Reinforcing Agentic Search Via Reward Density Optimization

Anonymous ACL submission

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) is a promising approach for enhancing agentic search. However, its performance is often hindered by reward sparsity, whereby agents receive very limited positive feedback despite incurring significant exploration costs. In this paper, we formalize this challenge as a new research problem termed **Reward Density Optimization**, which aims to improve the reward obtained per unit of exploration cost. To address this problem, we introduce InfoFlow, a systematic framework that operates along three complementary dimensions: 1) **Sub-goal Scaffolding**: which decomposes long-horizon tasks into intermediate objectives and assigns process-level rewards to provide denser learning signals; 2) **Pathfinding Hints**: which injects corrective guidance into stalled trajectories to increase the ratio of successful trials; and 3) **Dual-agent Refinement**: which employs a dual-agent architecture to offload the cognitive burden of deep exploration. We evaluate InfoFlow on several popular agentic search benchmarks, where it significantly outperforms strong baselines and enables lightweight LLMs to achieve performance comparable to that of advanced proprietary models.

1 Introduction

Large language models (LLMs) have become essential tools for information seeking in the daily life (Zhao et al., 2023; Gao et al., 2023). As their applications expand, users increasingly expect LLMs to handle not only factual queries but also complex, multi-step tasks requiring knowledge discovery and synthesis. However, because an LLM's internal knowledge is limited and quickly outdated, relying solely on parametric memory is insufficient for knowledge-intensive tasks (Vu et al., 2023; Luo et al., 2024). Addressing such challenges requires integrating external knowledge sources and moving beyond surface-level retrieval toward deeper reasoning and information synthesis (Shi et al., 2023). Most existing approaches follow the retrieval-augmented generation (RAG) paradigm (Gao et al., 2023), which treats the input as a query and retrieves relevant documents for generation. While effective for factual questions, RAG struggles with hierarchical or implicit information needs (Asai et al., 2023; Qian et al., 2025). Extensions such as query rewriting, iterative retrieval, and self-refinement (Ma et al., 2023; Jiang et al., 2023; Madaan et al., 2023) improve flexibility but remain bound to a *pre-inference* design that retrieves information before reasoning begins, limiting adaptability in dynamic, multi-step tasks.

Inspired by reasoning-centric models (OpenAI, 2024; DeepSeek-AI, 2025), recent studies adopt the *search-integrated reasoning* paradigm (Yao et al., 2023; Chen et al., 2025a; Xue et al., 2025; Huang et al., 2025), which interleaves reasoning and search to adaptively incorporate external knowledge at each step (Li et al., 2025d; Jin et al., 2025b; Li et al., 2025c). However, current LLMs lack native mechanisms to invoke external search tools. Early implementations relied on manually crafted prompts and exhibited limited generalization (Li et al., 2025a). To overcome this, *Reinforcement Learning with Verifiable Rewards* (RLVR) has emerged as an effective approach for training LLMs to conduct agentic search, enhancing the capability of search interleaved reasoning via on-policy rollouts and final reward-driven optimization (Jin et al., 2025b,a).

Despite its promise, the application of RLVR to deep research is hindered by **reward sparsity**, whereby agents receive only limited positive feedback after incurring high exploration costs (formally defined in Eq. 1). Deep search tasks typically require *long-horizon, multi-turns* of interleaved reasoning and search before giving a final answer. Longer context imposes a heavier cognitive burden on the model and accumulates inter-

mediate reasoning errors, substantially increasing the likelihood of an incorrect final answer. Consequently, even after multiple attempts for the same task sample, on-policy rollouts often fail to produce any correct outcomes, rendering many samples ineffective for policy updates. Moreover, recent work highlights that training robust search agents *requires more complex, reasoning-intensive tasks* (Xia et al., 2025; Tao et al., 2025; Bae et al., 2025; Yan et al., 2025a). However, our preliminary experiments (Fig. 2) show that on difficult tasks successful rollouts become rare (less than 10% of initial accuracy), further reducing reward density and increasing computational inefficiency.

To address these issues, we formulate **Reward Density Optimization** and propose **InfoFlow**, a reinforcement learning framework that improves reward accessibility and stabilizes learning in search-integrated reasoning. InfoFlow increases reward density and learning efficiency via three core components: **(1) Sub-goal Scaffolding.** To make deep search more tractable for agents with limited initial capabilities, InfoFlow decomposes complex search queries into sub-goals and awards intermediate rewards for solving them. Deep search tasks naturally exhibit hierarchical structure: reaching the final answer typically requires identifying intermediate key facts or anchor entities. Rather than assigning rewards only for full task success, InfoFlow grants partial rewards for resolved sub-goals, providing denser feedback for policy updates. **(2) Pathfinding Hints.** To guide agents toward full solutions, InfoFlow incorporates explicit guidance during RL exploration in the form of pathfinding hints. We employ LLM (Gemini 2.5 and Qwen3-8B (Gemini Team, 2025; Yang et al., 2025)) as annotators to enrich training data (§ D) by generating search queries that guide the agent toward reaching key sub-goals. When the agent struggles to reach final answers within a predefined turns during on-policy rollouts, InfoFlow inserts guiding queries into the next turn to suggest more informative search directions. These pathfinding hints make intermediate key facts and anchor entities easier to discover, increasing sub-goal rewards and the likelihood of a correct final answer. They also help the agent learn improved search strategies via learning from expert demonstrations. **(3) Dual-agent Refinement.** To reduce the cognitive burden associated with long trajectories, InfoFlow adopts a dual-agent design for deep search. A *research agent* performs reasoning and search, while a *refiner agent* condenses retrieved information into concise, structured summaries that are fed back to the research agent. We observe up to 59.5% higher initial rewards, 16.4% reduced inference time, and 44.8% shorter trajectories (§ 3.1), substantially increasing reward density. Together, these techniques enable InfoFlow to overcome the reward sparsity bottleneck in RLVR.

We evaluate InfoFlow on a suite of knowledge-intensive agentic search benchmarks as well as the challenging complex benchmark BrowseComp-Plus (Chen et al., 2025b). Experimental results demonstrate that our method consistently outperforms strong baselines. Notably, on BrowseComp-Plus, our optimized small-scale model achieves performance comparable to much larger LLMs. Our main contributions are summarized as follows: (1) We propose InfoFlow, a dual-agent framework for agentic search, where a researcher agent is responsible for central reasoning and planning, while a refiner agent synthesizes retrieved evidence into coherent knowledge. (2) We introduce a well-tailored reward density optimization strategy, comprising sub-goal reward shaping and off-policy pathfinding hints. (3) Through extensive experiments, we verify the effectiveness of InfoFlow. In particular, on the challenging BrowseComp-Plus benchmark, InfoFlow enables a small-scale model to achieve performance competitive with much larger LLMs.

2 Preliminaries and Data Preparation

2.1 Agentic Deep Search

We address the task of **Deep Search Question Answering (DSQA)**, which involves solving complex queries that require multi-step information seeking and synthesis (Wei et al., 2025). Following Xia et al. (2025), such tasks can be represented as a reasoning tree where nodes are sub-problems and edges denote logical dependencies. Task complexity is defined by the tree's *depth* (sequential reasoning) and *width* (parallel information gathering). *Detailed task description is in § B.*

We formalize the LLM agent's problem-solving process as a Markov Decision Process (MDP). A trajectory is a sequence $\tau = (q, a_0, i_0, \ldots, a_{K-1}, i_{K-1}, a_K)$, where q is the initial question. At each step k, the agent in state S_k (the history) generates an action a_k, which includes a reasoning trace (*thinking*) and a set of parallel search queries (*searching*). The environment returns retrieved information

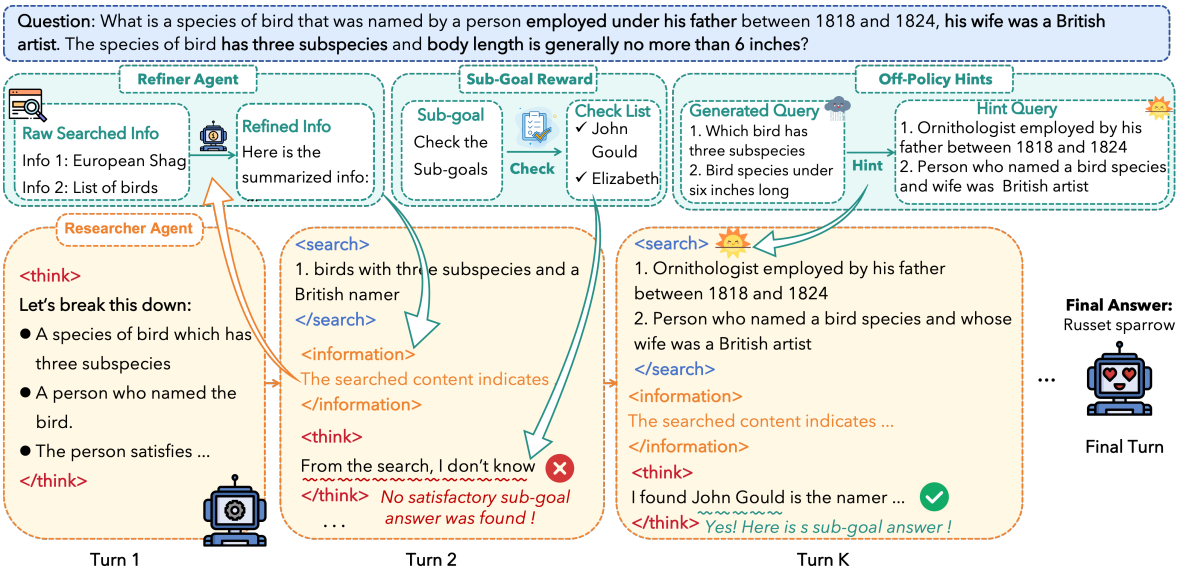

Figure 1: The framework of InfoFlow and example of DSQA task. Researcher agent focuses on reasoning and planning, refiner agent synthesizes massive searched content into condensed info.

i_k, leading to the next state S_{k+1}. The process terminates with an action a_K containing the final answer. A final reward $R(\tau)$ is given based on the answer's correctness. The agent's goal is to learn a policy $\pi(a|S)$ that maximizes the expected return. *Detailed task formulation is in § C.*

2.2 Data Preparation with Enriched Process Information

Reinforcement learning (RL) for agentic search is often hindered by sparse rewards, particularly in complex, multi-step search tasks where successful outcomes are rare. This scarcity of feedback renders policy gradient methods ineffective, as they learn little from predominantly unsuccessful exploratory trajectories (Dong et al., 2025; Jin et al., 2025b; Sun et al., 2025).

To address this challenge, we introduce dense, process-level supervision by augmenting the *InfoSeek* dataset (Xia et al., 2025). Unlike datasets such as Natural Questions or HotpotQA (Kwiatkowski et al., 2019a; Yang et al., 2018) that focus on few-hop reasoning, InfoSeek is designed for multi-step information seeking, providing a more suitable foundation for our work. We enrich its 18,000 training instances with two forms of off-policy supervision generated via LLMs, designed to facilitate the RL strategies detailed in § 3. Empirically, both proprietary (e.g., Gemini 2.5 (Gemini Team, 2025)) and open-source models (e.g., Qwen3-8B (Yang et al., 2025)) are capable of effectively performing this annotation task (§ 4.2.3).

First, we provide **Sub-goal Scaffolding**. For each problem, we employ an LLM to identify critical entities in the reasoning tree, designating them as mandatory sub-goals. To avoid reward hacking, we prompt LLM to select only important nodes such as milestones necessary for solving the full problem. Only these selected nodes receive sub-goal reward. This process yields a ground-truth set $\mathcal{G}_q = \{g_1, \ldots, g_M\}$, where each sub-goal g_i is assigned a normalized importance weight s_i reflecting its contribution, with $\sum_{i=1}^{M} s_i = 1$. The resulting set of weighted sub-goals, $\{(g_i, s_i)\}_{i=1}^{M}$, enables a granular reward shaping scheme that encourages structured task decomposition.

Second, to lower the exploration barrier at long-horizon multi-turn search interleaved reasoning process, we generate **Pathfinding Hints** for critical and difficult edges in the reasoning tree. These edges are judged by its importance for solving the whole problem and its difficulty to prepare suitable query for effective search. These hints are formulated as high informative **guiding queries** that decompose complex constraints into actionable search steps. They are designed to teach the agent effective strategies for overcoming non-obvious reasoning bottlenecks, serving as adaptive off-policy guidance to mitigate unproductive exploration during on-policy RL (Yan et al., 2025a; Zhang et al., 2025; Wu et al., 2025b). *Further details on data construction and examples are available in § D.3 & § D.4.*

3 Method

We formally define reward density as the expected reward obtained per unit exploration cost, which reflects how efficiently a search agent transforms

exploratory computation into verifiable learning signals. Given a dataset of n deep search QA instances, each solved by a leading search agent coupled with an external search engine, we conduct k rollouts per instance under a non-zero sampling temperature to ensure exploration diversity. For the j-th rollout of instance i, we denote the final reward as $r_{i,j} \in \{0, 1\}$, indicating correctness of the final answer, and the trajectory length as $l_{i,j}$, representing the trajectory length of the search agent. The reward density τ is computed as:

$$\tau = \frac{\sum_{i=1}^{n} \sum_{j=1}^{k} r_{i,j}}{\sum_{i=1}^{n} \sum_{j=1}^{k} l_{i,j}}. \quad (1)$$

Reward density is the key to the efficiency and scalability of Reinforcement Learning (RL) stage. Higher τ provides stronger and more stable gradient signals for policy optimization in RL. *We provide theoretical analysis on how reward density affects gradient variance in § E.*

InfoFlow addresses the challenge of *low reward density* in deep search training by formulating learning as a **Reward Density Optimization** problem. We enhance the reward density through three comprehensive and complementary mechanisms: (i) **Sub-goal Scaffolding** (§ 3.3), (ii) **Pathfinding Hints** (§ 3.4), and (iii) **Dual-agent Refinement** (§ 3.1 & § 3.2).

3.1 Dual-agent Refinement

The cognitive burden of managing long, noisy trajectories in deep search is a key driver of low reward density. To mitigate this, our framework (Figure 1) decouples this process into a ***Researcher Agent*** (π_θ) for planning and exploration, and a ***Refiner Agent*** ($\mathcal{F}_\phi$) for information synthesis.

The *Researcher* navigates the reasoning tree by generating actions $a_k = a_k^{\text{think}} \circ a_k^{\text{search}}$, where a_k^{search} can issue parallel queries $\{q_{k,j}\}_{j=1}^{N_k}$ to explore multiple lines of inquiry. For each query, the *Refiner* (driven by a LLM described in § J.1) processes the resulting noisy evidence $e_{k,j}$ and distills it into a concise summary: $sum_{k,j} = \mathcal{F}_\phi(q, q_{k,j}, e_{k,j})$. These summaries form the structured information $i_k = \{(q_{k,j}, sum_{k,j})\}_{j=1}^{N_k}$ that updates the researcher's state to S_{k+1}.

The advantage of the decoupled architecture lies in its ability to enhance the **reward density** (higher accuracy with less context length), which lays the foundation for later stable on-policy RL. As shown in Figure 2, we conduct experiments on InfoSeek

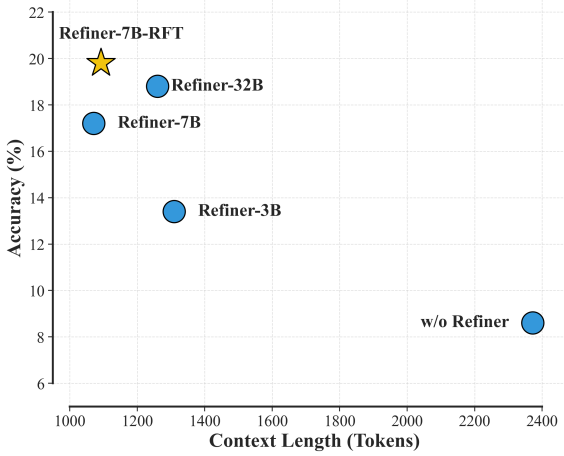

Figure 2: Dual agent framework enhances reward density: achieving higher accuracy with less context.

evaluation set using Qwen2.5-3B-Instruct as the researcher, and compare varying refiner configurations. The 3B refiner improves the success rate by 5.0 points while reducing the researcher's context length by nearly 45% (from 2372 to 1310 tokens). The context reduction frees up the researcher's limited context window to focus on high-level reasoning and planning rather than being overwhelmed by verbose, unprocessed evidence. More detailed efficiency analysis is conducted in § F.

3.2 Rejection Sampling Fine-tuning for Reward-Dense Initialization

Preliminary experiments in Figure 2 show less than 10% accuracy for untrained agents, yielding extremely sparse rewards. To mitigate this cold-start issue, we construct a high-quality corpus using rejection sampling (Xiong et al., 2025) and use it to jointly fine-tune both the Researcher and Refiner.

Trajectory collection and verification. We start from 18,000 DSQA tasks in the *InfoSeek* dataset (Xia et al., 2025). Using the base dual-agent framework (Qwen2.5-7B-Instruct for both roles), we perform two rollouts per task and retain only trajectories that produce correct final answers. We then apply a powerful verifier (Gemini-2.5-Pro (Gemini Team, 2025)) to filter out trajectories that succeed by chance or contain flawed reasoning; the final corpus contains ≈3,450 high-quality trajectories. This corpus encodes step-level reasoning and search-grounded evidence, providing dense supervised signals absent in standard pretraining data. *We conduct analytical experiments on the robustness of RFT data in § G.*

Joint fine-tuning objective. We co-train the Researcher policy π_θ and the Refiner $\mathcal{F}_\phi$ on the

Configuration	Mean@4	Solve None(%)	Context (Tok.)	Search Calls
Base Agents (researcher-3B + refiner-7B)	17.2	76.7	1071.2	2.83
+ RFT on researcher only	31.0	50.3	2489.3	3.92
+ RFT on Both (Co-training)	34.3	46.0	2612.0	4.17
+ Simple Prompt	35.0	45.6	2650.6	4.35
+ Pathfinding Hints	39.1	42.1	2892.7	4.72

Table 1: Analytical experiments on InfoSeek eval set. Co-training (RFT on both) and pathfinding hints improves Mean@4 and reduces the fraction of unsolved ("Solve None") samples.

verified trajectories. The Researcher is trained with token-level negative log-likelihood on demonstrated actions:

$$\mathcal{L}_{\text{SFT}}^{\text{researcher}}(\theta;\tau) = -\sum_{k=0}^{K}\sum_{t=1}^{|a_k|}\log \pi_\theta(a_{k,t} \mid S_k, a_{k,<t}), \tag{2}$$

while the Refiner is trained to map raw evidence to compact summaries:

$$\mathcal{L}_{\text{SFT}}^{\text{refiner}}(\phi;\tau) = -\sum_{k=0}^{K-1}\sum_{j=1}^{N_k}\log P_\phi(\mathit{sum}_{k,j} \mid q, q_{k,j}, e_{k,j}). \tag{3}$$

Joint RFT yields a substantially higher initial success rate and reduces trajectory verbosity, making subsequent RL more stable. Empirical comparisons are reported in Table 1.

3.3 Sub-goal Scaffolding

After the RFT initialization, we conduct RLVR to further enhance deep search capability of InfoFlow. The sparse-reward challenge in deep search RL arises from both task complexity and outcome reward. Single binary reward for final answer offers limited guidance for the intermediate steps of a long reasoning trajectory, particularly when early-stage success rates are low.

To provide informative learning signals inside long trajectories, we decompose complex question into a set of weighted sub-goals $\{(g_i, s_i)\}_{i=1}^{M}$ (e.g., find anchor entities, verify key facts) as described in § 2.2. For a trajectory τ, let $\mathcal{G}_{\text{solved}}(\tau)$ denote the sub-goals resolved by checking the Exact Match (EM) of the ground-truth entities within the agent's reasoning trace. The total reward is

$$R(\tau) = \max\left(R_{\text{final}}(\tau) + w \cdot \sum_{g_i \in \mathcal{G}_{\text{solved}}(\tau)} s_i,\ 1\right) \tag{4}$$

where w is a hyperparameter balancing outcome and process (we use $w = 0.3$). This shaped reward provides gradient information for partially correct trajectories and encourages decomposed reasoning. *We provide theoretical analysisin § E.1*

3.4 Pathfinding Hints

While sub-goal rewards densify the learning signal, on-policy exploration alone remains a bottleneck for the more challenging problems. Even after RFT, a significant portion of difficult samples are never solved through multiple rollouts (as suggested by the solve none ratio with four rollouts in Table 1), hindering any signal to policy gradient updates. This is because the agent can become trapped in unproductive exploration loops, failing to discover the critical reasoning paths necessary for success.

To overcome this exploration barrier, we introduce *pathfinding hints* to provide help during on-policy rollouts. We leverage the guiding queries prepared in § 2.2, which are high-leverage search actions designed to bridge difficult logical steps. The pathfinding hints injection is triggered when a trajectory exceeds a predefined turn threshold, K_h, without reaching a terminal state. At this step ($k = K_h$), the executed action a'_k is constructed by combining the agent's original reasoning trace a_k^{think} with the pre-constructed hint queries $a_{k,\text{hint}}^{\text{search}}$:

$$a'_k = a_k^{\text{think}} \circ a_{k,\text{hint}}^{\text{search}}. \tag{5}$$

The agent then receives the information retrieved using these hint queries and continues its trajectory from the new state. We set $K_h = 5$ in practice.

As shown in Table 1 (bottom), to validate the effectiveness of pathfinding hints, we conduct a comparative analysis on the InfoSeek evaluation set before RL training. We compare our approach with a baseline heuristic that simply prompts the agent to "Wait, let's try again from another angle." The results demonstrate that pathfinding hints clearly yields higherMean@4 and fewer unsolved cases. We conduct further analysis on the effects of pathfinding hints during RL training (§ 4.3.2) and no hints during inference after RL (§ 4.2.2).

This mechanism offers two-fold benefits. First, as an exploration corrective, it rescues the agent from unproductive loops, increasing the yield of successful trajectories for policy optimization. Second, as an explicit demonstration, it exposes the

Model	NQ	TQA	PopQA	HQA	2Wiki	MSQ	Bamb	Avg.
Qwen2.5-3B Based Models								
Search-o1-3B	23.8	48.2	26.2	22.1	21.8	5.4	32.0	25.6
Search-R1-3B	40.8	59.1	42.8	30.8	31.1	8.4	13.0	32.3
ZeroSearch-3B	41.2	61.5	44.0	31.2	33.2	12.6	14.3	34.0
AutoRefine-3B	43.6	59.7	44.7	40.4	38.0	16.9	33.6	39.6
InForage-3B	42.1	59.7	45.2	40.9	42.8	17.2	36.0	40.6
InfoFlow-3B	**44.5**	**63.7**	**47.0**	**44.6**	**45.2**	**21.0**	**41.2**	**43.9**
Qwen2.5-7B Based Models								
Self-RAG-7B	36.4	38.2	23.2	15.7	11.3	3.9	5.6	19.2
Search-o1-7B	27.7	47.4	29.4	34.8	35.6	4.8	15.2	27.1
Searcn-R1-7B	38.3	59.3	39.9	37.6	31.7	15.1	38.1	37.0
ZeroSearch-7B	43.6	65.2	**48.8**	34.6	35.2	18.4	27.8	39.1
ParallelSearch-7B	46.2	62.8	42.9	42.9	42.4	19.7	41.1	42.5
InfoFlow-7B	**47.2**	**68.1**	48.1	**44.3**	**47.2**	**21.9**	**47.6**	**46.2**

Table 2: Performance comparison on QA tasks with agentic search methods. The best result in each column is highlighted in **bold**.

agent to an informative off-policy search action and its positive outcome for better learning. *We provide theoretical analysis on how pathfinding hints improve exploration stability in § E.2*

3.5 Policy Optimization

We fine-tune the researcher via reinforcement learning that integrates the shaped reward $R(\tau)$ and hint-guided exploration after RFT. We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a PPO-style algorithm that normalizes advantages within trajectory groups to reduce variance. For a batch of G trajectories $\{\mathcal{Y}_i\}$ with returns $\{R_i\}$, the group-normalized advantage is $A_i = \frac{R_i - \text{mean}(\mathbf{R})}{\text{std}(\mathbf{R})}$, and the GRPO objective is:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\min\Big(r_i A_i, \text{clip}(r_i, 1-\epsilon, 1+\epsilon)A_i\Big) - \beta \mathrm{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\Bigg], \tag{6}$$

where r_i is the importance ratio and π_{ref} is a reference policy used for KL regularization.

4 Experiments

In this section, we empirically validate InfoFlow. Our experiments are designed to demonstrate that by systematically optimizing for *reward density*, our framework achieves strong performance and generalization for agentic search tasks, particularly on complex deep search tasks.

4.1 Experimental Setup

Datasets and Evaluation Metrics. To assess the **general information-seeking and agentic search** capability, we test InfoFlow on a suite of widely-used single-hop and multi-hop QA benchmarks with external search corpus: Natural Questions (NQ) (Kwiatkowski et al., 2019b), TriviaQA (TQA) (Joshi et al., 2017), PopQA (Mallen et al., 2022), HotpotQA (HQA) (Yang et al., 2018), 2WikiMultihopQA (2Wiki) (Ho et al., 2020), Musique (MSQ) (Trivedi et al., 2022), and Bamboogle (Bamb) (Press et al., 2022). We use E5 (Wang et al., 2024) as the embedding model, the 2018 Wikipedia dump (Karpukhin et al., 2020) as the corpus, and set the number of retrieved passages to 3. We report Exact Match (EM) as the metric for these datasets. To evaluate **deep search capability**, we employ the BrowseComp-Plus benchmark (Chen et al., 2025b), a refined version of BrowseComp (Wei et al., 2025) with 830 challenging problems and a fixed 100K webpage corpus. This benchmark is an ideal testbed for DSQA as its problems inherently demand the deep, iterative reasoning and search. Following the official implementation, accuracy is judged by an LLM (we use deepseek v3.1 (DeepSeek-AI, 2024) to judge).

Baselines and Implementation Details. We compare InfoFlow against recent agentic search methods, including Self-RAG (Asai et al., 2023), Search-o1 (Li et al., 2025d), Search-R1 (Jin et al., 2025b), Zero-Search (Sun et al., 2025), AutoRefine (Shi et al., 2025), InForage (Qian and Liu, 2025), and ParallelSearch (Zhao et al., 2025). These methods employ multi-turn interactions but differ in their training strategies and agentic framework. For the complex BrowseComp-Plus bench-

Model	Accuracy (%)	Search Calls
Gemini 2.5 Flash	15.5	10.6
Gemini 2.5 Pro	19.0	7.4
Sonnet 4	14.3	10.0
GPT-4.1	14.6	11.2
GPT-5	55.9	23.2
Qwen3-32B	3.5	0.9
SearchR1-32B	3.9	1.8
InfoFlow-3B	18.5	8.1
InfoFlow-7B	23.2	7.9

Table 3: Performance and search calls on the complex BrowseComp-Plus benchmark.

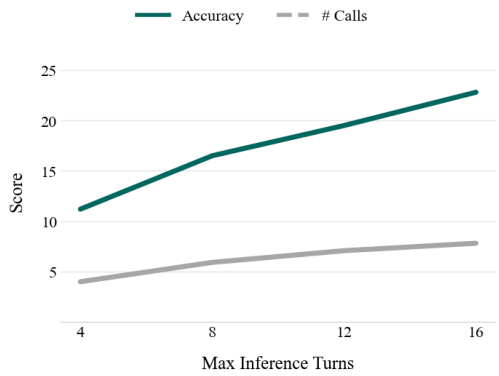

Figure 3: Analysis of reasoning depth on the complex BrowseComp-Plus benchmark.

mark, we include proprietary models like Gemini 2.5 Pro (Comanici et al., 2025), Sonnet 4 (Anthropic, 2025), GPT-5 (OpenAI, 2025), and larger open-sourced Qwen3-32B (Yang et al., 2025) and Search-R1-32B (Jin et al., 2025b). Our model is initialized with the framework described in § 3.1. For InfoFlow-3B/7B, we use Qwen2.5-3B-Instruct/Qwen2.5-7B-Instruct (Group, 2025) as initialization for researcher agent respectively. We use Qwen2.5-7B-Instruct as initialization for refiner agent. Training details are provided in § I.

4.2 Main Results

4.2.1 InfoFlow Demonstrates Superior Generalization on QA tasks

As shown in Table 2, InfoFlow demonstrates strong performance and generalization ability on standard agentic search and information-seeking benchmarks, outperforming all baseline models. Notably, InfoFlow maintains robust and transferable performance without training on in-domain datasets, unlike baseline methods which rely on in-domain NQ and HQA training data. This result highlights the effectiveness of our reward density optimization approach with the enriched InfoSeek dataset, which encourages more resilient and generalizable reasoning by providing dense, process-level rewards. These rewards enable the model to capture the compositional structure of multi-step reasoning. The benefit is particularly evident on multi-hop datasets such as HQA and 2Wiki, where the method explicitly trains the agent to synthesize information step by step, a critical capability for complex information-seeking tasks.

4.2.2 InfoFlow Excels at Long-Horizon Deep Search and Reasoning Depth

We evaluate InfoFlow on the challenging BrowseComp-Plus benchmark to assess its capability for complex, long-horizon information seeking. As shown in Table 3, InfoFlow substantially outperforms all existing open-source LLMs, including those based on larger 32B models, and also surpasses strong proprietary systems such as Gemini 2.5 Pro and GPT-4.1. We observe the following two findings. First, the dual-agent architecture maintains a clear separation between high-level strategic planning and low-level search execution. This design effectively alleviates the cognitive burden of reasoning over long and noisy context and enables InfoFlow to invoke three times more tool calls than SearchR1-32B in complex information seeking tasks. Second, since hints are withheld during inference after RL, the robust performance indicates that pathfinding hints during RL training effectively internalize long-horizon, persistent search interleaved reasoning. This enables InfoFlow to sustain deep exploration, rather than providing speculative answers after merely one or two search steps.

We further analyze how InfoFlow's performance scales with reasoning depth by varying the number of reasoning–search turns at inference time. As illustrated in Figure 3, increasing the allowed turns consistently improves accuracy, from 11.2% with 4 turns to 22.8% with 16 turns. This trend indicates that InfoFlow learns a generalizable, iterative reasoning policy rather than overfitting to the fixed maximum horizon used during training.

4.2.3 InfoFlow Generalizes across Training Datasets and Annotators

We investigate the generalizability of InfoFlow-7B by varying both the training dataset and annotator of sub-goals and hints. Although InfoSeek provides explicit tree-structured annotations, we hypothesize that the hierarchical nature of deep search (characterized by horizontal constraints and vertical reasoning) is intrinsic to the task definition rather than specific to a dataset format. To evaluate this hypothesis, we apply our framework to

Train Set	Annotator	NQ	TQA	PopQA	HQA	2Wiki	MSQ	Bamb	Avg.
InfoSeek	None	45.6	67.1	46.5	42.2	45.7	20.2	45.8	44.7
	Qwen3-8B	46.9	68.4	47.8	43.5	47.0	21.4	46.9	46.0
	Gemini 2.5	47.2	68.1	48.1	44.3	47.2	21.9	47.6	46.2
HQA	None	40.7	59.6	44.4	45.1	50.4	19.1	46.9	43.7
	Qwen3-8B	42.5	60.9	47.3	48.7	53.5	19.7	44.8	45.3

Table 4: Generalization analysis across training datasets and annotator of sub-goal and hints.

Configuration	QA Average	BrowseComp-Plus	InfoSeek-Eval
InfoFlow-7B	46.2	23.2	47.8
w/o Dual-Agent RFT	38.4	10.2	32.5
w/o Sub-Goal Reward	44.9	21.4	44.5
w/o Off-Policy Hints	45.8	20.1	42.1

Table 5: Ablation study of InfoFlow components. We report average accuracy on seven general QA tasks, accuracy on the BrowseComp-Plus and InfoSeek-Eval benchmarks.

HotpotQA (Yang et al., 2018), a dataset lacking explicit decomposition labels and thus requires to annotate from the question and intermediate hop. Furthermore, to assess the robustness of sub-goal and hint mechanism to different annotators, we replace the proprietary Gemini 2.5 with the open-source, lightweight Qwen3-8B for annotating.

Results in Table 4 reveal two findings: First, the marginal performance gap between using Qwen3 and Gemini for annotation indicates that the presence of structural guidance itself is the primary driver of improvement, *rather than relying on the annotator model*. Second, the consistent gains observed when training on HotpotQA confirm that InfoFlow is not bound to a specific dataset. Instead, it effectively exploits the *latent hierarchical nature* of deep search tasks (e.g., decomposing constraints and multi-hop reasoning).

4.3 Discussion

4.3.1 Ablation Study

We perform ablations on InfoFlow-7B to evaluate the contribution of each component: 1) Removing dual-agent RFT causes the largest performance degradation. The combination of low success rates and long trajectories results in extremely low reward density, which is insufficient for stable policy optimization. 2) Removing sub-goal reward shaping also yields a consistent decrease. This finding underscores the importance of sub-goal supervision for deep search. 3) Without pathfinding hints has a relatively minor effect on general QA but leads to a drop on BrowseComp-Plus, indicating that hints are especially valuable for difficult information-seeking tasks requiring deep search, intensive reasoning, and long-horizon exploration.

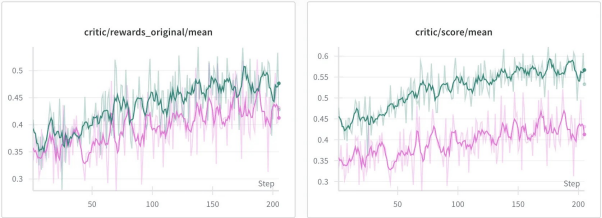

Figure 4: RL training dynamics with and without hints and sub-goal rewards.

4.3.2 RL Training Dynamics Analysis

We examine the RL training dynamics of InfoFlow-7B *with* (green curve) and *without* (pink curve) sub-goal scaffolding and pathfinding hints. We report both original final reward (Fig. 4 left) and shaped reward (Fig. 4 right). Both metrics rise consistently, indicating that the policy learns from the off-policy information provided by the sub-goal and hints during RL training effectively. If reward hacking were present, the shaped reward would increase while the final reward stagnated or declined.

5 Conclusion

We introduced InfoFlow, a dual-agent framework designed to address the critical challenge of low reward density in training LLM for agentic search tasks. By integrating sub-goal reward shaping, adaptive off-policy hints, and dual-agent architecture initialized with RFT, InfoFlow provides dense, process-level supervision that makes learning tractable. Our experiments demonstrate that this approach enables even lightweight LLMs to achieve performance competitive with proprietary models on challenging deep search benchmarks.

6 Limitations

While InfoFlow demonstrates strong performance in deep search tasks, there are several avenues for future improvement. First, due to computational constraints, our experiments primarily focus on models up to 7B parameters. Investigating the scaling effects of our reward density optimization on larger foundation models (e.g., 14B or 32B) remains a direction for future work. Second, our current framework is tailored for text-centric information seeking. Extending the dual-agent architecture to handle multi-modal evidence or interact with a broader range of tools (e.g., code interpreters) requires further adaptation. Finally, like most search-integrated methods, InfoFlow relies on the quality of the underlying search engine; extremely noisy or adversarial web content may still pose challenges for the refiner agent during information synthesis.

7 Ethical Consideration

Our research explores the use of various Large Language Models (LLMs) as backbone for agentic search tasks. Despite undergoing additional fine-tuning in various experiments, these models retain ethical and social risks inherent in their pretraining data. Notably, open-source LLMs may incorporate private or contentious data during the training phase, thereby raising additional ethical concerns.

References

Anthropic. 2025. Claude sonnet 4. https://www.anthropic.com/claude/sonnet. Accessed: 2025-08-24.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

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

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. 2025. Online difficulty filtering for reasoning oriented reinforcement learning. *arXiv preprint arXiv:2504.03380*.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, Fan Yang, et al. 2025a. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, et al. 2025b. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Yong Deng, Guoqing Wang, Zhenzhe Ying, Xiaofeng Wu, Jinzhen Lin, Wenwen Xiong, Yuqin Dai, Shuo Yang, Zhanwei Zhang, Qiwen Wang, et al. 2025. Atom-searcher: Enhancing agentic deep research via fine-grained atomic thought reward. *arXiv preprint arXiv:2508.12800*.

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

Tianqing Fang, Zhisong Zhang, Xiaoyang Wang, Rui Wang, Can Qin, Yuxuan Wan, Jun-Yu Ma, Ce Zhang, Jiaqi Chen, Xiyun Li, et al. 2025. Cognitive kernel-pro: A framework for deep research agents and agent foundation models training. *arXiv preprint arXiv:2508.00414*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Google Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Qwen Group. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Haoyang Hong, Jiajun Yin, Yuan Wang, Jingnan Liu, Zhe Chen, Ailing Yu, Ji Li, Zhiling Ye, Hansong Xiao, Yefei Chen, et al. 2025. Multi-agent deep research: Training multi-agent systems with m-grpo. *arXiv preprint arXiv:2511.13288*.

S Hong, X Zheng, J Chen, Y Cheng, C Zhang, Z Wang, SKC Yau, Z Lin, L Zhou, C Ran, et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. arxiv. *arXiv preprint arXiv:2308.00352*.

Ziyang Huang, Wangtao Sun, Jun Zhao, and Kang Liu. 2025. Improve rule retrieval and reasoning with self-induction and relevance reestimate. *arXiv preprint arXiv:2505.10870*.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7969–7992. Association for Computational Linguistics.

Bowen Jin, Jinsung Yoon, Priyanka Kargupta, Sercan O Arik, and Jiawei Han. 2025a. An empirical study on reinforcement learning for reasoning-search interleaved llm agents. *arXiv preprint arXiv:2505.15117*.

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

Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. 2025c. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 737–740.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019a. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019b. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Chengpeng Li, Mingfeng Xue, Zhenru Zhang, Jiaxi Yang, Beichen Zhang, Xiang Wang, Bowen Yu, Binyuan Hui, Junyang Lin, and Dayiheng Liu. 2025a. Start: Self-taught reasoner with tools. *arXiv preprint arXiv:2503.04625*.

Haitao Li, Yifan Chen, Hu YiRan, Qingyao Ai, Junjie Chen, Xiaoyu Yang, Jianhui Yang, Yueyue Wu, Zeyang Liu, and Yiqun Liu. 2025b. Lexrag: Benchmarking retrieval-augmented generation in multi-turn legal consultation conversation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3606–3615.

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

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025d. Search-o1: Agentic search-enhanced large reasoning models. *CoRR*, abs/2501.05366.

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

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, et al. 2025. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*.

Kun Luo, Zheng Liu, Shitao Xiao, Tong Zhou, Yubo Chen, Jun Zhao, and Kang Liu. 2024. Landmark embedding: a chunking-free embedding method for retrieval augmented long-context large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3268–3281.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Rafael Rafailov, Ivan Laptev, Philip HS Torr, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. 2024. Malt: Improving reasoning with multi-agent llm training. *arXiv preprint arXiv:2412.01928*.

Liangbo Ning, Ziran Liang, Zhuohang Jiang, Haohao Qu, Yujuan Ding, Wenqi Fan, Xiao-yong Wei, Shanru Lin, Hui Liu, Philip S Yu, et al. 2025. A survey of web-agents: Towards next-generation ai agents for web automation with large foundation models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6140–6150.

OpenAI. 2024. Openai o1 system card. *CoRR*, abs/2412.16720.

OpenAI. 2025. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/. Accessed: 2025-08-24.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Martin L Puterman. 1990. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434.

Hongjin Qian and Zheng Liu. 2025. Scent of knowledge: Optimizing search-enhanced reasoning with information foraging. *arXiv preprint arXiv:2505.09316*.

Hongjin Qian, Zheng Liu, Chao Gao, Yankai Wang, Defu Lian, and Zhicheng Dou. 2025. Hawkbench: Investigating resilience of rag methods on stratified information-seeking tasks. *arXiv preprint arXiv:2502.13465*.

Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. 2024. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591*, 1.

Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, et al. 2025. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents. *arXiv preprint arXiv:2509.13309*.

Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, et al. 2025. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

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

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. 2025. Search and refine during think: Autonomous retrieval-augmented reasoning of llms. *arXiv preprint arXiv:2505.11277*.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.

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

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, et al. 2025. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*.

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

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. 2023. Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun Li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Shuguang Ma, Fei Yu, and Yanghua Xiao. 2025a. Hint: Helping ineffective rollouts navigate towards effectiveness. *arXiv preprint arXiv:2510.09388*.

Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. 2025b. Stepsearch: Igniting llms search ability via stepwise proximal policy optimization. *arXiv preprint arXiv:2505.15107*.

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

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

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

Jinyang Wu, Chonghua Liao, Mingkuan Feng, Shuai Zhang, Zhengqi Wen, Pengpeng Shao, Huazhe Xu, and Jianhua Tao. 2025b. Thought-augmented policy optimization: Bridging external guidance and internal capabilities. *arXiv preprint arXiv:2505.15692*.

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

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. 2025. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*.

Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. 2025. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *arXiv preprint arXiv:2509.02479*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025a. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025b. Learning to reason under off-policy guidance. *Preprint*, arXiv:2504.14945.

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

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

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

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. 2024. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. 2025. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. *arXiv preprint arXiv:2508.11408*.

Shu Zhao, Tan Yu, Anbang Xu, Japinder Singh, Aaditya Shukla, and Rama Akkiraju. 2025. Parallelsearch: Train your llms to decompose query and search subqueries in parallel with reinforcement learning. *arXiv preprint arXiv:2508.09303*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. 2024. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37:62557–62583.

A Related Work

From Retrieval Augmentation to Search-Integrated Reasoning. To mitigate the limitations of static parametric knowledge, Retrieval-Augmented Generation (RAG) has become a standard practice (Lewis et al., 2020). Early RAG methods follow a static retrieve-then-generate pipeline, which struggles with complex, multi-hop queries. Recent efforts have made this process more dynamic through query rewriting, iterative retrieval, or self-critique mechanisms that assess the relevance of retrieved information (Asai et al., 2024; Qian et al., 2024; Jin et al., 2025c; Li et al., 2025b). A more advanced paradigm, Search-Integrated Reasoning (SIR), moves beyond this separation by deeply interleaving reasoning steps with tool actions like web engine and local knowledge base search (Yao et al., 2023; Chen et al., 2025a; Xue et al., 2025; Huang et al., 2025). Foundational frameworks such as ReAct (Yao et al., 2023) demonstrated the effectiveness of this approach using in-context learning. Our work, InfoFlow, adopts the SIR paradigm but focuses on explicitly training models to acquire these capabilities, rather than relying solely on prompt engineering at inference time.

Multi-Agent Collaboration. Decomposing complex problems for multi-agent systems is a powerful strategy. Most current approaches focus on inference-time orchestration, where a central planner LLM delegates sub-tasks to specialized tools or other LLM instances without altering their weights (Qiu et al., 2025; Hong et al., 2025). Frameworks like MetaGPT (Hong et al., 2023; Zhang et al., 2024; Motwani et al., 2024) assign distinct roles to different LLM agents to collaboratively solve complex tasks. InfoFlow advances this concept by introducing a co-trained dual-agent framework. We partition the cognitive load for planning, execution and evidence synthesis and guidance. Crucially, unlike inference-time frameworks, our agents are jointly optimized, allowing them to develop a specialized and synergistic protocol that enhances reasoning efficiency and stability.

Training Agents for Search and Reasoning. A prominent research direction focuses on fine-tuning LLMs to learn robust policies for interacting with search engines (Li et al., 2025d; Jin et al., 2025b; Wu et al., 2025a; Li et al., 2025c). While Supervised Fine-Tuning (SFT) on expert trajectories provides a strong initialization (Zeng et al., 2023; Li et al., 2025e), Reinforcement Learning (RL) is crucial for teaching agents to explore and discover effective strategies for unseen problems (Guo et al., 2025; Fang et al., 2025). Several works have successfully applied RL to train search agents (Jin et al., 2025b; Song et al., 2025; Liu et al., 2025; Qiao et al., 2025). However, a fundamental obstacle is reward sparsity: complex tasks yield infrequent terminal rewards, providing poor learning signals for the long sequence of intermediate steps (Ning et al., 2025; Wong et al., 2025). This makes policy optimization unstable and inefficient. While some methods attempt to mitigate this by learning a separate reward model or using offline policy optimization (Wang et al., 2025b; Deng et al., 2025; Yan et al., 2025b; Wang et al., 2025a), InfoFlow addresses the problem directly through a novel combination of sub-goal reward shaping to provide dense, intermediate signals and adaptive off-policy hints to increase the rate of successful trajectory completion during online training.

B Deep Search Question Answering Task Definition

The task of **Deep Search Question Answering (DSQA)** involves addressing complex queries that require multi-step reasoning and extensive information seeking. Benchmarks such as *BrowseComp* (Wei et al., 2025) exemplify this challenge, evaluating agentic search capability to navigate large-scale corpora such as the internet and synthesize information into coherent answers.

To enable principled optimization, we formalize DSQA as a reasoning tree $\mathcal{T}$, following the framework of Xia et al. (2025). In this formulation, each node denotes a sub-problem, either an entity to be identified or a constraint imposed on its parent entity. The root node is the final answer to the DSQA problem, which is a fact or entity to be discovered. Directed edges from child to parent nodes encode logical dependencies that must be validated. The complexity of DSQA problem is characterized by two structural properties of the reasoning tree. The *depth*, defined as the length of the longest root-to-leaf path, captures the extent of sequential reasoning required to resolve all sub-problems. The overall *width*, measured as the sum of children across all non-leaf nodes, reflects the degree of parallel information aggregation necessary to complete the task.

C Formulation of agentic search Process

The process of an LLM agent solving DSQA task can be formalized as a Markov Decision Process (MDP) (Puterman, 1990). An agent's trajectory τ is a sequence of interactions with a search environment: $\tau = (q, a_0, i_0, a_1, i_1, \ldots, a_{K-1}, i_{K-1}, a_K)$. Here, q is the initial question, a_k is the agent's action at step k, i_k is the information retrieved from the environment, and a_K is the terminal action containing the final answer. The MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where:

Action (a_k). The agent generates an action a_k, which involves two components: *Thinking (a_k^{think}):* A reasoning trace where the agent analyzes the current state S_k, synthesizes retrieved knowledge, and plans its next steps. This corresponds to *depth-wise progress* in the reasoning tree by exploiting available information and is enclosed in <think> tags. *Searching (a_k^{search}):* The agent generates a set of N_k parallel search queries $\{q_{k,j}\}_{j=1}^{N_k}$ to acquire new information. This facilitates *width-wise exploration* of the reasoning tree and is enclosed in <search> tags. The full action is the concatenation $a_k = a_k^{\text{think}} \circ a_k^{\text{search}}$. The action space also includes a terminal action a_K, where the agent provides the final answer within <answer> tags.

Transition ($\mathcal{T}$). The transition function $\mathcal{P}(S_{k+1}|S_k, a_k)$ is determined by the environment's response to the search action. An external search tool processes the queries $\{q_{k,j}\}$ and returns a set of retrieved evidence $i_k = \{(q_{k,j}, e_{k,j})\}_{j=1}^{N_k}$. This information is presented to the agent within <information> tags. The subsequent state is formed by appending the action and observation to the history: $S_{k+1} = S_k \circ (a_k, i_k)$.

Reward (R). A final reward $R(\tau)$ is assigned based on the correctness of the final answer a_K, evaluated by a rule-based reward model. The agent's objective is to learn a policy $\pi(a|S)$ that maximizes the expected return.

D Off-Policy Information Construction with InfoSeek Dataset

As introduced in Section 2.2, our process-based reinforcement learning approach relies on densely supervised data. This appendix details how we construct this off-policy supervision, specifically the weighted sub-goals and hints, by leveraging the unique structure of the InfoSeek dataset (Xia et al., 2025).

D.1 InfoSeek Dataset Information

The InfoSeek dataset was specifically designed to address the scarcity of benchmarks for *Deep Search* tasks, which demand complex, multi-step reasoning beyond simple multi-hop question answering. Its core innovation lies in its data synthesis paradigm, which generates questions grounded in a verifiable and explicit reasoning structure called a research tree. The generation process begins by mining entities and their relationships from a large-scale text corpus. From these, a research Tree is recursively constructed for each data point, where the root denotes the final, unique answer, internal nodes represent intermediate sub-goals, and edges encode their logical dependencies. To ensure complexity, the descriptions of these internal nodes are blurred with additional constraints. Finally, a powerful LLM is prompted with the entire tree structure to generate a high-level, natural language question whose resolution requires traversing the entire reasoning path. This tree-based structure provides a ground-truth decomposition of a complex problem into a hierarchy of verifiable sub-goals, making it an ideal foundation for generating process-level supervision.

InfoSeek-Evaluation The InfoSeek-Evaluation set contains 300 high-quality, human-checked samples to evaluate agentic search capability. Qwen2.5-72B-Instruct with a CoT prompting achieves lower than 8% accuracy in this evaluation set.

D.2 InfoSeek Task Examples

Figure 5, 6 and 7 provide three task examples of infoseek dataset.

D.3 Sub-Goals Construction

We utilize the InfoSeek research tree's topology to define sub-goals and assign an importance weight s_i to each. Our process begins by extracting a subset of high-value internal nodes from the research tree to form the set $\mathcal{G}_q = \{g_1, \ldots, g_M\}$, deliberately excluding simple confirmatory facts. We leverage LLM (Gemini 2.5 and Qwen3-8B; § 4.2.3) to select these critical entities (typically 2-4 per tree) and assign an importance weight to each. This selection process distinguishes between pivotal intermediate nodes (core entities unlocking subsequent paths) and secondary supporting nodes (necessary evidence), ensuring sparse yet targeted supervision. The assigned weights are constrained to sum to one ($\sum_{i=1}^{M} s_i = 1$), providing the final set

of weighted sub-goals $\{(g_i, s_i)\}_{i=1}^{M}$ for our reward shaping scheme. The prompt used for sub-goal annotation is detailed in Appendix J.

D.4 Pathfinding Hints Constructions

Hints are formulated as high-leverage guiding queries that act as off-policy information bridges. They are designed to assist the agent when it is unable to make progress through autonomous exploration, thereby mitigating unproductive reasoning loops. These hints are generated using Gemini 2.5 (see Appendix J) based on the critical edges of the Research Tree. Notably, open-source LLMs can also handle this annotation task as shown in § 4.2.3. During policy optimization, these hints are instrumental in teaching the agent several crucial search skills. They foster **purposeful search** by providing direct queries for specific sub-problems, guiding the agent onto a productive path. Furthermore, they help the agent **break through key points** in the reasoning chain where identifying the next step is non-obvious. Finally, by reframing or combining constraints in novel ways, the hints encourage **creative search**, training the agent to formulate more effective queries beyond simple keyword matching.

Figure 5, 6 and 7 provide three examples. The main question contains multiple, intertwined constraints. The generated hints effectively decompose this complexity by isolating and combining key constraints into actionable search queries. The first hint focuses on identifying the person, while the second provides an alternative, more robust query by combining the person's profession with their marital information.

Through this process, we enrich the original InfoSeek dataset with a structured layer of off-policy supervision. This augmented data, containing both quantitative sub-goal importance and qualitative reasoning hints, provides a robust foundation for training more capable and efficient Deep Research agents using our proposed method.

Question: What is a literary genre that was defined by a novelist who wrote a novel incorporating elements of the legendary origins of the Hope Diamond, and was mentored by Charles Dickens, characterized as a 'novel-with-a-secret'?

Answer: Sensation novel

Hint Queries:
novelist mentored by Charles Dickens who wrote The Moonstone
author whose novel incorporated elements of the Hope Diamond and was mentored by Charles Dickens
author of 'The Woman in White' mentored by Charles Dickens

Sub Goals:
Wilkie Collins: weight 0.6
Charles Dickens: weight 0.2
The Moonstone: weight 0.2

Figure 5: Case study 1 (Sensation novel): An example of enriched InfoSeek dataset. The hints decompose the main question into more manageable, high-leverage search queries that serve as off-policy guidance.

Question: What is an album that was created by a musician who played piano in Gus Arnheim's band, created a jazz camp, was recorded in 1955, and features drumming by Mel Lewis?

Answer: Contemporary Concepts

Hint Queries:
musician who played piano in Gus Arnheim's band and later created a jazz camp
bandleader whose 1955 album featured Mel Lewis on drums
jazz pianist who once played for Gus Arnheim and founded a music education program

Sub Goals:
Stan Kenton: weight 0.7
Gus Arnheim: weight 0.3

Figure 6: Case study 2 (Contemporary Concepts): An example of enriched InfoSeek dataset.

Question: What is a British Thoroughbred racehorse that was sired by a horse who won the 1941 Epsom Derby, was the leading British two-year-old of 1959, was a dark bay horse with a white blaze standing 16.1 hands high, and had considerable success as a sire of sprinters?

Answer: Sing Sing (horse)

Hint Queries:
horse that won the 1941 Epsom Derby
1941 Epsom Derby winner

Sub Goals:
Tudor Minstrel: weight 0.5
Owen Tudor: weight 0.5

Figure 7: Case study 3 (Sing Sing (horse)): An example of enriched InfoSeek dataset.

E Theoretical Analysis of Reward Density Optimization

In this section, we analyze how the two core components of InfoFlow, Sub-goal Scaffolding and Pathfinding Hints, address the limitations of sparse rewards in the RL of agentic search. We analyze the impact of Reward Density Optimization within the framework of Group Relative Policy Optimization (GRPO) (Shao et al., 2024), especially in the perspective of gradient variance reduction and exploration efficiency.

E.1 Sub-goal Scaffolding: Variance Reduction via Dense Signals

Standard Reinforcement Learning for deep search suffers from extreme reward sparsity. In the GRPO formulation, for a query q, the policy π_θ generates a group of G trajectories $\{\tau_1, \ldots, \tau_G\}$. The gradient update relies on the normalized advantage A_i:

$$A_i = \frac{R(\tau_i) - \mu_{\mathbf{R}}}{\sigma_{\mathbf{R}}} \tag{7}$$

where $\mu_{\mathbf{R}}$ and $\sigma_{\mathbf{R}}$ are the group mean and standard deviation.

The Vanishing Gradient Problem. Without scaffolding, the reward is binary, $R_{\text{final}} \in \{0, 1\}$. For complex, multi-step reasoning tasks, the probability of reaching the correct final answer a_K without guidance is negligible during early training stages. This leads to the *Zero-Variance Dilemma*:

$$P(\forall i, R(\tau_i) = 0) \to 1 \implies A_i \approx 0, \quad \nabla_\theta \mathcal{J} \approx 0 \tag{8}$$

Even when the group contains a mixture of correct and incorrect answers, the binary signal fails to distinguish between sensible failures (trajectories that solved partial sub-problems) and complete failures, leading to high variance in gradient estimation and unstable convergence.

Process Reward Formulation. To densify the learning signal, Sub-goal Scaffolding introduces a composite reward function. Let $\mathcal{G}_q = \{g_1, \ldots, g_M\}$ be the set of ground-truth sub-goals for query q introduced in § 2.2 and § D.3. We define the sub-goal reward R_{sub} based on the exact match (EM) of entities in the reasoning trace against $\mathcal{G}_q$:

$$R_{\text{sub}}(\tau) = \sum_{m=1}^{M} w_m \cdot \mathbb{I}(\text{EM}(g_m, \tau)) \tag{9}$$

where $\mathbb{I}(\cdot)$ is the indicator function and w_m is the weight of the sub-goal. The total reward is defined as:

$$R(\tau) = R_{\text{final}}(\tau) + w \cdot R_{\text{sub}}(\tau) \tag{10}$$

We enforce a reward ceiling of 1.0, ensuring that if the final answer is correct (i.e., $R_{\text{final}}(\tau) = 1$), the agent receives the maximum reward regardless of the sub-goal alignment. This design acknowledges that the annotated sub-goals $\mathcal{G}_q$ constitute a *sufficient* but not *necessary* path for solving the problem; since alternative valid reasoning paths may exist, the agent should not be penalized for correctly solving the problem via a trajectory that differs from the annotation.

Variance Reduction. By introducing R_{sub}, we ensure that even in groups where no trajectory solves the final task (i.e., $\forall i, R_{\text{final}}(\tau_i) = 0$), the reward distribution within the group is not uniform:

$$\sigma_{\mathbf{R}} > 0 \quad \text{due to variations in } R_{\text{sub}}(\tau_i) \tag{11}$$

This creates valid, non-zero advantages A_i, providing informative gradients that guide the policy toward intermediate milestones, effectively smoothing optimization process.

E.2 Pathfinding Hints: Overcoming the Exploration Barrier

While Sub-goal Scaffolding densifies rewards for partially successful trajectories, it cannot help if the policy fails to reach even the first sub-goal. This is the *Exploration Barrier*, manifesting as *Zero-Information Groups* where $\mathcal{R} = \mathbf{0}$ for the entire group.

Hints as Off-Policy Exploration guidance. Pathfinding Hints act as a mechanism to inject high-quality exploration guidance. By conditionally forcing a subset of trajectories to follow informative guiding queries, we artificially induce successful outcomes within a sampled group. Let τ_{hint} be a trajectory augmented with hints. The presence of τ_{hint} in group G ensures:

$$\exists \tau_k \in \{\tau_1, \ldots, \tau_G\} \text{ s.t. } R(\tau_k) > \mu_{\mathbf{R}} \tag{12}$$

This guarantees that $\mu_{\mathbf{R}} > 0$ and $\sigma_{\mathbf{R}} > 0$. Consequently, unguided trajectories that fail receive a meaningful negative advantage ($A_i < 0$), while the guided trajectory provides a positive learning signal ($A_k > 0$). This contrastive signal is essential for the policy to learn *what not to do* versus *what to do*.

Refiner Agent	Accuracy (%)	Search Calls (#)	Context Length (Tok.)	Time (min.)
w/o refiner	8.4	1.93	2372.4	12.2
Qwen2.5-3B-Inst	13.4	3.07	1309.6	10.2
Qwen2.5-7B-Inst	17.2	2.83	1071.2	10.5
Qwen2.5-32B-Inst	18.8	3.01	1260.4	11.3

Table 6: Analysis of the dual-agent framework on the InfoSeek evaluation set. The Researcher Agent is fixed as Qwen2.5-3B-Instruct. Context Length is the average number of tokens processed by the researcher per trajectory. Time denotes the average inference time per task.

Stability via GRPO/PPO Clipping. A theoretical concern with off-policy hints is the risk of distribution shift, where the policy might overfit to the hinted actions rather than learning the reasoning logic. However, the GRPO (Shao et al., 2024) objective (derived from PPO (Schulman et al., 2017)) inherently mitigates this via the clipping mechanism:

$$\mathcal{L}(\theta) = \mathbb{E}\left[\min(r_t A_t, \text{clip}(r_t, 1-\epsilon, 1+\epsilon)A_t)\right] \tag{13}$$

where r_t is the probability ratio.

If the hinted trajectory τ_{hint} induces a large policy update (i.e., r_t deviates significantly from 1), the gradient is clipped. This ensures that pathfinding hints serve as a nudge towards promising regions of the search space rather than forcing a hard imitation update, allowing the policy to gradually internalize search logic while maintaining training stability.

Pathfinding hints serve a dual purpose: they provide the necessary variance to make GRPO mathematically valid in sparse-reward settings, especially for more difficult deep search tasks, while the algorithm's clipping mechanism ensures that this powerful guidance translates into stable, generalized policy improvements rather than brittle overfitting. Since hints are strictly removed during inference, the final policy π_θ remains fully autonomous, relying solely on the generalized reasoning patterns internalized during this stabilized training process. Analysis of RL training dynamics in § 4.3.2 validates the stabilization of pathfinding hints.

F Framework Efficiency Analysis

As introduced in Section 3.1, our dual-agent framework decouples high-level reasoning from low-level evidence gathering to enhance performance and efficiency. This section provides a detailed analysis of this design.

Dual-agent Framework Improves Reward Density. As shown in Table 6, we conduct experiments employing a fixed Qwen2.5-3B-Instruct researcher to isolate the impact of the refiner with InfoSeek evaluation set. The baseline without a refiner struggles, achieving only 7.4% accuracy. The introduction of a 3B refiner dramatically improves accuracy to 13.4% while simultaneously reducing the researcher's average context length per trajectory by 45% (from 2372 to 1310 tokens). We quantify this gain using the reward density metric τ defined in Eq. (1). By approximating the expected return via accuracy and exploration cost via context length, the dual-agent framework achieves an approximately **3.3× improvement in reward density** (0.010 v.s. 0.003).

Quantitative Efficiency Analysis. Beyond performance improvements, the proposed dual-agent framework also demonstrates clear computational advantages. As reported in Table 6, we compare inference-time costs under identical computational resources (8×H100 GPUs) between a *single-agent* baseline and our *dual-agent* framework on the InfoSeek evaluation set. Despite employing two models, the dual-agent system achieves **faster end-to-end inference**, requiring 10.2 minutes compared to 12.2 minutes for the single-agent setup.

The primary bottleneck in LLM is the quadratic complexity ($O(n^2)$) of self-attention with respect to context length. By delegating the processing of verbose evidence to the refiner, we substantially reduce the peak context length and overall inference time.

Feasibility of Practical Deployment. In practical deployment, this architecture is highly feasible. A standard setup for information-seeking tasks already requires a researcher agent and a retrieval service (10% VRAM) in a single 8xH800 node. Adding a dedicated refiner, optimized with frameworks like vLLM (Kwon et al., 2023), incurs a manageable overhead of approximately 20% more VRAM, making the entire system viable on

a single 8xH800 node. In conclusion, although deploying two models may appear to introduce additional system complexity, advances in inference infrastructure (Kwon et al., 2023; Zheng et al., 2024) largely mitigate this concern. In practice, the dual-agent framework achieves higher accuracy and faster inference under identical computational budgets, **without increasing deployment resource requirements**.

Adaptability for Optimization. A key advantage of our approach is its implementation simplicity and adaptability. Unlike complex multi-agent reinforcement learning schemes, our refiner can be aligned with the researcher via a straightforward joint RFT process. This involves sampling trajectories from the researcher and using them to train the refiner, ensuring it learns to distill information in a manner tailored to the researcher's reasoning patterns. Consequently, the refiner is not a static, prompt-engineered module but a dynamic component that **co-evolves with the researcher adaptively**. This training methodology provides a scalable path toward building more capable, collaborative agent systems without incurring prohibitive complexity.

G Robustness Analysis of RFT

In § 3.2, we emphasized the importance of filtering trajectories to retain only those with rigorous reasoning chains, reducing the training pool from 18,000 raw rollouts to a high-quality subset of approximately 3,450. To empirically validate this design choice and assess the trade-off between data quantity and process fidelity, we conducted a robustness analysis by removing the verification step. We trained a control variant, denoted as RFT3B-Noisy, utilizing the complete set of approximately 16,000 trajectories that reached the correct final answer. This unfiltered dataset represents a substantial increase in scale but introduces significant noise, may including search shortcuts, spurious correlations, and correct answers derived from flawed intermediate reasoning.

Table 7 compares the performance of this noisy variant against our standard RFT3B-Cleaned model across eight QA benchmarks. The results reveal a compelling pattern: despite being trained on nearly five times less data, the model fine-tuned on the curated subset consistently outperforms the one trained on the larger, noisy corpus, achieving a higher average score (40.6 vs. 39.8). This performance gap highlights that for agentic search tasks, where the objective is to learn a generalizable reasoning policy rather than merely memorizing answer patterns, **the quality of the supervision signal is far more critical than the number of demonstrations**.

H Error Analysis on Failed Trajectories

To better understand the limitations of current agentic search models and identify directions for future improvement, we conducted a detailed manual inspection of failed trajectories from the **BrowseComp-Plus** and **InfoSeek-Eval** benchmarks. We categorize the failure modes into three primary types:

Retrieval and Evidence Acquisition Bottlenecks (48.2%). The dominant source of failure stems from the inability to acquire necessary evidence from the external environment. In these cases, despite the agent generating semantically reasonable search queries, the retrieval system either fails to surface relevant documents (low recall) or ranks them below the cutoff threshold. We observed that even when the agent attempts to recover via query reformulation, the information gain often remains stalled. This suggests that the bottleneck for nearly half of the errors lies not in the agent's reasoning policy, but in the inherent difficulty of high-recall retrieval for obscure or long-tail knowledge.

Incomplete Reasoning and Verification (33.7%). This category reflects the challenge of sustaining rigorous logic over long horizons. Here, agents often formulate a partially correct intermediate hypothesis but fail to fully validate all constraints before committing to a final answer. These failures typically manifest as *premature termination*, where the agent treats a plausible-looking entity as the ground truth without conducting the necessary cross-verification steps required by the complex query structure.

Early Planning and Decomposition Errors (18.1%). A smaller but fatal proportion of errors arises from incorrect initial task decomposition. In these instances, the agent commits to a suboptimal search strategy or misidentifies the entry point of the reasoning graph within the first few turns. Due to the cumulative nature of deep search, such early drift is difficult to correct; the subsequent trajectory often diverges irreversibly from the target solution space, rendering later reasoning steps ineffective regardless of their individual quality.

Model	NQ	TQA	PopQA	HQA	2Wiki	MSQ	Bamb	Avg.
RFT3B-Noisy	39.0	55.6	**45.4**	39.8	42.3	16.7	**39.4**	39.8
RFT3B-Cleaned	**40.4**	**57.6**	45.0	**41.6**	**43.5**	**18.0**	37.9	**40.6**
Diff.	*+1.4*	*+2.0*	*-0.4*	*+1.8*	*+1.2*	*+1.3*	*-1.5*	*+0.8*

Table 7: Robustness analysis of the RFT data filtering process. We compare the model trained on the raw, noisy set of all correct rollouts (**Noisy**, ≈16k samples) against the one trained on the verified, high-quality subset (**Cleaned**, ≈3.4k samples).

In summary, while policy optimization (as addressed by InfoFlow) significantly improves planning and reasoning, the high prevalence of retrieval, induced errors highlights that *search utility*, the ability of the environment to support the agent's reasoning, remains a critical limiting factor for deep search tasks.

I Implementation Details

For research agent RFT, we fine-tune for 3 epochs with a learning rate of 1e-5, L2 normalization of 0.01(important for stablizing training), and a context length of 16,384, using a single 8×H100 node. For refiner agent RFT, we fine-tune for 2 epochs with a learning rate of 1e-5, L2 normalization of 0.01 , and a context length of 8,192, using a single 8×H100 node.

RL training is conducted with a batch size of 256, a maximum of 10 turns, rollout size 8, temperature 0.8, and a search engine restricted to the top-5 retrieved contents. The training is conducted on two 8×H100 nodes.

J Prompts

J.1 Refiner Agent

The complete prompt template used for our Refiner Agent is presented in Figure 8. We use this template both to drive the refiner and for refiner RFT training.

J.2 Prompt for Dataset Enrichment

We use the Gemini 2.5 API (Gemini Team, 2025) and Qwen3-8B (Yang et al., 2025) with the prompt shown in Figure 9 to conduct InfoSeek dataset enrichment.

J.3 Prompt for Evaluation

The specific prompt used for DeepSeek-as-a-judge is shown in Figure 10.

Refiner Agent Template

<|im_start|>user
TASK:
Synthesize the key information from the **[Retrieved Documents]** that is relevant to the **[Current Query]**. The synthesis should be guided by conducting deep research to uncover the **[Original Question]**.
INSTRUCTIONS:
1. **Extract & Merge:** Identify all relevant facts and combine them. Eliminate redundancy. You should provide information for deep research, not answer to current query or original question.
2. **Provide Information, Not an Answer:** Your output should be a self-contained block of information, NOT a direct, short answer to the original question or the current query.
3. **Handle Insufficient Information:** If the documents do not contain relevant information for the query, state that the provided sources are insufficient and suggest that further investigation may be needed. You can also provide some further investigation direction and query rewrite suggestions.
4. **Format:** Enclose the entire synthesized output within <information> and </information> tags. Add no other text. For example, <information> Synthesized information for deep research here </information>.

CONTEXT:
- **[Original Question]:** {original_question}
- **[Current Query]:** {query}
- **[Retrieved Documents]:** {documents}

TASK:
Synthesize the key information from the **[Retrieved Documents]** that is relevant to the **[Current Query]**. The synthesis should be guided by conducting deep research to uncover the **[Original Question]**.
INSTRUCTIONS:
(Repeated instructions omitted for brevity...)
Format: Enclose the entire synthesized output within <information> and </information> tags.
SYNTHESIZED INFORMATION:
<|im_end|>
<|im_start|>assistant

Figure 8: The prompt template for the Refiner Agent.

AI Data Augmentation Expert Prompt

```
<|im_start|>user
Role: You are an AI Data Augmentation expert. Your mission is to extract and
expand key information from a Research Tree to optimize reinforcement learning
for training an LLM as a deep research agent.
Objective: From the input Research Tree, complete the two tasks below and
return results in one unified JSON output.

### Task 1: Extract High-Value Entities & Assign Weights (for Reward Shaping)
Identify pivotal breakthroughs to reward in PPO training.
Steps:
1. Select 2-4 most critical entities from the Research Tree.
2. Assign each a weight (float), with all weights summing to 1.0.
3. Prioritize:
     • Pivotal Nodes (0.6-0.8): Core breakthroughs, usually direct children of
       the root, resolving major clauses.
     • Supporting Nodes (0.2-0.4): Necessary for pivotal nodes, smaller but
       still important.
     • Exclude trivial confirmatory facts.
Output: JSON array of objects with id, entity, and weight.

### Task 2: Generate Early-Stage Guiding Queries (for Strategic Hints)
Provide hints to guide initial exploration without leaking answers.
Steps:
1. Generate 1-2 critical guiding queries.
2. Focus on leaf nodes, using their parent's entity + claim.
3. Queries must not contain the child node's entity.
4. Queries should be natural, strategic, and yield high information gain.
Output: JSON array of objects with target_id and generated_queries (array of
strings).
Background:
• Research Tree = hierarchical structure of questions/answers (nodes).
• Root = original complex question.
• Children = sub-questions.
• Claims = relationship between parent and child entities.
Execute both tasks on this Research Tree:
{research_tree_structure}
Output:
<|im_end|>
<|im_start|>assistant
```

Figure 9: The prompt used for the AI Data Augmentation expert to process the Research Tree.

DeepSeek Evaluation Prompt

```
<|im_start|>user
You are an evaluation assistant. Please determine if the predicted answer is
equivalent to the labeled answer.
Question: {question}
Labeled Answer: {labeled_answer}
Predicted Answer: {pred_answer}
Are these answers equivalent? Please respond with "Correct" if they are
equivalent, or "Incorrect" if they are not equivalent. Do not include any other
text.
<|im_end|>
<|im_start|>assistant
```

Figure 10: The prompt used for DeepSeek-as-a-judge to evaluate correctness.