OPEN DATA SYNTHESIS FOR DEEP RESEARCH

Ziyi Xia[*] **Kun Luo**[*] **Hongjin Qian**[*] **Zheng Liu**[†]
BAAI
{ziyixia85,luokun695,chienqhj,zhengliu1026}@gmail.com

ABSTRACT

Large language models (LLMs) are increasingly expected to go beyond simple factual queries toward Deep Research—tasks that require decomposing questions into sub-problems, coordinating multi-step reasoning, and synthesizing evidence from diverse sources. We formalize Deep Research tasks with verifiable answers as Hierarchical Constraint Satisfaction Problems (HCSPs), which are fundamentally different from single-constraint, multi-hop, or flat CSP formulations. However, existing benchmarks (e.g., Natural Questions, HotpotQA) fail to capture this complexity, while recent synthetic datasets often introduce shortcut reasoning, knowledge leakage, or lack sufficient structural depth.

To address this gap, we introduce InfoSeek, a scalable framework for synthesizing complex Deep Research tasks. InfoSeek uses a dual-agent system to recursively build a Research Tree from large-scale webpages, blurring intermediate nodes into valid sub-problems, and converting these trees into natural language questions that require traversing the full hierarchy. It also enables rapid scaling, yielding over 50K training examples, a curated test set, and reasoning trajectories generated via reject sampling. Experiments show that models trained on InfoSeek consistently outperform strong baselines. On a challenging benchmark BrowseComp-Plus, 3B LLMs optimized with InfoSeek surpass much larger 32B models and lightweight commercial APIs (e.g., Gemini2.5-Flash), while achieving performance comparable to stronger APIs (e.g., Gemini2.5-Pro). By preserving meta-information such as intermediate steps and retrieval labels, InfoSeek further supports advanced optimization strategies, including compound reward design and trajectory-level exploration. We provide our codes and datasets in this repository.
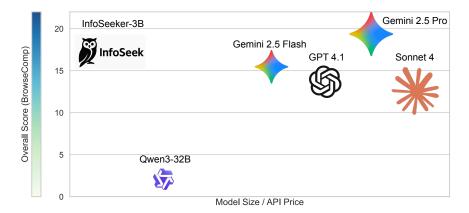
Figure 1: Performance comparison on the BrowseComp-Plus benchmark. InfoSeeker-3B, a compact LLM trained with the InfoSeek dataset, significantly outperforms Qwen3-32B and achieves performance on par with leading commercial LLMs (ordered in API prices), highlighting the strong potential of InfoSeek for advancing Deep Research tasks. Notably, the InfoSeek data synthesis framework is fully open-source, enabling convenient and scalable dataset construction.

---

[*]Equal contribution.
[†]Corresponding Author

Table 1: Comparison of the open-source status of classical QA datasets and recent data synthesis approaches for Deep Research. While prior datasets either lack structural depth or remain limited in scale, InfoSeek provides the first large-scale dataset dedicated to Deep Research scenarios, capable of generating hierarchical constraint satisfaction problems with controllable complexity, and supporting easy scalability for diverse research needs.

Name	Problem	Data Source	QA pairs	Trajectories	Framework
NQ	Single-hop	Wiki	300k+	–	–
HotpotQA	Multi-hop	Wiki	100k+	–	–
WebWalkerQA	Multi-hop	Web	14.3k	–	–
InForage	Multi-hop	Web	–	–	–
SimpleDeepSearcher	Multi-hop	–	–	871	Open
Pangu DeepDiver	Multi-hop	Web	–	–	–
WebDancer	Multi-hop	Wiki&Web	200	200	–
WebShaper	Complex	Wiki	500	–	–
InfoSeek	HCSP	Wiki&Web	50k+	16 .5k	Open

1 INTRODUCTION

Recently, large language models (LLMs) have transformed AI by generating and interpreting language with unprecedented fluency and contextual depth (OpenAI, 2023; Gemini Team, 2025). Beyond simple factual queries, emerging advances point to a new frontier: *Deep Research*—where models must decompose complex tasks, generate sub-queries, and reason across diverse sources of information (OpenAI, 2025). Unlike conventional information seeking, which suffices for straightforward retrieval, Deep Research demands synthesizing heterogeneous evidence, coordinating multi-step reasoning, and often interacting with external tools (Wu et al., 2025c; Zhang et al., 2024). Such capabilities are essential for domains like scientific discovery and policy analysis, where problems are open-ended and knowledge landscapes continuously evolve. Consequently, Deep Research is increasingly viewed as a cornerstone for the next generation of LLMs, shifting them from conversational assistants to autonomous knowledge engines (Li et al., 2025c).

A *deep research question* goes beyond simple factual lookup (OpenAI, 2025). It requires navigating multiple layers of knowledge integration, and is best understood in contrast with several simpler problem types. A constraint satisfaction problem is solved by combining several independent conditions to narrow the candidate set. A multi-hop problem demands a sequence of dependent inferences and search (Yang et al., 2018; Zhao et al., 2024; Qian et al., 2025). Deep research questions extend beyond both by involving a hierarchy of interdependent constraints that intertwine both parallel conditions and sequential steps. The solution emerges only through progressively resolving this hierarchy of sub-questions. When the final answer is unique and verifiable, the reasoning process can be naturally represented as a tree, with intermediate vertices denoting sub-questions and branches encoding their logical dependencies. In this work, rather than long-form open-ended tasks such as report writing, we focus on deep research questions that yield a unique and verifiable answer.

Some recent approaches propose carefully designed workflows for planning and tool use (Li et al., 2025b; Wu et al., 2025c; Soni et al., 2025; Zhang et al., 2025). While effective in narrow domains, these workflows lack the flexibility required for diverse Deep Research tasks (Li et al., 2025c). Another line of work enhances models' reasoning and search capabilities through supervised fine-tuning (Sun et al., 2025b) or reinforcement learning (Jin et al., 2025; Song et al., 2025a; Zheng et al., 2025). Although such methods achieve gains on traditional and multi-hop QA benchmarks, their training still depends heavily on datasets like Natural Questions (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018), which remain far simpler than real Deep Research scenarios. Other efforts (Wu et al., 2025b; Shi et al., 2025a) attempt to construct open-source QA datasets or trajectories, but these remain focused on multi-hop QA. More recent studies (Qian & Liu, 2025; Shi et al., 2025a; Wu et al., 2025a; Tao et al., 2025) explore harder question types involving web pages or Wikipedia, yet neither their datasets nor their workflows are publicly released. Table 1
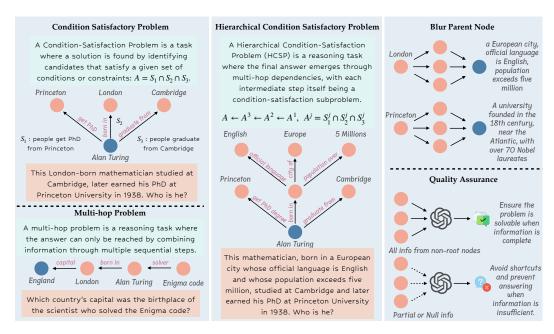
Figure 2: Illustration of Constraint Satisfaction Problems (CSP), Multi-hop Problems (MHP), and Hierarchical Constraint Satisfaction Problems (HCSP), the latter formalized as Deep Research questions with unique, verifiable answers. The right panel depicts the blurred parent node technique and quality assurance process employed in our InfoSeek data synthesis framework.

compares the open-source availability of classical QA datasets and recent data synthesis efforts for Deep Research. The results highlight a scarcity of high-quality, large-scale datasets explicitly designed for Deep Research in the open-source community.

To overcome the limitations of existing benchmarks, we propose **InfoSeek**, a novel data synthesis paradigm that generates structurally complex Deep Research questions and answers from large-scale unstructured text. As shown in Fig 2, the process begins by mining entities and their relations from raw text corpora. Leveraging these relations, we randomly sample a root entity and recursively construct a *Research Tree*, where the root denotes the final answer, internal vertices represent intermediate sub-problems, and edges encode logical dependencies among entities. Accompany with adding entities, we enrich the tree structure by blurring the parent vertices with more constraints. This helps enhance the difficulty and ensure the uniqueness of the answer to the corresponding question. Through the process, every internal vertex naturally becomes a constraint satisfaction problem (CSP), and only by solving all sub-questions and integrating multiple such layers can the reasoning path eventually converge to the root answer. To convert these structured trees into natural language questions, we provide the blurred vertex descriptions to a powerful LLM such as DeepSeek V3 or GPT-4.1. The model is prompted to generate queries whose solution requires traversing the entire tree. This ensures that each synthesized question enforces genuine multi-step reasoning, prevents potential shortcut and yields a unique, verifiable answer grounded in factual evidence. By design, InfoSeek produces datasets that are structurally diverse, complexity-controllable, and intrinsically verifiable, providing a scalable foundation for training and evaluating Deep Research agents.

To assess whether InfoSeek supports model optimization, we conduct supervised fine-tuning and reinforcement learning experiments. An SFT trajectories dataset is first constructed via rejection sampling to ensure correctness, followed by standard RL training on InfoSeek to balance exploration and exploitation. Under this pipeline, models trained with InfoSeek already outperform strong baselines. In addition, since InfoSeek preserves meta-information such as intermediate steps and retrieval labels, it offers valuable signals for designing more sophisticated RL rewards, which we leave for future work.

In brief, the work present three contributions: (1) We formalize Deep Research questions with verifiable answers as *Hierarchical Constraint Satisfaction Problems* (HCSPs), and provide a principled

distinction from simpler multi-hop and flat CSP problem types. (2) We introduce *InfoSeek*, an autonomous and scalable data synthesis framework that adheres to this definition, enabling high-quality dataset construction with explicit control over structural complexity and principled scalability. (3) Following this framework, we build a large-scale Deep Research dataset that fully records the construction process. Empirically, we verify the effectiveness of InfoSeek by fine-tuning and optimizing models on the dataset, achieving performance that consistently surpasses strong baselines. (4) The dataset with more than 50k QA pairs, 16.5k reasoning trajectories, and data construction framework of InfoSeek are fully open-sourced. Thus encouraging further research or development in the community.

2 PRELIMINARY

In this section, we lay out the foundational concepts required to formalize deep research questions. We begin by classifying fundamental problem types that serve as building blocks for more complex reasoning tasks. These classifications provide the basis for defining Hierarchical Constraint Satisfaction Problems (HCSPs), a framework that captures the layered structures and interdependent constraints characteristic of deep research. Finally, we present a method for constructing HCSPs systematically from tree representations.

2.1 FUNDAMENTAL PROBLEM TYPES

Constraint Satisfaction Problem (CSP). A group of simple research questions can be formalized as a constraint satisfaction problem, where the goal is to identify the unique answer set A that simultaneously satisfies all constraints extracted from the question. Formally, given a set of constraints $C_q = \{c_1, c_2, \ldots, c_n\}$, we define

$$A = \bigcap_{i=1}^{n} S(c_i) \quad \text{s.t. } |A| = 1, \ |S(c_i)| \geq 1 \ \forall i, \tag{1}$$

where $S(c_i)$ denotes the set of entities that satisfy constraint c_i. When $n = 1$, the CSP reduces to the base case of a *single-constraint problem*, which involves exactly one condition and yields a unique ground-truth answer. For example, the question "Who developed the theory of relativity?" corresponds to $C_q = \{c_1\}$ with $c_1 =$ "developed the theory of relativity", whose solution is $A = \{$Albert Einstein$\}$. For $n > 1$, such as in Fig. 2.a with constraints c_1 : got PhD from Princeton University in 1938, c_2 : born in London, and c_3 : graduated from University of Cambridge. The intersection $S(c_1) \cap S(c_2) \cap S(c_3)$ yields the correct answer $A = \{$Alan Turing$\}$. Solving a CSP thus requires both accurate retrieval of entities for each constraint and reasoning over their intersection.

Multi-hop Problem (MHP). A multi-hop problem is a reasoning task where the answer can only be obtained by sequentially chaining together multiple inference steps, with each step depending on the output of the previous one. Formally, given an initial constraint c, the solution is derived through a composition of reasoning functions:

$$A = S^{(k)}(c) = \underbrace{S \circ S \circ \cdots \circ S}_{k \text{ times}}(c), \tag{2}$$

where k denotes the number of reasoning hops.

Take the 3-hop question illustrated in Fig. 2.b as an example: (1) starting with $c =$ "scientist who solved the Enigma code", we obtain $S(c) = \{$Alan Turing$\}$; (2) using this entity, we resolve $S($"birthplace of Alan Turing"$) = \{$London$\}$; (3) finally, the problem reduces to a single-constraint query: "which country has London as its capital", yielding $A = \{$England$\}$.

Unlike constraint satisfaction problems, where the answer arises from intersecting parallel constraints, multi-hop problems require strictly ordered reasoning. A central challenge is that errors at intermediate steps propagate forward, potentially invalidating the final result.

2.2 DEFINITION OF DEEP RESEARCH TASK

A *deep research task* (Citron, 2024; OpenAI, 2025; Perplexity, 2025) is a complex information-seeking activity characterized by multi-layered information dependencies. It evaluates the ability of

an agent—whether a single LLM or a system of collaborating LLMs—across several dimensions: (1) multi-step reasoning, (2) decomposition of a complex question into manageable sub-questions, (3) strategic query generation and iterative search for relevant information, and (4) integration of evidence from multiple sources into a coherent final output. In this work, we study complex question answering as a representative instance of deep research, since it naturally supports both verifiability and reproducibility.

Building on fundamental problem types, we formalize a deep research question as a **Hierarchical Constraint Satisfaction Problem (HCSP)**. In an HCSP, the final answer is not directly accessible but must be progressively uncovered by satisfying a hierarchy of interdependent constraints. Solving such problems requires systematically pruning the search space at each level—eliminating candidates inconsistent with the accumulated evidence—until the root level converges to a unique valid answer. Formally, given a question x with a set of constraints $C_x = \{c_1, \ldots, c_k\}$ and a set of sub-questions $Y_x = \{y_1, \ldots, y_m\}$, we define a hierarchical decomposition $H(\cdot)$ as:

$$H(x) = \bigcap_{i=1}^{k} S(c_i) \ \cap \ \bigcap_{j=1}^{m} H(y_j), \quad \text{with} \ \bigcap \varnothing := \mathbb{U}, \tag{3}$$

where $\mathbb{U}$ denotes the universal set. The final answer A of a hierarchical constraint satisfaction problem q_H is then given by $A = H(q_H)$.

In this framework, both constraint satisfaction problem and multi-hop problem emerge as special cases of HCSP. Unlike classical CSPs that operate on flat, independent constraints, HCSPs impose structured, multi-level reasoning: the validity of higher-level conclusions depends on satisfying all lower-level constraints. This hierarchical pruning process not only parallels algorithmic paradigms such as constraint propagation in AI, but also echoes human reasoning, where complex judgments arise from integrating multiple, interdependent strands of evidence.

2.3 Constructing HCSP from Research Tree

With a clear definition of HCSP in hand, we now argue that every HCSP admits an underlying tree structure. In this subsection, we first formalize the research tree representation and the basic operations for constructing it. We then explain how to derive an HCSP from a given research tree, and finally discuss potential issues that may arise during this construction.

2.3.1 A Research Tree from Entities

In graph theory (Bondy et al., 1976), a *tree* is defined as a connected, acyclic graph. Or equivalently, a tree $T = (V, E)$ is a graph in which, for any two vertices $u, v \in V$, there exists exactly one simple path between them. Here we define a *research tree* $\mathcal{T} = (V, E)$, where each vertex $v \in V$ represents a knowledge entity (e.g., "Alan Turing", "University of Cambridge") or trivial fact (e.g., "1910s", "summer of 1925"), and each edge $(v, w) \in E$ connects two vertices and represents their relationship (e.g., "Alan Turing graduated from the University of Cambridge", "Alan Turing was born in 1910s).

A research tree can be constructed recursively as follows:

- **Base case:** A single vertex r with no edges as the root:

$$\mathcal{T} = (\{r\}, \varnothing). \tag{4}$$

- **Recursive expansion:** Given a tree $\mathcal{T} = (V, E)$, we may expand it by introducing a new vertex $w \notin V$ and connecting it with exactly one edge to some existing vertex $v \in V$:

$$\mathcal{T} = (V \cup \{w\}, E \cup \{(v, w)\}). \tag{5}$$

2.3.2 From Research Tree to HCSP

Once a research tree has been defined, we can recursively construct an HCSP that reflects its hierarchical structure:

- **Base case:** For a node v of height 1 (i.e., all children $child(v) = \{w_1, \ldots, w_n\}$ are leaves), each edge (v, w_i) is converted into a constraint c_i. A question q_v is then formed by:

$$q_v = Q(C_v), \quad \text{where } v = H(q_v), \ C_v = \{c_1, \ldots, c_n\}, \tag{6}$$

with $Q(\cdot)$ denoting the function that combines constraints into a question, and $H(\cdot)$ the decomposition operator defined in Eq. 3. This case reduces to constructing a standard CSP.

- **Recursive step:** For a node v of height ≥ 1, partition its children into leaves $\{w_1, \ldots, w_k\}$ and internal nodes $\{w_{k+1}, \ldots, w_n\}$. Each edge (v, w_i) with $i \leq k$ is converted into a constraint c_i, while each internal child w_j with $j > k$ recursively yields a sub-question $Q(w_j)$. The resulting question can be formalized by:

$$q_v = Q(C_v \cup \{Q(w_j) \mid j = k+1, \ldots, n\}). \tag{7}$$

Finally, for a research tree $\mathcal{T}$ with root r, the corresponding HCSP is obtained as: $q = Q(r)$.

2.4 POTENTIAL ISSUES IN TREE-BASED HCSP CONSTRUCTION

While the tree-based construction of HCSP provides a systematic framework, it also introduces two potential issues. *First*, the problem may be **underdetermined**: even after combining multiple constraints, the answer set may remain non-unique, leaving ambiguity in the solution space. *Second*, the problem may be **overdetermined**: in some cases, a single constraint (or a small subset of constraints) already yields a unique solution, leading to premature convergence and diminishing the role of hierarchical reasoning. Both phenomena challenge the intended multi-level structure of HCSP, making it challenge to synthesize high quality data. In practice, however, we adopt specific data construction techniques to mitigate these issues and preserve the richness of hierarchical reasoning.

3 INFOSEEK: SCALABLE DATA SYNTHESIS FOR DEEP RESEARCH

In this section, we present **InfoSeek**, a novel data synthesis framework for Deep Research tasks. Building on the theoretical foundations in Sec. 2, we design a dual-agent framework that generates structurally complex Hierarchical Constraint Satisfaction Problems (HCSPs) in an automatic and scalable manner, and incorporates mechanisms to ensure the quality and controllability of the generated questions. We first discuss the designed actions for the two agents. Then we introduce how we ensure data quality. Lastly, we present the statistics of our constructed dataset.

The tree construction process is orchestrated by two cooperating agents incrementally based on webpages and the Wikipedia dump. The Planner maintains a global view of the partially constructed tree, selecting target vertices and specific actions according to global complexity objectives. This ensures balanced growth across sequential and parallel reasoning demands. The Browser executes the Planner's chosen action by browsing webpages of the selected vertices, extracting candidate hyperlinks (for depth) or atomic claims (for constraints), and validating them for relevance. A complete iteration begins with the first step of action 1, follows by a series of actions alternating between action 2 and 3, and ends with action 4. Each step expansion is recorded with an explicit evidence trace, guaranteeing verifiability.

3.1 ACTION 1: INITIALIZATION FROM RESEARCH ANCHORS

The first step of InfoSeek data sythesis pipeline is to select an entity to create the first vertex as the root of the research tree. We use webpages and full Wikipedia dump as the underlying knowledge base. After filtering out invalid or trivial pages, each remaining page—identified by its title and containing sufficient content—is treated as an entity. Within each page, all mentioned entities (validated by the presence of a hyperlink) or facts are related vertices to the current entity vertex. And sentences that describe a relationship between the current entity and any other entities or facts are treated as candidate edges connecting them.

At the beginning of each iteration, the Planner chooses action 1 to initiate the construction of the research tree. Specifically, the Browser sample a valid entity from the Wikipedia knowledge base to serve as the final answer, and creates a vertex r as the root to form the very first research tree $\mathcal{T}_0$. The browser then performs the first extension by selecting a related entity to create a child vertex w of r, along with the connecting edge (r, w).

3.2 ACTION 2: BLURRING PARENT WITH CONSTRAINTS

Given a research tree $\mathcal{T}_t$ at step t, action 2: blurring parent with constraints, is available to expand a specific vertex with constraints. The Planner agent first identifies a vertex v whose current constraints are insufficient to uniquely derive v. The Browser then selects k claims from v's webpage that, taken together, yield the unique and determinate answer v. These claims form the child vertices $w_1, \ldots, w_k$, along with their corresponding edges. To avoid the **overdetermination** issue, we ensure that the resulting candidate sets are mutually exclusive, i.e., without inclusion relations.

3.3 ACTION 3: EXTENDING THE TREE

To increase the depth of the research tree, the Planner can perform a vertical expansion by choosing action 3. Starting from an existing vertex v that represents an entity, the Browser agent extracts a hyperlink from the corresponding article that indicates a dependency (e.g., "v was discovered by w"). This operation creates a new child node w of v, thereby extending the logical dependency chain. Such an expansion increases the height of the tree and yields a structure that requires the model to perform an additional step of reasoning.

3.4 ACTION 4: TERMINATION AND GENERATION OF THE QUESTION

Unlike prior work that often relies on surface-level heuristics for difficulty, our framework directly controls and monitors the structural complexity of the research tree. Termination is triggered only when the research tree achieves the desired complexity and all vertices have sufficient constraints. At this point, the Planner agent constructs the final complete question based on the research tree. Because the expansion operations rely on lightweight rules (i.e., hyperlink and fact extraction), the process is highly scalable and enables the cost-efficient synthesis of large-scale datasets.

3.5 DATA QUALITY ASSURANCE

To ensure our dataset is robust and effective for training, we instituted a rigorous, two-pronged quality assurance protocol focused on **Difficulty** and **Verifiability**.

The **Difficulty** criterion ensures that problems are non-trivial and cannot be solved by relying on a LLM's parametric memory alone. To validate this, we challenged Qwen2.5-32B-Inst (Team, 2024) to answer the questions directly. Our validation confirms the dataset's high degree of challenge: the model was able to correctly answer only 2% of the questions. We remove these samples to further enhance difficulty.

The **Verifiability** criterion confirms that each question is factually grounded and solvable via the generated search trajectory. We present Gemini 2.5 Flash (Comanici et al., 2025) API with the ground-truth web pages from the constructed search path, intermixed with a set of distractor documents. The LLM is then tasked with deriving the correct answer from this provided context. We filter out the questions with wrong answer, multiple possible answers, or unable to solve. This process filter out the questions that contain ambiguity or even unsolvable, effectively preventing the **underdetermined** issue.

3.6 STATISTICS

As shown in Table 2, InfoSeek comprises more than 50K samples, with the total data curation cost as \$571.8, provided for reproducibility. The distribution concentrated on problems requiring 4 to 6 reasoning vertices. To further quantify the dataset's complexity, we measure the failure rate of a powerful baseline, Qwen2.5-72B (Yang et al., 2024), using CoT prompting (Wei et al., 2022). This approach serves as a reliable proxy for deep research difficulty, as established by prior work (Wei et al., 2025). The results reveal a high overall failure rate of 92.7%, confirming that our dataset poses a significant challenge even for powerful models. Crucially, the failure rate exhibits a strong positive correlation with the number of vertices, increasing from 88.1% for 3-vertex problems to 94.1% for problems with 7 or more vertice. This trend validates that our synthesis process effectively controls for reasoning complexity.

Table 2: Analysis of costs, failure rates (Qwen2.5-72B, CoT), and token lengths by vertex count for constructed Research Tree Data.

# Vertices	Count	Failure (%)	Cost ($)	Question Len (tok)	Answer Len (tok)
3	3,841	88.1	43.9	31.97	6.17
4	15,263	91.7	142.8	43.38	5.91
5	15,051	91.0	160.4	54.35	5.75
6	17,714	92.6	214.4	65.52	5.64
≥ 7	269	94.1	10.3	81.59	5.23
Total	**52138**	**91.6**	**571.8**	**53.43**	**5.79**

4 METHOD

We introduce **InfoSeeker**, a framework for advanced agentic search and deep research, built upon our high-quality InfoSeek dataset. The core of our methodology is a novel workflow featuring parallel multi-query search and a dedicated Refiner Agent, designed to efficiently gather and synthesize vast amounts of information (Section 4.1). We then detail our two-stage training process: supervised fine-tuning (SFT) using rejection sampling to learn from successful reasoning trajectories (Section 4.2), followed by reinforcement learning (RL) to further enhance its reasoning and search capabilities (Section 4.3).

4.1 WORKFLOW WITH MULTI-QUERY SEARCH AND REFINER AGENT

Recent studies (Jin et al., 2025; Chen et al., 2025a; Li et al., 2025a) have demonstrated the potential of LLMs to reason and invoke external search engines for complex tasks. A central challenge lies in effectively searching and integrating web information into the model's context. Increasing snippet length or search depth (top-k) enhances recall by covering more information, but in multi-turn, agentic ReAct rollouts (Yao et al., 2023), this quickly leads to bloated contexts filled with redundant or noisy evidence, causing the model to lose focus. As shown in Fig. 3, we propose a novel workflow that addresses this limitation. At each step, InfoSeeker generates multiple queries in parallel, and a dedicated *Refiner Agent* condenses the corresponding retrievals into concise summaries. This design maintains high recall while keeping the working context compact and tractable.

Think Before Action. Each reasoning turn in INFOSEEKER begins with an explicit "thinking" phase, delimited by `<think>` and `</think>`. This stage encourages the model to reflect on what has already been gathered and to plan what information remains necessary, leading to more targeted query generation and robust reasoning (Yao et al., 2023; Jaech et al., 2024; Guo et al., 2025).

Parallelized Multi-Query Search To enhance the efficiency and broader coverage of information exploration, InfoSeeker generates multiple, diverse queries within a single reasoning step. These queries, enclosed within `<search>` and `</search>` tags, are designed to comprehensively address the current information-seeking intent from various angles. This parallelized approach broadens the informational coverage and accelerates the exploration process compared to a sequential, single-query strategy.

Refiner Agent for Summarization. To enable InfoSeeker to efficiently exploit massive information from the multi-query search, we introduce a Refiner Agent. For each query, the search engine returns the top-k retrieved results, which are then passed to the Refiner Agent. The agent extracts salient evidence and produces a concise summary aligned with the query's intent, while also offering recommendations for subsequent reasoning steps. Summaries are paired with their originating queries and encapsulated within `<information>` and `</information>`. In practice, we employ Qwen2.5-7B-Inst (Yang et al., 2024) as the Refiner Agent, which is both efficient and effective enough across our validations.

Output Answer. Once the model determines that sufficient information has been accumulated, or the maximum number of search steps has been reached, it produces the final answer enclosed within `<answer>` and `</answer>`.
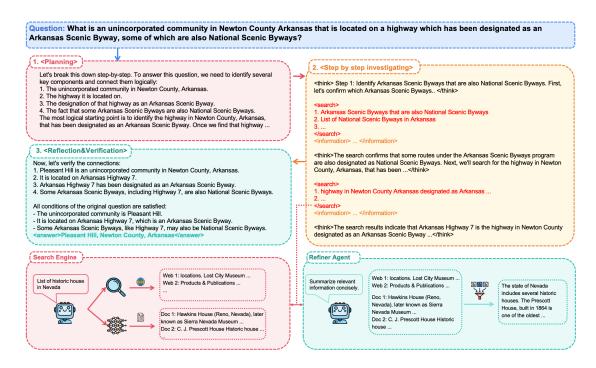
Figure 3: Framework of InfoSeeker, which decomposes target tasks, conducts step-by-step investigation, and synthesizes final answer through coordination with the Search Engine and Refiner Agent.

4.2 REJECTION SAMPLING AS SFT

A central challenge in developing deep research agents is navigating the vast exploration space of complex, multi-step reasoning tasks. Direct reinforcement learning is unstable and inefficient due to sparse rewards and a combinatorial action space, making it difficult for an agent to independently discover intricate workflows of planning, search, and verification in complex problems (Tao et al., 2025; Wu et al., 2025a).

To address this, we employ rejection sampling to construct a dataset of successful and executable reasoning trajectories for supervised fine-tuning (SFT). This approach filters for high-quality data by retaining only those trajectories that lead to correct outcomes. Specifically, we utilize both teacher model (Qwen2.5-72B (Group, 2025)) and preview version of InfoSeeker to solve tasks from our well-curated InfoSeek dataset with the workflow proposed in Sec 4.1. Only trajectories that successfully complete the task and yield a demonstrably correct final answer are accepted into our fine-tuning dataset. We further use Gemini 2.5 Flash (Gemini Team, 2025) to check if there exist search or reasoning shortcuts. SFT training details are described in Sec A.1 thoroughly.

4.3 REINFORCEMENT LEARNING WITH REASONING AND SEARCH

Reinforcement learning (RL) has been widely used in post-training phase to align models with human preferences (Ouyang et al., 2022) or strengthen LLMs' specific functionalities such as reasoning and tool using (Jaech et al., 2024; Guo et al., 2025). Previous works (Jin et al., 2025; Li et al., 2025a) have shown distinct effects on leveraging RL to enhance the capability in performaing complex searching tasks. We adopt an appropriate checkpoint from the SFT phase to start RL training, further make use of our constructed data to reinforce the model's ability in reasoning and writing precise queries.

4.3.1 ALGORITHM

Group Relative Policy Optimization (Shao et al., 2024) introduced Group Relative Policy Optimization (GRPO), a novel policy gradient based algorithm optimized for large scale training.

Table 3: Performance comparison on Single-Hop and Multi-Hop QA benchmarks. Best results in each column are highlighted in **bold**.

Model	Single-Hop QA			Multi-Hop QA				
	NQ	TQA	PopQA	HQA	2Wiki	MSQ	Bamb	Avg.
RAG-based Models								
RAG	34.8	54.4	38.7	25.5	22.6	4.7	8.0	27.0
IRCoT	11.1	31.2	20.0	16.4	17.1	6.7	8.0	15.8
RQRAG	32.6	52.5	39.4	28.5	30.7	10.1	12.9	29.5
Self-RAG	36.4	38.2	23.2	15.7	11.3	3.9	5.6	19.2
Agentic Search Models								
Search-o1-3B	23.8	48.2	26.2	22.1	21.8	5.4	32.0	25.6
Searcn-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
InfoSeeker								
InfoSeeker-3B	42.7	57.1	**48.0**	**44.6**	**52.0**	**20.5**	**39.8**	**43.5**

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=0}^{G}\frac{1}{|\mathcal{Y}|-|\mathcal{K}|}\sum_{t=1:I(\mathcal{Y}_t)=1}^{|\mathcal{Y}|}\min\left(r_{i,t}A_{i,t}\ ,\ \text{clip}\left(r_{i,t}, 1-\epsilon, 1+\epsilon\right)A_{i,t}\right)\right.$$

$$\left. -\beta\mathbb{D}_{KL}(\pi_\theta||\pi_{ref})\right]$$

As $r_{i,t}$ defined as $\frac{\pi_\theta(\mathcal{Y}_{i,t}|\mathcal{X},\mathcal{Y}_{i,<t},\mathcal{K}_{i,<t})}{\pi_{\theta_{\text{old}}}(\mathcal{Y}_{i,t}|\mathcal{X},\mathcal{Y}_{i,<t},\mathcal{K}_{i,<t})}$, GRPO adopts the clipped surrogate objective from PPO (Schulman et al., 2017), while in company with a KL divergence penalty. Instead of using GAE (Schulman et al., 2015), GRPO uses a new method to compute the advantages without value model. For each question input $\mathcal{X}$, we use the old policy model π_{old} to generate a group of outputs $\{\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_G\}$, and get G rewards $\mathbf{R} = \{R_1, R_2, \ldots, R_G\}$. Then the advantage is computed as a normalized reward: $A_{i,t} = \frac{R_i - \text{mean}(\mathbf{R})}{\text{std}(\mathbf{R})}$.

4.3.2 REWARD DESIGN

In reinforcement learning, reward serves as the fundamental signal that guides a model's learning and decision-making process. Instead of learning directly from labeled examples, the LLM generates responses and receives feedback in the form of rewards that indicate how desirable or useful those responses are. In our approach, since the model has already possess a degree of capability solving Deep Reserach tasks in desired format after the SFT phase, we design a relatively straightforward reward:

$$R = \begin{cases} 1 & \text{if format and extracted answer are both correct} \\ 0 & \text{otherwise} \end{cases}$$

This binary design reward only those responses that both follow the required structure and contain the correct extracted answer, providing a clear signal to guide the model's optimization. RL training details are described in Sec A.1 thoroughly.

Table 4: Model performance on the **BrowseComp-Plus** benchmark for complex reasoning tasks.

Model	Retriever	Accuracy (%)	Search Calls
Gemini 2.5 Flash	BM25	15.5	10.56
Gemini 2.5 Pro	BM25	19.0	7.44
Sonnet 4	BM25	14.3	9.95
GPT-4.1	BM25	14.6	11.22
GPT-5	BM25	55.9	23.23
Qwen3-32B	BM25	3.5	0.92
SearchR1-32B	BM25	3.9	1.78
InfoSeeker-3B	BM25	16.5	8.24

5 EXPERIEMENTS

5.1 EXPERIMENT SETTING

Datasets We evaluate InfoSeeker on both single-hop benchmarks: Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017), PopQA (Mallen et al., 2022); and multi-hop benchmarks: HotpotQA (HQA) (Yang et al., 2018), 2WikiMultihopQA (2Wiki) (Ho et al., 2020), Musique (MSQ) (Trivedi et al., 2022b), and Bamboogle (Bamb) (Press et al., 2022). We use Exact Match (EM) is the evaluation metric for all these datasets. For evaluating more advanced deep research capability, we use the complex BrowseComp (Wei et al., 2025) benchmark, with the filtered 830 problems and fixed webpage corpus (100K) from BrowseComp-Plus (Chen et al., 2025b). We use LLM to judge accuracy following the official setting.

Baselines We compare with RAG-based methods: (1) Vanilla RAG, which retrieves top-k documents once and prepends them to the prompt; (2) IRCoT (Trivedi et al., 2022a), alternating retrieval with chain-ofthought reasoning; (3) RQRAG (Chan et al., 2024), which refines initial queries through rewriting and decomposition to improve retrieval accuracy; (4) Self-RAG (Asai et al., 2023), which introduces a self-reflection mechanism allowing the model to critique and revise its own outputs based on retrieved evidence. We compare with more advanced agentic search methods: (1) Search-o1 (Li et al., 2025b), which enhances LLMs with an agentic retrieval module and a Reason-in-Documents component for structured document reasoning; (2) Search-R1 (Jin et al., 2025), which learns to generate multiple search queries during reasoning via reinforcement learning to optimize multi-turn retrieval interactions. (3) Zero-Search (Sun et al., 2025a), which trains search agents using reinforcement learning without real search engines by simulating retrieval with another LLM. (4) AutoRefine (Shi et al., 2025b), which introduces a "search-and-refine-during-think" paradigm using reinforcement learning with retrieval-specific rewards to iteratively filter and distill information before answering. (5) InForage (Qian & Liu, 2025) incorporate intermediate retrieval reward into agentic RL training. For BrowseComp-Plus, we compare with the following models: Gemini 2.5 Flash, Gemini 2.5 Pro (Comanici et al., 2025), Sonnet 4 (Anthropic, 2025), GPT-4.1 (OpenAI, 2023), GPT5 (OpenAI, 2025), Qwen3-32B (Yang et al., 2025), Search-R1-32B (Jin et al., 2025). The detailed evaluation details can be found in Appendix A.2.

5.2 MAIN RESULTS

InfoSeeker exhibits strong general agentic search capability. Table 3 reports results on both single-hop and multi-hop QA benchmarks. We observe that InfoSeeker consistently outperforms all baselines, including both RAG methods and recent agentic search approaches. This indicates that our training pipeline enables the model to generalize effectively across diverse reasoning scenarios. Notably, most baselines rely heavily on large amounts of in-domain supervision (i.e., more than 100K NQ&HQA), while our approach focuses on leveraging purpose-built InfoSeek dataset for training, yet still attains stronger general performance.

Table 5: Performance on BrowseComp-Plus with different training datasets.

Training Set	Retriever	Accuracy (%)	Search Calls
NQ+HQA	BM25	3.0	1.39
InfoSeeker	BM25	16.5	8.24

InfoSeeker exhibits strong deep research capability. Table 4 highlights performance on the BrowseComp-Plus benchmark, which emphasizes open-ended, search-intensive reasoning. InfoSeeker-3B (16.5%) surpasses several closed-source systems, including Gemini 2.5 Flash, Sonnet 4, and GPT-4.1, while also vastly outperforming open-source baselines such as Qwen3-32B and SearchR1-32B. Considering that InfoSeeker contains only 3B parameters, this result underscores the efficiency of our pipeline in scaling down deep research capabilities to compact LLMs.

5.3 COMPARISON WITH OTHER TRAINING DATASETS

We further evaluate the effectiveness of the InfoSeek dataset by comparing it with widely used datasets for agentic search, namely NQ and HotpotQA. Both datasets are frequently employed to train models with external information-seeking and multi-hop reasoning capabilities (Jin et al., 2025; Sun et al., 2025a; Shi et al., 2025b). We conduct RL on NQ+HQA to compare with InfoSeek, using Qwen2.5-3B-Inst (Group, 2025) as backbone LLM. As shown in Table 5, training on InfoSeek yields substantially stronger deep research performance and more effective use of search tools.

6 RELATED WORKS

Inference-Time Agentic Frameworks A significant body of work aims to enhance the problem-solving capabilities of pre-trained LLMs without altering their weights. These approaches construct agentic frameworks that operate at inference time, typically employing a central planning model to decompose complex problems and delegate sub-tasks to specialized tools or other LLM instances. For example, Wu et al. (2025c) introduced Agentic Reasoning, a framework that dynamically assigns tasks to agents specializing in web search, coding, and memory management. Similarly, AgentOrchestra (Zhang et al., 2025) proposes a hierarchical multi-agent system where a central planner delegates tasks to a suite of sub-agents. Other approaches, such as ALITA (Qiu et al., 2025), explore runtime self-evolution, enabling agents to dynamically generate and reuse tools on the fly. The primary focus of this research area is on the design of effective inference-time scaffolding and orchestration to leverage the existing capabilities of LLMs.

Training Agents for Search and Retrieval Another prominent research direction focuses on explicitly training agents to interact with external information sources, most notably search engines. Reinforcement learning (RL) is a common paradigm in this domain. For instance, R1-Searcher (Song et al., 2025b) and Search-R1 (Jin et al., 2025) both employ RL frameworks to teach LLMs how to interleave search queries with their reasoning steps, with the latter introducing techniques like retrieved token masking for more stable training. AutoRefine (Shi et al., 2025b) presents a "search-and-refine" model where an agent learns to iteratively distill information from retrieved documents, guided by both retrieval-specific and final-answer rewards. Frameworks like Search-o1 (Li et al., 2025b) emphasize modularity by separating the search workflow from a document refinement module. The central goal of these works is to develop robust algorithms and policies for agents to effectively seek and utilize external knowledge.

Automated Data Synthesis for Deep Research Agents To cultivate more advanced reasoning skills, recent research has turned to automated data synthesis for training highly capable agents. These methods span a range of strategies. Some leverage RL within complex, open-ended environments; DeepResearcher (Zheng et al., 2025), for example, scales RL for agents interacting directly with the open web, while WebSailor (Li et al., 2025a) synthesizes high-uncertainty web navigation tasks to train specialized agents. Other frameworks provide comprehensive ecosystems for agent development, such as Cognitive Kernel-Pro (Fang et al., 2025), or explore novel inference-time processes, like the iterative denoising approach of TTD-DR (Han et al., 2025). A key challenge lies in

ensuring the quality and logical consistency of the synthesized data. WebShaper (Tao et al., 2025), for instance, adopts a formalization-driven approach where a reasoning graph is defined before the corresponding question is generated. Our work contributes to this line of research on data synthesis, focusing on methods to automatically generate large-scale datasets with controllable and verifiable structural complexity to foster deep, multi-step reasoning.

7 CONCLUSION

We presented InfoSeek, a data-centric framework for advancing Deep Research with large language models. By formalizing verifiable Deep Research questions as *Hierarchical Constraint Satisfaction Problems* (HCSPs), we established a principled distinction from simpler tasks such as multi-hop and flat CSP problems, and highlighted the need for high quality data that reflects hierarchical reasoning. InfoSeek operationalizes this formulation through a scalable data synthesis pipeline: entities and relations are mined from large-scale webpages, organized into Research Trees incrementally with sufficient constraints to blur parent vertices, ensuring that intermediate vertices form valid sub-problems and that solutions require traversing the full hierarchy. This design yields datasets that are structurally diverse, complexity-controllable, and intrinsically verifiable. Empirical evaluation confirms that models trained on InfoSeek dataset with supervised fine-tuning and reinforcement learning outperform strong baselines, validating its effectiveness for enabling more robust reasoning and tool use capabilities. Furthermore, because InfoSeek preserves meta-information such as intermediate steps and retrieval labels, it opens new opportunities for compound reward design and trajectory-level optimization.

REFERENCES

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

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

John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Macmillan London, 1976.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. RQ-RAG: learning to refine queries for retrieval augmented generation. *CoRR*, abs/2404.00610, 2024. doi: 10.48550/ARXIV.2404.00610. URL `https://doi.org/10.48550/arXiv.2404.00610`.

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

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

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

Dave Citron. Try deep research and our new experimental model in gemini, your ai assistant. `https://blog.google/products/gemini/google-gemini-deep-research/`, 2024.

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

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

Google Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL `https://arxiv.org/abs/2507.06261`.

Qwen Group. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

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

Rujun Han, Yanfei Chen, Zoey CuiZhu, Lesly Miculicich, Guan Sun, Yuanjun Bi, Weiming Wen, Hui Wan, Chunfeng Wen, Solène Maître, et al. Deep researcher with test-time diffusion. *arXiv preprint arXiv:2507.16075*, 2025.

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

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

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

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

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

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

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *CoRR*, abs/2501.05366, 2025b. doi: 10.48550/ARXIV.2501.05366. URL `https://doi.org/10.48550/arXiv.2501.05366`.

Yuchen Li, Hengyi Cai, Rui Kong, Xinran Chen, Jiamin Chen, Jun Yang, Haojie Zhang, Jiayi Li, Jiayi Wu, Yiqun Chen, et al. Towards ai search paradigm. *arXiv preprint arXiv:2506.17188*, 2025c.

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

OpenAI. Gpt-4 technical report. `https://cdn.openai.com/papers/gpt-4.pdf`, 2023.

OpenAI. Introducing deep research. `https://openai.com/index/introducing-deep-research/`, 2025.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Perplexity. Introducing perplexity deep research. `https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research`, 2025.

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

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

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

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

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

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

Wenxuan Shi, Haochen Tan, Chuqiao Kuang, Xiaoguang Li, Xiaozhe Ren, Chen Zhang, Hanting Chen, Yasheng Wang, Lifeng Shang, Fisher Yu, et al. Pangu deepdiver: Adaptive search intensity scaling via open-web reinforcement learning. *arXiv preprint arXiv:2505.24332*, 2025a.

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

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning, 2025a. URL `https://arxiv.org/abs/2503.05592`.

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

Aditya Bharat Soni, Boxuan Li, Xingyao Wang, Valerie Chen, and Graham Neubig. Coding agents with multimodal browsing are generalist problem solvers. *arXiv preprint arXiv:2506.03011*, 2025.

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

Shuang Sun, Huatong Song, Yuhao Wang, Ruiyang Ren, Jinhao Jiang, Junjie Zhang, Fei Bai, Jia Deng, Wayne Xin Zhao, Zheng Liu, et al. Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis. *arXiv preprint arXiv:2505.16834*, 2025b.

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

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*, 2022a.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

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

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webdancer: Towards autonomous information seeking agency, 2025a. URL `https://arxiv.org/abs/2505.22648`.

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

Junde Wu, Jiayuan Zhu, and Yuyuan Liu. Agentic reasoning: Reasoning llms with tools for the deep research. *arXiv preprint arXiv:2502.04644*, 2025c.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

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

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

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

Weinan Zhang, Junwei Liao, Ning Li, Kounianhua Du, and Jianghao Lin. Agentic information retrieval. *arXiv preprint arXiv:2410.09713*, 2024.

Wentao Zhang, Ce Cui, Yilei Zhao, Rui Hu, Yang Liu, Yahui Zhou, and Bo An. Agentorchestra: A hierarchical multi-agent framework for general-purpose task solving. *arXiv preprint arXiv:2506.12508*, 2025.

Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K. Qiu, and Lili Qiu. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely, 2024. URL `https://arxiv.org/abs/2409.14924`.

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

A APPENDIX

A.1 TRAINING DETAILS

In this section, we introduce the detailed training pipeline and implementation details of InfoSeeker. Training deep research agents is non-trivial, particularly for small scale LLMs. The key challenges lie in the scarcity of high-quality, complex data, and the lack of a clear, reproducible training pipeline. To address this, we construct the InfoSeek-50K dataset and design a two-stage training pipeline, enabling us to train a 3B LLM (Qwen2.5-3B-Inst) (Group, 2025) that approaches the performance of proprietary models.

Distill Teacher Model Since small LLMs are inherently weaker, we begin with knowledge distillation from a larger teacher model. Specifically, we distill trajectories from Qwen2.5-72B (Group, 2025) executing research workflows proposed in Sec 4.1, which are then used for SFT of Qwen2.5-3B-Inst (Group, 2025). Concretely, we utilize 50K InfoSeek samples (For training advanced deep research capability) and 5K NQ & HQA samples (For preserving general agentic search capability), each rolled out twice. After filtering incorrect executions, we obtain 24K valid trajectories, implying that the teacher model achieves 21.8% accuracy under our carefully designed workflow. Importantly, we deliberately retain "shortcut" cases among the correct trajectories, as preserving diverse solution strategies offers valuable learning signals for small LLMs during the early stages of training.

Two Round Training Following distillation, we conduct Round 1 training to bootstrap the deep research capability of the backbone Qwen2.5-3B-Inst model. Using the 24K trajectories, we fine-tune the model for 2 epochs with a learning rate of 1e-5, weight decay of 0.01, and a context length of 16,384. Training on a single 8×H100 node completes in 2 hours, yielding InfoSeeker-3B-SFT-Round1. We then perform reinforcement learning using GRPO with outcome-based rewards. Training is conducted with a batch size of 256, a maximum of 10 turns, rollout size 5, temperature 0.8, and a search engine restricted to the top-5 retrieved contents. After 200 RL steps, we obtain InfoSeeker-3B-RL-Round1.

To further strengthen deep research abilities, we proceed to Round 2 training. We first perform rejection sampling on InfoSeeker-3B-RL-Round1, generating 16,494 trajectories from 55K source samples. These are filtered via the Gemini 2.5 Flash (Gemini Team, 2025) API, producing 3,450 high-quality trajectories characterized by multi-turn search, finer-grained task decomposition, and more accurate step-by-step reasoning. The statistics are shown in Figure 4. Using the same hyperparameters as Round 1, we obtain InfoSeeker-3B-SFT. Next, we conduct the second stage of reinforcement learning. From the original 55K data pool, we select 17K harder samples (15K InfoSeek, 2K NQ & HQA). Before training, the model generates preliminary answers, and we keep only 14K samples it fails on. Using GRPO **without KL loss** for 100 steps, we derive the final InfoSeeker-3B model.

A.2 EVALUATION DETAILS

For both single-hop and multi-hop QA tasks (NQ, TQA, PopQA, HQA, 2Wiki, MSQ, and Bamb), we employ Wikipedia-25 as the corpus, segmented into chunks of 512 tokens. Document retrieval is performed using BGE-M3 (Chen et al., 2023), with the top-5 documents selected. For the BrowseComp-Plus benchmark, we utilize the 100K web page corpus provided by the official release (Chen et al., 2025b), with BM25 (Robertson et al., 2009) serving as the retrieval method.

A.3 CASE STUDY ANDFURTHER STATISTICS

SFT Trajectory Data Figure 4 provide detailed statistics for the constructed SFT Trajectory data from Research Tree data.

Case Study Figure 5 and Figure 6 provide examples of constructed research tree structure and their visualization of InfoSeek.

Figure 4: Statistics for SFT trajectory data.

A.4 ANNOUNCEMENT

The code and data accompanying this work are released under the Apache License, Version 2.0. This permits use, modification, and distribution for research and commercial purposes, provided that proper attribution is given and the terms of the license are followed.

```
Research Tree Structure Case 1:

{"root": {
    "id": "A",
    "entity": "Russet sparrow",
    "question": "What is a species of bird that was
named by a person employed under his father between
1818 and 1824, whose wife was a British artist, and
which has three subspecies and body length is generally
no more than 6 inches?",
    "claims": [
        { "target_id": "B", "claim": "A was named by B" },
        { "target_id": "C", "claim": "A has three
subspecies" },
        { "target_id": "D", "claim": "A's body length is
generally no more than 6 inches" }
    ],
    "children": [
        {
        "id": "B",
        "entity": "John Gould",
        "claims": [
            { "target_id": "E", "claim": "B was employed
by his father between 1818 and 1824" },
            { "target_id": "F", "claim": "B's wife was
F" }
        ],
        "children": [
            { "id": "E", "entity": "None", "claims": [],
"children": [] },
            { "id": "F", "entity": "Elizabeth Gould",
"claims": [], "children": [] }
            ]
        },
        { "id": "C", "entity": "None", "claims": [],
"children": [] },
        { "id": "D", "entity": "None", "claims": [],
"children": [] }]}}
```

```
Reseach Tree visualization:

(A: Russet sparrow)
 |
 |
 |— [claim] "was named by" —> (B: John Gould)
 |   |
 |   |
 |   |— [claim] "was employed by his father (1818-1824)"
 |   |
 |   |— [claim] "wife was" —> (F: Elizabeth Gould)
 |
 |
 |— [claim] "has three subspecies"
 |
 |— [claim] "body length is generally no more than 6 inches"
```

Figure 5: Research tree structure of Case One in InfoSeek.

```
Research Tree Structure Case 2:

{"root": {
    "id": "A",
    "entity": "SV Werder Bremen (women)",
    "question": "What is a women's football team whose
first goals in the 2. Bundesliga were scored by a
player born in Korogocho, who was discovered and
developed by the Mathare Youth Sports Association?",
    "claims": [
        { "target_id": "B", "claim": "A's first goals in
the 2. Bundesliga were scored by B" }
    ],
    "children": [
        {
        "id": "B",
        "entity": "Doreen Nabwire",
        "claims": [
            { "target_id": "C", "claim": "B was
discovered and developed by C" },
            { "target_id": "D", "claim": "B was born in
D" }
        ],
        "children": [
            { "id": "C", "entity": "Mathare Youth Sports
Association", "claims": [], "children": [] },
            { "id": "D", "entity": "Korogocho", "claims":
[], "children": [] }
            ]
        }]}}
```

```
Reseach Tree visualization:

(A: SV Werder Bremen (women))
 |
 |
 |— [claim] "first goals scored by" —> (B: Doreen Nabwire)
      |
      |
      |— [claim] "discovered and developed by"
      |           —> (C:Mathare Youth Sports Association)
      |
      |
      |— [claim] "was born in" —> (D: Korogocho)
```

Figure 6: Research tree structure of Case Two in InfoSeek.