages to inspect, and when to answer. Training directly on the live web requires repeated Google Search API calls and URL fetching across RL rollouts, incurring substantial API expenditure and interaction latency. Given a question q, the policy π_θ conditions on an interaction history h_t, which contains the question and preceding reasoning, tool actions, and observations. At each step, the policy either produces a final answer or issues a tool action a_t. We consider two tool types: `search`, which returns ranked URLs with titles and snippets, and `visit`, which returns content from a specified URL. Executing an action in the real-web environment produces an observation $o_t^{\mathrm{R}} \sim \mathcal{E}_{\mathrm{R}}(\cdot \mid h_t, a_t)$.

During policy training, SEA mediates tool execution, returning either a locally constructed observation or a response obtained through real execution. The policy remains responsible for reasoning, generating its actions and producing the final answer. At evaluation, the trained policy interacts directly with the real-web environment, without SEA or Oracle Memory. Our objective is to maximize the trained policy's real-web performance under a limited training budget.

Achieving this objective requires more than factually correct feedback: the training environment must not misrepresent how the policy's actions affect its access to evidence. On the one hand, omitting clues that an effective query would retrieve can make productive search behavior appear unsuccessful. On the other hand, supplying decisive evidence that the current action would not reasonably obtain can create spurious success. Both errors can distort the feedback from which the policy learns to search. We therefore focus on discrepancies that can alter evidence acquisition or subsequent search decisions, rather than requiring an exact reproduction of real-web responses.

2.2 SEARCH ENVIRONMENT AGENT FOR SELECTIVE ENVIRONMENT CALIBRATION

As shown in Figure 1, SEA manages the training environment through memory-informed routing and grounded response construction. It maintains Oracle Memory as an internal record of real-tool experience, using retrieved records to assess action–feedback compatibility and, when applicable, to supply reusable evidence. Conditioned on the current history, action, and local evidence, SEA

predicts whether real execution is necessary after accounting for possible local correction. When local execution suffices, it constructs a response through source selection, reranking, and excerpt extraction.

Local evidence. SEA consults *Oracle Memory* to interpret and complement candidates retrieved from a local webpage index. The index contains approximately 2.79 million cached pages and supports query search and URL lookup. We use BGE-M3 (Chen et al., 2023) as search backend for local webpage index. Memory stores verified real tool responses from prior deep search policy SFT and RL trajectories, together with their actions, contexts, and sources. Retrieval uses the current action and search history. The local webpage index supplies candidate pages, while memory records what similar actions retrieved. Before reusing a memory response, SEA checks whether its tool type, query constraints, and source are compatible with the current action. Relevance to the question and the policy action alone is not sufficient for reuse.

Selective execution. Let E_t denote the retrieved evidence and $x_t = (h_t, a_t, E_t)$. SEA predicts a binary route $z_t = g_\phi(x_t)$, where 0 denotes local execution and 1 denotes real execution. For the local route, f_ϕ selects sources and excerpts to construct the response:

$$o_t = \begin{cases} \mathrm{Render}(f_\phi(x_t); E_t), & z_t = 0, \\ o_t^{\mathrm{R}}, & z_t = 1. \end{cases} \tag{1}$$

The two functions share parameters ϕ. Routing precedes real execution: SEA predicts whether local correction is sufficient using only x_t.

Grounded response construction. For `search`, SEA combines the top-20 local candidates with applicable memory evidence, selects and orders the final top-10 results, and chooses their source excerpts. For `visit`, it selects content associated with the requested URL. SEA outputs source identifiers and text spans; a program copies the selected content into the real tool's response format. This restricts calibration to evidence selection rather than generation of unsupported content.

When SEA chooses real execution, the observation from the real-web environment is returned to the policy. Verified responses enter Oracle Memory, and newly obtained webpage content expands the index. Each real interaction can thus support the current rollout and supply evidence for future local responses. Implementation details of the local index and Oracle Memory, including retrieval, action-compatibility checks, update schedules, and data-contamination safeguards, are provided in Appendix F.

2.3 Learning from Paired Feedback

We use Qwen3.5-35B-A3B (Yang et al., 2025) to initialize SEA. SEA is trained with two complementary SFT tasks: deciding when local evidence is insufficient and real execution is needed, and constructing calibrated responses when local execution suffices. Both tasks use paired local and real feedback for the same history and action, targeting differences that affect evidence acquisition rather than surface-form similarity.

Single-step feedback probes for annotation. We prioritize actions that reveal key evidence or resolve subproblems, supplemented with early and intermediate exploration and actions that yield no new clue on the real web. For each history–action pair (h, a), we freeze the available local evidence E before obtaining the paired real response, excluding subsequent observations from the source trajectory. We construct initial local feedback o^{S} from E and collect real feedback o^{R} for the same action.

The SFT deep search policy continues independently from each observation under matched decoding settings and continuation budgets. All subsequent tool calls use the real web, and subsequent actions are generated anew rather than replayed. The intervention is therefore confined to the current feedback source, avoiding additional discrepancies from later local responses. These continuations provide behavioral evidence for routing annotation; they do not evaluate sustained interaction with the local environment. The SEA training data annotation details are provided in Appendix G.1.

Routing supervision. The router learns whether real execution is necessary *after accounting for possible local correction*. An LLM judge identifies omitted clues, factual errors, and extra evidence not justified by the current action, grounding each diagnosis in source passages. A mismatch is decision-relevant when it materially changes the evidential basis for acquiring further evidence, ruling out hypotheses, or ending search. The judge uses the continuations to assess whether the discrepancy obstructs evidence gathering, induces avoidable tool calls, or leads to premature answering, rather than merely producing an alternative valid search path. Continuation outcomes inform this assessment but do not determine the label on their own: the annotator must also establish a source-supported discrepancy and assess its local repairability.

Let $d \in \{0, 1\}$ indicate a decision-relevant mismatch and $c \in \{0, 1\}$ indicate whether it can be corrected using only E through reranking, excerpt selection, or removal. Any correction must remain consistent with the current action and be supported by identifiable local sources. The routing target is

$$z^\star = d(1 - c). \tag{2}$$

Thus, both adequate feedback and locally repairable mismatches receive the local label ($z^\star = 0$). Real execution ($z^\star = 1$) is reserved for decision-relevant mismatches that the available evidence cannot resolve.

Response supervision. For the local route, SEA must also learn how to reduce harmful omissions and unwarranted evidence exposure. For each selected `search` action, we combine the top-20 local candidates with action-compatible Oracle Memory evidence. Using the paired real response o^{R} as a reference, an LLM annotator produces an ordered list of candidate identifiers specifying which results to retain in the final top-10 response. These targets teach SEA to surface relevant clues omitted from the initial local response, reorder retained results, and remove unjustified additions. Supervision is restricted to evidence already available in E: the real response guides selection but does not supply new content to the local target. Decision-relevant omissions that cannot be repaired from E are instead handled by the routing objective.

This supervision also governs Oracle Memory reuse. A historical real-tool response is not automatically appropriate feedback for the current action. The targets train SEA to retain action-supported memory evidence and suppress unwarranted clues that would compensate for missing query constraints. Thus, SEA learns how to expose historical evidence, rather than treating all retrieved memories as valid feedback.

We construct $3{,}000$ routing examples and $2{,}000$ response-selection examples, and fine-tune the shared SEA parameters ϕ with task-specific instructions. Both tasks condition only on $x = (h, a, E)$; paired real responses and counterfactual continuations are used for annotation, never as SEA inputs. SEA supervision is drawn from the training question pool, which is also used for policy SFT and RL. Paired real responses and annotation continuations are never included in SEA's inputs. The SEA training details are provided in Appendix G.

2.4 Policy Training and Evaluation

We first initialize the policy with SFT and train SEA as described above. We then optimize the policy with Group Relative Policy Optimization (GRPO) (Shao et al., 2024), sampling eight trajectories per question through the environment in equation 1. An LLM (Qwen3.5-122B-A3B) judge assigns a binary terminal reward for final-answer correctness. The loss applies only to policy-generated tokens, with tool observations serving as context. SEA's parameters remain fixed during policy RL, while verified real interactions continue to update the index and memory. At evaluation, the policy uses real-web tools exclusively, without SEA or Oracle Memory. Detailed SFT and RL training settings are described in Appendix E.

Table 1: Performance comparison on deep search and deep research benchmarks.

Model	BrowseComp	GAIA	DSQA	xBench	WideSearch-EN	AR-Deep
Research agents						
LiteResearcher-4B	27.5	71.3	17.2	12.0	20.3	4.1
AgentCPM-Explore-4B	24.1	63.9	32.8	34.0	5.1	0.0
Tongyi-DR-30B	43.4	70.9	55.0	40.0	57.8	6.1
Quest-35B	64.6	80.8	59.2	52.0	60.6	4.7
MiroThinker-1.7-mini	67.9	80.3	67.9	57.2	69.4	10.1
General models						
Qwen3.5-4B	19.0	56.3	49.9	39.0	56.0	0.0
Qwen3.5-35B-A3B	31.0	77.5	54.5	45.0	59.7	6.2
Qwen3.5-122B-A10B	37.0	81.6	67.7	51.6	69.6	9.7
DeepSeek-V4-Flash	73.2	83.2	90.6	69.0	76.4	24.8
Kimi-K2.6	83.2	80.6	90.4	60.0	80.8	30.5
GPT-5.4	82.7	78.6	83.6	88.5	77.5	35.7
Ours						
SEA-RL-4B	52.6	78.4	67.4	51.0	68.8	14.3
SEA-RL-35B	74.0	85.4	80.6	60.0	73.8	39.0

3 Experiments

3.1 Experimental Setup

Policy Training Configs. We adopt Qwen3.5-4B and Qwen3.5-35B-A3B (Yang et al., 2025) as the backbone search policy for SEA-RL-4B and SEA-RL-35B. We use the SEA-managed training environment to conduct RL training on the search policy, as described in Section 2 and Appendix E.

Benchmarks. We evaluate trained policy on six benchmarks covering complementary regimes of search-augmented reasoning. **BrowseComp** (Wei et al., 2025) and **DeepSearchQA** (Gupta et al., 2026) (DSQA) emphasize deep research, requiring long-horizon and persistent deep search, evidence aggregation, and answer synthesis. **GAIA** (Mialon et al., 2023) and **xbench-DeepSearch-2510** (Chen et al., 2025) (xBench) further stress agentic and general information seeking. **WideSearch** (Wong et al., 2025) evaluates broad coverage research and synthesis over a large search space, for which we report results on the English subset. **AutoResearchBench** (Xiong et al., 2026) evaluates scientific paper research capability. We use the deep research subset for evaluation (AR-Deep). Following the standard evaluation protocols, we report accuracy for BrowseComp, GAIA, AutoResearchBench-Deep, and xbench-DeepSearch-2510, and F1 for DeepSearchQA, Item-F1 for WideSearch-EN.

Baselines. We compare trained policy against both specialized deep-research agents and strong general-purpose models, including LiteResearcher (Li et al., 2026), AgentCPM-Explore (Chen et al., 2026), Tongyi-DeepResearch (Team et al., 2025), Quest (Xie et al., 2026), MiroThinker (Team et al., 2026), Qwen3.5 (Yang et al., 2025), DeepSeek-V4-Flash (Xu et al., 2026), Kimi-K2.6 (Moonshot AI, 2026), and GPT-5.4 (OpenAI, 2026). The detailed evaluation protocol is presented in Appendix C.

3.2 Main Results

Table 1 shows that SEA yields strong end-to-end improvements on real-web deep search benchmarks. With the same 35B backbone scale, SEA-RL-35B improves substantially over Qwen3.5-35B-A3B on BrowseComp, from 31.0 to 74.0, and on DeepSearchQA, from 54.5 to 80.6. It also outperforms the evaluated open-source research agents on these two representative deep-search bench-

Table 2: Comparison of search-policy RL training environments at the 4B and 35B-A3B scales. All policies are evaluated with real-web tools across 3 independent runs (mean$_{\pm\text{std}}$).

	Training Cost		Evaluation Perf.				Evaluation Stat.	
Training Setting	**API ($)**	**Step Time (min)**	**BC**	**GAIA**	**DSQA**	**AR-Deep**	**Avg. Turn**	**Avg. Tok.**
4B Policy								
SFT	–	–	$34.3_{\pm1.2}$	$72.8_{\pm1.5}$	$64.1_{\pm0.9}$	$7.4_{\pm0.5}$	24.5	84,167
Cached-Web RL	0.0	28.2	$47.3_{\pm1.4}$	$69.6_{\pm1.6}$	$63.7_{\pm1.1}$	$8.1_{\pm0.6}$	31.9	111,921
Real-Web RL	675.8	39.7	$51.7_{\pm1.1}$	$\mathbf{80.6}_{\pm1.3}$	$66.2_{\pm0.8}$	$10.1_{\pm0.7}$	29.4	103,486
SEA RL	32.1	32.9	$\mathbf{52.6}_{\pm0.8}$	$78.4_{\pm1.1}$	$\mathbf{67.4}_{\pm0.6}$	$\mathbf{14.3}_{\pm0.8}$	29.0	98,251
A3B Policy								
SFT	–	–	$49.2_{\pm1.1}$	$82.2_{\pm1.3}$	$70.5_{\pm0.8}$	$15.7_{\pm0.7}$	21.3	67,334
Cached-Web RL	0.0	37.5	$58.5_{\pm1.3}$	$83.4_{\pm1.5}$	$74.1_{\pm1.0}$	$20.4_{\pm0.9}$	27.4	100,490
Real-Web RL	567.7	50.2	$72.1_{\pm1.0}$	$85.2_{\pm1.0}$	$\mathbf{81.7}_{\pm0.7}$	$36.9_{\pm1.1}$	23.6	83,824
SEA RL	26.2	42.1	$\mathbf{74.0}_{\pm0.7}$	$\mathbf{85.4}_{\pm0.8}$	$80.6_{\pm0.6}$	$\mathbf{39.0}_{\pm0.8}$	22.6	73,688

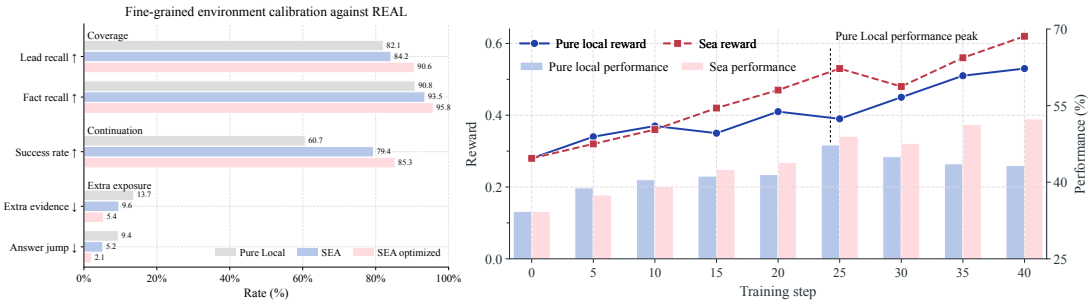


Figure 2: Environment calibration and reward–transfer decoupling. Left: Fine-grained feedback calibration across Pure Local, SEA (before optimization), and SEA environments. Right: Training reward and real-web BrowseComp accuracy across GRPO updates

marks, surpassing MiroThinker-1.7-mini on BrowseComp by 6.1 points and on DeepSearchQA by 12.7 points.

Notably, the advantage is not limited to the larger model. SEA-RL-4B already achieves 52.6 on BrowseComp and 67.4 on DeepSearchQA, outperforming or matching much larger general models on several search-intensive settings. For example, SEA-RL-4B substantially exceeds Qwen3.5-122B-A10B on BrowseComp, and nearly matches it on DeepSearchQA. This suggests that the main benefit of SEA does not come merely from backbone scaling, but from exposing the policy to training feedback that better reflects how search actions affect access to evidence in the real web.

Together, these results establish strong real-web performance at both policy scales. Since evaluation uses real-web tools exclusively, the improvements do not rely on SEA or Oracle Memory at inference time. We next use controlled training-environment comparisons and feedback diagnostics to examine the contribution of environment management and its relationship to real-web transfer.

3.3 Training Environments and Efficiency Analysis

We investigate how different RL training environments affect real-web transfer performance. All policies are initialized from the same SFT model and trained with identical RL optimization settings. The only difference is the environment providing tool feedback during training: *Real-Web RL* directly interacts with costly real-web search and browsing tools. *Cached-Web RL* relies on the 2.79 million cached pages as training corpus and BGE-M3 (Chen et al., 2023) as search engine. SEA RL uses our Search Environment Agent described in Section 2 to provide adaptive search feedback while reducing unnecessary real-world interactions. All models are evaluated exclusively with real-web tools.

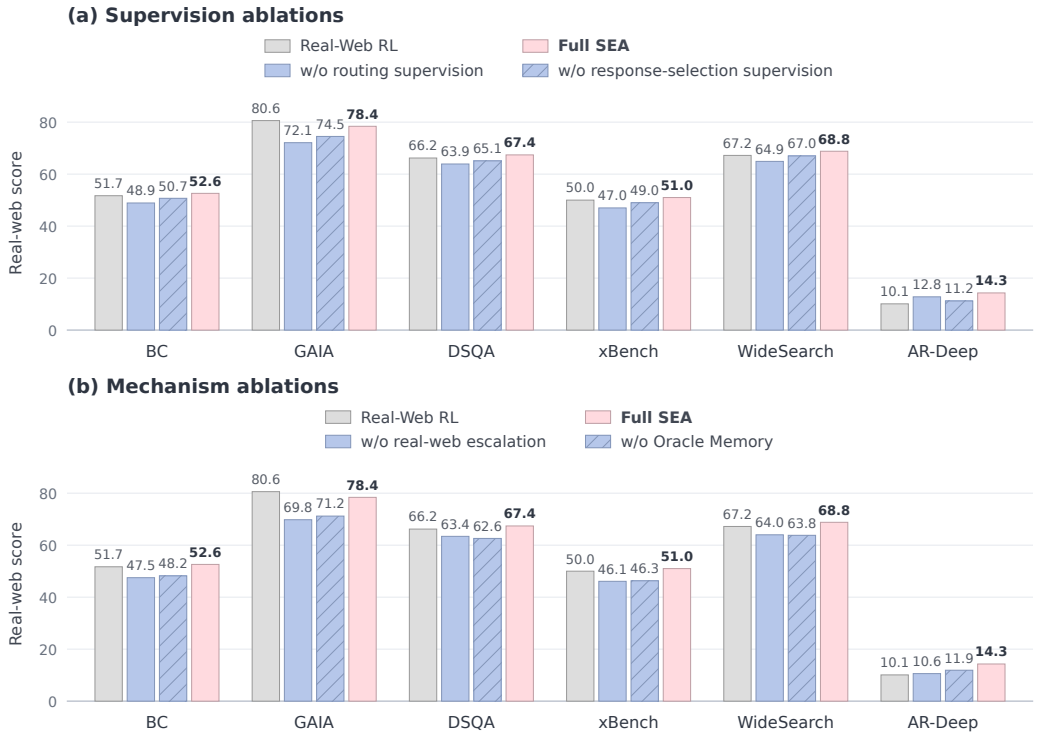


Figure 3: Ablations of SEA supervision and environment mechanisms for 4B policy RL.

Real-web transfer under different training environments. Table 2 summarizes the impact of training environments on downstream real-web performance. Training exclusively with cached webpages provides limited transferability: although Cached-Web RL improves over SFT on BrowseComp, its gains do not consistently extend to other real-web benchmarks. This indicates that static offline retrieval feedback cannot fully capture the dynamic interaction patterns required for real-world search and browsing.

SEA is competitive with Real-Web RL at substantially lower external API cost. It achieves higher scores on BrowseComp, DeepSearchQA, and AutoResearchBench-Deep, while trailing on GAIA by 2.2 points. These results suggest that effective RL environments should not only provide accurate evidence retrieval, but also model the decision process of searching, selecting, and utilizing information under realistic constraints.

RL Training Efficiency Analysis. Beyond superior transfer capability, SEA substantially reduces the RL training overhead. As shown in Table 2 (left), compared to Real-Web RL, SEA achieves an over **95% reduction in API costs** (32.1$ vs. 675.8$) along with a 17% reduction in step time (32.88 mins vs. 39.76 mins). This reduction comes from selective execution rather than removing real feedback: SEA reserves real calls for consequential mismatches that cannot be corrected from local evidence, and reuses the resulting observations to support subsequent interactions. This demonstrates that SEA achieves an exceptional efficiency–performance trade-off: it preserves the fidelity of dynamic online interactions while bypassing the prohibitive economic and latency costs of continuously hitting real-world web services during iterative RL exploration. We detail the cost scope, measurement setup, and SEA-specific preprocessing and inference overhead in Appendix H.

3.4 Feedback Calibration and Real-Web Transfer

We examine the local–real feedback gap and its relationship to search completion and real-web transfer. Detailed protocols and metric definitions appear in Appendix B.

Feedback calibration. Using the SFT-initialized 4B policy, we collect real-web trajectories on 100 questions held out from our constructed QA dataset and select 300 decision-critical search

actions. These questions are excluded from both policy and SEA training and are not used for model selection. All environments receive identical pre-action histories and actions, with the corresponding real response serving as the feedback reference.

Figure 2 (left) measures under-support through evidence recall and potential over-assistance through excess evidence exposure. Relative to Pure Local, SEA improves lead and fact recall by 8.5 and 5.0 percentage points, while reducing Extra Evidence and Answer Jump by 60.6% and 77.7%, respectively. These results characterize the complete environment, including selective real execution, rather than local response construction alone.

Continuation outcomes. From each selected decision point, we continue the fixed policy, with each branch retaining its assigned environment throughout. SEA achieves 85.3% continuation success, compared with 60.7% for Pure Local, 79.4% for SEA (before optimization), and 90.2% for Real-Web. It therefore closes 83.4% of the Pure Local–Real-Web success-rate gap. This improvement accompanies lower answer-level over-exposure in the feedback evaluation. The continuation comparison measures cumulative environment effects, including subsequent routing decisions, rather than the isolated effect of a single corrected response.

Reward–transfer alignment. Figure 2 reveals late-stage reward–transfer decoupling under local-only RL. Between updates 25 and 40, training reward increases from 0.39 to 0.53, but real-web BrowseComp performance falls from 47.3 to 43.3. Over the same interval, SEA improves reward from 0.53 to 0.62 and BrowseComp performance from 49.0 to 52.6. Its final score exceeds the best observed checkpoint under local-only RL by 5.3 points, so the advantage is not solely attributable to the late-stage decline under local-only RL. Within the observed training horizon, SEA better aligns continued reward optimization with real-web improvement.

3.5 Ablation Study

We study two groups of ablations for 4B policy RL: removing task-specific SEA supervision and disabling environment mechanisms. Figure 3 includes Real-Web RL as a reference. All policies are evaluated with real-web tools, without SEA or Oracle Memory.

Supervision ablations. Omitting either routing or response-selection supervision reduces performance on all six benchmarks, although both runtime functions remain active. These objectives address complementary decisions: routing supervision targets whether a decision-relevant mismatch requires real execution after accounting for local repairability, whereas response-selection supervision targets how to expose available evidence without harmful omissions or unwarranted assistance. The results suggest that retaining these mechanisms alone does not recover the benefits of learning action-dependent feedback.

Mechanism ablations. Removing Oracle Memory or disabling real-web escalation likewise degrades transfer across all benchmarks. The memory ablation supports the value of historical real-tool experience as a reference for routing and local response construction. In SEA, however, historical records are references rather than responses to replay indiscriminately: verified provenance alone does not justify exposing their evidence to the current action. The degradation without escalation is consistent with the limits of local calibration when consequential discrepancies cannot be resolved from available evidence. This ablation removes both immediate real feedback and the new observations that would update memory and the local index.

4 Conclusion

We presented SEA, a memory-augmented environment manager for low-cost deep search RL. Using Oracle Memory as an internal record of real-tool experience, SEA learns to calibrate locally supported feedback and reserve real execution for consequential discrepancies that cannot be repaired locally. SEA-trained policies achieve strong performance across six benchmarks using real-web tools exclusively at evaluation. In the controlled 4B comparison, SEA incurs approximately 4.8% of the measured training API cost of Real-Web RL for the RL training process while achieving higher scores on three of four benchmarks. This work highlights the importance of assessing training feedback by the search behavior it teaches, rather than its factual correctness alone.

AI USE STATEMENT

We used generative AI tools for auxiliary judging and annotation during data construction, including identifying feedback mismatches and assessing final-answer correctness. These outputs were used only as intermediate signals and were subject to human verification and consistency checks, as described in the appendix. We also used generative AI tools for limited language polishing and code assistance. We did not use generative AI tools to fabricate experimental results, citations, or unsupported claims. The authors reviewed all AI-assisted materials and take full responsibility for the final content of the paper.

ETHICS STATEMENT

This work studies training environments for web-based search agents. We use publicly available benchmarks and construct training data only from publicly accessible web sources and tool interactions. To reduce privacy risk, cached webpages and derived training examples are processed with privacy filtering to remove detected personally identifiable or sensitive information. We do not intentionally collect private user data, account-restricted content, or non-public documents. For released artifacts, we provide code, prompts, configurations, constructed training data, and redistributable resources, while excluding raw third-party content that cannot be redistributed. The proposed method is intended for improving reliable evidence acquisition in research settings, and should not be used for privacy-invasive data collection, bypassing access restrictions, or generating unsupported claims.

REPRODUCIBILITY STATEMENT

We provide implementation details to support reproducibility throughout the paper and appendix. Section 2 describes the SEA framework, including selective execution, grounded response construction, paired-feedback learning, and policy optimization. Section 3.1 specifies the evaluated benchmarks, baselines, metrics, and evaluation protocols. The appendix further reports data construction procedures, counterfactual comparison and judging prompts, SEA training targets, hyperparameters, cost accounting, and additional ablations. We also include anonymized source code and configuration files in the supplementary material, covering policy RL and example policy RL training data. We will release the code, constructed training data, and redistributable cached-web resources. During evaluation, all reported policies use real-web tools only, without SEA or Oracle Memory, to ensure that the measured gains reflect transferable search behavior rather than access to the learned training environment.

REFERENCES

Anthropic. Introducing claude opus 4.8, 2026. URL https://www.anthropic.com/news/claude-opus-4-8.

ByteDance Seed. Seed2.0 model card: Towards intelligence frontier for real-world complexity. *arXiv preprint arXiv:2607.00248*, 2026.

Haotian Chen, Xin Cong, Shengda Fan, Yuyang Fu, Ziqin Gong, Yaxi Lu, Yishan Li, Boye Niu, Chengjun Pan, Zijun Song, et al. Agentcpm-explore: Realizing long-horizon deep exploration for edge-scale agents. *arXiv preprint arXiv:2602.06485*, 2026.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2023.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, et al. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations. *arXiv preprint arXiv:2506.13651*, 2025.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026. URL https://arxiv.org/abs/2606.19348.

Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, pp. 1486–1494. PMLR, 2017.

Bernd Frauenknecht, Artur Eisele, Devdutt Subhasish, Friedrich Solowjow, and Sebastian Trimpe. Trust the model where it trusts itself - model-based actor-critic with uncertainty-aware rollout adaption. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 13973–14005. PMLR, 2024. URL https://proceedings.mlr.press/v235/frauenknecht24a.html.

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. *arXiv preprint arXiv:2508.07976*, 2025. doi: 10.48550/arXiv.2508.07976. URL https://arxiv.org/abs/2508.07976.

Google DeepMind. Gemini 3.1 pro: Model card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, February 2026. Published February 2026.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. *Advances in neural information processing systems*, 33:5541–5552, 2020.

Nikita Gupta, Riju Chatterjee, Lukas Haas, Connie Tao, Andrew Wang, Chang Liu, Hidekazu Oiwa, Elena Gribovskaya, Jan Ackermann, John Blitzer, et al. Deepsearchqa: Bridging the comprehensiveness gap for deep research agents. *arXiv preprint arXiv:2601.20975*, 2026.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516, 2025. doi: 10.48550/ARXIV.2503.09516. URL https://doi.org/10.48550/arXiv.2503.09516.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025.

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

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

Moonshot AI. Kimi k2.6 tech blog: Advancing open-source coding, 2026. URL https://www.kimi.ai/blog/kimi-k2-6.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-Searcher: Incentivizing the search capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025. doi: 10.48550/arXiv.2503.05592. URL https://arxiv.org/abs/2503.05592.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025.

MiroMind Team, S Bai, L Bing, L Lei, R Li, X Li, X Lin, E Min, L Su, B Wang, et al. Mirothinker-1.7 & h1: Towards heavy-duty research agents via verification. *arXiv preprint arXiv:2603.15726*, 2026.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, et al. Widesearch: Benchmarking agentic broad info-seeking. *arXiv preprint arXiv:2508.07999*, 2025.

Ziyi Xia, Kun Luo, Hongjin Qian, and Zheng Liu. Open data synthesis for deep research. *arXiv preprint arXiv:2509.00375*, 2025.

Jian Xie, Tianhe Lin, Zilu Wang, Yuting Ning, Yuekun Yao, Tianci Xue, Zhehao Zhang, Zhongyang Li, Kai Zhang, Yufan Wu, et al. Quest: Training frontier deep research agents with fully synthetic tasks. *arXiv preprint arXiv:2605.24218*, 2026.

Lei Xiong, Kun Luo, Ziyi Xia, Wenbo Zhang, Jin-Ge Yao, Zheng Liu, Jingying Shao, Jianlyu Chen, Hongjin Qian, Xi Yang, et al. Autoresearchbench: Benchmarking ai agents on complex scientific literature discovery. *arXiv preprint arXiv:2604.25256*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Xichen Zhang, Ziyi He, Yinghao Zhu, Sitong Wu, Shaozuo Yu, Meng Chu, Wenhu Zhang, Haoru Tan, and Jiaya Jia. SearchGym: Bootstrapping real-world search agents via cost-effective and high-fidelity environment simulation. 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. 18633–18665, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.848. URL https://aclanthology.org/2026.acl-long.848/.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. DeepResearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025. doi: 10.48550/arXiv.2504.03160. URL https://arxiv.org/abs/2504.03160.

A Related Work

Reinforcement learning for search agents. Search-R1 (Jin et al., 2025) and R1-Searcher (Song et al., 2025) train language models to interleave reasoning and retrieval through outcome-based RL, without annotated search trajectories. DeepResearcher (Zheng et al., 2025) extends RL to real-web search and browsing, while ASearcher (Gao et al., 2025) supports long-horizon exploration through fully asynchronous training. More recent systems combine RL with richer training pipelines. Tongyi DeepResearch (Team et al., 2025) integrates agentic mid-training and post-training with automated data synthesis, while QUEST (Xie et al., 2026) uses rubric-based synthetic tasks with verifiable rewards for multi-stage deep research training. These works advance search policy learning and scalable exploration. SEA addresses a complementary environment-side problem: providing affordable training feedback that supports real-web transfer. Rather than introducing a new policy optimizer, it learns to manage the observations from which the search policy learns.

Local and simulated search environments. Local retrieval and tool simulation reduce reliance on live-web interactions. Search-R1 uses a local Wikipedia index, while LiteResearcher (Li et al., 2026) co-constructs training tasks and a cached-web corpus supporting local search and browsing. SearchGym (Zhang et al., 2026) builds verified knowledge graphs, aligned documents, and solvable tasks, explicitly examining sim-to-real transfer. ZeroSearch (Sun et al., 2025) learns a query-conditioned search-response generator with a curriculum over document quality. These approaches address evidence availability, task–environment consistency, and controllable feedback generation. SEA focuses on an additional requirement: preserving *action-dependent access to evidence*. Even grounded responses can under-support effective queries or over-assist underspecified ones, distorting which behaviors RL reinforces. SEA calibrates such feedback through source selection, reranking, and rendering over cached webpages and action-compatible records of verified real interactions.

Decision-aware modeling and selective model use. Model-based RL balances inexpensive simulated experience against model bias. MBPO uses short model rollouts branched from real data (Janner et al., 2019), while MACURA adapts rollout lengths to local model uncertainty (Frauenknecht et al., 2024). A related line asks what models should preserve. Value-aware model learning evaluates transition-model errors through their effects on value functions (Farahmand et al., 2017), while value equivalence characterizes models that induce identical Bellman updates over specified policy and function classes (Grimm et al., 2020). SEA adopts a related decision-aware perspective empirically, rather than enforcing Bellman equivalence. Its routing supervision uses paired same-action feedback and counterfactual policy continuations to assess both decision relevance and repairability using pre-existing local evidence. Real execution is reserved for predicted consequential mismatches that cannot be corrected locally, rather than discrepancies in response appearance alone.

B Details of Feedback Calibration and Real-Web Transfer

B.1 Action-level Feedback Evaluation

We use the SFT-initialized 4B policy to collect real-web trajectories on the 100 held-out analysis questions described in Section 3.4. For each question, we select three decision-critical search actions, yielding 300 history–action pairs. Selection is based on evidence acquisition in the real-web trajectory, including new leads, resolved subquestions, and answer-supporting evidence. We conduct the evaluation three runs for average metric.

The same questions and decision points are used for both action-level feedback metrics and environment-level continuation evaluation. Each environment receives the identical pre-action history and action. SEA routing remains enabled, and real responses obtained through routing are included in the evaluated feedback. The comparison therefore concerns complete environments.

Annotation and adjudication. DeepSeek-V4-Flash (DeepSeek-AI, 2026) provides the primary structured annotations, and Qwen3.5-122B-A10B (Yang et al., 2025) independently annotates all samples. Gemini-3.1-Pro (Google DeepMind, 2026) adjudicates disagreements. The annotations identify newly available leads and key facts in the real response relative to the shared history, their

retention in the evaluated response, and additional key or answer-level evidence. Feedback metrics are computed from the final, adjudicated annotations.

Under-support metrics. Lead Recall and Fact Recall are macro-averaged over actions. For information type $T \in \{L, F\}$, let $n_{i,T}$ denote the number of reference items at action i, and $m_{i,T}^{(v)}$ the number retained by environment v. With $\mathcal{I}_T = \{i : n_{i,T} > 0\}$, we compute

$$R_T(v) = \frac{100}{|\mathcal{I}_T|} \sum_{i \in \mathcal{I}_T} \frac{m_{i,T}^{(v)}}{n_{i,T}}. \tag{3}$$

Each selected action contains a lead or key fact. An action without reference items of a particular type is excluded only from that type's recall average. Thus, recall weights actions equally rather than pooling all reference items.

Over-assistance metrics. Extra Evidence is the percentage of evaluated actions for which the environment exposes additional key evidence absent from the paired real response. Answer Jump is the percentage for which the environment exposes answer-level evidence while the real response provides at most intermediate clues. Both metrics use all evaluated actions as their denominator; Answer Jump is not normalized only over actions with clue-only real feedback.

These metrics concern evidence exposure rather than factual correctness. A real response is an observed reference, not an exhaustive account of retrievable evidence. Extra exposure therefore indicates potential over-assistance rather than necessarily incorrect or unobtainable information. Moreover, because the sample emphasizes decision-critical actions, these rates should not be interpreted as averages over all tool calls.

B.2 Environment-Level Continuation Evaluation

At each selected history–action pair, we branch from the saved real-web prefix using the same SFT policy checkpoint. Each branch receives its assigned environment's response and retains that environment for all subsequent interactions: REAL uses real tools, Pure Local uses local retrieval, and each SEA variant follows its corresponding routing policy. Decoding settings and continuation budgets are matched across environments. Success is determined by final-answer correctness.

This protocol differs from the single-step feedback probes used to construct routing supervision (Appendix G.1). Those probes vary the current response and use real tools thereafter; this evaluation sustains the assigned environment throughout the remaining episode. It therefore measures cumulative environment effects from a common prefix, rather than the isolated effect of the current response or preservation of action-value ordering.

REAL serves as an empirical reference, not a theoretical upper bound. The relative reduction in the observed Pure Local–REAL success-rate gap is

$$\frac{85.3 - 60.7}{90.2 - 60.7} \approx 83.4\%. \tag{4}$$

Environment-state isolation. Each evaluation point starts from a fixed environment snapshot containing only eligible pre-analysis evidence. Reference-trajectory observations are not used to populate the initial cache or Oracle Memory. Each continuation runs on a separate copy of the corresponding snapshot. Verified real observations may update that branch's memory and index for subsequent decisions, but these updates are discarded after the continuation and are never shared across branches or evaluation points. Diagnostic interactions are not written back to the training evidence stores.

B.3 RL Training Dynamics

In Table 3, Local Cached Web denotes local-only RL, whereas SEA denotes RL with the optimized environment. Steps are actual GRPO updates. Reward is the mean terminal correctness reward in the training environment. BC denotes the mean BrowseComp score, evaluated with real-web tools only, without SEA or Oracle Memory. Reward is reported on a 0–1 scale and BC on a 0–100 scale.

Environment	Metric	GRPO Update Step								
		0	5	10	15	20	25	30	35	40
Local Cached Web	Reward	0.28	0.34	0.37	0.35	0.41	0.39	0.45	0.51	0.53
	BrowseComp (BC)	34.3	38.9	40.5	41.2	41.5	47.3	45.0	43.6	43.3
SEA (Ours)	Reward	0.28	0.32	0.36	0.42	0.47	0.53	0.48	0.56	**0.62**
	BrowseComp (BC)	34.3	37.5	39.1	42.5	43.9	**49.0**	**47.6**	**51.3**	**52.6**

Table 3: Training reward and real-web BrowseComp performance across GRPO updates. SEA significantly outperforms the Local Cached Web baseline in long-term performance. All BC evaluations use real tools.

C Evaluation Protocol

We evaluate SEA-RL-4B and SEA-RL-35B through a unified agent interface, with identical tool access and inference budgets for each benchmark. Evaluation uses real-web search and browsing exclusively, without SEA or Oracle Memory. We allow at most 300 interaction turns per episode, with temperature 1.0 and a context window of $262{,}144$ tokens. The agent system prompt is provided in Appendix I.1. We use the official benchmark evaluators without modification and block access to benchmark answer pages and evaluation artifacts, including relevant Hugging Face pages, during inference.

D Deep Search Training Data Construction

We construct $2{,}525$ question–answer pairs for deep search policy training, comprising 989 *browse-intensive problems* and $1{,}536$ *scientific literature problems*. We split $2{,}425$ constructed data into training set for SFT and RL training, 100 into dev set for analytical experiments. Motivated by prior work on challenging information-seeking and long-horizon agent tasks (Li et al., 2025; ByteDance Seed, 2026), we prioritize task difficulty and evidence quality over dataset size. The questions are designed to exercise sustained exploration, query refinement, and cross-source verification, while retaining reference answers that support reliable correctness assessment.

Minimal-sufficiency objective. Both categories follow a *minimal-sufficiency objective* (Xia et al., 2025; Xiong et al., 2026): provide enough verifiable clues to distinguish the reference answer, without redundant information that creates retrieval shortcuts. Starting from a source-supported answer, we refine its identifying constraints through clue removal and searches for plausible alternatives. Each retained clue should contribute to disambiguation: removing it should leave an otherwise excluded candidate plausible. This is a practical construction objective evaluated against examined alternatives, not a guarantee of uniqueness over the entire open web. Because minimal sufficiency alone does not establish search difficulty, we separately screen for shallow solutions and assess difficulty through independent solving attempts.

D.1 Browse-Intensive Problems

We formulate browse-intensive questions as progressive candidate-disambiguation problems (Xia et al., 2025; Li et al., 2025). Human experts specify the expected answer type and verification requirements, then identify a documented target and gather complementary evidence from webpages, public repositories, and structured knowledge sources. The selected clues connect attributes or relationships that must be resolved across sources. Rather than exposing the target through a directly identifying phrase, the question provides partial descriptions that support intermediate searches and subsequent query refinement.

Question revision alternates between checking retrieval shortcuts and examining competing answers. Clues that immediately reveal the target are replaced with indirect but source-faithful descriptions. Conversely, when multiple candidates satisfy the question, we introduce a supported distinction and reassess which existing clues remain necessary. The intended search process is therefore to acquire

Table 4: Difficulty assessment of the constructed training questions under a common ReAct setup with a maximum sequence length of 128K tokens. Browse, Literature, and Browsecomp report answer accuracy (%); Avg. Turns reports the mean number of interaction turns.

Model	Constructed-Browse Acc. (%)	Literature Acc. (%)	Browsecomp Acc. (%)	Avg. Turns
Qwen3.5-397B-A17B	37.5	14.2	34.3	17.4
Gemini-3.5-Flash	51.7	20.5	–	30.5
Kimi-K2.6	55.9	30.9	65.4	26.8

partial evidence, narrow the candidate set, and verify the remaining constraints, rather than recover an answer from a single lexical match.

D.2 Scientific Literature Problems

We construct literature questions around technical distinctions that separate a target paper from closely related work. Targets are substantive computer science papers, with preference given to less prominent studies rather than broad surveys. Annotators examine the relevant full text to identify evidence such as a methodological condition, an assumption underlying a derivation, or a localized experimental finding. These details provide discriminative clues without directly revealing the paper's title or headline contribution.

Construction centers on comparing plausible paper candidates against their supporting passages. Annotators retain distinctions that rule out competing matches and express them without copying uniquely identifying wording. Where appropriate, a clue refers to a property established in a cited paper, requiring the solver to connect evidence across publications. Such dependencies are retained when they contribute to identifying the target, rather than merely lengthening the search. The resulting questions emphasize locating, interpreting, and reconciling technical evidence instead of matching bibliographic metadata.

D.3 Quality Control and Difficulty Assessment

Human–model verification. Human experts collaborate with Opus-4.8 (Anthropic, 2026) to construct and revise the questions, with human review of every retained question–answer pair. Verification checks answer correctness, consistency between clues and source passages, evidence availability, and plausible competing answers. Independent research rollouts further identify shallow retrieval solutions and cases requiring revision. Instances with unresolved ambiguity, contradictory constraints, or insufficient supporting evidence are removed. Model failure alone is not treated as evidence of desirable difficulty: a retained problem must still have a source-supported solution.

Empirical difficulty. We evaluate the questions using three strong models under a common ReAct framework (Yao et al., 2023), with a maximum sequence length of 128K tokens. Table 4 reports answer accuracy for each category and average interaction turns.

Both categories remain challenging for the tested agents, with literature questions yielding consistently lower accuracy. These results complement source-based quality checks by characterizing empirical difficulty; interaction length alone is not used as a quality criterion. The resulting question–answer pairs provide task-level supervision for policy training, distinct from SEA's action-level routing and response-selection examples described in Appendix G. Policy SFT and RL settings are provided in Appendix E.

Benchmark separation. We screen the constructed questions and reference answers against all six evaluation benchmarks for exact question overlap, semantic near-duplication, and identical or semantically equivalent answers. The screening identifies no remaining question or answer matches in the retained dataset. We additionally perform source-level deduplication against identified sources supporting benchmark reference answers and exclude these sources from the training-source pool. These checks separate the constructed tasks from the evaluation benchmarks at the question, answer,

and identified evidence-source levels. Our evaluation therefore measures transfer to held-out benchmark instances rather than performance on the constructed training tasks. Additional corpus- and memory-level safeguards are described in Appendix F.

E Deep Search Policy Training Detail

We train Qwen3.5-4B and Qwen3.5-35B-A3B through supervised fine-tuning (SFT) followed by policy RL. SFT provides a cold start for iterative search, browsing, and answer synthesis, while RL optimizes the policy using newly sampled interactions.

E.1 Supervised Fine-Tuning

We use Qwen3.5-35B-A3B to generate candidate trajectories for the training questions described in Appendix D. For each question, we sample 16 trajectories and retain at most five with correct final answers through rejection sampling, yielding $7{,}900$ SFT demonstrations. Each demonstration contains the reasoning, tool actions, observations, and final answer. The same demonstration corpus is used to initialize both policy scales. Filtering is based on final-answer correctness and does not require every intermediate search action to be optimal. We fine-tune each policy for three epochs with a learning rate of 5×10^{-6} and weight decay of 10^{-2}. Training uses Megatron on eight NVIDIA H100 GPUs with 80GB memory each.

E.2 Reinforcement Learning

RL uses the same underlying question–answer collection as SFT demonstration construction. Unlike SFT, it samples new trajectories from the current policy rather than replaying accepted demonstrations. For each question, we generate a group of eight trajectories and optimize the policy with GRPO (Shao et al., 2024). During SEA RL, tool calls receive feedback through the environment in equation 1, while the policy generates its own reasoning, search and browsing actions, and final answer.

An LLM judge evaluates the final answer against the reference answer, assigning a binary terminal reward of 1 for correctness and 0 otherwise. The RL loss applies only to policy-generated tokens; tool observations serve as conditioning context and are excluded from the loss. SEA is trained separately as described in Appendix G, and its parameters remain frozen throughout policy RL. Verified real-tool interactions nevertheless continue to update Oracle Memory and the local webpage index.

We implement policy RL using a modified `slime` framework with Megatron for model training. Each policy is trained for 100 updates with a learning rate of 10^{-6} and a GRPO group size of eight, using eight NVIDIA H100 80GB GPUs. Within each policy scale, the controlled training-environment comparisons start from the same SFT checkpoint and use identical RL optimization settings, varying only the environment that supplies tool feedback. At evaluation, all policies use real-web tools exclusively, without SEA or Oracle Memory.

F Environment Implementation and Oracle Memory

Our training environment combines a cached webpage index, action-conditioned Oracle Memory, and selective real-web execution. SEA constructs local feedback from available source evidence and invokes real tools when it predicts that a decision-relevant discrepancy cannot be corrected locally.

Cached webpage corpus. The local index contains approximately 2.79 million webpage records collected throughout data construction and training, subject to the benchmark exclusions below. During training-question construction (Appendix D), we retain answer-supporting pages, evidence for intermediate subquestions, and other pages returned in the top-10 search results. Query rewriting and difficulty refinement additionally provide hard negatives. The corpus is expanded with webpages encountered during question-validation rollouts and real-tool interactions in SFT and RL. It therefore contains supporting evidence, partially relevant pages, and plausible distractors, rather than only answer-bearing pages.

Each record preserves its source URL and webpage evidence. The index uses BGE-M3 (Chen et al., 2023) for query retrieval and supports URL lookup for page visits.

Oracle Memory. Oracle Memory is a trajectory-level store of verified responses obtained from prior SFT and RL interactions with the real tools. An entry associates the observed response and source evidence with the action that produced it, together with the relevant search history and interaction context. Thus, Memory stores reusable action–evidence relationships rather than an unconditioned collection of answers. The term "Oracle" refers to the verified provenance of these responses; it does not denote access to evaluation labels at test time. Query similarity alone does not justify reuse: evidence must also be compatible with the current tool type, query constraints, and source.

At each decision step, the environment retrieves candidates using both the current action and the preceding history. A memory entry is admitted only when it is compatible with the requested operation, such as a search query or a visit to a particular URL. Relevance to the question alone is insufficient. The retrieved memory entries are combined with pages from the local index, after which SEA estimates whether the available evidence can support an action-consistent response. It may reorder sources, select relevant excerpts, or remove unsupported candidates. If the corrected local evidence is sufficient, the action is answered locally; otherwise, SEA routes the action to the real tool. Verified real observations can subsequently extend the webpage index and Memory for later interactions.

Action-conditioned memory reuse. Real-tool provenance establishes where evidence came from, not whether the current action should expose it. Retrieved records therefore serve as historical references, rather than responses to replay unconditionally. SEA assesses their evidence against the current action and history; only evidence judged compatible with the tool type, query constraints, and source is eligible for local feedback. In particular, a clue retrieved by a more specific historical query is not automatically justified for an underspecified current query, even when both concern the same question. The paired-feedback objectives in Section 2.3 train this action-conditioned evidence selection. Memory remains internal to SEA: the policy receives the resulting tool observation, not the retrieved memory records.

Benchmark isolation. The benchmark-source exclusions in Appendix D apply to both the webpage corpus and Oracle Memory, including subsequent updates. We prohibit caching webpages or tool responses collected through evaluation-benchmark rollouts and exclude identified benchmark answer sources through source-level deduplication. Thus, evaluation trajectories do not populate the training evidence stores, and identified benchmark answer sources are not retained as reusable local evidence.

Temporal evidence boundaries. At each decision, SEA can retrieve only evidence already available before that decision. For paired-feedback supervision, the local evidence is frozen before collecting the corresponding real response and excludes later steps of the source trajectory. The paired real response and counterfactual continuations serve only as annotation references: they are not SEA inputs and cannot supply new content to the local target for that example. Real observations obtained after routing may support later interactions, but cannot retroactively alter the evidence available for the current routing decision.

Real-web evaluation. All reported transfer evaluations use real-web tools exclusively, without SEA, Oracle Memory, or the local webpage index. Evaluation interactions are not written back to the training stores. This separates benchmark execution from the training environment and tests the policy's search behavior without access to its training-time memory.

G Search Environment Agent Training Detail

We construct 3,000 routing examples and 2,000 response-selection examples to train SEA. Both tasks use paired local and real feedback for the same history and action, with DeepSeek-V4-Flash providing annotations. The supervision targets action-consistent evidence access rather than textual similarity between tool responses. The 3,000 routing examples and 2,000 response-selection examples are constructed from trajectories on the training question pool used for policy SFT and RL.

Sharing training questions supports complementary supervision: the policy learns task-solving behavior, whereas SEA learns action-level routing and evidence selection. All action examples derived from held-out analysis questions are excluded from SEA training.

G.1 Routing Supervision

Action selection for Supervision Human inspection assisted by an LLM identifies actions associated with new answer clues, changes in the policy's reasoning, or the resolution of a subproblem. We supplement these decision-critical actions with early and intermediate exploration and actions that yield no new clue on the real web.

Single-step feedback intervention. For each selected (h, a), we freeze the local evidence E before obtaining the paired real response. The initial local response o^{S} is constructed only from E; the real response o^{R} is collected for the same action. The fixed probe policy then continues independently from each observation with matched decoding settings and budgets. All subsequent calls in both branches use real-web tools, and later actions are generated anew.

The annotator uses these continuations to examine whether source-supported feedback differences affect evidence gathering, candidate elimination, search effort, or stopping. Neither a difference in final-answer correctness nor a different valid search path is sufficient on its own to determine the routing label. Local repairability is assessed separately using only E. The resulting label follows equation 2.

These continuations are annotation-only artifacts. They neither serve as SEA inputs nor contribute evaluation samples to the held-out environment analysis.

Decision relevance and local repairability. The annotator examines both the feedback discrepancy and its downstream consequences. At the feedback level, it identifies missing clues, factual errors, and decisive evidence not justified by the current action, citing supporting passages and sources. At the behavioral level, it checks whether the policy uses these differences in ways that obstruct evidence gathering, incorrectly eliminate candidates, induce avoidable tool calls, or trigger premature answering. A different but valid search path is not sufficient evidence of a harmful mismatch.

Crucially, the annotator also checks whether the discrepancy can be corrected using only the frozen evidence E. Permitted corrections include reranking candidates, selecting different source excerpts, and removing unwarranted information; they cannot introduce evidence obtained from the paired real call. Following equation 2, the target is `REAL` only when a decision-relevant mismatch cannot be repaired locally. Both adequate feedback and locally repairable mismatches receive `SIM`. This distinction trains the router to account for SEA's correction capabilities rather than escalating every imperfect local response.

G.2 Response-Selection Supervision

The second task trains SEA to calibrate feedback when local execution is sufficient. For each selected `search` action, we combine the top-20 local search candidates with Oracle Memory evidence compatible with the current tool type, query constraints, and sources. The annotator uses the same-action real response as a reference and produces an ordered list of candidate identifiers for the final top-10 response.

Targets prioritize locally supported clues missing from the initial response, reorder retained results, and remove evidence whose exposure is not justified by the action. A memory record is therefore not retained solely because it is relevant to the question or contains the correct answer. All selected content must already exist in E: the real reference guides selection but cannot supply new facts, pages, or excerpts. Decision-relevant omissions that exceed local support are handled by routing rather than fabricated local corrections. Response text is rendered from the selected sources as described in Section 2.2, keeping supervision focused on evidence selection and ordering.

G.3 Human Verification and SFT Configuration

Label-blinded human audit. Three human experts independently assess 100 examples spanning both tasks, presented in randomized order with the automatic labels hidden. For routing, they apply the decision-relevance criterion together with the local-repairability requirement. For response selection, they assess which candidates should be retained or excluded to preserve supported clues without introducing action-incompatible evidence. Inter-annotator agreement reaches Fleiss' $\kappa = 0.79$. Using majority votes as the human reference, agreement with the automatic annotations is 91.0% overall: 92.0% for routing and 90.0% for response selection. These results provide a sampled assessment of annotation consistency.

Training configuration and information separation. We initialize SEA from Qwen3.5-35B-A3B and fine-tune its shared parameters on the combined $5{,}000$ examples using task-specific instructions. Training runs for three epochs with a learning rate of 10^{-6} and weight decay of 10^{-2}, using Megatron on eight NVIDIA H100 GPUs with 80GB memory each.

For both tasks, SEA conditions only on (h, a, E) and the task instruction, predicting either a routing label or an ordered candidate list. Paired real responses and counterfactual continuations are annotation-only information and are never included in SEA's inputs. After SFT, SEA's parameters remain frozen throughout policy RL, while verified real interactions continue to update Oracle Memory and the local webpage index.

H Cost Accounting and Resource Usage

Accounting scope. The API costs in Table 2 cover external `search` and `visit` calls during each policy RL run. They exclude prior resource collection, LLM-based annotation and reward judging, GPU computation, and local indexing and storage. We account for reusable data-collection costs separately below. The reported API savings therefore do not imply an equivalent reduction in total training expenditure.

Per-run API expenditure. We use Serper[1] for Google search, accounting for queries at \$1 per $1{,}000$ requests, and Jina[2] for webpage retrieval. Browsing expenditure is estimated using $20{,}000$ tokens per retrieved page. The resulting cost in USD is

$$C_{\mathrm{API}} \approx 0.001 N_{\mathrm{search}} + 0.02 p_{\mathrm{J}} N_{\mathrm{visit}}, \tag{5}$$

where N_{search} and N_{visit} count externally executed queries and page retrievals, respectively, and p_{J} is the applied Jina price in USD per million tokens.

The 4B run comprises 100 RL updates, with 64 questions per update and eight rollouts per question, totaling $51{,}200$ trajectories. Under this accounting, Real Web RL costs approximately \$675.8, compared with \$32.1 for SEA RL, a 95.3% reduction. Approximately 5% of SEA-mediated tool actions invoke real tools. The routing fraction and expenditure ratio need not coincide, since search and visit calls have different unit costs.

Amortizing real-web interaction. Oracle Memory provides a mechanism for amortizing real-web interaction costs across repeated policy rollouts. Our SFT and RL stages use the same $2{,}425$ training questions, and repeated attempts can revisit search actions executed in earlier trajectories. For recurring actions, compatible historical evidence can support local feedback without another real-tool call, while the policy still generates its own actions and subsequent continuation. When exploration moves beyond memory coverage, the local webpage index may still provide sufficient evidence: a memory miss alone does not trigger real execution. SEA invokes real tools only when it predicts a decision-relevant discrepancy that cannot be corrected from the available local evidence. The resulting observations update Memory and can support later rollouts. This design reuses the results of prior exploration while reserving real interaction for unresolved feedback needs, subject to the same action-compatibility constraints.

[1] `https://serper.dev/`
[2] `https://jina.ai/reader/`

Reusable collection costs and amortization. We retain search responses and retrieved webpage content throughout training-question construction, annotation, SFT trajectory collection, and RL development. This process yields approximately 2.79 million cached webpages, with cumulative search-and-browsing API expenditure of approximately $3,400. These resources support subsequent experiments without requiring repeated collection from scratch.

To illustrate amortization, we conservatively charge the entire collection cost to SEA and none to the real-web baseline. For K additional, non-overlapping 4B training runs after collection, assuming unchanged per-run expenditure,

$$\begin{aligned} C_{\mathrm{SEA}}(K) &\approx 3400 + 32.1K, \\ C_{\mathrm{Real}}(K) &\approx 675.8K. \end{aligned} \tag{6}$$

Under these assumptions, SEA has lower cumulative web-tool API expenditure from the sixth additional run onward. Its cost advantage thus extends to repeated ablations and hyperparameter studies, although this API-only projection does not include computation or personnel costs. We plan to release the training questions, SFT trajectories, and redistributable cached-web resources to facilitate reuse.

Computational resources and serving overhead. In our 8xH100 training setup, SEA is deployed with SGLang, occupying approximately 10% of GPU memory per device in the measured configuration. Routing and response-selection requests use at most 10K tokens across input and output, and we observe stable serving at a concurrency of 128 requests. SEA remains parameter-frozen during policy RL, so its online model-computation overhead comes from inference rather than parameter updates.

The end-to-end step times in Table 2 include SEA inference and simulated environment processing. For the 4B policy, SEA requires 32.9 minutes per update, compared with 28.2 minutes for Cached Web RL and 39.7 minutes for Real Web RL. This corresponds to approximately 17% overhead relative to cached-only execution, but a 17% reduction in step time relative to real-web execution. Thus, SEA introduces additional local computation while reducing end-to-end latency in the measured setting.

I PROMPTS

I.1 DEEP SEARCH POLICY SYSTEM PROMPT

Figure 4 illustrates the complete system prompt and tool definitions designed for the deep-search policy agent.

System Prompt: Deep Search Policy

[Role & Mission]
You are a dedicated worker agent. Your primary role is to plan and orchestrate comprehensive, multi-step research to deliver an accurate answer supported by verified evidence. Analyze the user query, carry out multi-turn tool interactions, and deliver substantiated findings.

[Research Guidelines]
- **Exploration**: Start broad to map the landscape, then narrow down. Do not rely merely on snippets; use `visit` to inspect full pages for critical claims.
- **Persistence**: The question is guaranteed to have a verifiable answer. If a search direction fails, pivot perspectives, expand keywords, and diversify sources.
- **Verification**: An explicit verification step is mandatory. If the evidence does not fully satisfy the query, resume research instead of terminating prematurely.
- **Fact-Grounded**: All statements must be strictly derived from visited sources, citing exact, authoritative URLs.

[Operational Constraints]
- **Strict Tool Execution**: Always invoke tools via the predefined format. **Never** hallucinate or simulate tool outputs.
- Call exactly **one** tool per turn. Natural-language reasoning is permitted *before* the tool call, but strictly forbidden *after*.

[Available Tools]
`<tools>`
- `search(query: string[])`: Performs batched web searches; retrieves the top-10 results per query.
- `visit(url: string | string[], goal: string)`: Visits webpage(s) and extracts a goal-directed content summary.
- `finish(answer: string, evidences: object[], confidence: string)`: Submits the final result. Each item in `evidences` must contain a concise verified fact (`evidence`) and its exact source (`url`).

`</tools>`

[Response Format]
When invoking a function, strictly conform to the following schema without any suffix:

```
<tool_call>
<function=function_name>
<parameter=parameter_name>
parameter_value (valid JSON for structured fields)
</parameter>
</function>
</tool_call>
```

Figure 4: System prompt and operational schema for the Deep Search Policy Agent.

I.2 SEA Inference Prompts

SEA Inference: Selective Routing (g_ϕ)

[Role & Objective]
You are SEA, the environment routing controller. Given the current interaction history h_t, policy action a_t, and retrieved candidate evidence E_t, determine whether the action can be sufficiently served via local evidence (`SIM`, synthesizing the local webpage index and Oracle Memory) or requires real-web execution (`REAL`).

[Routing Criteria]

- **SIM** (Local Execution, $z_t = 0$): Select when available local evidence E_t (from cached pages or action-compatible Oracle Memory) provides actionable clues, valid navigation paths, or sufficient grounding to sustain policy progress without distorting evidence acquisition.
- **REAL** (Real Execution, $z_t = 1$): Reserve strictly for decision-relevant gaps that cannot be repaired locally (e.g., missing critical live facts, exact verbatim text/tables, exhaustive lists, or ungrounded local candidates).

[Output Format]
Return a single JSON object. For `REAL` or insufficient local coverage, set `"selections": []`:

```
{
  "route": "SIM" | "REAL",
  "reason_code": "SUFFICIENT" | "NO_COVERAGE" | "NO_NEW_EVIDENCE" |
                 "WRONG_ENTITY_OR_RELATION" | "STALE_OR_CONFLICTING" |
                 "UNGROUNDED",
  "selections": [
    {
      "unit_id": 0,
      "ranked_result_ids": ["S1"],
      "selected_span_ids": {"S1": ["P1"]}
    }
  ]
}
```

Figure 5: Inference prompt for SEA selective routing ($z_t = g_\phi(x_t)$).

I.3 Offline Annotation Prompts

Offline Annotation: Routing Supervision ($z^\star = d(1-c)$)

[Role & Task]
You are an offline expert annotator supervising SEA routing decisions. You are provided with: (1) current interaction (h, a), (2) candidate evidence E, (3) paired real-web observation o^{R}, and (4) auxiliary counterfactual rollouts. Diagnose whether the local mismatch is decision-relevant and whether it can be locally repaired.

[Supervision Logic]

- **Decision-Relevance** (d): Evaluate whether the gap between local feedback and real feedback materially alters the policy's evidential basis (e.g., misleads reasoning or triggers unnecessary queries). Minor stylistic differences are non-decision-relevant ($d = 0$).
- **Local Repairability** (c): Determine whether the gap can be resolved using available evidence in E through reranking, excerpt selection, or filtering.
- **Label Assignment**: Label as `LOCAL` (`MEMORY` or `SIM`) if $d = 0$ or $c = 1$. Assign `REAL` ($z^\star = 1$) only for irrecoverable decision-relevant discrepancies ($d = 1, c = 0$).

[Output Format]
Output a concise JSON object containing the ground-truth route and evidence citations:

```
{
  "status": "OK" | "UNRESOLVED" | "EXCLUDED",
  "route": "SIM" | "REAL",
  "reason_code": "SUFFICIENT" | "NO_COVERAGE" | "NO_NEW_EVIDENCE" |
                 "WRONG_ENTITY_OR_RELATION" | "STALE_OR_CONFLICTING" |
                 "UNGROUNDED",
  "basis": "evidence" | "observed_behavior" | "risk",
  "reason": "<Brief justification linking discrepancy to decisions>",
  "evidence": [
    {"source_id": "<exact_id>", "quote": "<verbatim substring>"}
  ]
}
```

Figure 6: Offline annotation prompt for supervising the routing objective.