ASCT: AGENTIC SEARCH CRITICAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) holds great promise for training autonomous agents in sequential decision-making, yet outcome-based methods struggle in long-horizon interactions because sparse terminal rewards provide little guidance for learning effective intermediate actions. Existing remedies are costly. Process Reward Models (PRMs) require a separately trained scorer and repeated step-level evaluation during rollout collection or test-time search, increasing training or inference cost depending on how they are deployed. Reward shaping relies on hand-engineered multi-objective functions, and value-based credit assignment suffers from unstable value estimation over long horizons. We propose Agentic Search Critical Training (ASCT), a two-stage RL framework built on a simple insight: teaching an agent to *critically judge which candidate action is better and to reason about why it is better* yields a higher performance ceiling than teaching it to *imitate* correct answers. Rather than relying on an auxiliary step scorer, ASCT decomposes multi-step search trajectories into three hierarchical perception levels—environment cognition, strategic decision-making, and tactical execution—and constructs multiple-choice discriminative tasks at each level via trajectory truncation. In Stage 1, hierarchical rewards inject these critical priors in a controlled setting; in Stage 2, all scaffolding is removed and the agent explores autonomously with only outcome rewards, leveraging its internalized analytical capabilities. Experiments on multi-hop QA and DeepResearch benchmarks show that ASCT consistently outperforms GRPO and other strong baselines across multiple model scales.

1 INTRODUCTION

Reinforcement learning (RL) has emerged as a powerful paradigm for training autonomous agents to interact with complex environments (Sutton & Barto, 2018; Chen et al., 2021). In information-seeking and deep research tasks, agents must navigate multi-turn interactions, making strategic decisions about when to search, which sources to consult, and when to synthesize findings (Yao et al., 2023; Zhou et al., 2024). Large language models (LLMs) have enabled such agents to handle increasingly sophisticated reasoning, from multi-hop question answering to open-domain research (Asai et al., 2024; Li et al., 2025c), and recent breakthroughs in reasoning models such as OpenAI o1 (OpenAI, 2024), DeepSeek-R1 (DeepSeek-AI et al., 2025), and QwQ (Qwen Team, 2024) have further demonstrated the power of RL in enhancing complex reasoning.

The prevailing approach for training these agents is outcome-based RL with algorithms such as GRPO (Shao et al., 2024), PPO (Schulman et al., 2017), or REINFORCE++ (Hu, 2025), which optimize agents to maximize final outcome rewards (e.g., answer accuracy). However, learning from such sparse outcome supervision becomes increasingly difficult in long-horizon agentic interactions because the terminal reward provides no direct guidance for intermediate search decisions.

In complex information-seeking tasks, an agent may execute a long and variable sequence of search and visit actions before producing a final answer. The sparse terminal reward (correct/incorrect) does not indicate which intermediate actions acquired useful evidence, repeated known information, or led exploration astray. This causes *inefficient exploration* (effective strategies must be discovered through many rollouts), *credit ambiguity* (success may coexist with poor search behavior, while sound search may still end in an incorrect answer), and *training instability* (trajectories with similar search quality can receive different terminal rewards).

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.

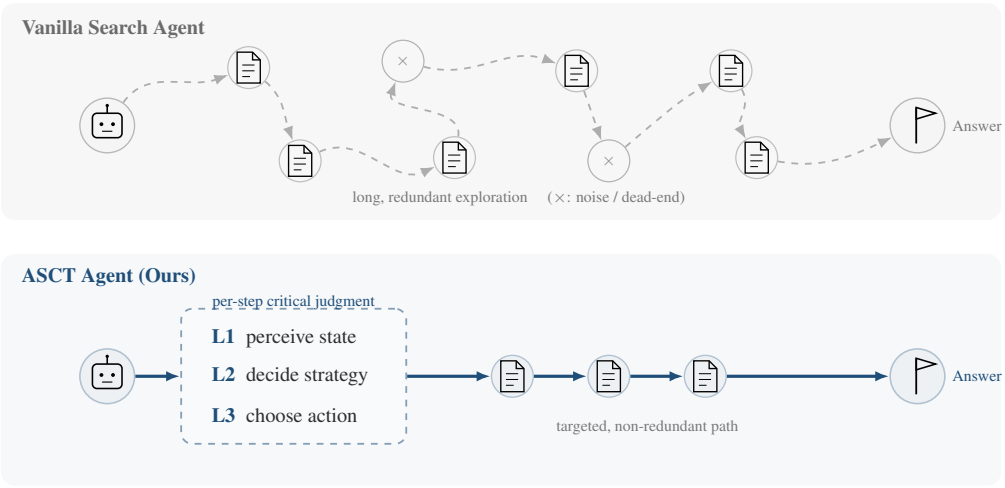

Figure 1: ASCT vs. conventional outcome-based agentic RL. A vanilla agent propagates one sparse outcome reward over a long, redundant trajectory; ASCT instead makes a per-step three-level judgment (L1 cognition, L2 strategy, L3 tactics) that yields a targeted path.

Prior work has attempted to address these challenges through various mechanisms. Process Reward Models (PRMs) (Lightman et al., 2023; Wang et al., 2023) provide step-level supervision, but require costly labels or Monte Carlo estimates and an auxiliary scorer during rollout collection; deployments that use PRM-guided reranking or search also incur test-time overhead. Reward shaping approaches engineer complex multi-objective functions but demand extensive domain expertise. Supervised fine-tuning on expert trajectories, as used in web-agent pipelines such as WebSailor (Li et al., 2025a), teaches imitation but can constrain subsequent exploration. Recently, ACT (Liu et al., 2026) explored training agents to discriminate action quality through RL, establishing critical training as a viable paradigm, though their approach is confined to world model scenarios with single-step decisions. Scalable RL systems like DAPO (Yu et al., 2025) still rely on outcome-level rewards without directly supervising intermediate action quality.

In this paper, we propose a novel perspective on agentic RL training: **instead of directly optimizing for outcomes, we first instill critical judgment—learning which candidate action is better and why—through hierarchical discriminative tasks, then leverage these internalized capabilities for efficient autonomous exploration**. We introduce Agentic Search Critical Training (ASCT), a two-stage RL framework designed to improve long-horizon learning under sparse outcome rewards without an auxiliary reward model or complex reward engineering (Figure 1). Our key insight is that multi-step agentic decision-making admits a natural decomposition into hierarchical cognitive levels: rather than explicitly assigning terminal credit to every action, we truncate expert trajectories and construct verifiable multiple-choice tasks that teach the policy to assess intermediate states and candidate actions before sparse-reward exploration.

Our contributions are threefold:

1. We address sparse-reward learning in long-horizon agentic search by turning intermediate state and action assessment into concrete, verifiable discriminative tasks via trajectory truncation.

2. We propose ASCT, a two-stage RL framework that first instills critical judgment—which candidate action is better and why—through hierarchical discriminative tasks, then enables autonomous exploration with sparse outcomes, requiring neither an auxiliary reward model nor complex reward engineering.

3. Experiments on multi-hop QA and DeepResearch benchmarks show that ASCT consistently outperforms GRPO and other strong baselines at multiple model scales.

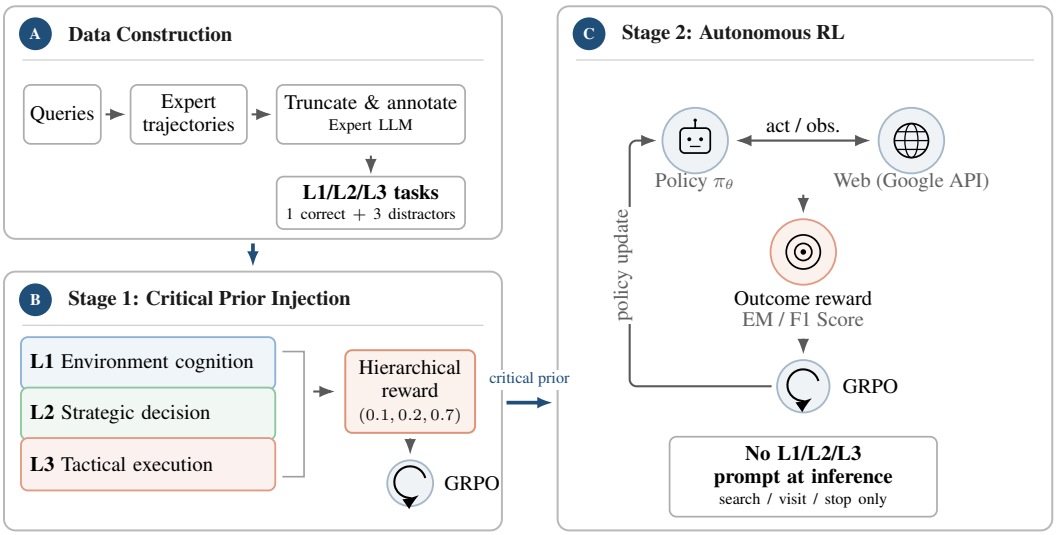

Figure 2: Overview of ASCT. (a) Expert trajectories are truncated into three-level discriminative tasks. (b) Stage 1 injects a critical prior via a hierarchical reward $(0.1, 0.2, 0.7)$ over L1/L2/L3. (c) Stage 2 removes all scaffolding and optimizes the same policy with outcome rewards only.

2 Related Work

Search Agents and Agentic RL. ReAct (Yao et al., 2023) interleaves reasoning with actions, WebArena (Zhou et al., 2024) provides a realistic web environment, and Self-RAG (Asai et al., 2024) adds reflection tokens for self-assessment; Search-o1 (Li et al., 2025b) and WebThinker (Li et al., 2025c) integrate agentic search into reasoning models. Search-R1 (Jin et al., 2025a) applies outcome-only RL to search agents, ARPO (Dong et al., 2025c) adapts rollouts for tool use, and AEPO (Dong et al., 2025a) balances rollout and policy entropy; these methods improve exploration but still optimize terminal outcomes without explicit intermediate-action supervision. Scalable systems such as DAPO (Yu et al., 2025), data-rich pipelines such as WebSailor (Li et al., 2025a) and Tool-Star (Dong et al., 2025b), and hierarchical reasoning frameworks (Jin et al., 2025b) provide complementary advances in search-agent training and structured reasoning.

Process Supervision and Critical Training. PRMs score intermediate reasoning steps (Lightman et al., 2023; Wang et al., 2023; Uesato et al., 2022), but require step-level targets and auxiliary scoring during RL rollouts; PRM-guided test-time search additionally incurs inference overhead. ASCT instead constructs discriminative tasks directly from expert trajectories and deploys no separate reward model. Reflexion (Shinn et al., 2023) and Self-Refine (Madaan et al., 2024) are inference-time verbal refinement methods without supervised parameter updates. Closest to our work, ACT (Liu et al., 2026) shows that critical judgment—discriminating better from worse actions—can be learned through RL with verifiable rewards, but only in single-step world-model settings; ASCT extends this critical-training paradigm to long-horizon, multi-turn tool use via a three-level hierarchical decomposition.

3 Method

We propose Agentic Search Critical Training (ASCT), a two-stage RL framework that teaches search agents to reflect through interaction. Figure 2 gives an overview. This section details our approach to trajectory truncation, three-level perception decomposition, and the progressive training pipeline.

3.1 Problem Formulation

In autonomous information-seeking tasks, a search agent interacts with a search environment over multiple turns. At each step t, the agent observes state s_t (current task, search history, and retrieved

content) and executes one of three environment actions: **search** (issue a query), **visit** (navigate to a URL), or **stop** (produce a final answer). BACKTRACK is not a fourth tool action; it is an internal L2 routing decision that abandons the current direction and is realized by a subsequent reformulated **search** action. For search and visit actions, the agent receives feedback o_t (search results or webpage content). After the agent decides to stop at step T, it produces a final answer and receives outcome reward R_{outcome}.

With terminal-only supervision, R_{outcome} provides no direct information about the quality of intermediate actions $a_1, a_2, \ldots, a_{T-1}$. A successful trajectory may contain redundant searches or irrelevant visits, while a failed trajectory may have a sound search strategy but flawed final reasoning. This sparse-reward ambiguity makes exploration and policy optimization inefficient over long horizons.

3.2 TRAJECTORY TRUNCATION AND THREE-LEVEL PERCEPTION

Rather than training agents to generate complete trajectories end-to-end, we truncate expert trajectories at intermediate states and construct discriminative subtasks. Given an ordered expert trajectory $\tau = \langle (s_1, a_1, o_1), (s_2, a_2, o_2), \ldots, (s_T, a_T) \rangle$, we sample truncation points and present the agent with partial state s_t.

At each truncation point, we decompose the agent's cognitive process into three hierarchical levels, following the perception–decision–execution paradigm: the agent first perceives what information is available, then makes a strategic judgment about whether to continue searching or stop, and finally executes a concrete action such as formulating a query.

L1: Environment Cognition. The agent must classify newly retrieved content o_t into categories: *New Fact*, *Redundant*, *Conflict*, or *Noise*. This forces the agent to build accurate state awareness rather than blindly accumulating information.

L2: Strategic Decision. Given the current state and cognition, the agent chooses among strategic options: **Search** (issue a new query), **Visit** (navigate to a specific URL), **Backtrack** (discard the current search direction and reformulate strategy), or **STOP** (produce final answer). This directly addresses three critical decisions: (1) when to stop searching, (2) whether to search broadly or visit specific pages, and (3) when to abandon an unproductive line of inquiry. The model emits one of four surface labels, but for reward accounting **Search** and **Visit** are mapped to the same SEARCH/VISIT class because the concrete tool and argument are resolved at L3; L2 is therefore scored as a 3-way judgment over {SEARCH/VISIT, STOP, BACKTRACK}.

L3: Tactical Execution. If the agent decides to search or visit, it must specify the concrete action. We turn this from open-ended generation into a multiple-choice task: the expert's actual query (or URL) serves as ground truth, contrasted with distractors that instantiate typical failure modes—*vague* queries (key entities or constraints removed, retrieving excessive noise), *hallucinated* queries (assumptions not supported by the history), *redundant* queries (re-requesting already-obtained information), and plausible-but-wrong links with similar anchor text. Distractors are generated by GLM-5.2, which produces three typed negatives per instance (for queries: vague, hallucinated, and redundant; for URLs: shallow-match, low-quality, and off-topic). This contrastive formulation is crucial: by weighing correct actions against typical failures, the agent learns discriminative features of good search behavior rather than merely imitating surface patterns.

3.3 STAGE 1: CRITICAL PRIOR INJECTION

In Stage 1, we train the agent on the three-level discriminative tasks using GRPO with a hierarchical proportional reward scheme. For each truncation point, the agent produces predictions $(\hat{y}_1, \hat{y}_2, \hat{y}_3)$ for the three levels, receiving reward:

$$R_{\text{stage1}} = w_1 \cdot \mathbb{I}(\hat{y}_1 = y_1) + w_2 \cdot \mathbb{I}(\hat{y}_2 = y_2) + w_3 \cdot \mathbb{I}(\hat{y}_3 = y_3) \tag{1}$$

where we set $(w_1, w_2, w_3) = (0.1, 0.2, 0.7)$, prioritizing tactical execution as the ultimate goal; when the ground-truth strategy is STOP or BACKTRACK, L3 is inapplicable and we renormalize to

$(w_1, w_2) = (0.4, 0.6)$ to prevent premature termination. L3 receives the highest weight because tactical execution quality directly determines information acquisition and is the primary bottleneck of search performance, whereas environmental cognition, though a necessary prerequisite, is rarely the main cause of failure. Three techniques ensure robust learning: (i) *naturalistic label distribution*—we preserve the deployment-like frequency of each L1/L2 category rather than artificially balancing them, retaining rare but operationally important CONFLICT/BACKTRACK cases (Section 4.4); (ii) *reshuffling*—options are randomly permuted to block positional shortcuts; (iii) *typed distractors*—each incorrect L3 option instantiates a distinct, labeled failure mode, so the agent must discriminate against realistic errors rather than random noise. During Stage 1, the reward rises steadily from $\sim$0.66 to $\sim$0.90, indicating thorough internalization of the three-level perception logic.

3.4 STAGE 2: AUTONOMOUS EXPLORATION

In Stage 2, all scaffolding is removed and we optimize the policy with GRPO using the outcome reward as the sole supervision signal. The critical prior learned in Stage 1 is now encoded in the model parameters: even without explicit hierarchical rewards, the agent implicitly runs the cognition $\rightarrow$ strategy $\rightarrow$ execution process at each step, so it enters exploration already knowing how to analyze search states, when to stop, and what constitutes a good action—the only difference from baseline GRPO is this fundamentally better prior.

Conceptually, the behavioral prior induced by Stage 1 can be interpreted through two latent judgments $\hat{y}_1$ (cognition) and $\hat{y}_2$ (strategy):

$$\pi(a_t|s_t) = \sum_{\hat{y}_1, \hat{y}_2} \pi_{\text{L3}}(a_t|s_t, \hat{y}_2)\, \pi_{\text{L2}}(\hat{y}_2|s_t, \hat{y}_1)\, \pi_{\text{L1}}(\hat{y}_1|s_t) \tag{2}$$

Equation (2) is an explanatory factorization, not the computational graph of separate latent-variable heads: Stage 2 remains a standard autoregressive policy optimized by a scalar terminal reward. Stage 1 directly supervises the three judgments and thereby initializes the same policy toward better search behavior; we do not claim that Stage 2 explicitly attributes credit to individual levels.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and Models. We evaluate on two task families. For multi-hop QA, we use Qwen2.5-7B (Yang et al., 2024) as the base model and follow the Search-R1 evaluation suite (Jin et al., 2025a), which covers three general QA sets—NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023)—and four multi-hop sets—HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), Musique (Trivedi et al., 2022), and Bamboogle (Press et al., 2023); we report Exact Match (EM), matching the Search-R1 evaluation protocol. For DeepResearch, following ARPO (Dong et al., 2025c), we adopt GAIA (Mialon et al., 2024), WebWalkerQA (Wu et al., 2025), Humanity's Last Exam (HLE) (Phan et al., 2025), and XBench (Chen et al., 2025) with Qwen3-8B/14B (Yang et al., 2025); these results use Pass@1 (%) under the AEPO/ARPO protocol. Since these benchmarks are evaluated on different subset versions across the literature, we take *all* baseline numbers from AEPO (Dong et al., 2025a) and evaluate ASCT on exactly those subsets (GAIA 103, WebWalkerQA 200, HLE 500 text-only, and XBench 100; per-split sizes are listed in Table 2). Table 1 reports the unweighted mean over its seven datasets, whereas each benchmark Avg. in Table 2 is the sample-weighted aggregate over that benchmark's sub-splits.

Baselines. We compare ASCT against two categories of baselines across two task families. For multi-hop QA (Table 1), the RL-trained reference points are Search-R1-base/instruct (Jin et al., 2025a), which apply outcome-only policy optimization to the same search-agent setting; we additionally report prompting and supervised baselines (Direct Inference, CoT, IRCoT (Trivedi et al., 2023), Search-o1 (Li et al., 2025b), RAG (Lewis et al., 2020), SFT, R1-base/instruct, and Rejection Sampling). A controlled comparison against plain GRPO (Shao et al., 2024) under our own codebase is reported as the w/o Stage 1 ablation in Table 3, since it shares every hyperparameter with ASCT and therefore isolates the effect of Stage 1. For DeepResearch (Table 2), we compare against prompting-based ReAct (Yao et al., 2023), Search-o1 (Li et al., 2025b), WebThinker (Li

Table 1: Exact Match (EM) on General QA and Multi-Hop QA benchmarks with Qwen2.5-7B. Avg. is the unweighted mean over the seven datasets. Best in **bold**, second-best underlined.

Methods	General QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	
Qwen2.5-7B-Base/Instruct								
Direct Inference	0.134	0.408	0.140	0.183	0.250	0.031	0.120	0.181
CoT	0.048	0.185	0.054	0.092	0.111	0.022	0.232	0.106
IRCoT (Trivedi et al., 2023)	0.224	0.478	0.301	0.133	0.149	0.072	0.224	0.226
Search-o1 (Li et al., 2025b)	0.151	0.443	0.131	0.187	0.176	0.058	0.296	0.206
RAG (Lewis et al., 2020)	0.349	0.585	0.392	0.299	0.235	0.058	0.208	0.304
SFT	0.318	0.354	0.121	0.217	0.259	0.066	0.112	0.207
R1-base	0.297	0.539	0.202	0.242	0.273	0.083	0.296	0.276
R1-instruct	0.270	0.537	0.199	0.237	0.292	0.072	0.293	0.271
Rejection Sampling	0.360	0.592	0.380	0.331	0.296	0.123	0.355	0.348
Search-R1-base (Jin et al., 2025a)	0.480	0.638	0.457	0.433	0.382	0.196	0.432	0.431
Search-R1-instruct	0.393	0.610	0.397	0.370	**0.414**	0.146	0.368	0.385
ASCT (Ours)	**0.492**	**0.651**	**0.468**	**0.447**	0.395	**0.208**	**0.445**	**0.444**

et al., 2025c), as well as RL-based GRPO, ARPO (Dong et al., 2025c), and AEPO (Dong et al., 2025a).

Implementation Details. All runs use 16 NVIDIA H200 GPUs, with HybridFlow (Sheng et al., 2024) for RL training and LLaMAFactory (Zheng et al., 2024) for SFT baselines. The training data are sourced from the public training resources released with Tool-Star (Dong et al., 2025b) and WebSailor (Li et al., 2025a), rather than from the evaluation benchmarks. Stage 1 uses 10K truncated trajectories sampled from these resources; we sample truncation points and generate distractors as described in the Method section. Stage 2 draws questions from a 50K-question pool built from the same two sources for autonomous exploration. With 128 questions per step, a GRPO group size of 16, and KL penalty coefficient 0.01, Stage 1 covers 3 epochs over the 10K samples in $\approx$240 GRPO steps at learning rate 1e-5, and Stage 2 runs 300 GRPO steps ($\approx$38K questions) at learning rate 5e-6; these are exactly the horizons plotted in Figures 3 and 5. All RL-based methods share the same hyperparameters for fair comparison. Offline annotation (Prompt A) and the DeepResearch answer judge both use GLM-5.2, and the search backend is the Google Search API returning the top-5 results per query, held fixed across our GRPO, SFT+GRPO, and ASCT runs.

4.2 Main Results

Table 1 reports multi-hop QA results with Qwen2.5-7B, and Table 2 reports the more challenging DeepResearch benchmarks with Qwen3-8B/14B.

Key observations:

1. **Consistent improvements over strong RL baselines:** On multi-hop QA, ASCT attains the best average EM (0.444) and the best score on 6 of 7 subsets (2Wiki excepted, where Search-R1-instruct leads 0.414 vs. 0.395). On DeepResearch, ASCT improves the average of *all four* benchmarks at *both* scales over AEPO, the strongest published baseline: GAIA +1.0/+2.0, WebWalkerQA +1.0/+1.5, HLE +1.0/+1.0, and XBench +2.0/+2.0. The margins are deliberately modest—one to two points on 100–500 question subsets corresponds to a handful of additional correct answers—but they are systematic across 8 benchmark–scale pairs.
2. **Where ASCT does not win:** the gains are not uniform. At 8B, ASCT ties AEPO on GAIA Lv.2 and WebWalkerQA Easy, and falls behind on GAIA Lv.3 (8.3 vs. 16.7 for ARPO) and HLE NS (11.8 vs. 12.1 for AEPO); at 14B it trails AEPO on WebWalkerQA Med. (48.8 vs. 50.0). GAIA Lv.3 contains only 12 questions, so this split differs by a single item and is dominated by sampling noise; the HLE NS and WebWalkerQA Med. losses are analyzed in Section 5.
3. **Improved search behavior:** As discussed in the failure analysis (Section 5), ASCT agents produce more targeted queries and avoid redundant searches, reflecting the internalized three-level perception from Stage 1.

Table 2: DeepResearch results with Qwen3 models (Pass@1, %). All baseline numbers from AEPO (Dong et al., 2025a) following the ARPO protocol (Dong et al., 2025c). ASCT is evaluated on the identical subsets (GAIA 39/52/12, WebWalkerQA 45/84/71, HLE 330/75/95, XBench 100); each Avg. is the sample-weighted aggregate of the displayed sub-splits. Best per column in **bold** (ties both bolded).

Method	General AI Assistant				WebWalkerQA				Humanity's Last Exam				XBench
	Lv.1	Lv.2	Lv.3	Avg.	Easy	Med.	Hard	Avg.	NS	CE	SF	Avg.	Avg.
Single-Enhanced Method (Qwen3-8B)													
Vanilla RAG (Lewis et al., 2020)	28.2	15.4	**16.7**	20.4	8.9	10.7	9.9	10.0	5.1	1.6	12.9	6.1	8.0
Search-o1 (Li et al., 2025b)	35.9	15.4	0.0	21.4	6.7	15.5	9.7	11.5	7.6	2.7	5.3	6.4	10.0
WebThinker (Li et al., 2025c)	43.6	11.5	0.0	22.3	6.7	13.1	16.9	13.0	7.3	4.0	6.3	6.6	13.0
ReAct (Yao et al., 2023)	35.9	17.3	8.3	23.3	8.9	16.7	18.3	15.5	4.2	4.0	6.3	4.6	16.0
RL-based Method (Qwen3-8B)													
Qwen3-8B	28.1	15.4	**16.7**	20.4	0.0	2.4	2.8	2.0	3.9	2.7	8.4	4.6	9.0
+ GRPO	48.7	25.0	8.3	32.0	28.9	32.1	28.2	30.0	7.9	4.0	10.5	7.8	20.0
+ ARPO (Dong et al., 2025c)	53.9	32.7	**16.7**	38.9	31.1	35.7	28.2	32.0	7.3	6.7	**15.8**	8.8	25.0
+ AEPO (Dong et al., 2025a)	61.5	**42.3**	8.3	45.6	**40.0**	39.3	35.2	38.0	**12.1**	5.3	11.6	11.0	28.0
+ **ASCT (Ours)**	**64.1**	**42.3**	8.3	**46.6**	**40.0**	**40.5**	**36.6**	**39.0**	11.8	**8.0**	**15.8**	**12.0**	**30.0**
Single-Enhanced Method (Qwen3-14B)													
Vanilla RAG (Lewis et al., 2020)	38.5	19.2	8.3	25.2	17.8	13.1	11.3	13.5	5.5	6.3	9.4	6.4	15.0
Search-o1 (Li et al., 2025b)	48.7	23.1	0.0	30.1	11.1	21.4	16.9	17.5	6.4	4.0	10.5	6.8	21.0
WebThinker (Li et al., 2025c)	48.7	26.9	8.3	33.0	13.3	23.8	18.3	19.5	7.0	4.0	9.5	7.0	23.0
ReAct (Yao et al., 2023)	48.7	25.0	8.3	32.0	11.1	20.2	12.7	15.5	5.8	5.3	10.5	6.6	20.0
RL-based Method (Qwen3-14B)													
Qwen3-14B	33.3	13.5	0.0	19.4	6.7	2.4	4.2	4.0	5.5	6.7	11.6	6.8	14.0
+ GRPO	51.3	34.6	0.0	36.9	35.6	42.9	35.2	38.5	7.9	6.7	12.6	8.6	27.0
+ ARPO (Dong et al., 2025c)	56.4	40.4	**16.7**	43.7	40.0	44.1	36.6	40.5	10.3	10.7	**13.7**	11.0	32.0
+ AEPO (Dong et al., 2025a)	61.5	44.2	**16.7**	47.5	40.0	**50.0**	40.9	44.5	10.6	**14.7**	10.5	11.2	35.0
+ **ASCT (Ours)**	**64.1**	**46.2**	**16.7**	**49.5**	**44.4**	48.8	**43.7**	**46.0**	**11.2**	**14.7**	**13.7**	**12.2**	**37.0**

4.3 Ablation Studies

To understand the contribution of each component, we conduct ablation studies (Table 3).

Importance of Two-Stage Training. Removing Stage 1 and training directly with outcome rewards (w/o Stage 1) is exactly vanilla GRPO: HotpotQA EM falls by 0.028 (0.447→0.419) and GAIA accuracy by 14.6 points (46.6→32.0, i.e., the + GRPO row of Table 2). The gap is far larger on GAIA than on multi-hop QA because GAIA requires successful end-to-end tool use over longer horizons—precisely the regime where credit assignment is most difficult. This confirms that reflective prior injection is crucial.

Three-Level Perception. Removing any level degrades performance, with L3 (Tactics) having the largest impact (-0.030 EM on HotpotQA, -15.5 points on GAIA), followed by L2 (-0.016 / -4.9) and L1 (-0.011 / -2.9). This monotone ordering justifies our hierarchical reward design (L3: 0.7 weight).

Hierarchical Rewards. Using uniform weights instead of our proportional scheme (0.1/0.2/0.7) costs 0.035 EM on HotpotQA (0.412) and 17.5 points on GAIA (29.1), falling below even the no-Stage-1 (pure GRPO) baseline of 0.419/32.0. This confirms that treating all three layers equally dilutes the learning signal: tactical execution (L3) requires substantially stronger gradient pressure to overcome the exploration difficulty of open-ended action generation, and an under-weighted L3 leaves the policy biased by Stage 1 without having acquired action discrimination—worse than no Stage 1 at all.

Sanity Checks. Candidate options are independently reshuffled with the correct index following the correct text, so a fixed-position policy cannot succeed. A separate label-corruption check (randomly reassigning the target index with option text fixed) drops L3 reward from 0.83 to 0.26 and overall reward from 0.90 to 0.64 with L1/L2 unchanged, confirming dependence on the option–label alignment.

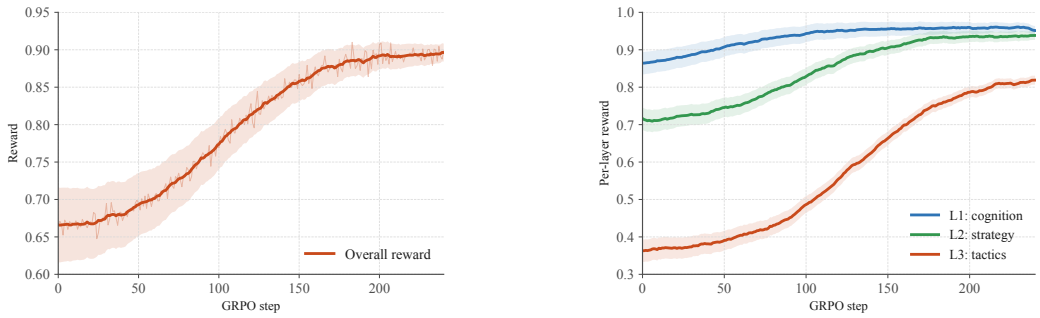

Figure 3: Stage 1 reward dynamics (≈240 GRPO steps). **Left:** overall weighted reward (∼0.66→∼0.90). **Right:** per-layer L1 (0.84 → 0.96), L2 (0.70 → 0.94), L3 (0.36 → 0.83).

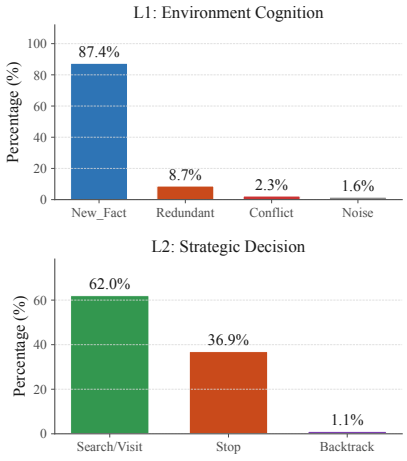

Figure 4: Naturally occurring L1 (4 classes) and L2 (3 classes) label distribution in the expert sessions, preserving deployment-like class prevalence.

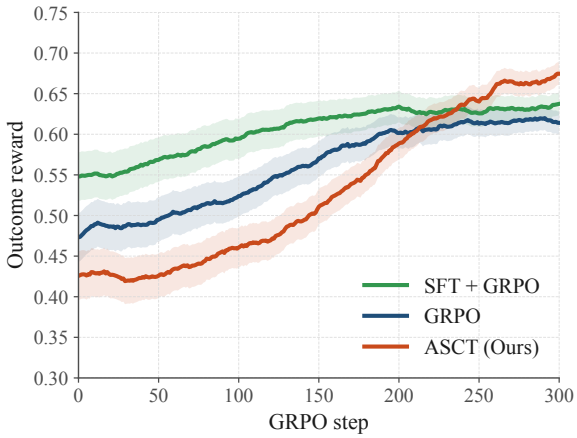

Figure 5: Stage 2 outcome-reward dynamics (Qwen3-8B). ASCT starts lowest (∼0.43) yet overtakes both baselines to the highest asymptote (∼0.68).

4.4 Training Dynamics Analysis

Figure 3 shows the Stage 1 reward dynamics. The overall mean reward rises from ∼0.66 to ∼0.90 over ≈240 GRPO steps with narrowing variance, and the per-layer decomposition reveals staggered convergence: L1 saturates near 0.96 within ∼50 steps (a straightforward 4-class categorization); L2 rises from 0.70 to 0.94; L3 starts at 0.36, above the 0.25 random-choice chance level, and grows slowest but most persistently, reaching 0.83 and still improving at the end. At each layer the agent must also *articulate why* the chosen option is correct—why a snippet is NEW_FACT rather than REDUNDANT, why SEARCH beats STOP, why one query dominates its alternatives—so the easiest layer plateaus quickly while the hardest keeps learning throughout, mirroring and justifying the reward weights.

Natural Label Distribution and Calibration. We keep the *naturally occurring* frequency of cognitive states observed in the expert sessions to preserve deployment-like class prevalence rather than to claim that imbalance itself prevents shortcuts. As shown in Figure 4, the L1 environment-cognition labels are dominated by NEW_FACT (87.4%) but retain meaningful mass on REDUNDANT (8.7%), CONFLICT (2.3%), and NOISE (1.6%); the L2 strategic-decision labels split between SEARCH/VISIT (62.0%) and STOP (36.9%), with a small but indispensable BACKTRACK tail (1.1%).

Table 3: Ablation study. HotpotQA/Musique EM follow Table 1; GAIA is the sample-weighted accuracy over 103 questions (Qwen3-8B). Ablating a level removes its Stage-1 output and renormalizes remaining weights to sum to one.

Variant	Qwen2.5-7B (EM)		Qwen3-8B
	HotpotQA	Musique	GAIA Avg.
Full ASCT	**0.447**	**0.208**	**46.6**
w/o L1 (Cognition)	0.436	0.199	43.7
w/o L2 (Strategy)	0.431	0.195	41.7
w/o Stage 1 (Stage 2 only)	0.419	0.184	32.0
w/o L3 (Tactics)	0.417	0.182	31.1
Uniform weights (0.33/0.33/0.33)	0.412	0.178	29.1

Because the L1 majority class alone reaches 0.87 and SEARCH/VISIT reaches 0.62, the converged 0.96/0.94 show the policy moves beyond constant prediction rather than exploiting the skew. We retain the rare BACKTRACK/CONFLICT labels for deployment fidelity, though per-class evaluation is still needed to quantify minority-class competence.

Stage 2 Transfer Dynamics. Figure 5 compares three initialization strategies under identical Stage 2 GRPO training. SFT+GRPO inherits the highest starting reward ($\sim$0.54) but saturates near $\sim$0.63 by step 130, as the behavior-cloning prior constrains exploration. Vanilla GRPO starts at $\sim$0.48 and converges to a similar ceiling ($\sim$0.62), indicating that outcome rewards alone provide insufficient credit assignment. ASCT begins *lowest* at $\sim$0.43—Stage 1 injects no answer-level shortcuts, so the initial policy has no head start on final accuracy—yet it overtakes GRPO around step 165 and SFT+GRPO around step 200, ending at $\sim$0.68. This crossover is the central empirical signature of our approach: Stage 1 transfers the ability to *compare* competing candidate actions rather than to imitate correct answers, which pays off once the policy has sufficient autonomy.

5 DISCUSSION

Why Two-Stage Training Works. ASCT addresses a practical tension in agentic RL: dense guidance is useful for learning intermediate search behavior, yet deployment tasks commonly provide only sparse outcome rewards. Stage 1 supplies that dense structure; once internalized, critical judgment becomes an implicit policy prior that supports Stage 2 exploration without an auxiliary scorer.

Failure Analysis. ASCT still exhibits weaknesses in specific scenarios, visible directly in Table 2. On splits that reward broad recall rather than selective querying, such as HLE NS at 8B and WebWalkerQA Med. at 14B, ASCT falls slightly behind AEPO: the L3 objective optimizes for precise, non-redundant actions, which can under-explore when the answer requires sweeping a wide topic space. Since removing L3 causes the largest ablation drop (-0.030 EM on HotpotQA, -15.5 points on GAIA), future work should prioritize strengthening L3 discrimination for query formulation and URL selection.

6 CONCLUSION

We proposed ASCT, a two-stage RL framework for long-horizon agent learning under sparse outcome rewards. Stage 1 instills critical judgment through verifiable hierarchical discriminative rewards, providing a stronger initialization for autonomous sparse-reward exploration in Stage 2. Across multi-hop QA and DeepResearch benchmarks, ASCT consistently outperforms GRPO and other strong baselines at multiple model scales.

AI USE STATEMENT

In this work, we used generative AI tools (GLM-5.2) for offline trajectory annotation (Prompt A) and as the LLM-as-judge for DeepResearch evaluation. We also used generative AI tools to assist

with paper editing, formatting, and scientific figure generation. We have not used generative AI tools for research ideation, experimental design, or result interpretation. All AI-assisted content has been reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics statement

This work studies reinforcement learning for information-seeking agents on publicly available research benchmarks (multi-hop QA and DeepResearch suites); it involves no human subjects, no private or personally identifiable data, and no sensitive attributes. The primary societal consideration is that stronger autonomous search agents can retrieve and synthesize web content more effectively, which shares the dual-use and misinformation risks common to all retrieval-augmented systems; ASCT introduces no new capability of this kind beyond the underlying base models. All datasets and baselines are used in accordance with their licenses. The authors have read and adhere to the ICLR Code of Ethics.

Reproducibility statement

We take several measures to support reproducibility. The main paper specifies the two-stage training procedure, the hierarchical reward with its exact weights, and all shared hyperparameters (Section 4.1). Appendix B reports the complete GRPO and optimizer configuration for both stages (Tables 4–6) together with the precise reward and advantage definitions, and Appendix C provides the three prompt templates (offline annotation, Stage-1 inference, and Stage-2 inference) that fully determine the pipeline. The 10K/50K training pools are sourced from the public Tool-Star and WebSailor training resources, while evaluation follows the public ARPO/AEPO protocol on the exact benchmark subsets enumerated in Section 4.1.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

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.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic entropy-balanced policy optimization. *arXiv preprint arXiv:2510.14545*, 2025a.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. *arXiv preprint arXiv:2505.16410*, 2025b.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025c.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

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

Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yang Zhao, Hongjin Qian, and Zhicheng Dou. Decoupled planning and execution: A hierarchical reasoning framework for deep search. *arXiv preprint*, 2025b.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

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*, 2025a.

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. *arXiv preprint arXiv:2501.05366*, 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.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Weize Liu, Minghui Liu, Sy-Tuyen Ho, Souradip Chakraborty, Xiyao Wang, and Furong Huang. Agentic critical training. *arXiv preprint arXiv:2603.08706*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Koustuv Mihaylova, Richard Pang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 9802–9822, 2023.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. *arXiv preprint*, 2024.

OpenAI. Learning to reason with llms. *OpenAI Blog*, 2024.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2023.

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown. *Hugging Face*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.

Noah Shinn, Frank Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.

Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. *MIT press*, 2018.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Jonathan Uesato, Nate Kushman, Vladimir Mishkin, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, et al. Webwalker: Benchmarking llms in web traversal. *arXiv preprint arXiv:2501.07572*, 2025.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

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*, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*, 2024.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024.

A SCOPE OF THIS SUPPLEMENT

This appendix provides the implementation details omitted from the main text for space reasons. It contains two parts:

1. **Complete RL configuration** for both training stages (Section B): shared GRPO and optimizer settings (Table 4), Stage-1 critical-prior injection (Table 5), and Stage-2 autonomous exploration (Table 6), together with the exact reward definitions.
2. **The three prompt templates** that fully determine our pipeline (Section C): the offline annotation prompt used to build the Stage-1 dataset (Section C.1), the Stage-1 inference prompt (Section C.2), and the Stage-2 inference prompt (Section C.3).

All values below are the ones used to produce the reported results; the quantities that also appear in the main text (learning rates, batch size, group size, KL coefficient, step counts, and reward weights) are reproduced here unchanged, so the appendix and the main text can be cross-checked directly.

B TWO-STAGE RL CONFIGURATION

B.1 SHARED SETTINGS

Both stages optimize the same policy with GRPO (Shao et al., 2024) under the HybridFlow framework (Sheng et al., 2024), using an identical optimizer, rollout engine, and advantage estimator; only the reward, the data, the learning rate, and the sequence budget differ. Table 4 lists the settings that are held fixed across stages and across all three base models (Qwen2.5-7B, Qwen3-8B, Qwen3-14B). Because GRPO estimates advantages from group-relative rewards, no value network is trained, which is what makes the two-stage schedule affordable on a single 16-GPU node group.

For a sampled group of G responses to the same input with rewards $\{r_1, \ldots, r_G\}$, the advantage of the i-th response is

$$A_i = \frac{r_i - \mathrm{mean}(r_1, \ldots, r_G)}{\mathrm{std}(r_1, \ldots, r_G) + \delta}, \qquad \delta = 10^{-6}, \tag{3}$$

and the policy is updated with the clipped surrogate objective plus a KL penalty against the frozen initial policy π_{ref},

$$\begin{aligned} \mathcal{L} = -\ \mathbb{E}\big[\min\big(\rho A,\ \mathrm{clip}(\rho, 1{-}\epsilon, 1{+}\epsilon)\, A\big)\big] \\ + \beta\, \mathbb{D}_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}]\,, \end{aligned} \tag{4}$$

where ρ is the token-level importance ratio, $\epsilon = 0.2$, and $\beta = 0.01$.

B.2 STAGE 1: CRITICAL PRIOR INJECTION

Stage 1 trains on 10K truncated-trajectory samples drawn from the public Tool-Star (Dong et al., 2025b) and WebSailor (Li et al., 2025a) training resources for 3 epochs, which at 128 samples per step amounts to ≈240 GRPO steps — the horizon plotted in Figure 3. Each sample is a single truncation point rendered with the prompt of Section C.2; responses are short (one reasoning block plus a JSON verdict), so the response budget is 1,024 tokens.

The reward is the hierarchical sum over the three levels,

$$R_{\text{stage1}} = \sum_{\ell=1}^{3} w_\ell\, \mathbb{I}\big(\hat{y}_\ell = y_\ell\big), \tag{5}$$

with $(w_1, w_2, w_3) = (0.1, 0.2, 0.7)$ and two special cases. First, when the ground-truth strategy is STOP or BACKTRACK, Layer 3 is undefined and the weights are renormalized to $(w_1, w_2) = (0.4, 0.6)$, which keeps the reward scale comparable across sample types while emphasizing the decision not to keep searching. Second, a response whose `<answer>` block cannot be parsed as valid JSON, or that omits a required field, receives $R_{\text{stage1}} = 0$ regardless of content; this format gate is what makes the reward fully verifiable without a learned scorer.

Table 4: Settings shared by Stage 1 and Stage 2 and by all base models. With 128 prompts per step and $G = 16$, each step consumes 2,048 rollouts.

Setting	Value
Algorithm	
RL algorithm	GRPO (critic-free)
Advantage estimator	group-normalized reward
Group size G	16
Prompts per step	128
Rollouts per step	2,048
Prompts per gradient update	32
Gradient updates per step	4
Clip ratio ϵ	0.2
KL coefficient β	0.01
KL estimator	low-variance (k_3)
Reference policy	frozen initial policy
Optimization	
Optimizer	AdamW
(β_1, β_2), ε	$(0.9, 0.95)$, 10^{-8}
Weight decay	0.01
LR schedule	constant, 10-step warmup
Gradient clipping	1.0
Precision	bf16
Sampling	
Rollout temperature / top-p	1.0 / 1.0
Eval. temperature / top-p	0.6 / 0.95 (Qwen3)
Eval. decoding (Qwen2.5-7B)	greedy
Samples per eval. question	1 (Pass@1)
Infrastructure	
Hardware	16× NVIDIA H200
Training backend	FSDP (full shard)
Rollout backend	vLLM, TP $= 2$
Rollout GPU mem. fraction	0.6

Table 5: Stage-1 (critical prior injection) configuration. L2 is scored as a 3-way judgment because SEARCH and VISIT share one label.

Setting	Value
Learning rate	1e-5
Training samples	10,000 truncation points
Data sources	Tool-Star + WebSailor
Epochs	3
GRPO steps	≈240
Max prompt length	8,192 tokens
Max response length	1,024 tokens
Reward weights (w_1, w_2, w_3)	$(0.1, 0.2, 0.7)$
Weights if L3 inapplicable	$(0.4, 0.6)$
Malformed-output reward	0
L1 label space	4 classes
L2 label space	3 classes
L3 options per sample	4 (1 correct, 3 distractors)
Option re-shuffling	every epoch
Distractor construction	3 typed negatives (GLM-5.2)

Options are re-shuffled at every epoch so that no position can be memorized. Distractors are the three typed negatives produced by GLM-5.2 (Prompt A); we impose no numeric similarity schedule. Table 5 summarizes the stage.

Table 6: Stage-2 (autonomous exploration) configuration. No hierarchical reward and no per-level output are used; the three-level judgment survives only as an implicit prior in the policy parameters.

Setting	Value
Learning rate	5e-6
Question pool	50,000
Data sources	Tool-Star + WebSailor
GRPO steps	300 (≈38K questions)
Max total context	32,768 tokens
Max tokens per turn	2,048
Max turns per trajectory	10
Reward	outcome only (F1 / judge)
Invalid or truncated trajectory	0
Loss on observation tokens	masked
Snippets per `<search>`	Google API, top-5
Page budget per `<visit>`	4,096 tokens
Tool timeout	30 s
Tool failure handling	error obs., turn used

B.3 Stage 2: Autonomous Exploration

Stage 2 removes every scaffold: no options are provided, no per-level label is requested, and the only supervision is the final answer reward. The policy interacts with the search environment through the two tools declared in Section C.3 for at most 10 turns, and the episode ends when the agent emits an `<answer>` block or exhausts its turn or context budget. Training runs for 300 GRPO steps over a 50K-question pool built from the same Tool-Star and WebSailor training resources, i.e. ≈38K questions, matching the horizon of the Stage-2 comparison figure in the main text.

The outcome reward is task-dependent:

$$R_{\text{outcome}} = \begin{cases} \mathrm{F1}(\hat{a}, a^*) & \text{QA}, \\ \mathbb{I}\big[\mathrm{judge}(\hat{a}, a^*)\big] & \text{DeepResearch}, \end{cases} \tag{6}$$

where $\hat{a}$ is the extracted answer span, a^* the reference answer, and the judge is a GLM-5.2 LLM-as-judge applied to our runs. Trajectories that never emit a well-formed answer, exceed the turn limit, or overflow the context receive reward 0.

Two implementation details matter for reproduction. Tool outputs are inserted into the context as observations but are *masked out of the loss*: gradients flow only through tokens the policy generated, otherwise the policy would be rewarded for predicting retrieved text. The search backend is the Google Search API (top-5 results per query), held fixed across our GRPO, SFT+GRPO, and ASCT runs so that the comparison isolates the policy rather than the search backend.

C Prompt Templates

Three prompts specify the pipeline. Placeholders written as `user_query`, `full_history_text`, `current_step_raw_data`, `agent_actual_action`, `layer_3_options`, and `max_turns` are substituted at runtime. The annotation prompt (Section C.1) is used *offline only*, to convert expert trajectories into labeled truncation points; it is never shown to the policy being trained. The Stage-1 prompt (Section C.2) is what the policy sees during critical prior injection. The Stage-2 prompt (Section C.3) is what it sees during autonomous exploration and at evaluation time.

The asymmetry between the last two is deliberate and is the mechanism behind our central claim. Stage 1 asks for an explicit three-level verdict with a written justification at each level; Stage 2 asks for none of it. Any three-level reasoning that survives into Stage 2 therefore has to come from the parameters rather than from the instructions.

C.1 PROMPT A: OFFLINE TRAJECTORY ANNOTATION

This prompt is run with GLM-5.2 as the teacher annotator over each truncation point of an expert trajectory. It emits the Layer-1 and Layer-2 ground-truth labels and, when the strategy is SEARCH or VISIT, constructs the four Layer-3 options with the correct index recorded. When the strategy is STOP or BACKTRACK it explicitly returns a null option list, which is exactly the case that triggers the $(0.4, 0.6)$ reward renormalization of Table 5.

```
# Role: Expert Data Annotator for Search Agent Training

## Task
You are an expert annotator. Your goal is to analyze a provided
**Search Trajectory** and output a structured JSON containing:
1. **Ground Truth Labels** for Layer 1 (Perception) and Layer 2
   (Strategy) based on fixed definitions.
2. **Constructed Multiple-Choice Options** for Layer 3 (Execution)
   ONLY IF the strategy requires an action (Search/Visit).

## Input Data
- **User Goal**: user_query
- **Full History**: full_history_text (All previous steps, facts,
  and decisions)
- **Current Step Context**: current_step_raw_data (Current search
  results, docs, or URL list)
- **Agent's Actual Action**: agent_actual_action (The ground truth
  query or url clicked)

## Fixed Definitions (For Layer 1 & 2)

### Layer 1: Perception Status (Choose ONE)
- `NEW_FACT`: Current doc provides key info missing in history.
- `REDUNDANT`: Current doc repeats info already in history.
- `CONFLICT`: Current doc contradicts confirmed history facts.
- `NOISE`: Current doc is irrelevant, ads, or a dead-end.

### Layer 2: Strategic Decision (Choose ONE)
- `SEARCH`: Information gap exists; need a new query.
- `VISIT`: A specific URL in current results looks promising;
  need to click.
- `STOP`: Info is sufficient to answer the user goal.
- `BACKTRACK`: Current path is invalid; need to revert/change
  direction.

## Instructions

### Step 1: Determine Layer 1 Ground Truth
Analyze `Current Step Context` vs `Full History`. Select the **ONE**
correct label from the fixed definitions above.
-> Output: `layer_1_gt`

### Step 2: Determine Layer 2 Ground Truth
Based on Layer 1 result and `User Goal`, select the **ONE** correct
strategic label from the fixed definitions above.
-> Output: `layer_2_gt`

### Step 3: Construct Layer 3 Options (CONDITIONAL)
**LOGIC CHECK**:
- IF `layer_2_gt_label` is `STOP` or `BACKTRACK`:
  - Set `layer_3_options` to `null`.
  - Set `layer_3_correct_index` to `-1`.
  - **Do not generate any options.**

- IF `layer_2_gt_label` is `SEARCH`:
  - **Correct Option**: Use `Agent's Actual Action` (the real query)
    as the correct answer.
  - **Generate 3 Distractors** specifically for this context:
    1. *Too Vague*: Missing key entities/constraints found in the
       gap analysis.
    2. *Hallucinated*: Includes assumptions not supported by history.
    3. *Repetitive*: Re-asks a question already answered in
       `Full History`.
  - Shuffle these 4 options and record the index of the correct one.

- IF `layer_2_gt_label` is `VISIT`:
  - **Correct Option**: The specific URL from `Current Step Context`
    that matches `Agent's Actual Action`.
  - **Generate 3 Distractors** from the available URL list or
    hypothetical ones:
    1. *Shallow Match*: Relevant title but likely lacks depth
```

```
       (e.g., news snippet).
    2. *Outdated/Low Quality*: From an unreliable source or old date.
    3. *Irrelevant*: Looks similar but addresses a different
       sub-topic.
  - Shuffle and record the correct index.

## Output Format
Output **ONLY** a valid JSON object.

{
  "analysis": {
    "reasoning": "Brief explanation of why L1 and L2 are chosen.",
    "gap_identified": "What specific info is still missing?
                       (Only if L2 is SEARCH/VISIT)"
  },
  "labels": {
    "layer_1_gt": "STRING (from fixed definitions)",
    "layer_2_gt": "STRING (from fixed definitions)"
  },
  "layer_3_construction": {
    "is_applicable": true | false,
    "options": [
      // If applicable: List of 4 objects
      //   {"label": "A", "text": "..."}
      // If not applicable: null
    ] | null,
    "correct_index": 0 | 1 | 2 | 3 | -1
    // Index corresponds to the position in the shuffled 'options'
    // list (0-based).
    // -1 if not applicable.
  }
}
```

C.2 PROMPT B: STAGE-1 INFERENCE

The policy receives the same context the annotator saw, minus the ground truth, and must commit to one label per level. For STOP/BACKTRACK instances, the prompt renderer independently injects four format-matched decoy actions whose L3 outputs are ignored by the reward, ensuring that the presence or absence of candidate options does not reveal the L2 label. The requirement that every level be justified *against a named rejected alternative* is what prevents Stage 1 from degenerating into three independent classification heads: the reward checks only the labels, but the justification requirement forces the comparison that produces them to be verbalized, and empirically it is the hardest part of Layer 3.

```
# Role: Critical Search Analyst

## Task
You are given one step of an in-progress search session. Judge the
situation at three levels. For EVERY level you must also explain why
your choice is better than at least one alternative you rejected.

## Input
- **User Goal**: user_query
- **History**: full_history_text
- **Current Observation**: current_step_raw_data
- **Candidate Actions**: layer_3_options   (four format-matched decoys
  are injected when Layer 3 is inapplicable)

## Layer 1 - Environment Cognition (choose exactly ONE)
- `NEW_FACT`   : the observation adds key info missing in history.
- `REDUNDANT`  : the observation repeats info already in history.
- `CONFLICT`   : the observation contradicts a confirmed fact.
- `NOISE`      : the observation is irrelevant, ads, or a dead-end.

## Layer 2 - Strategic Decision (choose exactly ONE; SEARCH and VISIT
## are scored as one SEARCH/VISIT class)
- `SEARCH`     : a gap remains; a new query is needed.
- `VISIT`      : a specific URL in the observation is worth opening.
- `STOP`       : the evidence already answers the user goal.
- `BACKTRACK`  : the current line of inquiry is invalid; change
                 direction.

## Layer 3 - Tactical Execution
- If Layer 2 is `SEARCH` or `VISIT`: pick the single best option from
  Candidate Actions and return its letter.
- If Layer 2 is `STOP` or `BACKTRACK`: return "N/A".
```

```
- Judge options on: does it target the identified gap, is it neither
  too narrow nor too broad, does it avoid re-asking something already
  answered, and is the source likely to be substantive.

## Rules
1. Think first inside <think> ... </think>.
2. The option order is randomized. Never choose by position; choose by
   content.
3. Do not invent facts that are absent from History or Current
   Observation.
4. Emit exactly one <answer> block containing valid JSON, nothing
   after it.

## Output Format
<think>
Compare the candidate labels and the candidate actions here.
</think>
<answer>
{
  "layer_1": {"label": "NEW_FACT|REDUNDANT|CONFLICT|NOISE",
              "why": "... rejected <alternative> because ..."},
  "layer_2": {"label": "SEARCH|VISIT|STOP|BACKTRACK",
              "why": "... rejected <alternative> because ..."},
  "layer_3": {"choice": "A|B|C|D|N/A",
              "why": "... better than <other option> because ..."}
}
</answer>
```

C.3 PROMPT C: STAGE-2 INFERENCE

This is the only prompt used during autonomous exploration and at evaluation. Note what is absent: no label vocabulary, no candidate options, no request for a per-level verdict, and no mention of perception, strategy, or tactics. The agent is asked only to think, act, and answer. It is also the prompt under which the vanilla-GRPO and SFT+GRPO baselines are trained, so the comparison in the main paper differs only in the initialization of the policy.

```
# Role: Autonomous Search Agent

## Task
Answer the user's question by interacting with a search environment.
You may take at most max_turns turns. In each turn, reason first, then
either call one tool or give the final answer.

## Question
user_query

## Tools
- <search>your query</search>
    Returns the top-5 retrieved snippets for the query.
- <visit>url</visit>
    Returns the cleaned main content of that page, truncated.

## Protocol
Each turn has this shape:

<think>
your reasoning for this turn
</think>
followed by EXACTLY ONE of:
<search>...</search>
<visit>...</visit>
<answer>...</answer>

The environment replies with <observation> ... </observation>.
- Never write an <observation> block yourself; only the environment
  produces them.
- Never emit two actions in the same turn.
- A tool that fails returns an error observation and still consumes a
  turn.

## Guidelines
- Search when a fact you need is missing; visit when a specific result
  looks like it holds the detail you need.
- Do not repeat a query whose answer is already in the transcript.
- If a line of inquiry is exhausted, change direction instead of
  rephrasing the same request.
- Stop as soon as the evidence is sufficient. Extra turns are not
```

```
  rewarded.

## Final Answer
Put the final answer inside <answer> ... </answer>. It must be short,
self-contained, and directly responsive to the question -- a span or a
single sentence, with no restatement of your reasoning.
```