OpenSeeker: Democratizing Frontier Search Agents by Fully Open-Sourcing Training Data

Yuwen Du[1,*] **Rui Ye**[1,*,#,†] **Shuo Tang**[1] **Xinyu Zhu**[1] **Yijun Lu**[1] **Yuzhu Cai**[1] **Siheng Chen**[1,†]

[1]Shanghai Jiao Tong University, [*]Equal Core Contributions, [#]Project Lead

†*CorrespondingAuthors:yr991129@sjtu.edu.cn,sihengc@sjtu.edu.cn*

Abstract

Deep search capabilities have become an indispensable competency for frontier Large Language Model (LLM) agents, yet the development of high-performance search agents remains dominated by industrial giants due to a lack of transparent, high-quality training data. This persistent data scarcity has fundamentally hindered the progress of the broader research community in developing and innovating within this domain. To bridge this gap, we introduce **OpenSeeker**, the first fully open-source search agent (i.e., model and data) that achieves frontier-level performance through two core technical innovations: (1) Fact-grounded scalable controllable QA synthesis, which reverse-engineers the web graph via topological expansion and entity obfuscation to generate complex, multi-hop reasoning tasks with controllable coverage and complexity. (2) Denoised trajectory synthesis, which employs a retrospective summarization mechanism to denoise the trajectory, therefore promoting the teacher LLMs to generate high-quality actions. Experimental results demonstrate that OpenSeeker, trained (a single training run) on only 11.7k synthesized samples, achieves state-of-the-art performance across multiple benchmarks including BrowseComp, BrowseComp-ZH, xbench-DeepSearch, and WideSearch. Notably, trained with simple SFT, OpenSeeker significantly outperforms the second-best fully open-source agent DeepDive (e.g., 29.5% v.s. 15.3% on BrowseComp), and even surpasses industrial competitors such as Tongyi DeepResearch (trained via extensive continual pre-training, SFT, and RL) on BrowseComp-ZH (48.4% v.s. 46.7%). We fully open-source the complete training dataset and the model weights to democratize frontier search agent research and foster a more transparent, collaborative ecosystem: https://github.com/PolarSeeker/OpenSeeker.

1. Introduction

In the era of information explosion, seeking accurate, real-time, and reliable information from the vast expanse of the internet has become a fundamental pillar of modern decision-making (Marchionini, 1995; Given et al., 2023). Consequently, the ability to perform *deep search* has emerged as a non-negotiable competency for frontier Large Language Model (LLM) agents (OpenAI, 2025a). The past year has witnessed a rapid rise in the development of search agents. As recently as April 10, 2025, even the most advanced LLMs, such as OpenAI's o1 (OpenAI, 2024), struggled to surpass a score of 10 on the representative BrowseComp (Wei et al., 2025) benchmark. Yet, by March 2026, the landscape has shifted dramatically, with over ten agentic LLMs now exceeding the 50-point threshold (OpenAI, 2025b; Team et al., 2026a; Zeng et al., 2026), signaling a new era of autonomous web intelligence.

However, despite this rapid progress, the training of high-performance search agents has remained a "closed-door game" played almost exclusively by well-funded corporate entities (OpenAI, 2026; Team et al., 2026a). The most capable search agents are currently dominated by proprietary models from giants such as Google and OpenAI. While prominent labs including Kimi and Minimax have contributed open-weights models, they have remained silent regarding their training data. Even within the research community, existing works either open-source the model without data (Li et al., 2025b), provide only a fraction of data (Li et al., 2025c), or fail to achieve competitive performance (Lu et al., 2025). This persistent lack of complete high-quality training data has stifled the growth of the open-source community for nearly a year.

To bridge this gap, we, a purely academic team, introduce **OpenSeeker**, the first fully open-source search agent that achieves frontier-level performance in web search tasks. OpenSeeker is not merely an open-weights model; it is a comprehensive democratization of the search agent pipeline, providing the community with all of training data, including both complex question-answer (QA) pairs and detailed trajectories.

The high-fidelity data behind OpenSeeker is powered by two core technical innovations: **fact-grounded scalable controllable QA synthesis** and **denoised trajectory synthesis**. Specifically, (1) our QA synthesis framework is designed to move beyond simple retrieval-based tasks that current models often solve through superficial pattern matching. To ensure queries demand genuine multi-hop reasoning, we reverse-engineer the web graph starting from randomly sampled seed pages within a massive web corpus. Specifically, we perform *topological graph expansion* to identify interconnected information clusters, which are then distilled into *entity subgraphs*. By applying *entity obfuscation* to these subgraphs, we transform straightforward facts into complex reasoning puzzles that structurally mandate multi-step navigation. This approach ensures our data is fact-grounded (anchored in real-world web topology), scalable (terabytes of web archives available), and controllable (modulating difficulty through subgraph complexity). (2) Our trajectory synthesis method is designed to overcome the distractions inherent in raw web content. During generation, we employ a secondary LLM to summarize preceding tool response, providing the teacher LLM with a cleaner/denoised history to produce superior reasoning and actions. In the training phase, however, we supervise the model to predict these expert decisions while conditioning it on the *original, raw* historical trajectory. This decoupling compels the agent to internalize robust information-extraction capabilities, learning to "see through the noise" to identify the essential signals required for frontier-level performance.

To validate the efficacy of our data, we synthesize a dataset comprising 10.3k English and 1.4k Chinese samples and perform Supervised Fine-Tuning (SFT) on the Qwen3-30B-A3B (Yang et al., 2025). Despite utilizing only SFT, OpenSeeker demonstrates remarkable competitiveness against models trained by corporate entities across benchmarks including BrowseComp (Wei et al., 2025) (29.5%), BrowseComp-ZH (Zhou et al., 2025) (48.4%), xbench-DeepSearch (Xbench-Team, 2025) (74.0%), and WideSearch (Wong et al., 2025) (59.4% item F1)[1]. Notably, on the BrowseComp-ZH, OpenSeeker surpasses Alibaba's Tongyi DeepResearch (Team et al., 2025d), a model trained with extensive continual pre-training, SFT and RL (48.4 v.s. 46.7). Among other models of equivalent scale trained via only SFT, our OpenSeeker achieves the best performance on average, proving the high-quality nature of our training data.

Our primary contributions are summarized as follows:

- We propose two effective techniques: fact-grounded, scalable, controllable QA synthesis and denoised trajectory synthesis, enabling the automated generation of frontier-level training data.
- We develop and release OpenSeeker, a search agent that achieves state-of-the-art performance among open-source agents, matching or exceeding frontier solutions developed by corporate.
- We fully open-source the entire synthesis solution, the final training dataset (QA pairs and full trajectories), and the model weights, aiming to accelerating the development of search agents.

Ultimately, to the best of our knowledge, **OpenSeeker** represents the first work by a purely academic team to achieve state-of-the-art performance on frontier search benchmarks while fully open-sourcing the entirety of its training data. Developed exclusively by an academic team, our work aims to democratize search intelligence by demonstrating that strategic data synthesis can effectively bridge the performance gap with industrial-scale efforts. By providing full data transparency, we hope OpenSeeker serves as a catalyst for the research community to participate in a more open, collaborative, and healthy development of autonomous agents.

2. Related Work

The evolution of LLM-base search agents has shifted the paradigm of information retrieval from simple keyword matching to autonomous, multi-turn synthesis (Marchionini, 1995). Most contemporary search agents are architected upon the ReAct paradigm (Yao et al., 2023), which utilizes a reasoning-action-observation loop to interact with web environments[2]. Historically, this path has been dominated by corporate entities. (1) OpenAI's Deep Research (OpenAI, 2025a) pioneers the fully closed-source path, followed by a series of proprietary agents including Kimi-Researcher (Kimi, 2025), Gemini's Deep Research (DeepMind, 2025), and Perplexity's Deep Research (Perplexity, 2025). (2) Within the past six months, a wave of "open-weights" models capable of search has emerged, such as the

[1] It is worth highlighting that, due to resource constraints, these results are achieved in a single training run using default hyperparameters, without any heuristic filtering or hyperparameter optimization, leaving a large room for future research.

[2] While some parallel efforts focus on context management for agents (Ye et al., 2025; Team et al., 2025c), our work primarily focuses on the fundamental challenge of data quality.

Kimi K2/2.5 series (Team et al., 2025b; 2026a), Zhipu GLM 4.5-5 (Zeng et al., 2025; 2026), MiniMax M2-2.5 (MiniMax, 2025; 2026), and Alibaba's Tongyi DeepResearch (Team et al., 2025d). However, none of these industrial efforts have disclosed their training data, effectively maintaining a "data moat" that preserves frontier performance as a corporate secret. (3) While the research community has made significant strides with frameworks such as WebDancer (Wu et al., 2025), WebSailor (Li et al., 2025c), WebSailor-V2 (Li et al., 2025c), WebLeaper (Tao et al., 2025), AgentFounder (Su et al., 2026), DeepDive (Lu et al., 2025), and MiroThinker (MiroMind AI Team, 2025), they either lack public releases, provide only a small fraction of the data, or suffer from low data fidelity that fails to achieve competitive performance.

This status quo has left the research community lacking of the high-quality data necessary to train high-performance agents. **OpenSeeker** explicitly addresses this void by fully open-sourcing its entire synthesis pipeline and high-fidelity training data, democratizing the "recipe" for frontier search intelligence[3]. **To the best of our knowledge, OpenSeeker represents the first work by a purely academic team to achieve state-of-the-art performance on frontier search benchmarks while simultaneously open-sourcing the full training data.** Notably, our SOTA results are achieved within a single training trial without any iterative refinement, underscoring the high quality of our synthesized data and leaving substantial room for future exploration.

3. Methodology

3.1. Overview & Problem Formulation

Our primary objective is to synthesize a high-fidelity dataset $\mathcal{D} = \{(q, y, \tau^*)\}$ comprising complex queries q, ground truth answers y, and optimal tool-use trajectories τ^*. This dataset aims to empower an agent π_θ to master long-horizon tool invocation for deep search tasks.

We model the web as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes web pages and $\mathcal{E}$ denotes hyperlinks. The synthesis challenge is to derive pairs (q, y) from $\mathcal{G}$ such that solving q necessitates a trajectory $\tau = [a_1, o_1, \ldots, a_T, o_T]$ of length $T \gg 1$, where a_t are search actions and o_t are observations. We argue that to effectively train deep search agents, one must address two pivotal challenges: (1) **High-difficulty QA:** Only sufficiently complex queries compel the system to engage in a rigorous multi-turn interaction cycle involving "Reasoning → Tool Call → Tool Response". This process is essential to generate long-horizon trajectories characterized by explicit decision points and extended tool invocation chains. (2) **High-quality trajectories:** The synthesis of solution paths must rely on stable and reproducible methods to ensure that the distilled training signals represent "correct and generalizable" strategies rather than accidental successes derived from stochastic sampling.

To address these, we propose a **fact-grounded scalable controllable QA synthesis** framework and a **denoised trajectory synthesis** method. The QA synthesis framework operates on the premise of *reverse-engineering* the reasoning graph: we first identify a latent inference path within $\mathcal{G}$ and then construct a question q that structurally mandates traversing this path. Complementarily, our trajectory synthesis method utilizes dynamic context denoising to generate clear reasoning and precise tool calls. By subsequently training on raw trajectories, we enable the agent to intrinsically learn to denoise and extract relevant information from noisy tool responses.

3.2. Fact-Grounded Scalable Controllable QA Synthesis

We engineer a pipeline to construct question-answer pairs (q, y) directly from the web graph $\mathcal{G}$, as shown in Figure 1. By leveraging intrinsic connectivity, we transform static hyperlinks into dynamic reasoning paths, ensuring factual grounding and controllable complexity. This scalable framework operates in two distinct phases: *Generative Construction* to synthesize candidate pairs, and *Dual-Criteria Verification* to rigorously filter for difficulty and solvability.

3.2.1. Generative Construction: From Graph to Question

Graph Expansion. To mimic the natural process of information discovery where one clue leads to another, we initiate the pipeline by sampling a seed node $v_{seed} \sim \mathcal{V}$. Recognising that complex questions rarely reside on a single isolated page, we expand from v_{seed} by traversing its outgoing edges in $\mathcal{E}$ to gather a set of k connected nodes. This forms a local dependency subgraph $\mathcal{G}_{sub} = \{v_{seed}\} \cup \{v_i | (v_{seed}, v_i) \in \mathcal{E}\}_k$, which serves as a coherent, topologically-linked knowledge base for problem construction.

Entity Extraction. Synthesizing complex questions necessitates utilizing a generative model to reference the information cluster within the expanded subgraph $\mathcal{G}_{sub}$. However, the raw content of these nodes often contains excessive noise that can distract the generation model. To sharpen the focus, we identify the central theme y_{theme} of v_{seed} and execute an extraction function. This function distills a set of key entities from across the subgraph that are directly or indirectly related to the central theme y_{theme}, and reorganizes them into a condensed *Entity Subgraph* $\mathcal{G}_{entity}$. In this graph, nodes represent the extracted entities and edges preserve the original topological connections efficiently. This step effectively abstracts $\mathcal{G}_{sub}$ into a dense relational structure, removing textual noise while retaining the logic paths.

[3]We discuss with two concurrent works in Section A.

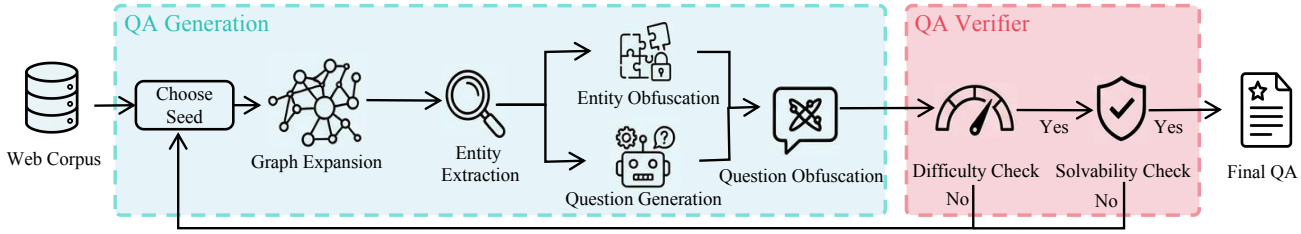

Figure 1. Overview of Fact-grounded scalable controllable QA synthesis. The pipeline begins with *Graph Expansion*, where a seed node is expanded into a subgraph of connected pages. *Entity Extraction* then distills key information themes into a structured Entity Subgraph. A generator synthesizes complex initial questions conditioned on this structure (*Question Generation*), ensuring multi-hop reasoning requirements. To enhance difficulty, we apply *Entity Obfuscation* to vagueify specific terms, finally producing a challenging question that necessitates deep graph traversal to solve.

Question Generation. To prevent the generation of questions that can be solved by simple look-up, we employ a generator P_{gen} to synthesize an initial question q_{init} conditioned explicitly on the structure of the Entity Subgraph $\mathcal{G}_{entity}$. We impose a hard structural constraint: the derivation of y_{theme} from q_{init} must necessitate traversing multiple edges within $\mathcal{G}_{entity}$. This explicitly forces the agent to engage in sequential multi-node deductive reasoning rather than single-step retrieval.

Entity Obfuscation. The synthesized questions are intended to drive agents to perform multi-step ReAct reasoning. However, agents often exploit specific keywords to shortcut the reasoning process via direct search. To simulate realistic user ambiguity and dismantle these shortcuts, we apply an obfuscation operator Φ directly to the entity nodes in $\mathcal{G}_{entity}$. Concrete entities e are mapped to vague, descriptive references $\tilde{e} = \Phi(e)$. This transformation yields a *Fuzzy Entity Subgraph* $\tilde{\mathcal{G}}_{entity}$, where the structural connectivity remains intact but the semantic nodes now demand disambiguation.

Question Obfuscation. The pipeline culminates in generating the final question $\tilde{q}$ by taking the initial question q_{init} and the fuzzy entity subgraph $\tilde{\mathcal{G}}_{entity}$ as inputs. This separation allows the generator to reference the pre-obfuscated descriptions in $\tilde{\mathcal{G}}_{entity}$ directly, thereby focusing exclusively on synthesizing the complex question structure. The generator rewrites q_{init} to incorporate the ambiguous descriptions while preserving the original reasoning logic, with the target answer remaining the invariant $y = y_{theme}$.

3.2.2. Dual-Criteria Verification via Rejection Sampling

To ensure the synthesized pair $(\tilde{q}, y)$ is both challenging and valid, we employ a rejection sampling scheme based on two indicator functions:

(1) Criterion 1: difficulty (strict tool necessity). Let π_{base} be a strong foundation model. We define the difficulty condition as $\mathbb{I}[\pi_{base}(\tilde{q}) \neq y]$, where π_{base} generates an answer in a closed-book setting (no external tools). If the model answers correctly using only parametric memory, the question is discarded. This guarantees that $\tilde{q}$ necessitates external information seeking.

(2) Criterion 2: solvability (logical consistency). We define the solvability condition as $\mathbb{I}[\pi_{base}(\tilde{q}|\mathcal{G}_{entity}) = y]$. Here, the model is provided with the full content of the Entity Subgraph $\mathcal{G}_{entity}$ as context (oracle setting). If the model fails to derive y, it implies the reasoning path is broken or hallucinated. Such samples are rejected to strictly enforce logical validity.

3.2.3. Discussions

Our data synthesis paradigm fundamentally advances agent training through three core strengths:

(1) **Factual grounding:** By anchoring queries in the real web's topology rather than relying on LLM generation, hallucination risks are significantly mitigated, if not entirely eliminated. Every training example is strictly grounded to verifiable, real-world data.

(2) **Scalability:** In this work, we leverage ∼68GB English and ∼9GB Chinese web data to testify our solution, demonstrating that it suffices to synthesize high-quality QA pairs for training high-performance search agents. With TB-scale web archives still largely untapped, our pipeline transforms the open web into an inexhaustible source. By continuously varying seed pages or adjusting graph configurations, we can generate an (almost) infinite stream of diverse, non-repeating samples, ensuring no data bottlenecks for model scaling.

(3) **Controllability:** In our solution, task difficulty is a deliberate design choice rather than a random variable By tuning the subgraph size (k), we can calibrate reasoning complexity and information coverage. This enables us to build tailored curriculums that progressively guide agents from straightforward retrieval to sophisticated, multi-hop investigations.

3.3. Denoised Trajectory Synthesis

Constructing high-quality search trajectories requires strictly balancing information retention with context window constraints. In web-scale search, raw observations are often dominated by irrelevant noise. To address this, we propose a synthesis framework that technically decouples the *generation context* (Teacher) from the *training context* (Student), employing a dynamic context denoising strategy.

3.3.1. Problem Formulation

Let a search trajectory be defined as a sequence $\tau = [q, (r_1, a_1, o_1), \ldots, (r_T, a_T, o_T), y]$, where q is the question, r_t is the reasoning step (chain-of-thought), a_t is the action (tool call), and o_t is the observation (tool response) at turn t, culminating in the final answer y. Our goal is to synthesize specific reasoning paths r_t and actions a_t that optimally lead to y.

3.3.2. Synthesis via Dynamic Context Denoising

During trajectory synthesis, we employ a **retrospective summarization** mechanism. This ensures that the agent utilizes the complete information from the immediate past while maintaining a concise long-term memory. Formally, at turn t, the agent generates the reasoning and action pair (r_t, a_t) based on the current context $\mathcal{H}_t$. Our context construction follows a "Summarized History + Raw Recent" protocol:

$$\mathcal{H}_t = \{q, \underbrace{(r_1, a_1, s_1), \ldots, (r_{t-2}, a_{t-2}, s_{t-2})}_{\text{Long-term summary}}, \underbrace{(r_{t-1}, a_{t-1}, o_{t-1})}_{\text{Short-term raw context}}\} \quad (1)$$

where $s_i = \text{Summarize}(o_i|\text{context})$ represents the compressed semantic summary of the observation o_i. This mechanism operates in a two-phase cycle:

(1) **Decision phase (information usage):** To generate the current decision (r_t, a_t), the agent is provided with $\mathcal{H}_t$, which includes the *full raw observation* o_{t-1} from the immediately preceding step. This guarantees that the agent has access to all potential signals in the most recent observation to inform its next move, preventing premature information loss.

(2) **Compression phase (context denoising):** Once step t is concluded and a new observation o_t is obtained, the system retrospectively invokes a summarizer to compress the previous observation o_{t-1} into s_{t-1}. This summary s_{t-1} then replaces o_{t-1} in the long-term history for the next step $\mathcal{H}_{t+1}$. This rolling window approach effectively filters noise and denoises the context, enabling the generation of extremely long horizons without performance degradation.

3.3.3. Asymmetric Context Training for Robust Denoising

To cultivate robustness in the final agent, we define a strategic asymmetry between the data format used for synthesis and that used for training, as shown in Figure 2. (1) **Synthesis data (teacher):** The trajectories are generated using the clean, denoised context $\mathcal{H}_t$ containing summaries. This acts as a scaffold, allowing the teacher model to produce "golden" reasoning paths unencumbered by excessive noise. (2) **Training data (student):** For the final training dataset, we strip away the summaries and revert to the full raw context:

$$\mathcal{H}_t^{train} = \{q, (r_1, a_1, o_1), \ldots, (r_{t-1}, a_{t-1}, o_{t-1})\} \quad (2)$$

The student model is supervised to predict the optimal r_t, a_t (derived from the Teacher) given the noisy, raw context $\mathcal{H}_t^{train}$. This forces the student to implicitly learn the *denoising* and *information extraction* capabilities, effectively internalizing the context denoising logic within its own parameters to handle real-world unstructured data.

4. Experiments

4.1. Experimental Setup

Implementation. We develop OpenSeeker, a deep search agent initialized from Qwen3-30B-A3B-Thinking-2507 (Team, 2025), featuring 30B total parameters with 3B activated during prediction. The maximum tool call limit is set to 200, with any trajectory exceeding this threshold being forcibly terminated. The context window size is set to 256k. Each training sample comprises a user question q and a sequence of raw reasoning steps, tool calls, and *full, uncompressed* tool responses. Note that due to resource constraints, we only train the model for a single run, without any heuristic data filtering or hyperparameter-tuning for training, leaving a large improving room for future research.

Benchmarks. We evaluate OpenSeeker on four key benchmarks: BrowseComp (Wei et al., 2025) and BrowseComp-ZH (Zhou et al., 2025), which test multi-step navigation and hard information location in English and Chinese, respectively (we evaluate BrowseComp results on a subset of 200 samples due to resource constraints); xbench-DeepSearch (Xbench-Team, 2025), assessing complex deep research capabilities like planning and synthesis; and WideSearch (Wong et al., 2025), measuring reliability in broad information seeking across extensive sources.

Baselines. To assess the efficacy of OpenSeeker, we compare it against a broad spectrum of state-of-the-art systems categorized into three groups: (1) closed-source proprietary models, representing the industry upper bound (e.g., Claude series (Anthropic, 2025), OpenAI-o3 (OpenAI, 2025b), OpenAI Deep Research (OpenAI, 2025a), GPT-5-High (Singh

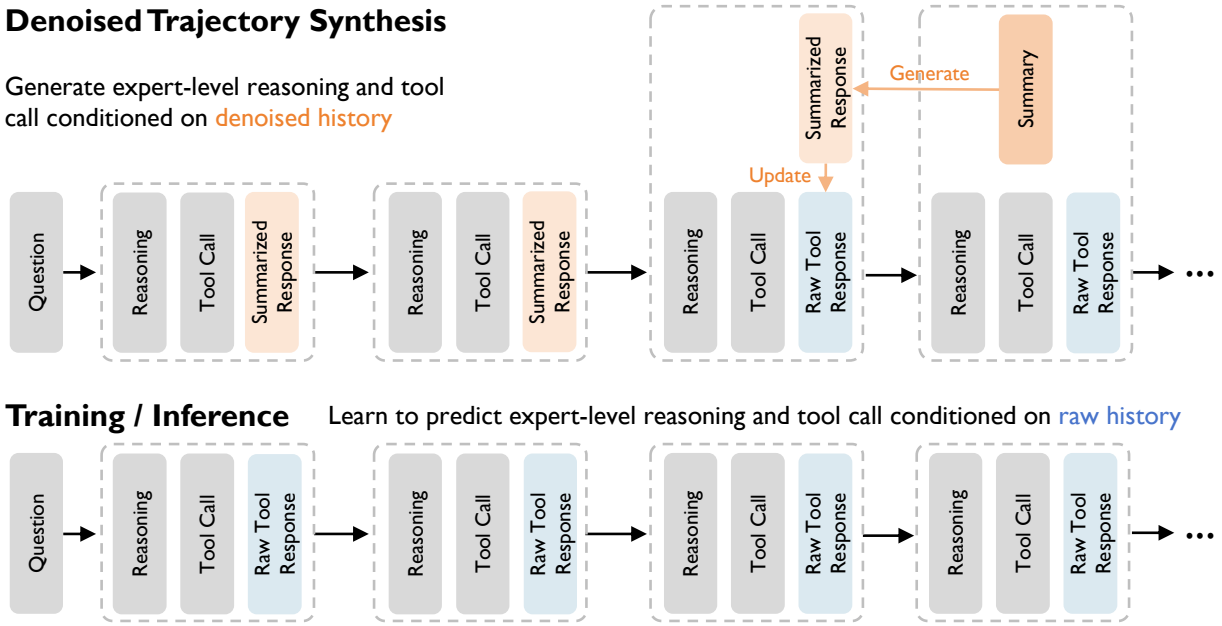

Figure 2. Overview of Denoised Trajectory Synthesis. We employ a retrospective summarization mechanism where, after each tool call, the raw tool response from the previous turn is condensed into a 'Summarized Response' that replaces the original raw tool response in the history window. This cleaner context enables the teacher to generate high-quality reasoning and actions. Note the asymmetry: while synthesis relies on summarized context, the training and inference phases operate on raw tool response to force the model to learn intrinsic denoising capabilities.

et al., 2025)); (2) large-scale open-source models, comprising massive parameter systems such as Kimi-K2 (Team et al., 2025b), DeepSeek (V3.1/V3.2 (DeepSeek-AI et al., 2025)), GLM-4 (4.6/4.7) (Team et al., 2025a), Minimax-M2 (MiniMax AI Team, 2025), and LongCat-Flash (Team et al., 2026b); and (3) ∼30B models, which serve as direct, comparable-scale benchmarks. This group includes representative search agents such as MiroThinker series (MiroMind AI Team, 2025), DeepDive-32B (Lu et al., 2025), WebDancer (Wu et al., 2025), WebSailor-V2 (Li et al., 2025b), WebLeaper (Tao et al., 2025), and Tongyi DeepResearch (Li et al., 2025a). Baseline performance metrics are derived from their respective technical reports or public leaderboards.

4.2. Main Results

Outperforming resource-intensive industry baselines. As shown in Table 1, our primary evaluation compares OpenSeeker against a spectrum of proprietary and open-source models, highlighting its superior efficiency and competitiveness achieved through just a single training run. Despite utilizing a modest dataset of only 11.7k samples and a standard SFT protocol, OpenSeeker consistently rivals or exceeds the performance of models backed by massive corporate resources. A standout result is observed on the BrowseComp-ZH benchmark, where OpenSeeker achieves a score of 48.4, surpassing Alibaba's Tongyi DeepResearch (46.7). This is particularly significant given that Tongyi DeepResearch employs a complex, resource-heavy training pipeline involving Continual Pre-Training (CPT), SFT, and Reinforcement Learning (RL), whereas OpenSeeker relies solely on high-quality SFT data.

Superior performance under identical training setup. As shown in Table 2, when evaluated under the same SFT training protocol in the ∼30B parameter class, OpenSeeker demonstrates a decisive advantage, highlighting the effectiveness of our data synthesis method. Most notably on BrowseComp-ZH, OpenSeeker (48.4) outperforms the runner-up WebSailor-V2-SFT (28.3) by nearly 20%. Additionally, models such as MiroThinker-32B-v0.1-SFT (13.8) lag significantly behind, confirming that data *quantity* (e.g., 147k for MiroThinker) is secondary to data *quality*. Our *Denoised Trajectory Synthesis* effectively teaches the model to denoise and extract "needle-in-a-haystack" information from raw, noisy web observation, a capability that standard SFT datasets often fail to cultivate.

Superior performance with comparable data volume. To further isolate the contribution of our synthesis methodology, we compare OpenSeeker against various configurations of WebSailor-V2 and WebLeaper data. As shown in Table 3, despite using a comparable or even smaller volume of data (∼11.7k samples vs. 10k–15k samples), OpenSeeker demonstrates superior performance across all benchmarks. Specifically, on xbench and WideSearch, it outperforms the

Table 1. Comparisons among our OpenSeeker and other search agents. '# Samples' denotes the number of total training data samples; '# OS Samples' denotes the number of open-source data samples; 'Training' denotes training techniques (CPT: continual pre-training, SFT: supervised fine-tuning, RL: reinforcement learning); 'Academic' denotes whether conducted by pure academic team (✓: Yes, ×: No); 'BC-ZH' denotes BrowseComp-ZH; 'WideSearch' denotes item F1 result on the English subset. With simple SFT, OpenSeeker-v1-30B-SFT even surpasses Tongyi DeepResearch on BrowseComp-ZH which is trained via CPT, SFT, and RL. **Among all SFT-based agents, OpenSeeker performs significantly best.**

Model Name	# Samples	# OS Samples	Training	Academic	BrowseComp	BC-ZH	xbench	WideSearch
Closed-Source Proprietary Models								
Claude-4-Sonnet	?	0	?	×	14.7	22.5	-	62.0
Claude-4.5-Sonnet	?	0	?	×	24.1	42.4	-	-
Claude-4-Opus	?	0	?	×	18.8	37.4	-	-
OpenAI-o3	?	0	?	×	49.1	68.7	65.0	60.0
OpenAI Deep Research	?	0	?	×	51.5	42.9	-	-
GPT-5-High	?	0	?	×	54.9	63.0	-	-
Open-Source Models > 30B								
Kimi-K2-Instruct-1T	?	0	?	×	14.1	28.8	-	59.9
DeepSeek-V3.1-671B	?	0	?	×	30.0	49.2	71.2	-
DeepSeek-V3.2-671B	?	0	?	×	51.4	65.0	-	-
GLM-4.6-357B	?	0	?	×	45.1	49.5	-	-
GLM-4.7-357B	?	0	?	×	52.0	66.6	-	-
Minimax-M2-230B	?	0	?	×	44.0	48.5	-	-
∼30B Models								
WebDancer-32B	?	0	SFT + RL	×	3.8	18.0	-	-
MiroThinker-32B-v0.1	147 k	147 k	SFT	×	10.6	13.8	-	-
MiroThinker-32B-v0.1	147 k	147 k	SFT + RL	×	13.0	17.0	-	-
DeepDive-32B	4.1 k	4.1 k	SFT + RL	×	15.3	29.7	51.8	-
WebSailor-32B	?	0	SFT + RL	×	10.5	25.5	53.3	-
WebSailor-V2-30B	?	0	SFT	×	24.4	28.3	61.7	-
WebSailor-V2-30B	?	0	SFT + RL	×	35.3	44.1	73.7	-
WebLeaper-30B	15 k	0	SFT	×	27.7	-	66.0	44.1
Tongyi DeepResearch	?	0	CPT + SFT + RL	×	43.4	46.7	75.0	-
OpenSeeker-v1-30B-SFT	11.7 k	11.7 k	SFT	✓	29.5	48.4	74.0	59.4

Table 2. Performance comparison of different models trained via SFT. Our OpenSeeker consistently and significantly performs the best across four benchmarks with only 11.7k training samples.

Data	# Samples	# OS Samples	Academic	BrowseComp	BC-ZH	xbench	WideSearch-EN
DeepDive-32B	4.1 k	4.1 k	×	9.5	23.0	48.5	-
MiroThinker-32B-v0.1	147 k	147 k	×	10.6	13.8	-	-
WebSailor-V2-30B	?	0	×	24.4	28.3	61.7	-
WebLeaper-30B	15 k	0	×	27.7	-	66.0	44.1
OpenSeeker-v1-30B-SFT	11.7 k	11.7k	✓	**29.5**	**48.4**	**74.0**	**59.4**

best baseline combination (utilizing 15k samples) by nearly 8% (74.0) and 15% (59.4), respectively. This result strongly validates the high quality and efficiency of our data, demonstrating that our synthesized samples provide significantly more effective supervision signals. In stark contrast to baselines relying on proprietary, company-synthesized datasets, OpenSeeker, developed by a purely academic research team, achieves this efficiency by leveraging our independently synthesized high-difficulty QA and high-quality denoised trajectories, while fully open-sourcing the entire dataset to the community.

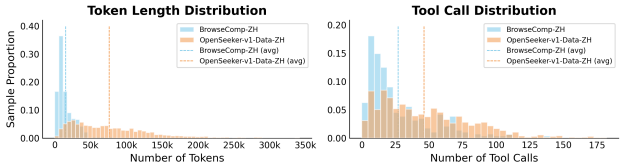

Figure 3. Comparison of difficulty between OpenSeeker-v1-Data-ZH and BrowseComp-ZH using the same model for inference. OpenSeeker-v1-Data-ZH exhibits significantly higher average token counts and tool call counts than BrowseComp-ZH.

Data statistics analysis. To quantitatively contrast the difficulty of our synthesized data with that of standard benchmarks, we employ an open-source model to perform inference on both our synthesized samples and the BrowseComp benchmarks (including BrowseComp-ZH and BrowseComp-EN). The comparison reveals that our synthesized data matches or even exceeds the difficulty of established benchmarks. Notably, although our Chinese dataset contains only approximately 1.4k samples, its complexity significantly surpasses that of BrowseComp-ZH. As illustrated in Figure 3, our synthesized Chinese data averages **46.35** tool calls per trajectory with an average token length of **76.1k**, whereas

Table 3. Performance comparison under comparable data volumes. OpenSeeker achieves significant advantages across three benchmarks, demonstrating the high quality of our data.

Data	# Samples	# OS Samples	Developer	BrowseComp	xbench	WideSearch-EN
WebSailor-V2-10k	10k	0	Tongyi	24.50	62.67	38.91
WebSailor-V2-5k + WebLeaper-Basic-5k	10k	0	Tongyi	20.67	58.33	32.26
WebSailor-V2-5k + WebLeaper-Union-5k	10k	0	Tongyi	27.50	62.33	41.70
WebSailor-V2-5k + WebLeaper-Reverse-Union-10k	15k	0	Tongyi	27.67	66.00	44.07
OpenSeeker-v1-Data-11.7k	11.7 k	11.7k	Academic	**29.50**	**74.00**	**59.40**

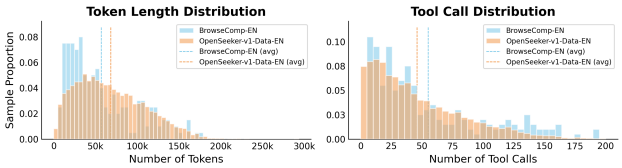

Figure 4. Comparison of difficulty between OpenSeeker-v1-Data-EN and BrowseComp-EN using the same model for inference. OpenSeeker-v1-Data-EN exhibits difficulty comparable to that of BrowseComp-EN.

BrowseComp-ZH averages only **26.98** tool calls and **15.1k** tokens. This not only demonstrates that our problems are inherently more challenging but also validates that despite the limited data volume, its high fidelity and complexity directly contribute to superior performance on Chinese benchmarks. Due to resource constraints, our English data has not yet been updated to the latest QA standards, resulting in slightly lower difficulty compared to the Chinese data (see Figure 4). We expect to release an updated version in the near future.

5. Discussions

Breaking the corporate data monopoly. For a long period, the development of high-performance search agents has been a "closed-door game" dominated by tech corporate, with high-quality data serving as their primary moat. Concurrently, existing open-source datasets often suffer from poor quality and inadequate reasoning complexity, leaving the academic community ill-equipped to train truly capable, frontier-level search models. OpenSeeker addresses this critical bottleneck. By open-sourcing a high-fidelity dataset that enables frontier-level performance, we provide the community with the necessary resources to replicate and build upon industrial-grade capabilities, breaking the long-standing "data moat".

Future work. While our current work demonstrates significant effectiveness, it represents merely a lower bound of OpenSeeker's potential. Due to resource constraints, we can only train for a single run, limiting not only the verification of effectiveness on more challenging data, but also the exploration of various parameters and data filtering strategies. In the next phase, we aim to optimize data distributions, implement rigorous quality filtering, and generate training data of even higher complexity to push the boundaries of performance. Furthermore, we plan to extend the agent's capabilities beyond pure web search by integrating a more diverse set of tools and data sources, ultimately advancing toward a more versatile and generalist agentic framework.

6. Conclusions

The growth of the open-source search agent community has long been stifled by the monopoly of high-quality training data held by industrial corporations. To bridge this gap, **OpenSeeker** represents the first work by a purely academic team to achieve state-of-the-art performance on frontier search benchmarks while simultaneously open-sourcing the full training data. Notably, our SOTA results are achieved using only 11.7k synthesized samples through a single supervised fine-tuning run, surpassing industrial baselines that rely on extensive resources and complex training pipelines. This efficiency validates the effectiveness of our proposed *fact-grounded scalable controllable QA synthesis* and *denoised trajectory synthesis* methods in producing high-fidelity training data. By openly releasing our the complete dataset, and model weights, we aim to dismantle the data barriers in this domain and foster a more inclusive, transparent, and collaborative ecosystem for future search agent research.

Impact Statement

This work aims to advance the open-source search agent community by releasing high-quality training data and model weights for search-oriented language agents. By making OpenSeeker publicly available, we seek to lower the data and resource barriers that have limited academic participation in this area, improve reproducibility, and foster a more transparent and collaborative ecosystem for future search agent research. The expected positive impacts include broader access to reliable information-seeking technologies, more efficient training pipelines, and better community understanding of search agent capabilities and limitations. As with other search agents, potential risks include misuse for large-scale automated information gathering and the propagation of biases or inaccuracies from retrieved sources; we therefore encourage careful downstream evaluation and responsible deployment, especially in high-stakes

settings.

References

Anthropic. Introducing claude 4, 2025. URL https://www.anthropic.com/news/claude-4.

Chu, Z., Wang, X., Hong, J., Fan, H., Huang, Y., Yang, Y., Xu, G., Zhao, C., Xiang, C., Hu, S., et al. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*, 2026.

DeepMind, G. Gemini 2.5, 2025. URL https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/.

DeepSeek-AI, Liu, A., Mei, A., Lin, B., Xue, B., Wang, B., Xu, B., Wu, B., Zhang, B., Lin, C., Dong, C., Lu, C., Zhao, C., Deng, C., Xu, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Li, E., Zhou, F., Lin, F., Dai, F., Hao, G., Chen, G., Li, G., Zhang, H., Xu, H., Li, H., Liang, H., Wei, H., Zhang, H., Luo, H., Ji, H., Ding, H., Tang, H., Cao, H., Gao, H., Qu, H., Zeng, H., et al. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, December 2025. doi: 10.48550/arXiv.2512.02556. URL https://arxiv.org/abs/2512.02556.

Given, L. M., Case, D. O., and Willson, R. *Looking for information: Examining research on how people engage with information*. Emerald Publishing Limited, 2023.

Kimi. Kimi-researcher: End-to-end rl training for emerging agentic, 2025. URL https://moonshotai.github.io/Kimi-Researcher/.

Li, B., Zhang, B., Zhang, D., Huang, F., Li, G., Chen, G., Yin, H., Wu, J., Zhou, J., Li, K., Su, L., Ou, L., Zhang, L., Xie, P., Ye, R., Yin, W., Yu, X., Wang, X., Wu, X., Chen, X., Zhao, Y., Zhang, Z., Tao, Z., Zhang, Z., Qiao, Z., Wang, C., Yu, D., Fu, G., Shen, H., Yang, J., Lin, J., Zhang, J., Zeng, K., Yang, L., Yin, H., Song, M., Yan, M., Liao, M., Xia, P., Xiao, Q., Min, R., Ding, R., Fang, R., Chen, S., Huang, S., Wang, S., Cai, S., Shen, W., Wang, X., Guan, X., Geng, X., Shi, Y., Wu, Y., Chen, Z., Li, Z., and Jiang, Y. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, Oct 2025a. doi: 10.48550/arXiv.2510.24701.

Li, K., Zhang, Z., Yin, H., Ye, R., Zhao, Y., Zhang, L., Ou, L., Zhang, D., Wu, X., Wu, J., et al. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. *arXiv preprint arXiv:2509.13305*, 2025b.

Li, K., Zhang, Z., Yin, H., Zhang, L., Ou, L., Wu, J., Yin, W., Li, B., Tao, Z., Wang, X., et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025c.

Li, Z., Jiang, D., Ma, X., Zhang, H., Nie, P., Zhang, Y., Zou, K., Xie, J., Zhang, Y., and Chen, W. Openresearcher: A fully open pipeline for long-horizon deep research trajectory synthesis. https://www.notion.so/OpenResearcher-A-Fully-Open-Pipeline-for-Long-Hor 2025d. Notion Blog.

Lu, R., Hou, Z., Wang, Z., Zhang, H., Liu, X., Li, Y., Feng, S., Tang, J., and Dong, Y. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn rl. *arXiv preprint arXiv:2509.10446*, 2025.

Marchionini, G. *Information seeking in electronic environments*. Number 9. Cambridge university press, 1995.

MiniMax. Minimax m2 & agent: Ingenious in simplicity, 2025. URL https://www.minimax.io/news/minimax-m2.

MiniMax. Minimax m2.5: Built for real-world productivity, 2026. URL https://www.minimax.io/news/minimax-m25.

MiniMax AI Team. MiniMax M2 & Agent: Ingenious in Simplicity, October 2025. URL https://www.minimax.io/news/minimax-m2. Open-sourced model weights on Hugging Face: https://huggingface.co/MiniMaxAI/MiniMax-M2.

MiroMind AI Team. Mirothinker: An open-source agentic model series trained for deep research and complex, long-horizon problem solving, 2025. URL https://github.com/MiroMindAI/MiroThinker.

OpenAI. Introducing openai o1-preview. https://openai.com/index/introducing-openai-o1-preview/, 2024. Accessed: 2025-01-22.

OpenAI. Deep research system card, 2025a. URL https://cdn.openai.com/deep-research-system-card.pdf.

OpenAI. Introducing openai o3 and o4-mini, 2025b. URL https://openai.com/index/introducing-o3-and-o4-mini/.

OpenAI. Introducing gpt-5.2, 2026. URL https://openai.com/index/introducing-gpt-5-2/.

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

Singh, A., Fry, A., Perelman, A., Tart, A., Ganesh, A., El-Kishky, A., McLaughlin, A., Low, A., Ostrow, A., Ananthram, A., Nathan, A., Luo, A., Helyar, A., Madry, A., Efremov, A., Spyra, A., Baker-Whitcomb, A., Beutel, A., Karpenko, A., Makelov, A., Neitz, A., et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, December 2025. doi: 10.48550/arXiv.2601.03267. URL https://arxiv.org/abs/2601.03267.

Su, L., Zhang, Z., Li, G., Chen, Z., Wang, C., Song, M., Wang, X., Li, K., Wu, J., Chen, X., Qiao, Z., Zhang, Z., Yin, H., Cai, S., Fang, R., Tao, Z., Yin, W., Ye, R., Jiang, Y., Zhang, N., Xie, P., Huang, F., Ye, K., Tu, K., Qian, C., and Zhou, J. Scaling agents via continual pre-training. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Dru5mm9anE.

Tao, Z., Shen, H., Li, B., Yin, W., Wu, J., Li, K., Zhang, Z., Yin, H., Ye, R., Zhang, L., et al. Webleaper: Empowering efficiency and efficacy in webagent via enabling info-rich seeking. *arXiv preprint arXiv:2510.24697*, 2025.

Team, G., Zeng, A., Lv, X., Zheng, Q., Hou, Z., Chen, B., Xie, C., Wang, C., Yin, D., Zeng, H., Zhang, J., Wang, K., Zhong, L., Liu, M., Lu, R., Cao, S., Zhang, X., Huang, X., Wei, Y., Cheng, Y., An, Y., Niu, Y., Wen, Y., Bai, Y., Du, Z., Wang, Z., Zhu, Z., Zhang, B., Wen, B., Wu, B., Xu, B., Huang, C., Zhao, C., Cai, C., Yu, C., Li, C., Ge, C., Huang, C., Zhang, C., Xu, C., Zhu, C., Li, C., Yin, C., Lin, D., Yang, D., Jiang, D., Ai, D., Zhu, E., Wang, F., Pan, G., Wang, G., Sun, H., Li, H., Li, H., Hu, H., Zhang, H., Peng, H., Tai, H., Zhang, H., Wang, H., Yang, H., Liu, H., Zhao, H., Liu, H., Yan, H., Liu, H., Chen, H., Li, J., Zhao, J., Ren, J., Jiao, J., Zhao, J., Yan, J., Wang, J., Gui, J., Zhao, J., Liu, J., Li, J., Li, J., Lu, J., Wang, J., Yuan, J., Li, J., Du, J., Du, J., Liu, J., Zhi, J., Gao, J., Wang, K., Yang, L., Xu, L., Fan, L., Wu, L., Ding, L., Wang, L., Zhang, M., Li, M., Xu, M., Zhao, M., Zhai, M., Du, P., Dong, Q., Lei, S., Tu, S., Yang, S., Lu, S., Li, S., Li, S., Shuang-Li, Yang, S., Yi, S., Yu, T., Tian, W., Wang, W., Yu, W., Tam, W. L., Liang, W., Liu, W., Wang, X., Jia, X., Gu, X., Ling, X., Wang, X., Fan, X., Pan, X., Zhang, X., Zhang, X., Fu, X., Zhang, X., Xu, Y., Wu, Y., Lu, Y., Wang, Y., Zhou, Y., Pan, Y., Zhang, Y., Wang, Y., Li, Y., Su, Y., Geng, Y., Zhu, Y., Yang, Y., Li, Y., Wu, Y., Li, Y., Liu, Y., Wang, Y., Li, Y., Zhang, Y., Liu, Z., Yang, Z., Zhou, Z., Qiao, Z., Feng, Z., Liu, Z., Zhang, Z., Wang, Z., Yao, Z., Wang, Z., Liu, Z., Chai, Z., Li, Z., Zhao, Z., Chen, W., Zhai, J., Xu, B., Huang, M., Wang, H., Li, J., Dong, Y., and Tang, J. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025a. URL https://arxiv.org/abs/2508.06471.

Team, K., Bai, Y., Bao, Y., Chen, G., Chen, J., Chen, N., Chen, R., Chen, Y., Chen, Y., Chen, Y., et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025b.

Team, K., Bai, T., Bai, Y., Bao, Y., Cai, S., Cao, Y., Charles, Y., Che, H., Chen, C., Chen, G., et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026a.

Team, M., Bai, S., Bing, L., Chen, C., Chen, G., Chen, Y., Chen, Z., Chen, Z., Dai, J., Dong, X., et al. Mirothinker: Pushing the performance boundaries of open-source research agents via model, context, and interactive scaling. *arXiv preprint arXiv:2511.11793*, 2025c.

Team, M. L., Gui, A., Li, B., Tao, B., Zhou, B., Chen, B., Zhang, C., Gao, C., Zhang, C., Han, C., et al. Longcat-flash-thinking-2601 technical report, 2026b.

Team, Q. Qwen3-30b-a3b-thinking-2507, 2025. URL https://huggingface.co/Qwen/Qwen3-30B-A3B-Thinking-2507.

Team, T. D., Li, B., Zhang, B., Zhang, D., Huang, F., Li, G., Chen, G., Yin, H., Wu, J., Zhou, J., et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025d.

Wei, J., Sun, Z., Papay, S., McKinney, S., Han, J., Fulford, I., Chung, H. W., Passos, A. T., Fedus, W., and Glaese, A. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Wong, R., Wang, J., Zhao, J., Chen, L., Gao, Y., Zhang, L., Zhou, X., Wang, Z., Xiang, K., Zhang, G., et al. Widesearch: Benchmarking agentic broad info-seeking. *arXiv preprint arXiv:2508.07999*, 2025.

Wu, J., Li, B., Fang, R., Yin, W., Zhang, L., Tao, Z., Zhang, D., Xi, Z., Jiang, Y., Xie, P., et al. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*, 2025.

Xbench-Team. Xbench-deepsearch, 2025. URL https://xbench.org/agi/aisearch.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Ye, R., Zhang, Z., Li, K., Yin, H., Tao, Z., Zhao, Y., Su, L., Zhang, L., Qiao, Z., Wang, X., et al. Agentfold: Long-horizon web agents with proactive context management. *arXiv preprint arXiv:2510.24699*, 2025.

Zeng, A., Lv, X., Zheng, Q., Hou, Z., Chen, B., Xie, C., Wang, C., Yin, D., Zeng, H., Zhang, J., et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

Zeng, A., Lv, X., Hou, Z., Du, Z., Zheng, Q., Chen, B., Yin, D., Ge, C., Xie, C., Wang, C., et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Zhou, P., Leon, B., Ying, X., Zhang, C., Shao, Y., Ye, Q., Chong, D., Jin, Z., Xie, C., Cao, M., et al. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*, 2025.

Table 4. Performance comparison of concurrent works trained via SFT.

Data	# Samples	# OS Samples	Academic	BrowseComp	BC-ZH	xbench
Openresearcher	96k	96k	×	26.3	-	65
REDSearcher	?	10k	×	**34.7**	26.8	-
OpenSeeker-v1-Data-11.7k	11.7 k	11.7k	✓	29.5	**48.4**	**74.0**

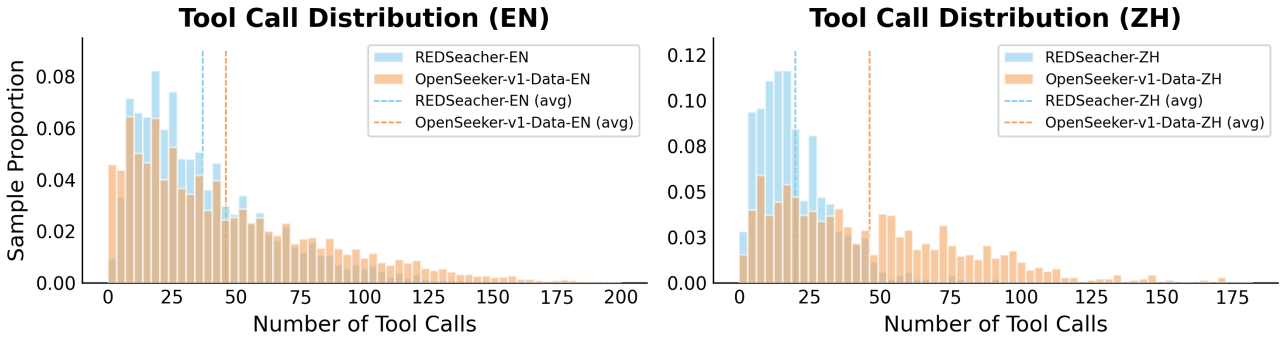

Figure 5. Comparison of tool call counts using the same model for inference: OpenSeeker's data vs. REDSearcher's data. OpenSeeker's data demonstrates a significantly higher average number of tool calls.

A. Concurrent Works

As the field moves toward more transparent search agent development, several concurrent efforts have emerged, yet they differ significantly from OpenSeeker in methodology and openness. (1) OpenResearcher (Li et al., 2025d) primarily aggregates QA pairs from existing open-source datasets and constructs trajectories within simulated environments. In contrast, OpenSeeker creates entirely new, high-difficulty QA pairs via our graph-grounded synthesis and collects trajectories within *real-world* web environments to ensure better generalizability. Furthermore, OpenSeeker demonstrates superior data quality, outperforming OpenResearcher (that is trained using 96k samples) with only 11.7k high-fidelity samples. (2) RedSearcher (Chu et al., 2026) adopts a multi-stage pipeline involving mid-training, SFT, and Reinforcement Learning (RL). However, it lacks full transparency regarding its training protocol and only provides a partial release of its SFT and RL data. Crucially, both OpenResearcher and RedSearcher involve significant corporate participation. OpenSeeker distinguishes itself as the first academic-led initiative to achieve state-of-the-art performance with a lean, SFT-only approach and 100% data transparency, proving that strategic data synthesis can bridge the gap traditionally filled by massive corporate compute and iterative RL cycles.

Quantitative results further validate these advantages. As shown in Table 4, when trained using the same SFT methodology, OpenSeeker comprehensively outperforms OpenResearcher across three benchmarks. Notably, on BrowseComp-ZH, OpenSeeker surpasses RedSearcher by 21.6% (48.4% vs 26.8%). Furthermore, as illustrated in Figure 5, when employing the same model for inference to compare tool call counts, OpenSeeker's data proves significantly more challenging than that of RedSearcher. Specifically, the average number of tool calls for OpenSeeker-v1-Data-EN averages 45.92 calls against 36.91 for RedSearcher-EN, while OpenSeeker-v1-Data-ZH is 46.35 compared to 20.02 for RedSearcher-ZH.