Beyond Stochastic Exploration: What Makes Training Data Valuable for Agentic Search

Anonymous ACL submission

Abstract

Reinforcement learning (RL) has become an effective approach for advancing the reasoning capabilities of large language models (LLMs) through the strategic integration of external search engines. However, current RL-based search agents often rely on a process of stochastic exploration guided by carefully crafted outcome rewards, leading to inefficient reasoning trajectories and unstable training. To address these issues, we propose a novel framework, Hierarchical Experience (HiExp), to enhance the performance and training stability of search agents. Specifically, we extract empirical knowledge through contrastive analysis and a multi-level clustering mechanism, transforming raw reasoning trajectories into hierarchical experience knowledge. By leveraging experience-aligned training, we effectively regularize stochastic exploration, evolving it into a strategic and experience-driven search process. Extensive evaluations on multiple complex agentic search and mathematical reasoning benchmarks demonstrate that our approach not only achieves substantial performance gains but also exhibits strong cross-task and cross-algorithm generalization.

1 Introduction

Large language models have demonstrated remarkable capabilities in task planning and agentic reasoning, with reinforcement learning significantly improving their performance on complex reasoning tasks (Shao et al., 2024; Guo et al., 2025; Yang et al., 2025a). However, reliance on static parametric knowledge presents notable limitations, often leading to hallucinations and inefficient reasoning (Yao et al., 2025; Kalai et al., 2025). To tackle these challenges, it is crucial to explore how to efficiently access diverse external information to support LLMs in achieving deliberate and well-substantiated reasoning. Therefore, a novel search paradigm termed *Agentic Deep Research Systems* has gradually become an important research task (Li et al., 2025; Jin et al., 2025; Zhang et al., 2025b).

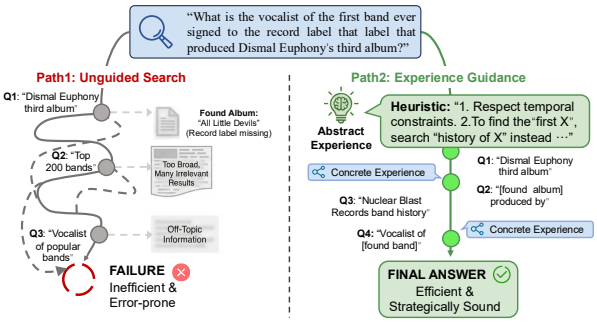

Figure 1: Comparison between stochastic exploration and experience-guided exploration. Experience-driven guidance facilitates more efficient reasoning trajectories, endowing LLMs with superior problem-solving capabilities for complex tasks.

Previous research has utilized Chain-of-Thought (CoT) (Wei et al., 2022) prompting to decompose complex problems into sequential sub-tasks, subsequently leveraging external information dynamically to bridge knowledge gaps and tackle intricate reasoning tasks (Trivedi et al., 2023; Yue et al., 2025; Feng et al., 2025b). Li et al. (2025) integrates agentic search into the reasoning process, enabling dynamic retrieval to address informational uncertainty or incompleteness. Recently, reinforcement learning has achieved remarkable success in mathematical reasoning and decision-making scenarios (Guo et al., 2025). Jin et al. (2025); Feng et al. (2025a) also utilize RL through environmental interactions to significantly enhance the capability of small language models (SLMs) in addressing intricate multi-hop and mathematical reasoning challenges. These training-based approaches integrate autonomous tool invocation into LLMs, facilitating dynamic environmental interaction (Zheng et al., 2025b; Chen et al., 2025). Due to their superior agentic abilities and strong generalization,

RL-based agentic reasoning approaches are increasingly emerging as a significant trend in deep research (Zhang et al., 2025b).

Existing RL-based search agents rely primarily on stochastic exploration guided by carefully crafted outcome rewards. However, these methods often struggle to execute global strategic planning and explore efficient reasoning trajectories, particularly when utilizing small language models for complex complex tasks, as shown in Figure 1. Furthermore, in multi-turn interaction scenarios, the inherent difficulty of providing consistent reward signals leads to significant instability during end-to-end RL training.

To address these limitations, we introduce the HiExp framework, which regularizes the exploration process of search agents with hierarchical experiences. By transforming stochastic exploration into a strategic, experience-aligned search, we significantly stabilize the reward signals and facilitate the discovery of optimal reasoning paths. Specifically, we extract empirical knowledge by performing contrastive analysis on pre-sampled rollouts, identifying the critical factors that differentiate successful reasoning paths from failures. We then employ a multi-level clustering strategy to abstract these instance-specific insights into high-dimensional reasoning strategies. These hierarchical experiences significantly bolster LLMs' performance across diverse task scenarios during the inference phase. Furthermore, throughout the critic-free RL training process, these systemic experiences are dynamically aligned with the rollout stage. This alignment effectively transforms conventional stochastic exploration into a strategic, experience-driven search, enhancing the effectiveness and stability of the optimization process. In summary, our main contributions are as follows:

- We introduce an endogenous scheme for hierarchical experience (HiExp) construction by leveraging self-reflection and agglomerative clustering over internal reasoning trajectories. This method facilitates the autonomous synthesis of meta-knowledge without the need for additional external factual information.

- Our proposed HiExp not only improves LLMs' performance in various tasks during the inference phase but also dynamically aligns with the rollout stage of RL training. This alignment transforms conventional stochastic exploration into a strategic, experience-driven search, enhancing the effectiveness and stability of policy optimization.

- Extensive evaluations demonstrate that HiExp consistently yields substantial performance gains over RL-based search agents. Furthermore, our approach exhibits robust generalization capabilities across various task domains and RL algorithms.

2 Related Work

2.1 Retrieval-Augmented Generation

Early retrieval-augmented generation (RAG) approaches employ various strategies such as branching, iteration, and adaptive retrieval to solve complex tasks. These methods rely on manually crafted workflows to guide LLMs in interacting with external knowledge sources. IRCoT (Trivedi et al., 2023) leverages CoT to steer the retrieval process, refining CoT with the retrieved information. Press et al. (2023a); Asai et al. (2024); Yue et al. (2025) refine intermediate queries to acquire valuable knowledge through multi-turn iterations. AirRAG (Feng et al., 2025b) applies Monte Carlo Tree Search to dynamically explore the reasoning paths. However, these approaches are limited to manually designed prompts and workflows, failing to fully unleash the inherent reasoning potential of LLMs.

2.2 Autonomous Search Agents

As the reasoning and decision-making capabilities of the foundation models continue to improve, Search-o1 (Li et al., 2025) significantly improves model performance in complex scenarios by designing an agentic search workflow, providing superior flexibility and generalization. DeepSeek-R1 (Guo et al., 2025) also demonstrates that outcome-based RL can significantly enhance the autonomous reasoning and decision-making capabilities of models. Therefore, RL has been applied to various complex reasoning tasks and agent-based scenarios. Complex multi-hop question answering represents a typical integrated application scenario that heavily relies on model-driven planning and reasoning. Chen et al. (2025); Jin et al. (2025) have successfully applied end-to-end RL to complex agentic search scenarios, further advancing the development of agentic deep research systems. These methods autonomously select retrieval tools during the reasoning process to interact with external environments. DeepResearcher (Zheng et al., 2025b) scales RL in real-world environments by

incorporating authentic web search interactions. s3 (Jiang et al., 2025) decouples the searcher from the generator and trains the searcher with fewer samples. EvolveSearch (Zhang et al., 2025a) further explores the self-evolution process of search agents. StepSearch (Wang et al., 2025) introduces fine-grained reward signals to steer strategic query planning and improve retrieval quality in complex search environments.

3 Methodology

In this section, we propose a framework designed to transition the stochastic exploration inherent in critic-free RL algorithms toward experience-aligned heuristic search. Beyond leveraging external factual knowledge bases, we conceptualize the historical trajectories generated during the rollout phase as an endogenous knowledge base. As shown in Figure 2, the framework consists of two primary components:

- Hierarchical Experience Construction: This phase extracts success-critical features from raw trajectories through contrastive sampling and subsequently refines fragmented insights into systematic principles using clustering algorithms.
- Experience-Aligned Training: This phase dynamically injects the distilled hierarchical knowledge into the training process of critic-free algorithms, effectively lifting the upper bound of the model's reasoning efficiency.

3.1 Hierarchical Experience Construction

In contrast to traditional static external knowledge sources, our HiExp framework introduces a self-evolving mechanism termed Self-Reflection Experience. This mechanism empowers the LLM to autonomously extract, abstract, and refine knowledge from its internal reasoning trajectories, as formalized in the hierarchical mining process of Algorithm 1. We further broaden the value of the training data beyond the annotated labels to encompass the entire exploration process.

3.1.1 Contrastive Distillation

For each sample x_i in the training set, we execute K independent rollouts to obtain a trajectory set $\mathcal{Y}_i$. Each trajectory comprises complete reasoning steps `<think>`, search actions `<tool_call>`, and the corresponding external environment responses `<tool_response>`. Guided by the outcome reward r_{orm}, we partition these trajectories into successful paths $\mathcal{Y}_i^+$ and failed paths $\mathcal{Y}_i^-$. We leverage the self-reflection capabilities of either the policy model or a superior teacher model to identify two critical features: key decision points and reasoning traps. The output of contrastive distillation is formalized as case-based experience e_i and its corresponding summary description d_i, which encapsulates high-value procedural knowledge extracted from pre-sampled trajectories. This process serves to transform raw rollout data into the foundational primitives necessary for the subsequent hierarchical clustering phase.

$$\mathrm{e}_i, \mathrm{d}_i = \text{Reflect}(x_i, y_i^+, y_i^-). \tag{1}$$

3.1.2 Hierarchical Clustering

Although case-based experiences e_i extracted from contrastive trajectories encapsulate valuable reasoning clues, their instance-specific nature often limits direct utility. Direct injection into the LLM may trigger overfitting or introduce significant retrieval noise due to an expansive search space. To mitigate these challenges, we propose a multi-level clustering mechanism that transforms fragmented experiences into strategic experience knowledge.

First, we employ a pre-trained semantic encoder $\phi(\cdot)$ to map all case-based experiences into a high-dimensional embedding space. For each experience entry e_i, its vector representation is denoted as $\mathbf{v}_i = \phi(\mathrm{d}_i)$. This transformation ensures that semantically equivalent but lexically distinct experiences (e.g., verifying director identity vs. confirming director uniqueness) are proximal within the vector space. Subsequently, we apply agglomerative clustering to $\mathbf{v}_i$ for subsequent aggregation. The detailed procedure is provided in phase 2 of Algorithm 1. By imposing a stringent similarity threshold τ_1, we consolidate experiences related to analogous problems. For each identified cluster $\mathcal{C}_j^{(1)}$, we leverage an LLM to distill multiple instance-level experiences into a generalized strategic experience. This distillation process is executed iteratively, leveraging systematic clustering and agentic self-reflection to progressively enhance the compactness and generalization of the experience repository.

3.2 Experience-Aligned Training

Inspired by Search-o1 (Li et al., 2025), current advanced RAG methods introduce an agentic search strategy, transforming the exploration process into

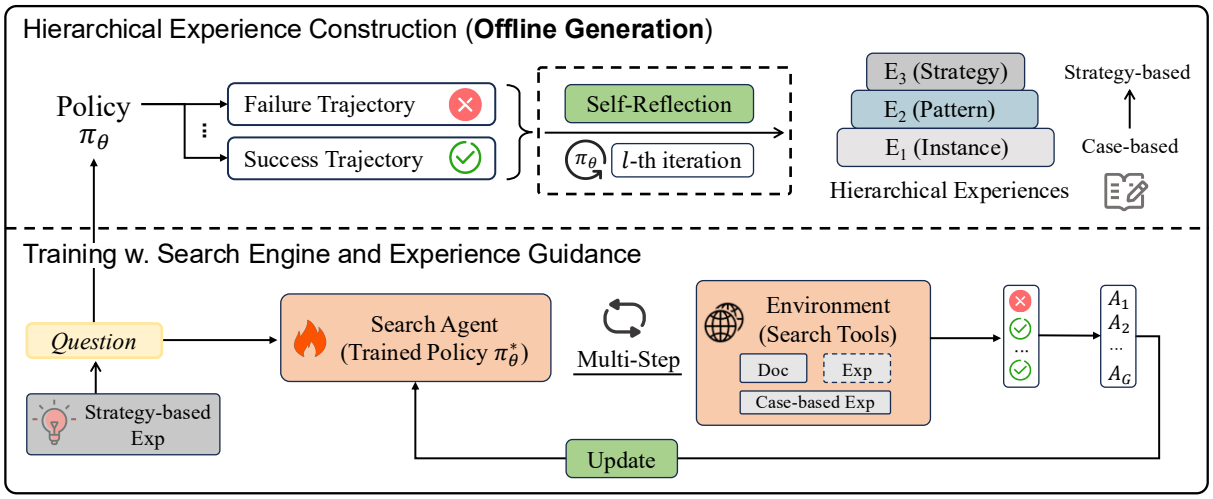

Figure 2: Overview of the offline hierarchical experience construction and the experience-guided policy optimization framework. The hierarchy spans from atomic instances to strategic principles, providing multi-granularity guidance for the search agent. During the training process, strategy-based experiences are leveraged to guide initial planning, while case-based experiences are employed to provide fine-grained support for intermediate reasoning steps.

an iterative interaction between the intrinsic reasoning of LLMs and the external environment, thus effectively activating their autonomous reasoning capabilities. During interactions with the external environment, these methods often rely on unstructured text retrieval systems to supplement information for intermediate reasoning steps. Irrelevant textual noise can easily result in inefficient intermediate queries and logical drift. RL-based search agents (Jin et al., 2025; Chen et al., 2025) typically rely on prior-free stochastic exploration during the rollout phase, which often suffers from low sample efficiency and limited convergence stability.

To alleviate these bottlenecks and surpass the performance ceilings of vanilla exploration, we introduce an Experience-Aligned Guidance mechanism. This framework empowers the agent to dynamically leverage high-fidelity strategic priors from Hierarchical Experience Knowledge (HEK) during trajectory generation, effectively transforming undirected search into an experience-guided exploration process. Within critic-free RL algorithms such as GRPO (Shao et al., 2024), the reasoning trajectory or intermediate query q_t generated at each rollout step t serves as a representation of the current state. The system utilizes a semantic encoder $\phi(\cdot)$ to compute the embedding vector of q_t, which is subsequently subjected to similarity matching against the hierarchical experience indices within the HEK. The retrieved experience e^* is defined as:

$$\mathrm{e}^* = \underset{e \in \mathrm{HEK}}{\operatorname{argmax}} \cos_\mathrm{sim}(\phi(q_t), \phi(\mathrm{d})). \quad (2)$$

Across different stages of the rollout process, our framework employs a dynamic guidance strategy. In the initial reasoning stage, global strategic experiences (E_2 or E_3) are prioritized and incorporated into the system prompt to provide strategic guidance that transcends specific task contexts, described in Table 6. During intermediate reasoning steps, the system adaptively provides top-k granular E_1 heuristics, filtered by a fixed semantic threshold to ensure high proximity to the current query. The agent can also adaptively refine its sub-query planning. This hierarchical retrieval allows the model to leverage the self-reflection experience to effectively navigate complex multi-step reasoning paths, transforming stochastic exploration into an experience-aligned heuristic search.

Combining the outcome reward with GRPO training objective, we propose a RL objective that explicitly incorporates a search engine $\mathcal{R}$ and a hierarchical experience knowledge base $\mathrm{HEK} = \{\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_L\}$ during optimization for the search agent training (Jin et al., 2025). Since reasoning trajectories are conditioned on hierarchical experiences, the advantage function derived from sampling trajectories possesses superior quality, facilitating more stable policy updates. The optimization objective is defined as:

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}\sim\pi_{\text{old}}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\min\Big(r_i(\theta)\hat{A}_i, \mathrm{clip}\big(r_i(\theta),\, 1-\epsilon,\, 1+\epsilon\big)\hat{A}_i\Big)\Bigg] - \beta\mathbb{D}_{\mathrm{KL}}, \quad (3)$$

where $r_i(\theta) = \frac{\pi_\theta(y_i|x,\mathrm{E}_l;\mathcal{R})}{\pi_{\text{old}}(y_i|x,\mathrm{E}_l;\mathcal{R})}$, π_θ denotes the trainable policy model, $\hat{A}_i$ represents the overall advantage function, $\mathbb{D}_{\text{KL}}$ denotes the KL divergence between the trained policy π_θ and the reference policy π_{ref} and β is a hyper-parameter. x are sampled from the dataset $\mathcal{D}$, and y denote the output sequence interleaving reasoning steps with search engine retrievals. During the loss calculation phase, we mask all retrieved document snippets and case-based experiences within the intermediate reasoning steps to prevent the training policy from being biased.

4 Experiments

4.1 Experimental Settings

Datasets and Evaluation Metrics. We conduct extensive experiments on six multi-hop datasets, including HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), Musique (Trivedi et al., 2022), Bamboogle (Bam) (Press et al., 2023b), MoreHopQA (MoreHQA) (Schnitzler et al., 2024), and Frames (Krishna et al., 2025). The first three datasets are in-domain datasets, with portions of their training sets used for training, while the latter three are out-of-domain datasets utilized to evaluate the model's generalization performance. Our evaluation is conducted on the full dev or test sets corresponding to the above datasets. For evaluation metrics, we employ the standard word-level F1 score (F1), Cover Exact Match (CEM), and Exact Match (EM). For more complex open-domain QA tasks, we additionally utilize LLM-as-Judge (LasJ) to ensure a fair evaluation. To evaluate domain generalization, we also perform experiments on six mathematical reasoning benchmarks, including AIME 2024/2025, AMC (Li et al., 2024), MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). For AIME, AMC benchmarks with a limited number of samples, we report Avg@32 over 32 independent runs; for others, we use Pass@1 metric.

Search Tools. An efficient search tool is essential for our search agent. We build a local retrieval environment using a dense retriever with the multilingual-e5-base (Wang et al., 2022) model, incorporating the 2018 Wikipedia corpus (Ho et al., 2020). To obtain more up-to-date information, we further utilize Tavily as a web search tool.

Baselines and Training Details. In our experiments, in addition to comparing with state-of-the-art LLMs such as *DeepSeek-R1-0528*, *Qwen3-235B-A22B*, *GPT-4.1-0414*, o4-mini-0416, and *Gemini-2.5-Pro-0325* (as shown in Table 1), we also benchmark against advanced RAG methods (Shao et al., 2023; Trivedi et al., 2023; Li et al., 2025) and RL-based agentic search models (Jin et al., 2025; Chen et al., 2025; Song et al., 2025; Zheng et al., 2025b; Wang et al., 2025). These experiments are primarily based on the Qwen2.5 models (Yang et al., 2025b), where Qwen2.5-7B and Qwen2.5-32B refer to their respective Instruct models. All training-based models are derived from their corresponding open-source models.

The training data of search agent consist of the stage-2 data from Song et al. (2025) and 8,000 randomly sampled instances from Musique. For mathematical reasoning tasks, we train on the OpenR1-Math 45k subset (Hugging Face, 2025; Yan et al., 2025). We utilize FSDP (Zhao et al., 2023) and vLLM (Kwon et al., 2023) in VeRL (Sheng et al., 2025) framework, with a sampling temperature of 1.0, top-p of 0.95 and a maximum response length of 8192. The detailed training process are shown in Appendix A.

4.2 Main Results

Table 1 provides a comprehensive evaluation of HiExp-Searcher against several strong baselines on four multi-hop benchmarks, while demonstrating the performance gains achieved by our approach when integrated into a prompt-based paradigm.

Achieves continuous performance gains. Our approach achieves significant performance improvements on multiple complex multi-hop benchmarks under all evaluation metrics. Unlike previous RL-based search agents that struggle with inefficient reasoning trajectories or redundant computations, HiExp-Searcher effectively guides the reasoning path to achieve a superior balance between response comprehensiveness and accuracy. Furthermore, our method is designed as a universal and pluggable enhancement that can be seamlessly integrated into various agentic frameworks and retrieval environments to achieve further performance boosts.

Achieve frontier LLM performance with small-scale models. The current state-of-the-art LLMs are evaluated on various multi-hop datasets. Interestingly, we find that these models struggle to effectively follow the Search-o1 series prompts to guide multi-step reasoning and retrieval. Therefore, we employ a standard RAG approach to achieve

Methods	HotpotQA†			2Wiki†			Musique†			Bamboogle‡			Average
	F1	CEM	EM	F1	CEM	EM	F1	CEM	EM	F1	CEM	EM	CEM
Prompt Based													
Qwen2.5-7B													
Vanilla RAG	29.0	22.4	20.5	32.5	27.9	27.0	11.2	5.1	3.4	17.6	12.8	10.4	17.1
Iter-RetGen	51.4	45.2	39.9	39.2	35.5	32.2	17.4	12.4	10.0	31.8	24.8	22.4	29.5
IRCoT	47.2	47.3	35.3	35.0	39.2	25.5	14.7	13.3	7.5	32.3	28.8	23.2	32.2
Search-o1*	44.4	41.2	34.2	50.8	51.0	41.8	18.1	15.5	11.1	37.5	31.2	27.2	34.7
+ HiExp	**48.7**	**47.7**	**37.1**	**54.8**	**56.8**	**45.2**	**22.4**	**18.6**	**15.1**	**44.6**	**37.6**	**33.6**	**40.2** (↑5.5)
Frontier LLMs													
DeepSeek-R1	62.5	54.0	48.0	65.7	65.0	54.0	39.9	33.0	27.5	63.0	52.8	52.0	51.2
Qwen3-235B-A22B	57.3	56.1	44.5	59.4	64.1	45.3	41.7	39.5	27.6	55.3	49.2	43.8	52.2
GPT-4.1	60.6	56.0	45.0	69.7	75.5	56.0	44.9	47.0	28.5	63.8	55.2	49.6	58.4
o4-mini	57.8	59.5	40.5	62.1	71.0	47.5	41.6	45.5	27.5	61.7	64.0	46.4	60.0
Gemini-2.5-Pro	55.6	60.5	39.5	71.8	83.0	60.5	37.0	47.0	24.5	59.7	69.6	52.0	65.0
Training Based													
Qwen2.5-7B													
Search-R1-v0.3	61.8	53.6	49.8	60.7	58.7	52.3	30.9	24.7	21.5	59.4	48.0	**47.2**	46.3
ReSearch	63.2	55.8	50.4	67.1	65.4	60.3	28.0	34.1	24.0	53.1	45.6	41.6	48.7
R1-Searcher	57.8	59.7	45.6	64.0	67.8	56.2	28.4	27.9	19.5	49.8	46.4	36.0	50.5
HiExp-Searcher	**65.4**	**60.4**	**52.4**	**74.6**	**76.5**	**66.9**	**41.7**	**36.7**	**30.7**	**61.0**	**50.4**	<u>46.4</u>	**56.0** (↑9.7)
Qwen2.5-32B													
Search-R1-v0.3	66.5	55.8	53.5	73.4	71.7	68.1	36.2	30.6	28.5	65.1	55.2	54.4	53.5
ReSearch	69.4	61.0	56.3	78.1	76.7	72.3	39.3	33.8	30.5	63.1	52.0	50.4	55.9
HiExp-Searcher	**71.2**	**62.9**	**57.8**	**81.5**	**80.4**	**75.8**	**49.2**	**41.1**	**36.2**	**68.2**	**57.2**	**54.8**	**60.4** (↑6.9)

Table 1: Overall evaluation results on the dev or test sets of four benchmarks. The best and second best results are bold and underlined, respectively. All methods are evaluated in the same local retrieval environment. * indicates the results reproduced by us. †/‡ represents in-domain/out-of-domain datasets. + indicates architectural updates, such as base model replacement or new module integration.

Methods	In-Domain		Out-of-Domain	
	F1	CEM	F1	CEM
Training-free				
(a) Doc Search only	37.8	35.9	31.3	24.2
(b) w/ E_2+E_1	**42.0**	**41.0**	**34.8**	**27.3**
(c) w/ E_3+E_1	41.0	38.9	33.4	26.8
Training				
(a) Baseline GRPO	54.2	49.6	34.4	30.3
(b) w/ E_2	59.3	56.8	36.8	33.6
(c) w/ E_3	56.1	53.2	34.8	31.5
(d) w/ E_2+E_1	**60.6**	**57.9**	**38.2**	**34.0**
(e) w/ E_3+E_1	57.7	53.8	35.5	32.7

Table 2: Ablation study on various multi-hop datasets. "w/" represent "with". Performance evaluations for all trained models utilize the corresponding optimal retrieval configurations of HEK and document.

Methods	Bam‡		Frames‡		MoreHQA‡		Avg.
	F1	LasJ	F1	LasJ	F1	LasJ	LasJ
Search-o1	60.4	66.2	26.6	33.5	25.4	36.6	45.4
ReSearch	71.9	73.8	38.7	48.5	30.9	46.5	56.3
R1-Searcher	67.2	71.3	33.4	42.6	23.5	37.9	50.6
Ours	**75.6**	**76.4**	**41.3**	**48.7**	**34.7**	**49.4**	**58.2**

Table 3: Generalization experiments on out-of-domain datasets using online search engine.

better performance. A primary contribution of our framework is the ability to empower small-scale models (e.g., 7B or 32B) to match or exceed the reasoning capabilities of much larger, frontier LLMs. Our trained 7B model achieves performance on par with GPT-4.1 and surpasses larger LLMs like DeepSeek-R1 and Qwen3-235B-A22B.

4.3 Further Analysis

4.3.1 Ablation Studies

The ablation study results presented in Table 2 underscore the substantial performance gains achieved by integrating HEK into training-free and training-based settings. In the training-free category, the inclusion of strategy-based (E_2) and case-based (E_1) experiences significantly elevates the in-domain F1 score from 37.8 to 42.0 and the CEM from 35.9 to 41.0, demonstrating the plug-and-play capability of the HiExp framework. This trend is even more pronounced in the training phase, where the full HEK configuration (E_2+E_1) propels the baseline GRPO's performance from an in-domain F1 of 54.2 to a peak of 60.6, with a correspond-

Methods	AIME24	AIME25	AMC	MATH500	Minerva	Olympia	Average
Base Model	13.2	6.1	44.5	58.4	25.4	29.0	29.4
+ HiExp	12.8	7.3	42.7	62.6	25.7	32.0	30.5 (↑1.1)
SFT	21.7	16.7	55.4	82.8	36.4	45.3	43.1 (↑13.7)
GRPO	24.1	17.8	58.8	83.4	35.3	47.4	44.5 (↑15.1)
GRPO + HiExp	26.7	23.3	62.7	84.2	37.1	46.8	**46.8** (↑17.4)

Table 4: Performance comparison across six mathematical reasoning benchmarks on Qwen2.5-Math-7B. "+ HiExp" uses the proposed hierarchical experience at inference phase without any training. "GRPO + HiExp" incorporates HiExp during GRPO training.

Methods	HotpotQA		2Wiki		Musique		Average
	F1	CEM	F1	CEM	F1	CEM	CEM
Search-o1*	44.4	41.2	50.8	51.0	18.1	15.5	35.9
+ HiExp	48.7	47.7	54.8	56.8	22.4	18.6	**41.0** (↑5.1)
GRPO	61.6	54.9	63.6	61.2	37.4	32.8	49.6
+ HiExp	65.4	60.4	74.6	76.5	41.7	36.7	**57.9**(↑8.3)
GSPO	52.3	62.7	57.9	60.0	29.6	35.7	52.8
+ HiExp	56.9	64.7	62.8	69.4	36.7	42.6	**58.9** (↑6.1)

Table 5: Performance comparison of HiExp integrated with different RL algorithms.

ing CEM increase from 49.6 to 57.9. These improvements validate that experience-aligned optimization effectively internalizes complex reasoning logic, allowing the agent to transcend the limitations of stochastic exploration and coarse outcome rewards.

A comparative analysis of experience granularities reveals that pattern-level induction (E_2) provides more effective guidance than higher-level E_3, particularly when combined with instance-level corrections (E_1). Across all benchmarks, the configuration E_2+E_1 consistently outperforms the alternative E_3+E_1 in the training section, achieving an out-of-domain F1 of 38.2 and a CEM of 34.0, which notably exceeds the out-of-domain GRPO baseline results of 34.4 and 30.3 respectively. This disparity suggests that specific task-structure patterns are more actionable for the model during the reasoning process than abstract meta-rules. Furthermore, the robust gains on out-of-domain datasets confirm that the framework facilitates the acquisition of generalized reasoning blueprints rather than mere memorization, ensuring high performance and stability even when encountering unfamiliar knowledge environments.

4.3.2 Generalization Performance Analysis

Cross-Task and Out-of-Domain Generalization. The framework demonstrates significant versatility by extending beyond multi-hop question answering into mathematical reasoning tasks. Experimental results in Table 4 show that integrating HiExp during GRPO training yields a substantial gain of +17.4 over the base model, while achieving consistent performance gains in out-of-domain scenarios. This advancement highlights that the hierarchical distillation of experience, specifically the transition from raw trajectories to strategic meta-principles, reinforces the model's fundamental logical processing and enables it to excel in domains requiring rigorous reasoning. Even in training-free scenarios, the addition of HiExp provides a consistent performance boost, confirming that the acquired experience-guided paradigms are robust enough to improve the model's reasoning ceiling without requiring further parameter updates.

Cross-Algorithm Generalization. The pluggable nature of the HiExp allows for significant performance gains across multiple RL algorithms. When integrated with various algorithms such as GRPO and GSPO (Zheng et al., 2025a), the framework yields consistent gains in CEM scores. This broad applicability across different optimization strategies and tasks confirms that experience-guided alignment is a generalizable solution for maximizing the training upper bound and inference reliability of agentic search agents.

Cross-Environment Generalization. The external information retrieval environment serves as a fundamental component for search agents. We also evaluate the agent in more realistic interactions by incorporating web search as shown in Table 3. The experimental results indicate that web search provides substantial performance gains by offering diverse and dynamic context. Throughout the training phase, the experience-guided mechanism within HiExp-Searcher effectively steers the agentic search process through hierarchical planning and grounding.

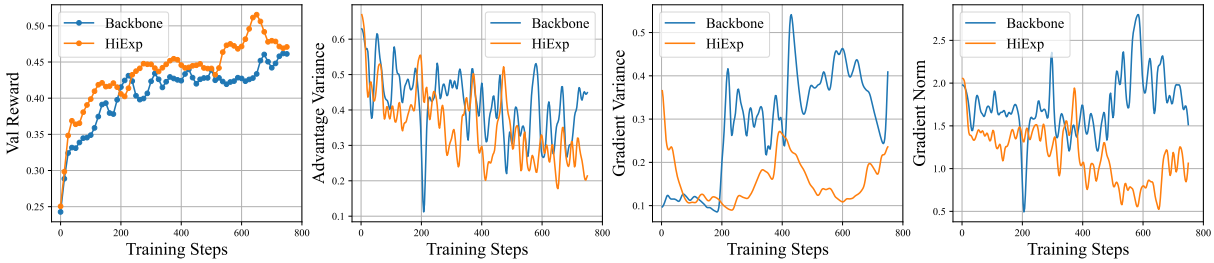

Figure 3: Training stability analysis of HiExp on multi-step retrieval benchmarks. Backbone denotes the performance of the base model trained via GRPO.

4.4 Training Stability Analysis

To evaluate the training stability of our framework, we analyze the evolution of reward signals and the variance across group rollouts, demonstrating the definitive advantages of hierarchical experience knowledge over the stochastic exploration typically observed in Figure 3. The proposed HiExp framework facilitates a more rapid and stable ascent in valid reward by leveraging hierarchical experience guidance to steer the policy model toward high-value reasoning paths. Unlike the stochastic exploration inherent in traditional reinforcement learning, which often leads to factually inconsistent queries and inefficient trajectories, HiExp ensures that sampled rollouts remain aligned with distilled reasoning principles. This strategic alignment effectively avoids the noisy or redundant trajectories that typically plague agentic search systems.

Consequently, our approach significantly reduces the variance in both advantages and gradients compared to the baseline. By providing more consistent and stable advantage estimates ($\hat{A}_i$) throughout the optimization process, HiExp suppresses gradient noise and stabilizes model updates. This improved stability allows the model to internalize efficient search behaviors more effectively, ultimately pushing the performance ceiling higher across diverse reasoning tasks.

4.5 Qualitative Analysis

To gain a deeper understanding of how hierarchical experience knowledge transforms the LLM's internal reasoning, we conduct a qualitative analysis of our experience-guided agent in Table 7. In this case, the strategy-based experience E_2 provides the "logic blueprint" by identifying a multi-hop constraint decomposition strategy. Instead of searching for plays from May 2016 in isolation, the experience instructs the model to resolve the temporal anchor first: identifying Natalie Diaz as the author of "Postcolonial Love Poem" (winning the MacArthur Fellowship in 2018), thereby establishing 2018 as the target fellowship year for the playwright. Simultaneously, the case-based E_1 experience serves as a "surgical correction" to maintain search precision during the execution phase. For example, once the agent identifies Dominique Morisseau as a 2018 MacArthur Fellow, E_1 prevents the common trap of confusing a play's premiere date with its specific composition or publication month, ensuring the model accurately targets the work written in May 2016. From an optimization perspective, this qualitative precision directly translates into the training stability discussed in Section 4.4. By suppressing redundant steps, HiExp provides more consistent and stable advantage estimates throughout the RL training process.

5 Conclusions

In this paper, we propose HiExp, an endogenous hierarchical experience construction framework tailored for search agents, which synthesizes meta-knowledge through self-reflection and agglomerative clustering over internal reasoning trajectories. HiExp facilitates the autonomous distillation of experiential priors, ensuring logical consistency while eliminating external data dependencies. Our framework not only bolsters LLM performance across diverse tasks during the inference phase but also dynamically aligns with the rollout stage of reinforcement learning. This alignment effectively transforms conventional stochastic exploration into a strategic, experience-guided search, significantly enhancing the stability and effectiveness of policy optimization. Extensive evaluations demonstrate that HiExp consistently yields substantial performance gains over state-of-the-art RL-based agents, exhibiting robust generalization across diverse task domains and reinforcement learning algorithms.

Limitations

Despite the substantial improvements in reasoning accuracy and training stability, our current framework possesses certain limitations that offer promising avenues for future research. Our current approach operates in a semi-decoupled manner, where the construction of hierarchical experience is isolated from the subsequent policy optimization. This static approach implies that the guidance distilled from the initial policy model may fail to synchronize with the model's evolving capabilities as training progresses. As the agent internalizes more sophisticated reasoning paradigms through reinforcement learning, it may encounter higher-order challenges. Therefore, a crucial future direction lies in establishing a dynamic closed-loop system where experience construction and model training are tightly coupled.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, Fan Yang, and 1 others. 2025. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025a. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Guochao Jiang, and Jingyi Song. 2025b. AirRAG: Autonomous strategic planning and reasoning steer retrieval augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 18934–18953, Suzhou, China. Association for Computational Linguistics.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3828–3850. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. s3: You don't need that much data to train a search agent via rl. *arXiv preprint arXiv:2505.14146*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 4745–4759. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

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

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023a. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023b. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.

Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. 2024. Morehopqa: More than multi-hop reasoning. *arXiv preprint arXiv:2406.13397*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 9248–9274. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM.

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*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Zijun Yao, Yantao Liu, Yanxu Chen, Jianhui Chen, Junfeng Fang, Lei Hou, Juanzi Li, and Tat-Seng Chua. 2025. Are reasoning models more prone to hallucination? *arXiv preprint arXiv:2505.23646*.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2025. Inference scaling for long-context retrieval augmented generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Dingchu Zhang, Yida Zhao, Jialong Wu, Baixuan Li, Wenbiao Yin, Liwen Zhang, Yong Jiang, Yufeng Li, Kewei Tu, Pengjun Xie, and 1 others. 2025a. Evolvesearch: An iterative self-evolving search agent. *arXiv preprint arXiv:2505.22501*.

Weizhi Zhang, Yangning Li, Yuanchen Bei, Junyu Luo, Guancheng Wan, Liangwei Yang, Chenxuan Xie, Yuyao Yang, Wei-Chieh Huang, Chunyu Miao, and 1 others. 2025b. From web search towards agentic deep research: Incentivizing search with reasoning agents. *arXiv preprint arXiv:2506.18959*.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. Pytorch FSDP: experiences on scaling fully sharded data parallel. *Proc. VLDB Endow.*, 16(12):3848–3860.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025a. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025b. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*.

A Implementation Details

Due to the large size of the dev sets in the 2WikiMultiHopQA and HotpotQA datasets, which affects iteration efficiency, we randomly sample 1,000 examples from their respective dev sets as our final test set, with a fixed random seed 42. We also verify that the performance on this subset is nearly identical to that on the full dev set, indicating that this approach can significantly improve iteration efficiency. To better understand the complexity of multi-hop reasoning in these datasets, we analyze the hop distribution of the HotpotQA, 2WikiMultiHopQA, MuSiQue, MoreHopQA, and Frames dev/test sets in Figure 4. The statistics show that there is a high proportion of complex reasoning queries with 3 hops or more. HotpotQA lacks explicit hop annotations, so we instead count the number of supporting facts. During the hierarchical experience construction, we employ the trained policy model for contrastive distillation and subsequent clustering of hierarchical experiences, thereby avoiding the introduction of external supervisory signals and enabling self-driven capability iteration and knowledge distillation.

In the retrieval process, we employ multilingual-e5-base as the retriever and use the widely used Wikipedia dump from December 2018 as the retrieval corpus, which comprises over 21 million passages. To improve retrieval efficiency, we combine the supporting document passages from five multi-hop datasets with one million randomly sampled documents from the 2018 Wikipedia dump to create our final retrieval corpus. All HEK are encoded using the same embedding model. We adopt a parent-child retrieval architecture, where succinct summary descriptions serve as child chunks for semantic matching. Upon a successful match, the corresponding detailed experiences are retrieved as parent chunks to provide the necessary context for the reasoning process. We apply a 0.8 similarity threshold for case-based experiences (E_1) to ensure high precision. For strategy-based experiences (E_2 or E_3), we select the top-5 candidates to maintain a diverse set of guidance strategies.

During the training phase of search agent, our training data consist of a total of 8,148 examples from HotpotQA and 2WikiMultiHopQA, which are selected through data selection in R1-Searcher. In addition, we randomly sample 8,000 examples from the training set of MuSiQue to form our final training set. The training consists of 2 epochs, with a `train_batch_size` of 16 and a learning rate of 1e-6. `ppo_mini_batch_size` is set to 16. The maximum lengths for prompt and response are set to 512 and 8192. Rollouts are conducted with a batch size of 8 and a temperature of 1.0 to encourage exploration. The KL-divergence regularization coefficient and the clipping ratio are set to 1e-3 and 0.2, respectively. All experiments are carried out on eight NVIDIA-H20-96G. In the inference stage, we use SGLang or vLLM as the underlying inference engines and set different maximum context lengths and maximum retrieval times to avoid the impact of outlier samples on training. For the evaluation of other prompt-based baselines, we use the implementations provided in the ReSearch GitHub repository[1]. For other training-based methods, we evaluate them using their publicly available trained models.

B Prompt Examples

Table 10 presents the implementation prompt for our LLM-as-Judge (LasJ) score. By leveraging larger and more powerful LLMs as judge models, we can achieve more accurate judgments of responses in the multi-hop QA scenarios. This more precise evaluation approach can be incorporated into the training process, which also introduces additional computational overhead for training. The prompts used for contrastive distillation and subsequent clustering of hierarchical experiences are provided in Tables 8 and 9.

C Quantitative Analysis

Table 1 in the main content presents the performance of state-of-the-art LLMs on multi-hop question answering tasks. Interestingly, we find that these models struggle to effectively follow instructions under the search-o1 paradigm, resulting in suboptimal performance. Additionally, in the basic RAG setting, where models are simply asked to answer questions based on retrieved documents, the models tend to respond that no relevant information is found when the answer is not present in the retrieved documents, failing to fully utilize their inherent capabilities. Therefore, after optimizing the prompt for the RAG scenario (see Table 11), the models are able to better integrate their own fundamental abilities with the retrieved information to jointly solve the original questions.

[1] https://github.com/Agent-RL/ReCall.

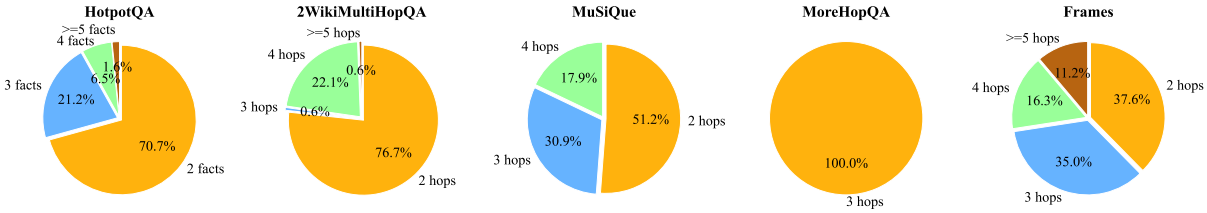

Figure 4: Overview of the distribution of query complexity over five multi-hop QA datasets.

Algorithm 1 Hierarchical Experience Construction

Require: Training set $\mathcal{D}$, rollout count K, reward function R, max depth L_{max}
Ensure: Hierarchical Experience Knowledge base $\text{HEK} = \{\text{E}_1, \text{E}_2, \ldots, \text{E}_L\}$

Initialize atomic experience set $\text{E}_1 \leftarrow \emptyset$
// Phase 1: Contrastive Experience Extraction
for each sample $x_i \in \mathcal{D}$ **do**
  $\mathcal{Y}_i \leftarrow \text{Sample_K_Rollouts}(x_i, K)$
  $\mathcal{Y}_{pos}, \mathcal{Y}_{neg} \leftarrow \text{Split_By_Reward}(\mathcal{Y}_i, R)$ ▷ Contrastive splitting
  if $\mathcal{Y}_{pos} \neq \emptyset$ **and** $\mathcal{Y}_{neg} \neq \emptyset$ **then**
    $\omega \leftarrow \text{LLM_Contrast}(x_i, \mathcal{Y}_{pos}, \mathcal{Y}_{neg})$ ▷ Extract success-critical insights
    $\text{E}_1 \leftarrow \text{E}_1 \cup \{\omega\}$
  end if
end for
// Phase 2: Self-Reflection & Iterative Hierarchical Abstraction
$\text{HEK} \leftarrow \{\text{E}_1\}$
for $l = 2$ **to** L_{max} **do**
  $Z \leftarrow \text{Encoder}(\text{E}_{l-1})$ ▷ Project insights from previous level
  $\mathcal{C}_{local} \leftarrow \text{Agglomerative_Clustering}(Z, \text{threshold}= \tau_l)$ ▷ Semantic clustering
  $\text{E}_l \leftarrow \emptyset$
  for each cluster $c \in \mathcal{C}_{local}$ **do**
    $\phi \leftarrow \text{LLM_Summarize_Cluster}(c, \text{level} = l)$ ▷ Pattern induction for current level
    $\text{E}_l \leftarrow \text{E}_l \cup \{\phi\}$
  end for
  $\text{HEK} \leftarrow \text{HEK} \cup \{\text{E}_l\}$ ▷ Append new level to knowledge base
  if $|\text{E}_l| \leq 1$ **then**
    break ▷ Convergence: Global principles reached
  end if
end for
return HEK

In this environment you have access to a set of tools you can use to assist with the user query. You may perform multiple rounds of function calls. In each round, you can call one or more functions.
Here are available functions in JSONSchema format: "'`json\n{func_schemas}\n`"'

Here are some relevant reasoning experience and examples to guide your decision-making process:
{experience}

In your response, you need to first think about the reasoning process in the mind and then conduct function calling to get the information or perform the actions if needed. The reasoning process and function calling are enclosed within `<think>` `</think>` and `<tool_call>` `</tool_call>` tags. The results of the function calls will be given back to you after execution, and you can continue to call functions until you get the final answer for the user's question. Finally, if you have got the answer, enclose it within `\boxed{}` with latex format and do not continue to call functions, i.e., `<think>` Based on the response from the function call, I get the weather information. `</think>` The weather in Beijing on 2025-04-01 is `\boxed{20C}`.

For each function call, return a json object with function name and arguments within `<tool_call></tool_call>` XML tags:
`<tool_call>\n{{"name":` <function-name>, `"arguments":` <args-json-object>`}}\n</tool_call>`

Table 6: System prompt for generating reasoning trajectories through interaction with the environments during training and inference stages.

Question: What is the name of the play written in May 2016 by a playwright who won the MacArthur Fellowship the same year as the poet who wrote "Postcolonial Love Poem"? **Answer**: Skeleton Crew

HiExp: **<think>** [Trigger E_2 Experience: Multi-hop Constraint Decomposition] This query requires a strict sequence: identify the poet → determine their award year → find a playwright from the same cohort → verify a specific play from May 2016. **</think>**
<tool_call> {"query": "which poet wrote the book 'Postcolonial Love Poem'?"} **</tool_call>**
<tool_response> ... Natalie Diaz is the author of Postcolonial Love Poem. In 2018, she received a MacArthur Fellowship ... **</tool_response>**
<think> ... **</think>**
<tool_call> {"query": "who is the playwright that won the MacArthur Fellowship in 2018?"} **</tool_call>**
<tool_response> [Trigger E_1 Experience: Temporal-Professional Attribute Intersection] ... **</tool_response>**
<think> ... **</think>**
<tool_call> {"query": "what is the name of the play written by Dominique Morisseau in May 2016?"} **</tool_call>**
<tool_response> [Trigger E_1 Experience: Temporal-Entity Work Attribution] ... **</tool_response>**
<think> ... **</think>**
<answer> The final answer is \boxed{Skeleton Crew} **</answer>**

Table 7: Quantitative analysis of the efficient reasoning process in Frames dataset.

An agent system is provided with a set of experiences and has tried to solve the question multiple times with both successful and wrong solutions. Review these problem-solving attempt and extract generalizable experiences. Follow these steps:

1. Trajectory Analysis:
- For successful steps: Identify key correct decisions, insights and formats used
- For errors: Pinpoint where and why the reasoning, answer or formatting went wrong
- Note any important patterns or strategies usedmissed
- Review why some trajectories fail? Is there any key steps are missed, or formats are wrong?
2. Experiences Summarization:
- Summarize and output with the following format:
{
"type": "The category to classify the question, including domain and solving method",
"title": "A one-sentence summary of the general experience",
"tags": ["Key words or tags, fewer than 5 words"],
"description": "Your analysis here, within 100 words",
"thinking": "Your thinking process here, especially comparing correct and incorrect solution attempts, within 100 words"
}

Table 8: Prompt for contrastive distillation.

You are given a set of experiences that an agent has accumulated while solving various questions. Your task is to cluster the similar experiences into generalized experiences that capture the core patterns and strategies. These generalized experiences should enable the agent to solve similar questions correctly and efficiently in the future.
The set of experiences is listing with the following format: [{"type": "", "title": "", "tags": "", "description": "", "thinking": "", "qa_groups": ["id": "", ...]}, ...], where qa_groups is the questions and answers in this experience group.

Summarize and output with the following format:
[{ "ids": ["all qa ids from the experiences in this cluster"], "type": "A category for this group of questions, including domain and solving method.", "title": "A one-sentence summary of the generalized strategy for this cluster.", "tags": ["A list of up to 5 keywords or tags."], "description": "Your analysis of the common patterns and core logic for this cluster, within 100 words.", "thinking": "Your thinking process here, especially differences within the group of experiences, within 100 words" }]

Table 9: Prompt for hierarchical experience clustering.

You will be provided with three pieces of content: the questioner's question, the user's response, and the reference answer list. Your task is to score the accuracy of the user's response based on the criteria outlined below. Please ensure that you carefully read and understand these instructions.
Evaluation Criteria:
Accuracy - Whether the user's answer is consistent with the reference answer and answers the questioner's question. We define this dimension as "whether the user's response includes all the key points from the reference answer and answers the questioner's question."
Evaluation Steps:
1. Carefully read the questioner's question and understand its key points.
2. Carefully read the reference answer and understand the key points relevant to the question.
3. Check whether the user's response includes all the key points from the reference answer and answers the questioner's question.
4. Based on the evaluation criteria, assign a score in the range of 0 to 5, where 0 indicates that the user's response does not include any of the key points from the reference answer and completely fails to answer the questioner's question; 5 indicates that the user's response includes all the key points from the reference answer and fully and correctly answers the questioner's question.
Example:
Questioner's question:
{question}
Reference answer:
{answer}
User's response:
{response}
Evaluation result (output only the score between 0 and 5):

Table 10: Judge prompt for LLM-as-judge scoring.

You are an expert in question answering. Given a question within <question> </question> and some contexts within <context> </context>, you first think about the reasoning process within <think> </think> and put the answer within <answer> </answer>. For example, <question> This is a question <question> <context> Here are contexts <context> <think> This is the reasoning process. </think> <answer> The final answer is \boxed{ answer here }</answer>. If the answer could not be deduced from the contexts or it's wrong, give the right answer based on your own knowledge. If the question is ambiguous or the contexts contain multiple possible answers, list all possible answers within \boxed{} with latex format, separated by commas.

Table 11: Prompt for vanilla retrieval augmented generation.