EG-GRPO: Evidence-Guided Step-level Credit Assignment for Deep Search Agents

Anonymous authors
Paper under double-blind review

Abstract

LLM-based search agents demonstrate strong capabilities in long-horizon information seeking and reasoning, typically trained via reinforcement learning. However, existing approaches predominantly rely on trajectory-level outcome rewards, providing limited supervision for complex search and reasoning processes. Although step-level credit assignment offers a promising solution, obtaining reliable and task-grounded supervision for open-ended search remains challenging. In this paper, we propose **E**vidence-**G**uided **G**roup **R**elative **P**olicy **O**ptimization (**EG-GRPO**), a framework that derives step-level supervision from the evidence annotations available in synthetic deep search data. EG-GRPO identifies the first supporting citation for each required evidence claim and uses the newly acquired evidence to construct step-level process rewards. By representing the accumulated evidence as an evolving knowledge state, EG-GRPO further performs state-conditioned relative advantage estimation and combines the resulting step-level signal with the standard trajectory-level outcome advantage. Experiments on five challenging deep search benchmarks show that EG-GRPO generally outperforms standard GRPO and existing reward-enhanced and step-level RL baselines across different model scales. These results demonstrate that task-grounded evidence annotations can provide effective process supervision for credit assignment in deep search RL.

1 Introduction

As the capabilities of Large Language Models (LLMs) continue to advance, deep search agents have demonstrated immense potential in tackling knowledge-intensive tasks (OpenAI, 2025; Xi et al., 2025; Wang et al., 2024). Grounded in the ReAct paradigm, these agents iteratively gather evidence within noisy web environments by alternating between reasoning, tool utilization, and observation, ultimately providing well-supported answers (Yao et al., 2023). To further enhance their long-horizon information retrieval capabilities, Reinforcement Learning (RL) has emerged as the prevalent paradigm for training deep search agents. Existing work typically uses a binary outcome reward as the training signal on synthetic multi-hop QA datasets, comparing the agent's final answer against the ground truth (Jin et al., 2025; Gao et al., 2025; Li et al., 2025; Liu et al., 2025b).

In deep search scenarios, a complete trajectory typically comprises anywhere from several to over a dozen retrieval iterations, with each retrieval action potentially exerting a critical impact on the final answer formulation. However, the entire trajectory receives only a single binary scalar signal at its conclusion, leaving the model unable to discern which specific retrieval step played the decisive role (Shao et al., 2025). To alleviate this reward sparsity issue, several studies have attempted to introduce richer reward signals at the trajectory level (Gunjal et al., 2026; Lu et al., 2025). E-GRPO (Zhao et al., 2026) leverages entity annotations in synthetic data to formulate an entity matching rate as a fine-grained reward; CaRR (Zhang et al., 2026) further decomposes questions into single-hop rubrics, combining context-aware rubric rewards and outcome rewards in RL. Although these methods yield substantial improvements in their respective evaluation settings, they fundamentally still calculate a scalar reward for the entire trajectory. Consequently, the core issue of differentiated step-level credit assignment remains unresolved.

Step-level credit assignment provides a more direct approach to alleviating the aforementioned problem and has been extensively studied in sequential reasoning (Zhang, 2026; Guo et al., 2026; Wang et al., 2026). Process Reward Models (PRMs) provide step-level supervision over intermediate

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.

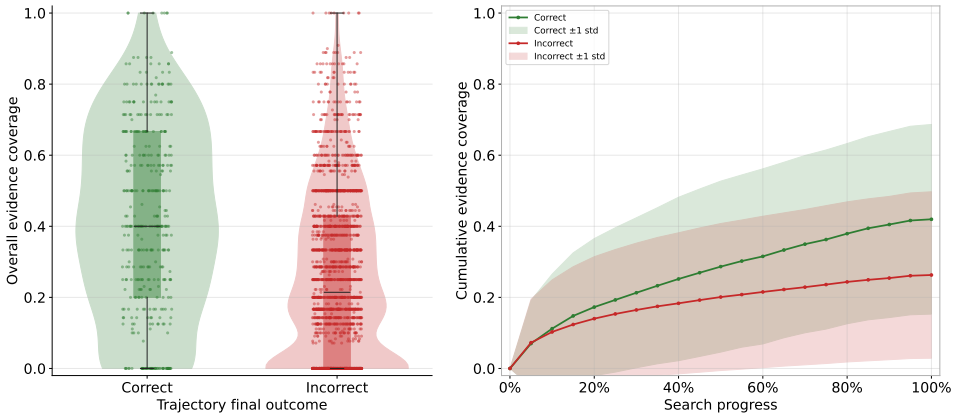

Figure 1: **Left:** Distribution of overall evidence coverage across correct and incorrect trajectories, showing that task success strongly correlates with higher coverage. **Right:** Trajectory-wise cumulative evidence coverage over normalized search progress, demonstrating that correct trajectories consistently accumulate evidence faster from the early search stages.

reasoning steps, offering a more direct training signal than outcome-only rewards (Lightman et al., 2024; Pala et al., 2025). Meanwhile, GiGPO (Feng et al., 2025) introduces state-based grouping to estimate step-level relative advantages for long-horizon LLM agents. Recently, STAMP (Xu et al., 2026) further extends step-level credit assignment to deep search agents by tracing evidence provenance and attributing credit to the first retrieval action that exposes supporting evidence. However, applying such approaches to open-ended deep search still faces two key challenges. First, search actions lack a well-defined notion of local correctness: a retrieval step may be valuable for discovering direct evidence, uncovering indirect clues, or narrowing the search space, making reliable and task-grounded step-level supervision difficult to obtain. Second, fine-grained credit assignment requires a meaningful representation of the evolving search state, whereas raw multi-turn search histories are highly heterogeneous and difficult to match reliably. These challenges motivate us to look beyond external process supervision and ask whether a task-grounded step-level signal is already available in the synthetic data used to train deep search agents.

We observe that such supervisory signals are inherently embedded in synthetic data pipelines. Current deep search datasets are often constructed through knowledge-graph random walks and information obfuscation, which produce structured annotations describing the factual evidence needed to answer each query (Lu et al., 2025; Du et al., 2026). Traditionally, only question-answer pairs are retained for reward computation, discarding this rich annotation. However, the annotated evidence claims provide a natural source of task-grounded step-level supervision, as they specify the factual information that a successful trajectory should progressively acquire. To test this hypothesis, we conducted an experiment on a subset of the training set (320 prompts) by sampling 8 trajectories per prompt and computing the evidence coverage rate for each trajectory. Figure 1 (left) shows that correct trajectories have significantly higher coverage (median 41.97% vs. 26.29%), confirming coverage as a strong predictor of final correctness. Figure 1 (right) further reveals that correct trajectories maintain higher cumulative coverage from the very beginning, and this gap widens as retrieval proceeds. These observations indicate that evidence acquisition is a continuous process signal highly correlated with task success. More importantly, the cumulative set of acquired evidence claims provides an explicit representation of the agent's evolving knowledge state, allowing rollouts to be compared under identical evidence states without manual process annotation. This observation motivates us to use task-grounded evidence annotations as a source of step-level supervision for deep search RL.

Building upon these insights, we propose **E**vidence-**G**uided **G**roup **R**elative **P**olicy **O**ptimization (**EG-GRPO**), a step-level credit assignment method for deep search agents. EG-GRPO transforms the evidence claims annotated in synthetic data into dense process rewards and combines them with GRPO's trajectory-level advantages, preserving outcome supervision while providing fine-grained guidance for intermediate retrieval actions. Specifically, an LLM-based judge identifies the first supporting citation for each required evidence claim, and newly supported claims at each step are converted into process rewards. We further define a cumulative knowledge state based on the

evidence claims acquired before each step and perform state-conditioned advantage normalization, enabling different rollouts to be compared under the same knowledge state. The resulting step-level advantages are then fused with trajectory-level advantages to produce token-level training signals. In this way, EG-GRPO provides task-grounded supervision for intermediate evidence acquisition while retaining the final outcome signal, enabling more precise credit assignment for deep search agents.

We systematically evaluate **EG-GRPO** on five deep search benchmarks: BrowseComp (Wei et al., 2025), BrowseComp-ZH (Zhou et al., 2025), GAIA (Mialon et al., 2023), xBenchDS-2505 and xBenchDS-2510 (Chen et al., 2025). EG-GRPO consistently outperforms the GRPO baseline as well as reward-enhanced and step-level baselines, demonstrating that step-level credit assignment provides a complementary training signal to trajectory-level rewards. Our main contributions are summarized as follows:

- Through step-level analysis of rollout trajectories, we reveal a strong correlation between the quality of incremental evidence acquisition and final answer correctness, with the most pronounced divergence emerging early in the search process. This provides empirical evidence for the need for step-level credit assignment in deep search RL.
- We propose **EG-GRPO**, which transforms task-grounded evidence claims from synthetic training data into dense step-level rewards and employs hierarchical fallback normalization for stable advantage estimation, without requiring additional human annotations or auxiliary model training.
- We conduct extensive experiments across five deep search benchmarks, demonstrating the effectiveness of EG-GRPO. Comprehensive ablation studies further validate the contribution of its key components.

2 Preliminary

2.1 Deep Search Agents

Multi-Turn Rollout. We adopt the ReAct paradigm (Yao et al., 2023) for deep search agents. Given a query q, an LLM agent policy π_θ iteratively interacts with an external environment, such as web search engines and browser tools, to gather factual evidence and derive the final answer. At each turn $t \in \{1, 2, \ldots, T\}$, the agent generates a textual reasoning trace $\tau_{i,t}$ and an action $a_{i,t} \in \mathcal{A}$ conditioned on the query and the interaction history:

$$(\tau_{i,t}, a_{i,t}) \sim \pi_\theta(\cdot \mid q, \mathcal{H}_{i,<t}), \tag{1}$$

where

$$\mathcal{H}_{i,<t} = (\tau_{i,1}, a_{i,1}, o_{i,1}, \ldots, \tau_{i,t-1}, a_{i,t-1}, o_{i,t-1}) \tag{2}$$

denotes the interaction history before turn t, and $o_{i,t}$ is the environment observation returned after executing $a_{i,t}$, such as retrieved search snippets or web page contents.

The multi-turn rollout terminates when the agent emits a stopping signal and produces a final answer y_i. We denote the resulting trajectory by

$$\tau_i = (\tau_{i,1}, a_{i,1}, o_{i,1}, \ldots, \tau_{i,T}, a_{i,T}, o_{i,T}, y_i). \tag{3}$$

2.2 Evidence Annotations in Synthetic Data

Synthetic Multi-Hop Data Generation. Modern deep search datasets (Lu et al., 2025; Du et al., 2026; Chu et al., 2026) leverage entity-centric knowledge graph traversal and information obfuscation to automatically construct complex multi-hop questions. During data generation, a structured sequence of factual relations is used to connect a seed entity to the target answer, yielding a set of intermediate factual claims required to resolve the query.

Evidence Claim Formalization. For each query q, we denote the annotated evidence claims produced during synthetic data generation as

$$C_q = \{c_1, \ldots, c_K\}, \tag{4}$$

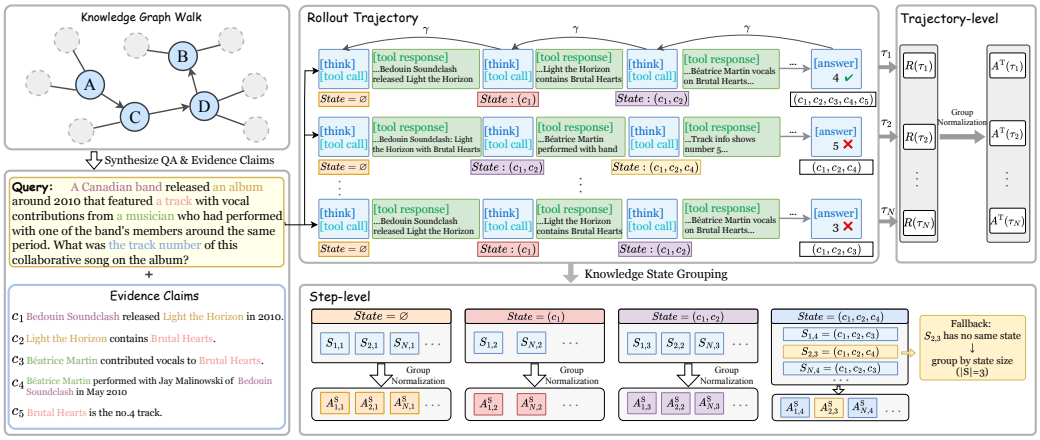

Figure 2: Overview of EG-GRPO. Step-level advantages (A^S) are computed through fine-grained knowledge-state grouping. Intermediate steps across different trajectories are aligned according to their cumulative knowledge states. A three-level fallback mechanism progressively relaxes the grouping criterion when exact states are sparse, enabling robust step-level advantage estimation.

where C_q contains K atomic evidence claims required to resolve q. Each claim c_k corresponds to an indivisible factual assertion. These claims are generated offline together with the synthetic training data and provide task-grounded supervision for evidence acquisition. In EG-GRPO, we use C_q to derive step-level evidence rewards and to construct the evolving knowledge state.

3 Methodology

3.1 Evidence Claims and Step-Level Rewards

Evidence Claim Construction. For each training query q, we obtain a set of atomic evidence claims

$$C_q = \{c_1, \ldots, c_K\} \tag{5}$$

from the synthetic data generation pipeline. These claims collectively specify the factual evidence required to resolve q. Each claim is assigned a uniform weight $1/K$. The evidence claims are used during training to provide task-grounded supervision for step-level evidence acquisition.

Evidence-to-trajectory attribution. Given a completed rollout τ_i, we first compress the tool responses into a set of numbered citation blocks. A single LLM judge reads the evidence claims C_q together with the citation contents and identifies, for each claim c_k, the earliest citation that explicitly supports it. By mapping each citation to its originating message and then to its corresponding turn, we obtain the first-exposure function:

$$\phi_i(c_k) = \min\{t : c_k \text{ is first supported at turn } t\}. \tag{6}$$

Each claim is therefore credited only at its earliest exposure turn, preventing repeated retrieval of the same evidence from receiving additional reward.

Incremental rewards and cumulative knowledge state. For each rollout τ_i, we define the *cumulative knowledge state* before turn t as the set of evidence claims already discovered:

$$S_{i,t} = \{c_k \in C_q : \phi_i(c_k) < t\}. \tag{7}$$

The newly discovered evidence at turn t is

$$\Delta_{i,t} = \{c_k \in C_q : \phi_i(c_k) = t\}, \tag{8}$$

and the corresponding step reward is

$$r_{i,t} = \frac{|\Delta_{i,t}|}{K} \in [0, 1]\,. \tag{9}$$

The trajectory-level evidence coverage is defined as

$$R_i^{\text{cov}} = \sum_{t=1}^{T} r_{i,t} = \frac{\left|\bigcup_{t=1}^{T} \Delta_{i,t}\right|}{K}. \tag{10}$$

Importantly, $S_{i,t}$ summarizes the evidence acquired by the agent before taking the action at turn t. Thus, two rollouts for the same query that share the same $S_{i,t}$ are aligned at the same knowledge state, providing a meaningful basis for state-conditioned advantage normalization (Section 3.3).

Special cases. If the final response is malformed or absent, we assign a terminal penalty $r_{i,T} = -1$. If the agent produces the final answer correctly without any further tool call, the final action is marked with a sentinel state $S_{i,T} = \{\texttt{SUMMARY_STEP}\}$. Rollouts that time out or encounter unrecoverable tool failures are excluded from policy updates.

3.2 Trajectory-Level Advantage: GRPO

For each query q, we sample N rollouts $\{\tau_i\}_{i=1}^{N}$ and apply standard GRPO group normalization to their outcome rewards. Denoting the outcome reward of trajectory τ_i by R_i^{out}, the trajectory-level advantage is

$$A_i^T = \frac{R_i^{\text{out}} - \text{mean}_{j=1}^{N} R_j^{\text{out}}}{\text{std}(\{R_j^{\text{out}}\}_{j=1}^{N})}. \tag{11}$$

The trajectory-level advantage A_i^T is broadcast to all decision-related tokens in trajectory τ_i, following the standard GRPO formulation.

3.3 Step-Level Advantage: Discounted Return and Three-Level Fallback Normalization

Discounted step return. To capture the delayed impact of evidence acquisition, we compute the discounted return from turn t:

$$G_{i,t} = \sum_{k=t}^{T} \gamma^{k-t} r_{i,k}, \tag{12}$$

where γ is the discount factor. When the terminal step carries a format-error penalty ($r_{i,T} < 0$), this penalty is retained at the terminal step but is not propagated backward through γ, preventing a formatting failure from contaminating the credit assigned to earlier evidence-acquisition steps.

Three-level fallback grouping. A naive strategy would normalize $G_{i,t}$ across all steps associated with the same query. However, steps taken under different knowledge states are not directly comparable: an action taken before any evidence has been acquired faces a different retrieval context from one taken after several claims have already been established.

We therefore normalize step returns using the finest available knowledge-state grouping and progressively relax the grouping criterion when the finer group contains insufficient samples. Specifically, we use the following three levels:

Level	Grouping key	Trigger condition
1 (exact state)	$(q,\ S_{i,t})$	$\geq M_1$ samples
2 (state size)	$(q,\ \lvert S_{i,t}\rvert)$	$\geq M_1$ samples
3 (query global)	q	≥ 2 samples

Table 1: Three-level fallback grouping for step-level advantage estimation.

Let $\mathcal{G}_{i,t}$ denote the first grouping level that satisfies its corresponding trigger condition for step (i,t). The step-level advantage is then computed by normalizing its discounted return within the selected group:

$$A_{i,t}^S = \frac{G_{i,t} - \text{mean}\left(\{G_{j,u} : (j,u) \in \mathcal{G}_{i,t}\}\right)}{\text{std}\left(\{G_{j,u} : (j,u) \in \mathcal{G}_{i,t}\}\right)}. \tag{13}$$

If the exact-state group contains insufficient samples, we fall back to the state-size group; if that group is also insufficient, we use the query-global group. The final level requires only two samples within the query group, ensuring that a normalization baseline is available for valid rollouts.

The hierarchy provides a trade-off between precision and statistical robustness. Level 1 compares steps taken from identical knowledge states and therefore provides the most fine-grained credit signal. However, exact states can be sparse because different rollouts often acquire evidence in different orders. Level 2 relaxes the criterion to states containing the same number of known claims, preserving a coarse notion of comparability while increasing the number of matched samples. Level 3 further falls back to all steps associated with the same query, ensuring broad coverage when finer-grained matches are unavailable.

After normalization, the step-level advantage is clipped to $[-C, C]$. The resulting scalar is broadcast to all decision-related tokens within the corresponding turn segment, providing a dense and forward-looking supervisory signal for policy optimization.

3.4 Training Objective

The two advantage streams provide complementary supervision: the trajectory-level advantage A_i^T captures whether the complete rollout successfully solves the query, while the step-level advantage $A_{i,t}^S$ provides finer-grained credit for evidence acquisition. We combine them into a per-token advantage:

$$A_{i,t} = A_i^T + w \cdot \text{clip}\left(A_{i,t}^S, -C, C\right), \tag{14}$$

where w controls the contribution of step-level supervision and C bounds the magnitude of the step-level signal.

The policy is optimized using a clipped surrogate objective:

$$\mathcal{J}_{\text{EG-GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{T}_i|}\sum_{t\in\mathcal{T}_i}\min\left(\rho_{i,t}(\theta)A_{i,t}, \text{clip}\left(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)A_{i,t}\right)\right], \tag{15}$$

where $\mathcal{T}_i$ denotes the decision-token positions in trajectory τ_i, and

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(x_{i,t} \mid x_{i,<t}, q)}{\pi_{\theta_{\text{old}}}(x_{i,t} \mid x_{i,<t}, q)} \tag{16}$$

is the token-level importance sampling ratio. Here, $x_{i,t}$ denotes a decision-related token, including reasoning, tool-call, or answer tokens.

The optimization masks out raw tool-response tokens, so gradients are propagated only through the agent's decision-related tokens. The same turn-level step advantage $A_{i,t}^S$ is assigned to all decision tokens within the corresponding turn segment.

4 Experiments

4.1 Experiment Setup

Tool server. The agent engages with the live web through two tools: the `search` tool, which queries the Serper API (ser, 2025) and retrieves top-10 snippets along with their associated URLs, and the `visit` tool, which extracts page content from a given URL via the Jina API (jin, 2025).

Baselines. We compare EG-GRPO with the following RL algorithms: (1) GRPO (Shao et al., 2024), which uses a binary outcome reward (0/1 score). (2) C-GRPO (Zhang et al., 2026), which combines a rubric-based citation score and outcome reward for training robust deep search agents. (3) GiGPO (Feng et al., 2025), which uses fine-grained credit assignment for LLM agents while preserving the appealing properties of group-based RL. (4) STAMP (Xu et al., 2026), which leverages provenance-guided supervision to perform fine-grained step-level credit assignment across search trajectories. Additionally, which leverages provenance-guided supervision to perform fine-grained step-level credit assignment across search trajectories. We also provide the reported scores of several state-of-the-art search agents for reference (see Appendix B for details).

Table 2: Overall performance of different agents on five challenging deep search benchmarks.

Model	Browse Comp	Browse Comp-ZH	GAIA	xBench-DS-2505	xBench-DS-2510
Advanced Agents with Proprietary Data					
GLM-4.7	67.5	66.6	61.9	72.0	52.3
Kimi-K2.5	74.9	62.3	–	–	46.0
DeepSeek-V3.2	67.6	65.0	75.1	78.0	55.7
Gemini 3.1 Pro	53.6	75.9	–	86.0	–
Tongyi-DeepResearch	43.4	46.7	70.9	75.0	–
Agents with Open-source Data					
DeepDive-32B	14.8	25.6	–	50.5	–
WebSailor-32B	10.5	25.5	53.2	53.3	–
WebExplorer-8B	15.2	32.0	50.0	53.7	–
OpenSeeker-30B	29.5	48.4	–	74.0	–
REDSearcher-30B-A3B	42.1	49.8	–	80.1	–
Our Agents					
Qwen-4B-SFT	5.7	13.7	32.7	49.0	14.0
+ GRPO	6.7	16.7	34.8	62.7	15.7
+ C-GRPO	7.0	20.1	38.2	64.0	13.7
+ GiGPO	7.0	16.6	34.6	54.7	11.3
+ STAMP	7.3	21.5	35.2	60.0	17.0
+ EG-GRPO (ours)	**8.0**	**23.5**	**45.9**	**65.0**	**19.0**
Qwen-30B-SFT	9.3	15.5	37.8	50.0	15.0
+ GRPO	10.3	22.4	42.7	57.6	20.1
+ C-GRPO	10.4	23.7	47.3	63.3	17.6
+ GiGPO	10.5	24.7	45.2	57.2	18.3
+ STAMP	10.8	24.1	45.2	62.5	**21.1**
+ EG-GRPO (ours)	**11.1**	**26.1**	**49.1**	**64.2**	20.8

Training Details. We evaluate EG-GRPO across two base model architectures: Qwen3-4B-Thinking-2507 (dense) and Qwen3-30B-A3B-Thinking-2507 (MoE) (Yang et al., 2025). The training pipeline consists of a cold-start Supervised Fine-Tuning (SFT) stage using 1,000 high-quality trajectories selected from OpenSeeker (Du et al., 2026), followed by RL training on 2,316 synthesized prompts. We employ Qwen-Flash (Yang et al., 2025) as the judge LLM for both final-answer reward scoring and step-level evidence verification. Full hyperparameters, training parameters, and hardware configurations are detailed in Appendix D.

Benchmarks and Evaluation Details. The evaluation of our trained agents is conducted on five challenging deep search benchmarks: BrowseComp (Wei et al., 2025), BrowseComp-ZH (Zhou et al., 2025), the text-only validation subset of GAIA (Mialon et al., 2023), xBenchDS-2505 and xBenchDS-2510 (Chen et al., 2025). We use Qwen-Flash (Yang et al., 2025) as judge LLM to determine if the agent's final response agrees with the ground truth. We evaluate each model at 128k context length.

4.2 Main Results

Table 2 summarizes the overall performance of EG-GRPO and various baselines across five challenging deep search benchmarks. Across both 4B and 30B model architectures, EG-GRPO generally outperforms the standard GRPO baseline as well as fine-grained reward-enhanced methods, achieving average accuracy gains of 4.96 and 3.64 over GRPO, respectively. Notably, GiGPO underperforms standard GRPO on both xBench-DS-2505 and xBench-DS-2510, suggesting that generic state-based grouping may be less effective for some long-horizon search tasks. In contrast, EG-GRPO demonstrates robust and consistent improvements; for instance, on the Qwen3-4B, it improves over

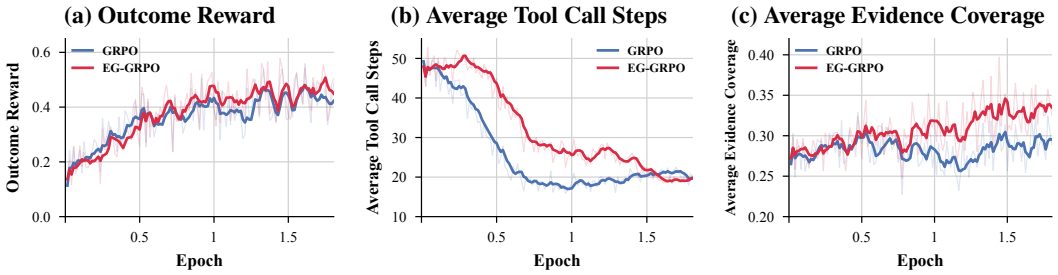

Figure 3: Training dynamics of GRPO and EG-GRPO on 4B models, including the changes of outcome rewards, average tool call steps and evidence coverage.

standard GRPO by 11.11 absolute points on GAIA (45.9 vs. 34.8). These results demonstrate that task-grounded evidence supervision provides an effective complementary signal to trajectory-level outcome rewards for deep search RL.

4.3 Analysis

Training Dynamics. Figure 3a demonstrates that both GRPO and EG-GRPO successfully optimize the outcome reward during training, with EG-GRPO exhibiting comparable or slightly better performance in later epochs. Concurrently, Figure 3b reveals a distinct difference in the agents' search behaviors. While the baseline GRPO shows a sharp decline in average tool calls, suggesting a tendency to terminate searches prematurely, EG-GRPO maintains a more gradual reduction. This indicates that step-level supervision helps mitigate the risk of the agent taking superficial shortcuts, encouraging sufficient interaction with the environment before formulating an answer.

Evidence Acquisition. The impact of this sustained exploration is reflected in the agents' evidence acquisition capabilities, as depicted in Figure 3c. While the standard GRPO model struggles to consistently increase its evidence coverage throughout training, EG-GRPO demonstrates a steady upward trend. This observation suggests that the evidence-guided step-level signals effectively guide the agent to progressively uncover the necessary factual evidence, rather than merely optimizing for the sparse outcome reward.

4.4 Ablation Studies

Weight of Step-Level Advantage. We investigate the balance between step-level evidence guidance and trajectory-level outcome supervision by varying the step advantage weight $w \in 0.0, 0.25, 0.5, 0.75, 1.0$. As shown in Table 3, introducing step-level guidance consistently improves performance over the $w = 0$ GRPO baseline, with $w = 0.5$ achieving the best results across all benchmarks. Notably, GAIA improves by 11.14% absolute from $w = 0$ to $w = 0.5$. However, further increasing w to 0.75 or 1.0 degrades performance, suggesting that excessive emphasis on local evidence acquisition can compromise global answer synthesis. Thus, $w = 0.5$ provides the best balance between evidence gathering and final task completion.

Table 3: Performance of EG-GRPO with different w.

Model	BC	BC-ZH	GAIA	xBenchDS-2505	xBenchDS-2510
Qwen3-4B-SFT	5.7	13.7	32.7	49.0	14.0
+ EG-GRPO(w=0)	6.7	16.7	34.7	62.7	15.7
+ EG-GRPO(w=0.25)	7.3	23.2	42.4	65.0	15.7
+ EG-GRPO(w=0.5)	**8.0**	**23.5**	**45.9**	**65.0**	**19.0**
+ EG-GRPO(w=0.75)	7.3	21.1	41.0	64.7	18.3
+ EG-GRPO(w=1.0)	7.0	23.2	37.2	64.2	17.0

Effect of Fallback Grouping Strategy. We evaluate the three-level fallback normalization by comparing the full strategy with two boundary cases: Level 1 only (exact-state grouping) and Level 3 only (prompt-global normalization). As shown in Table 4, the full strategy achieves the best overall performance across the five benchmarks, although prompt-global normalization slightly outperforms it on GAIA. Exact-state grouping suffers from sparse matches among divergent rollouts, while global normalization ignores differences in intermediate knowledge states and introduces noisier comparisons. The hierarchical fallback strategy combines the benefits of both, providing fine-grained state conditioning when sufficient samples are available while maintaining robust normalization through progressive fallback.

Table 4: Performance comparison of EG-GRPO under different fallback grouping strategies.

Model	BC	BC-ZH	GAIA	xBenchDS-2505	xBenchDS-2510
EG-GRPO (full)	**8.0**	**23.5**	45.9	**65.0**	**19.0**
EG-GRPO (level 3 only)	7.9	20.1	**47.1**	64.0	19.0
EG-GRPO (level 1 only)	7.9	22.8	39.4	62.0	18.3

5 Related Work

Reinforcement Learning for Deep Search Agents. Reinforcement learning has become an effective paradigm for training deep search agents to perform long-horizon information retrieval and reasoning. Early approaches such as Search-R1 (Jin et al., 2025) and DeepDive (Lu et al., 2025) employ outcome-based rewards on automatically constructed multi-hop search tasks, while subsequent systems further scale this paradigm through synthetic data generation and large-scale RL (Gao et al., 2025; Du et al., 2026; Chu et al., 2026). However, binary outcome rewards provide limited supervision for long search trajectories. Recent work therefore exploits auxiliary annotations from synthetic data to construct richer trajectory-level rewards. E-GRPO (Zhao et al., 2026) leverages ground-truth entity information to derive dense entity-aware rewards, while CaRR (Zhang et al., 2026) decomposes complex questions into verifiable single-hop rubrics and incorporates citation correctness into the reward. These approaches provide more informative supervision, but their rewards remain scalar signals assigned at the trajectory level.

Step-Level Credit Assignment. Process Reward Models (PRMs) provide intermediate-step supervision for mathematical reasoning (Zhang et al., 2025), while GiGPO (Feng et al., 2025) introduces state-based grouping to estimate step-level relative advantages for long-horizon LLM agents. More recently, STAMP (Xu et al., 2026) further assigns step-level credit through provenance tracing, identifying the first retrieval action that exposes supporting evidence. In contrast, EG-GRPO uses task-grounded evidence claims annotated in synthetic training data as the supervision units, rewarding newly acquired claims and representing their cumulative set as an explicit knowledge state for state-conditioned advantage estimation. This provides evidence-grounded process supervision without additional human process annotations or auxiliary reward-model training.

6 Conclusion

In this paper, we introduced EG-GRPO, a novel reinforcement learning framework that addresses the challenge of sparse rewards in training deep search agents by enabling step-level credit assignment. By leveraging task-grounded evidence claims available in synthetic training data, our method derives dense, task-grounded process rewards without requiring additional human process annotations. To ensure robust and fair advantage estimation, EG-GRPO aligns intermediate actions based on cumulative knowledge states and applies a hierarchical fallback normalization strategy. Extensive evaluations across multiple deep search benchmarks demonstrate that EG-GRPO consistently improves over standard trajectory-level RL and achieves strong performance against existing fine-grained baselines across different model scales. Ultimately, this step-level supervision effectively guides agents to sustain multi-turn exploration and systematically acquire the evidence needed to solve complex search tasks, paving the way for more capable and reliable long-horizon search models.

AI Use Statement

In this work, we used generative AI tools for LLM-based judging and annotation within the research pipeline, including final-answer correctness scoring, step-level evidence verification, evidence-claim matching, identification of the first supporting citation for each claim, and automated construction of synthetic evidence claims. Additionally, we used generative AI tools for code development assistance and language editing during manuscript preparation. We reviewed all AI-assisted work, manually verified the reliability of the LLM judge on sampled trajectories, and checked all generated data, annotations, judge outputs, code, and manuscript text. The authors designed the research and experiments, validated and quality-controlled the generated data and annotations, verified implementation correctness and experimental results, and interpreted the findings, and we take full responsibility for the final content of this work.

Reproducibility Statement

We provide sufficient detail to reproduce the proposed EG-GRPO training and evaluation pipeline. Section 3 specifies evidence-claim construction, evidence-to-trajectory attribution, incremental step rewards, cumulative knowledge states, three-level fallback grouping, and the final training objective. Section 4 describes the tool server, baselines, benchmarks, model scales, cold-start SFT and RL training setup, evaluation protocol, main results, analyses, and ablation studies. Appendix D reports the full SFT and RL configurations, including model initialization, optimization hyperparameters, hardware, batch sizes, context lengths, and EG-GRPO hyperparameters ($w = 0.5$, $\gamma = 0.95$, $C = 5$). Appendix E documents human verification of the LLM judge, and Appendix F provides the prompts used for outcome-reward scoring and evidence matching. We provide anonymized implementation code and scripts for reproducing the main experiments in the supplementary material.

References

Jina ai, 2025. URL https://jina.ai/.

Serper: Google search api, 2025. URL https://serper.dev.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, Chen Sun, Han Hou, Hui Yang, James Pan, Jianan Lou, Jiayi Mao, Jizheng Liu, Jinpeng Li, Kangyi Liu, Kenkun Liu, Rui Wang, Run Li, Tong Niu, Wenlong Zhang, Wenqi Yan, Xuanzheng Wang, Yuchen Zhang, Yi-Hsin Hung, Yuan Jiang, Zexuan Liu, Zihan Yin, Zijian Ma, and Zhiwen Mo. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations, 2025. URL https://arxiv.org/abs/2506.13651.

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

Yuwen Du, Rui Ye, Shuo Tang, Xinyu Zhu, Yijun Lu, Yuzhu Cai, and Siheng Chen. Openseeker: Democratizing frontier search agents by fully open-sourcing training data, 2026. URL https://arxiv.org/abs/2603.15594.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

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, 2025. 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/, 2026. Accessed: 2026-09-15.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations*, volume 2026, pp. 127924–127945, 2026.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *Advances in Neural Information Processing Systems*, 38:114399–114431, 2026.

Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, et al. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*, 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, 2025. URL https://arxiv.org/abs/2503.09516.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating super-human reasoning for web agent, 2025. URL https://arxiv.org/abs/2507.02592.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, Jiayuan Song, Zhengmao Zhu, Wenhu Chen, Pengyu Zhao, and Junxian He. Webexplorer: Explore and evolve for training long-horizon web agents, 2025b. URL https://arxiv.org/abs/2509.06501.

Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn rl, 2025. URL https://arxiv.org/abs/2509.10446.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants, 2023. URL https://arxiv.org/abs/2311.12983.

OpenAI. Deep research system card, 2 2025. URL https://deploymentsafety.openai.com/deep-research. Accessed: 2026-08-31.

Tej Deep Pala, Panshul Sharma, Amir Zadeh, Chuan Li, and Soujanya Poria. Error typing for smarter rewards: Improving process reward models with error-aware hierarchical supervision. *arXiv preprint arXiv:2505.19706*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Zhihong Shao, Yuxiang Luo, Chengda Lu, ZZ Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. Deepseekmath-v2: Towards self-verifiable mathematical reasoning. *arXiv preprint arXiv:2511.22570*, 2025.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Ziwei Chai, Y Charles, HS Che, Cheng Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 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.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, et al. Information gain-based policy optimization: A simple and effective approach for multi-turn search agents. In *International Conference on Learning Representations*, volume 2026, pp. 104502–104529, 2026.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), March 2024. ISSN 2095-2236. doi: 10.1007/s11704-024-40231-1. URL http://dx.doi.org/10.1007/s11704-024-40231-1.

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, 2025. URL https://arxiv.org/abs/2504.12516.

Yunjia Xi, Jianghao Lin, Yongzhao Xiao, Zheli Zhou, Rong Shan, Te Gao, Jiachen Zhu, Weiwen Liu, Yong Yu, and Weinan Zhang. A survey of llm-based deep search agents: Paradigm, optimization, evaluation, and challenges, 2025. URL https://arxiv.org/abs/2508.05668.

Ke Xu, Han Xu, Xinran Chen, Yuqian Wang, Zhixuan Li, Xiaojian Liu, Changwo Wu, Jianqiang Xia, and Yuchen Li. Stamp: Provenance-guided credit assignment for deep search agents, 2026. URL https://arxiv.org/abs/2607.11172.

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, 2023. URL https://arxiv.org/abs/2210.03629.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

Chenchen Zhang. From reasoning to agentic: Credit assignment in reinforcement learning for large language models. *arXiv preprint arXiv:2604.09459*, 2026.

Jiajie Zhang, Xin Lv, Ling Feng, Lei Hou, and Juanzi Li. Chaining the evidence: Robust reinforcement learning for deep search agents with citation-aware rubric rewards, 2026. URL https://arxiv.org/abs/2601.06021.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning, 2025. URL https://arxiv.org/abs/2501.07301.

Yida Zhao, Kuan Li, Xixi Wu, Liwen Zhang, Dingchu Zhang, Baixuan Li, Maojia Song, Zhuo Chen, Chenxi Wang, Xinyu Wang, Kewei Tu, Pengjun Xie, Jingren Zhou, and Yong Jiang. Repurposing synthetic data for fine-grained search agent supervision, 2026. URL https://arxiv.org/abs/2510.24694.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, Yuxin Gu, Sixin Hong, Jing Ren, Jian Chen, Chao Liu, and Yining Hua. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese, 2025. URL https://arxiv.org/abs/2504.19314.

A Trajectory Format

Our tool descriptions and trajectory format are shown as follows:

Tool Definitions

```
[
  {
    "name": "search",
    "description": "Performs web searches; the tool retrieves the top (default 10) results for each query in one call.",
    "parameters": {
      "type": "object",
      "properties": {
        "query": {"type": "array", "items": {"type": "string"}, "description": "Array of query strings. "},
        "num": {"type": "integer", "description": "Number of results to return"},
      },
      "required": ["query"]
    }
  },
  {
    "name": "visit",
    "description": "Visit webpages and return the summary of the content.",
    "parameters": {
      "type": "object",
      "properties": {
        "url": {"type": "array", "items": {"type": "string"}, "description": "The URL array of the webpages to visit."},
        "goal": {"type": "string", "description": "The goal of the visit for webpages"},
      },
      "required": ["url", "goal"]
    }
  }
]
```

Figure 4: Tool definitions for the search and visit tools.

Format

<think> thinking process here </think>
<tool_call>
{"name": "tool name", "arguments": {"parameter name": parameter value, "another parameter name": another parameter value, ...}}
</tool_call>
<tool_response>
tool_response here
</tool_response>
(more thinking processes, tool calls and tool responses here)
<think> thinking process here </think>
<answer> answer here </answer>

Figure 5: Multi-turn trajectory format for training and inference.

B Referred Deep Search Agents

For advance agents with proprietary data, we present the scores of GLM-4.7 (Zeng et al., 2025), Kimi-K2.5 (Team et al., 2026), DeepSeek-V3.2 (Liu et al., 2025a), Gemini 3.1 Pro (Google DeepMind, 2026), Tongyi-DeepResearch (Team et al., 2025) from their technical reports. For agents with open-source data, we present the score of DeepDive-32B (Lu et al., 2025), WebSailor-32B (Li et al., 2025), WebExplorer-8B (Liu et al., 2025b), OpenSeeker-30B (Du et al., 2026), REDSearcher-30B-A3B (Chu et al., 2026) from their previous paper.

C Data Synthesis Details

C.1 Synthetic QA Construction

We construct synthetic deep-search queries from structured factual paths derived from an entity-centric knowledge source. Starting from a seed entity, we traverse a multi-hop factual path to obtain a sequence of related entities and factual relations. The resulting path is then transformed into a natural-language question through entity obfuscation, where intermediate entities that should be recovered through search are replaced by descriptive clues while preserving the underlying factual dependencies.

This construction process produces questions that require the agent to progressively recover hidden information through multi-step web search rather than directly matching an entity mentioned in the query. The path length and the number of hidden entities are used to control the reasoning complexity of the synthesized questions.

C.2 Evidence Claim Extraction

The structured factual paths used for question synthesis are further converted into atomic evidence claims. For each synthesized query q, we obtain a set

$$C_q = \{c_1, \ldots, c_K\},$$

where each c_k is a self-contained factual assertion that can be independently verified from web evidence. Claims are normalized to a consistent granularity so that each claim represents a single verifiable fact required for resolving the query.

For example, for the query:

A European tour that concluded in a year when a film about a U.S. president was released, and which visited a continent where a writer published research on African literature containing the word 'polyphony' - what was the academic journal where that research appeared?

The original structured facts can be converted into claims such as:

- c_1: The Carpe Diem Tour concluded in 2012.
- c_2: The film *Lincoln* was released in 2012.
- c_3: The Carpe Diem Tour visited Europe.
- c_4: The Carpe Diem Tour visited Africa.
- c_5: Ken Walibora's research was related to Africa.
- c_6: Ken Walibora published research in *Research in African Literatures*.
- c_7: The relevant research article contains "polyphony" in its title.

These claims constitute the task-grounded evidence units used by EG-GRPO. Although the claims originate from the structured synthesis process, their internal dependency relations are not used by the training algorithm.

C.3 Claim Verification and Filtering

To ensure that the extracted evidence claims provide reliable supervision, we apply automatic verification and filtering to the synthesized data. Each claim is required to satisfy three criteria: (1) it represents a concrete factual assertion rather than a vague description; (2) it is independently verifiable from publicly available web evidence; and (3) it provides information that is relevant to resolving the corresponding query.

We further remove samples containing redundant, ambiguous, or unverifiable claims. Claims that require substantial inference beyond the retrieved evidence are also excluded. The resulting claim set is used as the task-grounded evidence annotation for subsequent RL training.

D Training Configuration

SFT. We filtered 1,000 high-quality SFT trajectories from the OpenSeeker (Du et al., 2026) dataset and adapted them to match the trajectory format used in this paper. Using this dataset, we fine-tuned both the 4B and 30B models for 1 epoch. The training was conducted on 8× NVIDIA H800 GPUs with a batch size of 32, a learning rate of 2×10^{-5}, and a maximum context length of 128k.

RL. Starting from the SFT checkpoints, we conducted RL training on 2,316 synthetic data across 16× NVIDIA H800 GPUs for 2 epochs, based on verl framework (Jiang et al., 2025). The training used a learning rate of 1×10^{-6} and a maximum context length of 64k. For experience collection, we set the prompt batch size to 32 and sampled 8 completions per prompt, yielding a global batch size of 256. Generation parameters during rollout were set to a temperature of 1.0, top_p = 0.95, top_k = 20, and a presence penalty of 1.1. For the EG-GRPO advantage calculation, we set the step-level advantage weight $w = 0.5$, discount factor $\gamma = 0.95$, and clipping threshold $C = 5$. We employed Qwen-Flash (Yang et al., 2025) as the judge LLM to perform both final-answer reward scoring and step-level evidence verification.

E Human Verification of LLM Judge

To assess the reliability of the LLM judge in identifying evidence acquisition within the EG-GRPO framework, we conducted a manual review of its judgments across 10 randomly sampled deep search trajectories. The sampled trajectories cover 67 evidence claims. Using human assessments as the gold standard, annotators independently verified whether each evidence claim was sufficiently supported by the retrieved information and, when supported, identified the first citation that provided sufficient evidence. We then compared the human annotations with the corresponding LLM judge outputs. The LLM judge achieves 91.0% accuracy in identifying whether evidence claims are supported and 94.0% accuracy in identifying the first supporting citation. These results indicate that the LLM judge can reliably identify evidence acquisition and its corresponding supporting citations in search trajectories.

F Prompts

Prompt for Outcome Rewards

Judge whether the following **[response]** to **[question]** is correct or not based on the precise and unambiguous **[correct_answer]** below.

[question]: {question}

[response]: {response}

Your judgement must be in the format and criteria specified below:

extracted_final_answer: The final exact answer extracted from the **[response]**. Put the extracted answer as 'None' if there is no exact, final answer to extract from the response.

[correct_answer]: {correct_answer}

reasoning: Explain why the **extracted_final_answer** is correct or incorrect based on **[correct_answer]**, focusing only on if there are meaningful differences between **[correct_answer]** and the **extracted_final_answer**. Do not comment on any background to the problem, do not attempt to solve the problem, do not argue for any answer different than **[correct_answer]**, focus only on whether the answers match.

correct: Answer 'yes' if **extracted_final_answer** matches the **[correct_answer]** given above, or is within a small margin of error for numerical problems. Answer 'no' otherwise, i.e. if there if there is any inconsistency, ambiguity, non-equivalency, or if the extracted answer is incorrect.

Figure 6: Prompt for outcome rewards.

Evidence Matching Prompt

You are an evidence matching expert. Given a list of evidence claims and citation content from a search trajectory, determine which citations first contain each piece of evidence.

Task

For each evidence claim, find the **earliest** citation number that explicitly contains the stated fact.
- Only count citations that explicitly contain the fact; the model's own inferences don't count.
- If multiple citations contain the same evidence, only take the smallest number (earliest discovery).
- If no citation contains a piece of evidence, its citation_ids should be an empty list.

Output Format

Matching Analysis of E1
Explanation: {{Clearly explain whether the citation content explicitly contains the fact stated in E1, citing relevant parts}}
Evidence Index: {{List the citation numbers that contain this evidence}}
Found: {{yes/no}}

Matching Analysis of E2
Explanation: {{Clearly explain whether the citation content explicitly contains the fact stated in E2, citing relevant parts}}
Evidence Index: {{List the citation numbers that contain this evidence}}
Found: {{yes/no}}

...

Final JSON-format Summary
{{
"E1": [3],
"E2": [5],
"E3": []
}}

Key is the evidence ID, value is a list of supporting citation numbers (starting from earliest discovery).

—

[Begin of Evidence List]
{claims_text}
[End of Evidence List]

[Begin of Citation Content]
{citation_content}
[End of Citation Content]