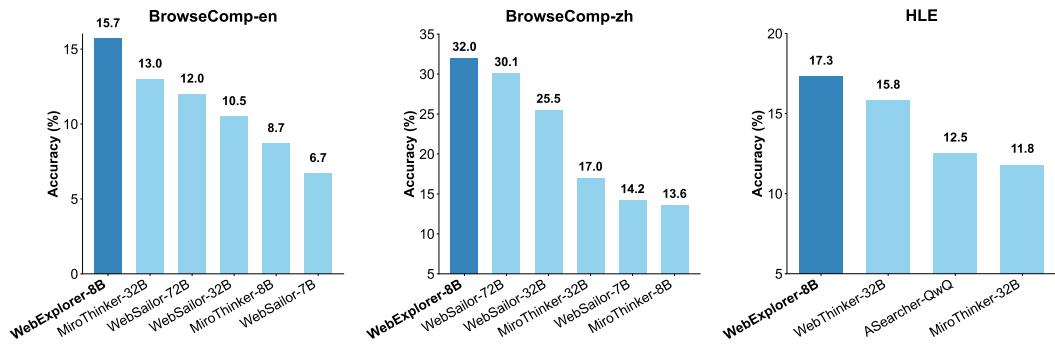


WEBEXPLORER: EXPLORE AND EVOLVE FOR TRAINING LONG-HORIZON DEEP RESEARCH AGENTS

000
001
002
003
004
005 **Anonymous authors**
006 Paper under double-blind review
007
008
009
010

ABSTRACT

011 The paradigm of Large Language Models (LLMs) has increasingly shifted toward
012 agentic applications, where web browsing capabilities are fundamental for retrieving
013 information from diverse online sources. However, existing open-source **deep**
014 **research** agents either demonstrate limited information-seeking abilities on com-
015 plex tasks or lack transparent implementations. In this work, we identify that the
016 key challenge lies in the scarcity of challenging data for information seeking. To
017 address this limitation, we introduce WEBEXPLORER: a systematic data gener-
018 ation approach using *model-based exploration* and iterative, *long-to-short query*
019 *evolution*. This method creates challenging query-answer pairs that require multi-
020 step reasoning and complex web navigation. By leveraging our curated high-
021 quality dataset, we successfully develop advanced **deep research** agent WEBEX-
022 PLOKER-8B through supervised fine-tuning followed by reinforcement learning.
023 Our model supports 128K context length and up to 100 tool calling turns, en-
024 abling long-horizon problem solving. Across diverse information-seeking bench-
025 marks, WEBEXPLORER-8B achieves the state-of-the-art performance at its scale.
026 Notably, as an 8B-sized model, WEBEXPLORER-8B is able to effectively search
027 over an average of 16 turns after RL training, achieving higher accuracy than
028 WebSailor-72B on BrowseComp-en/zh and attaining the best performance among
029 models up to 100B parameters on WebWalkerQA and FRAMES. Beyond these
030 information-seeking tasks, our model also achieves strong generalization on the
031 HLE benchmark even though it is only trained on knowledge-intensive QA data.
032 These results highlight our approach as a practical path toward long-horizon **deep**
033 **research** agents.
034



045 Figure 1: Performance comparison on BrowseComp-en, BrowseComp-zh and HLE benchmarks
046 across different models.
047

1 INTRODUCTION

048
049 The rapid advancement of Large Language Models (LLMs) has spurred a paradigm shift toward
050 agentic intelligence, where models are equipped with diverse tools to tackle increasingly complex
051 problems. Web browsing agents (OpenAI, 2025; Perplexity AI, 2025) represent a critical component
052 of this shift, enabling autonomous information retrieval from the vast landscape of online sources.
053

054 Despite significant progress, existing open-source models such as WebThinker (Li et al., 2025b)
 055 and WebSailor (Li et al., 2025a) exhibit limited performance on challenging search-based bench-
 056 marks like BrowseComp-en/zh (Wei et al., 2025; Zhou et al., 2025), BrowseComp-Plus (Chen et al.,
 057 2025) and Xbench (Xbench-Team, 2025), while stronger commercial models remain opaque in their
 058 training details (Zhipu AI, 2025; OpenAI, 2025).

059 We hypothesize that the fundamental challenge in developing capable **deep research agents** lies in
 060 the quality of training data. Current benchmarks have evolved to include queries of extraordinary
 061 difficulty – For example, over half of BrowseComp-en queries fail to be solved by human annota-
 062 tors. While such challenging queries are rare in typical use cases, constructing high-quality, difficult
 063 query-answer pairs is essential for developing agents that can achieve super-human performance on
 064 information-seeking tasks. Previous approaches to web navigation data construction have primarily
 065 employed two methodologies: Graph-based methods (Li et al., 2025a; Tao et al., 2025) construct
 066 explicit graphs through random walks where each node represents a website or entity and each
 067 edge represents a relationship, then utilize subgraphs to formulate QA pairs. Evolution-based ap-
 068 proaches (Wu et al., 2025a; Gao et al., 2025) iteratively modify existing queries to increase their
 069 complexity. However, these approaches face limitations. Previous graph construction introduces
 070 complexity in node expansion and selection heuristics. Meanwhile, evolution-based methods usu-
 071 ally increase the complexity by injecting new information to replace original content at each evolu-
 072 tion step, potentially resulting in unnatural query formulations and limited flexibility in synthesis.
 073 Our framework WEBEXPLORER addresses these limitations through a novel approach that leverages
 074 model-based exploration to construct the information space and iterative long-to-short query evolu-
 075 tion to reduce salient information. This enables more natural and flexible synthesis of challenging
 076 web navigation tasks.

077 Rather than constructing explicit graphs through rule-based methods, which is a complex process
 078 requiring decisions about expansion strategies and node selection, we adopt a simpler model-based
 079 approach to explore the information space purely through prompting. Starting from a seed entity, we
 080 leverage powerful LLMs to simulate the graph-building process internally through iterative search
 081 and browsing actions. This approach enables flexible, dynamic exploration of information spaces
 082 related to the seed entity without the overhead of explicit graph construction. The model then utilizes
 083 this explored information space to construct initial query-answer pairs.

084 In our preliminary experiments, however, we observed that these initially constructed QA pairs,
 085 while requiring multiple websites to solve, still remained relatively straightforward evidenced by
 086 the high success rates achieved by strong proprietary models. To address this limitation, we intro-
 087 duce a second-stage evolution process. Specifically, unlike previous work where evolution involves
 088 injecting new information (Wu et al., 2025a; Gao et al., 2025), we prompt models to systemati-
 089 cally increase query difficulty by *removing* explicit clues and introducing strategic obfuscation. In
 090 contrast to the initial QA pairs with clear search entry points, the evolved QA pairs require longer
 091 solution processes with more exploratory search attempts. This systematic evolution generates chal-
 092 lenging queries that lead to significantly lower success rates and require more reasoning steps from
 093 proprietary models. Through this process, we construct our WEBEXPLORER-QA data.

094 Our training utilizes a typical approach combining supervised fine-tuning for cold-start initializa-
 095 tion, followed by reinforcement learning using the GRPO algorithm (Shao et al., 2024). Our RL
 096 training scales to 128K context length and 100 maximum number of tool calling turns, where we
 097 observe consistent increases in both the number of tool calls and benchmark performance. Based on
 098 Qwen3-8B (Yang et al., 2025), our WEBEXPLORER-8B achieves state-of-the-art performance at its
 099 scale on multiple information-seeking benchmarks, including BrowseComp-en/zh (Wei et al., 2025;
 100 Zhou et al., 2025), GAIA (Mialon et al., 2024), WebWalkerQA (Wu et al., 2025b), Frames (Krishna
 101 et al., 2024), and XBench-DeepSearch (Xbench-Team, 2025). Notably, our model achieves 15.7%
 102 on BrowseComp-en and 32.0% on BrowseComp-zh, significantly outperforming the previous leading
 103 WebSailor 72B model despite a much smaller size. It also achieves 62.7% on WebWalkerQA and
 104 75.7% on FRAMES, establishing the best performance among models up to 100B parameters.
 105 Beyond superior performance on information-seeking tasks, WEBEXPLORER-8B demonstrates re-
 106 markable generalization to the academic benchmark HLE (Phan et al., 2025), scoring 17.3% and
 107 outperforming previous 32B models like WebThinker-32B (Li et al., 2025b). This validates the
 108 strong generalization capability of our approach beyond pure information-seeking tasks. Overall,
 109 the superior performance of WEBEXPLORER-8B across diverse benchmarks strongly validates the

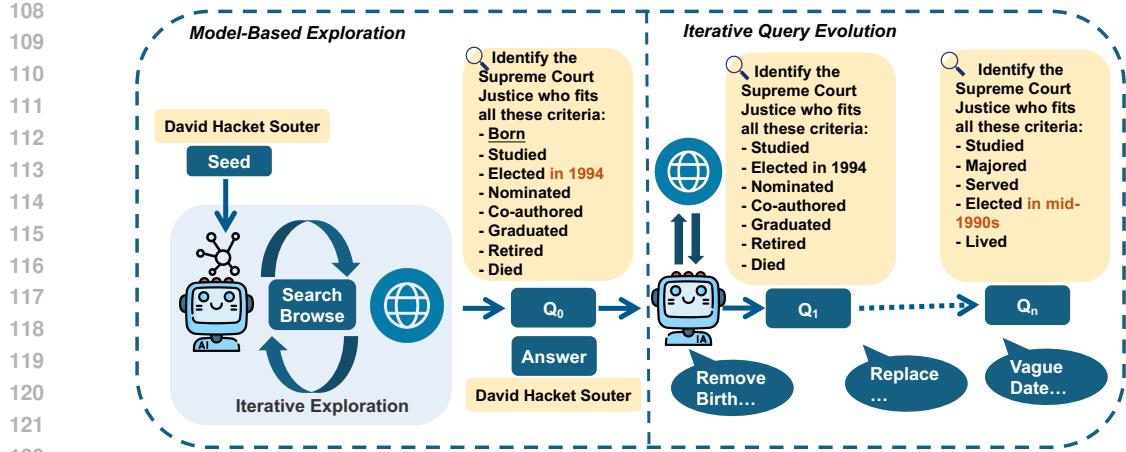


Figure 2: **Model-Based Exploration and Iterative Query Evolution Framework.** Starting from a seed entity (e.g., David Hackett Souter), the framework employs iterative search and browsing actions to construct the information space related to the seed entity. Initial queries (Q_0) and Answers are generated based on this explored information space. Through iterative evolution, salient information is systematically obfuscated (e.g., “Remove Birth...”, “Replace ...” or “Vague Date...”) to produce more challenging queries (Q_1 to Q_n). This process ensures the resulting queries require longer reasoning steps and explorations.

quality of our synthesized data and establishes a practical pathway toward building capable long-horizon [deep research agents](#).

2 WEBEXPLORER: SYNTHESIZING CHALLENGING QA PAIRS

Contemporary information-seeking benchmarks present great challenges, exemplified by BroweComp-en (Wei et al., 2025) where more than half of the queries remain unsolvable by human annotators despite two hours of dedicated effort. These challenging benchmarks rely predominantly on manual curation (Wei et al., 2025; Zhou et al., 2025; Mialon et al., 2024), resulting in prohibitively high annotation costs and severely limited dataset sizes that preclude their use in large-scale training. Concurrently, existing open-source training data exhibit some limitations: they either lack sufficient scale to enable large-scale model training or present overly simplistic queries that fail to match the difficulty of evaluation benchmarks (Wu et al., 2025a; Li et al., 2025a; Tao et al., 2025; Pahuja et al., 2025). Consequently, the autonomous synthesis of large-scale, challenging search-oriented query-answer pairs emerges as a pivotal research challenge in developing advanced deep research agents.

Several works have explored synthesizing challenging information-seeking QA pairs through two predominant methodological paradigms. Graph-based approaches explicitly construct knowledge graphs where each node represents a website or entity, then generate QA pairs by selecting subgraphs from the complete graph structure. Typically, larger subgraphs correspond to more challenging queries (Li et al., 2025a; Tao et al., 2025). Evolution-based methods iteratively modify existing queries by injecting new information to systematically increase their complexity (Wu et al., 2025a; Gao et al., 2025), typically evolving queries to become progressively longer. However, the previous rule-based graph construction introduces complexity of node expansion and selection heuristics. Instead, our method WEBEXPLORER utilizes existing LLMs to navigate the web and construct the information space autonomously. Additionally, traditional evolution approaches that inject new information at every step can become unnaturally complex. Rather than following short-to-long evolution, our iterative query evolution operates in a *long-to-short* manner by strategically removing salient information, creating natural obfuscation. As illustrated in Figure 2, through these two stages, WEBEXPLORER proposes a simple and flexible prompting-based framework for challenging QA pair generation.

162
163

2.1 PRELIMINARIES

164
165
166
167
168
169

Our agent framework builds upon the ReAct paradigm (Yao et al., 2023), where agents execute iterative cycles of reasoning and acting. When presented with a query, the agent conducts multiple rounds of Thought-Action-Observation sequences. During each round, the language model formulates a Thought τ_t based on the current context, executes a structured Action α_t , and receives corresponding environmental feedback as Observation o_t . The final round contains only Thought τ_t without further actions and observations.

170
171
172
173
174
175

Our action space encompasses two primary tools: $\alpha_t \in \{\text{search}, \text{browse}\}$. The search action enables querying search engines to retrieve relevant information, while the browse action facilitates direct webpage access for content extraction. We formalize each action as $\alpha_t = (\alpha_t^m, \alpha_t^p)$, where α_t^m specifies the action type and α_t^p contains the necessary parameters. For search operations, α_t^p encompasses a list of query strings, whereas browse operations require target URLs and extraction objectives within α_t^p . More details about these tools can be found in §4.1.

176
177

The complete execution trajectory spanning T iterations is represented as:

178
179
180

$$H_T = (\tau_0, \alpha_0, o_0, \tau_1, \alpha_1, o_1, \dots, \tau_T) \quad (1)$$

181
182
183
184

At each timestep t , the agent produces thought τ_t and selects action α_t according to policy $\pi(\tau_t, \alpha_t | H_{t-1})$, conditioning on the accumulated trajectory history H_{t-1} . We provide a detailed trajectory example in Appendix A.

185
186

2.2 MODEL-BASED EXPLORATION

187
188
189
190
191
192
193
194

Traditional graph-based methods (Li et al., 2025a; Tao et al., 2025) require heuristic rules of explicit graph construction, typically initiated from a root URL or seed entity, followed by systematic expansion through predefined traversal strategies. The resulting graph contains structured information where usually nodes represent entities and edges form the relationships. Such approaches involve iteratively identifying related entities, extracting their features, and expanding the graph until reaching predetermined size constraints. This process introduces some complexity, requiring careful, heuristic design of expansion strategies and node selection heuristics.

195
196
197
198
199
200
201
202
203
204
205
206

In contrast, we propose WEBEXPLORER, which employs a different model-based exploration approach by leveraging powerful LLMs to construct the information space autonomously. Our method operates purely through prompting: we provide a seed entity as the initial search entry along with three example QA pairs, then instruct the model to conduct iterative search and browsing actions to explore the information space before synthesizing QA pairs. Specifically, given an entity as a seed, the models conduct iterative search and browsing actions to construct a comprehensive information space encompassing entity-related content. This approach simulates the graph-building process internally, enabling flexible and dynamic exploration without the complexity of graph expansion strategy design and explicit graph maintenance. Subsequently, utilizing the explored information space, models generate QA pairs that necessitate reasoning across multiple websites to reach the correct solution. A detailed example demonstrating how the model iteratively explores and then forms a query-answer pair using multiple sources of searched information is provided in Appendix C.

207
208

Formally, starting from a seed entity e_0 , the exploration process conducts multiple rounds of actions and observations:

209
210
211
212

$$H_T = (e_0, \tau_0, \alpha_0, o_0, \tau_1, \alpha_1, o_1, \dots, \tau_T) \quad (2)$$

213
214
215

where H represents the complete information space encompassing all explored content, and τ_T contains the synthesized QA pair. The model autonomously determines when to stop searching to formulate a challenging QA pair, eliminating the need for predefined stopping criteria and allowing the model to explore flexibly.

216 2.3 ITERATIVE QUERY EVOLUTION
217218
219

Example of BrowseComp-en

220
221
222
223
224

Query: Please identify the fictional character who occasionally breaks the fourth wall with the audience, has a backstory involving help from selfless ascetics, is known for his humor, and had a TV show that aired between the 1960s and 1980s with fewer than 50 episodes. **Answer:** *Plastic Man*

225
226
227
228
229
230
231
232
233
234
235
236

While the initial QA pairs constructed from model-explored information spaces successfully incorporate content from multiple websites, we observe that proprietary models can still solve them with relatively high accuracy. For instance, as shown in Table 1, Claude-4-Sonnet achieves 86.6% accuracy on the initial QA pairs compared to 12.2% on BrowseComp-en and 68.3% on GAIA. This indicates that despite requiring multi-website reasoning, these initial QA pairs remain insufficiently challenging. Through systematic case analysis, we identify that the initial queries contain excessive explicit information and salient clues that substantially reduce their difficulty. For example, the initial query-answer example shown below demonstrates several salient clues such as “*the official attendance set a record*” and “*this player died at the age of 44*” that serve as clear and strong indicators. Such specific information—including dates, locations, and proper names—often provides direct entry points that enable straightforward solution trajectories without requiring exploratory detours or consideration of alternative reasoning paths.

237
238
239
240
241
242
243

In contrast, examination of challenging information-seeking benchmarks like BrowseComp-en reveals a critical distinguishing characteristic: these queries deliberately avoid providing clear, specific clues, instead employing vague descriptions. As the example above shows, the BrowseComp-en query “*TV show that aired between the 1960s and 1980s with fewer than 50 episodes*” demonstrates this obfuscation. Therefore, these challenging queries usually do not contain clear search entry points, which necessitate extensive exploration and multiple reasoning attempts before reaching the correct answer (Gao et al., 2025).

244
245
246
247
248
249
250
251
252
253
254
255

Drawing inspiration from BrowseComp’s design principles, we implement an iterative query evolution process to systematically increase QA pair difficulty. Unlike previous evolution methods that follow a short-to-long approach (Wu et al., 2025a; Gao et al., 2025), our evolution primarily reduces excessive information from initial QA pairs to increase difficulty. Given the complete initial QA construction information, we explicitly prompt models to refine queries through three strategic directions: (1) removing salient information, (2) introducing strategic obfuscation to specific details such as dates, locations, and proper names, and (3) identifying alternative descriptions to replace original explicit references. Through multiple iterative evolution cycles, we generate substantially more challenging QA pairs that require extensive exploration and demonstrate significantly reduced success rates. As demonstrated in the example below, the evolved query eliminates some salient descriptions in purple color like *this player died at the age of 44* and obscures concrete entities by replacing explicit references such as *Manchester United* with vague descriptors like *First Division giant*.

256
257
258

Example of Evolved Query

259
260
261
262
263
264

Initial Query: A football match took place in a stadium where *the official attendance set a record* that still stands today for FIFA World Cup matches. The referee of this match was *the oldest person to ever officiate a World Cup final*, and *exactly 26 years after this match*, he was the chairman of a club that defeated *Manchester United* in an *FA Cup final*. The player who scored the winning goal in that FA Cup final was born in an area that became part of its current city in *1920*, and *this player died at the age of 44*. In what minute of the FA Cup final was the winning goal scored? **Answer:** *83rd minute*

265
266
267
268
269

Evolved Query: In the *unique* FIFA World Cup tournament format that concluded without a knockout final, a match official later guided a Second Division club to victory over a *First Division giant* in the monarch’s final attendance at such an *occasion*. The match-winner had been rejected by the club he supported as a child, hailing from a district that joined a centuries-old Royal Naval stronghold *two decades* into the 20th century. In which minute did this decisive strike occur? **Answer:** *83rd minute*

Dataset	Initial QA	Evolved QA	WebDancer	SailorFog	WebShaper	ASearcher
Average Turns	7.9	9.9	5.4	8.2	8.4	6.5
Accuracy (%)	86.6	67.1	62.0	35.0	67.4	62.0

Table 1: Comparison of average tool calling turns and accuracy (%) of Claude-4-Sonnet across web navigation datasets, reporting our Initial QA, Evolved QA, and other QA datasets (WebDancer, SailorFog, WebShaper, ASearcher).

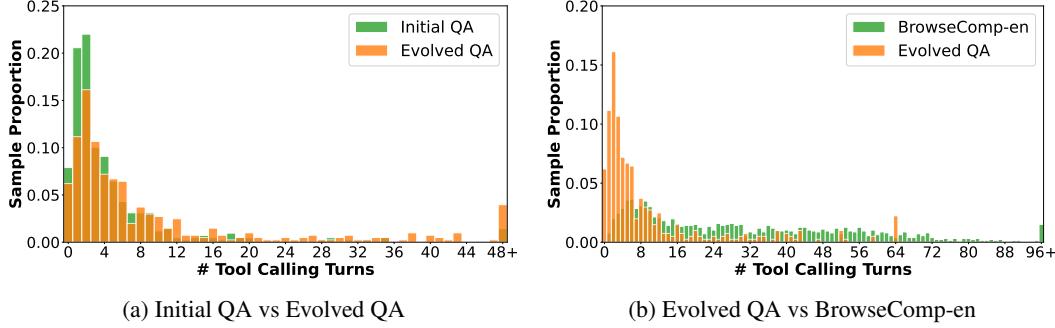


Figure 3: Tool calling turns distribution comparisons using OpenAI-o3: Initial QA vs Evolved QA (left) and Evolved QA vs BrowseComp-en (right).

Formally, starting with the initial query Q_0 extracted from the previous information space $H^{(0)}$, we iteratively evolve queries through multiple rounds. For $k = 0, \dots, K - 1$, the $(k+1)$ -th information space $H^{(k+1)}$ is obtained by appending the k -th round's augmentation to the entire history:

$$H^{(k+1)} = (H^k, \tau_1^{(k)}, \alpha_1^{(k)}, o_1^{(k)}, \dots, \tau_{m_k}^{(k)}), \quad k = 0, \dots, K - 1 \quad (3)$$

where each evolved query Q_{k+1} is contained in $\tau_{m_k}^{(k)}$ and maintains the same answer A . The sequence $(\alpha_1^{(k)}, o_1^{(k)}, \dots, \alpha_{m_k-1}^{(k)}, o_{m_k-1}^{(k)})$ represents the multiple action-observation cycles conducted during the k -th evolution step to further augment the information space.

2.4 THE WEBEXPLORER-QA DATASET

To ensure diversity in our synthesized QA pairs, we collect seed entities from Wikipedia and incorporate three QA pairs from BrowseComp-en as exemplars in our synthesis prompt. After generating initial QA pairs using our model-based exploration, we apply our iterative query evolution methodology for 5 iterations to synthesize the WEBEXPLORER-QA dataset, with detailed prompts description for the synthesis shown in Appendix B. **We use the Claude-4-Sonnet model (Anthropic, 2025) for both the exploration and evolution stages of data synthesis, with detailed tool calling statistics shown in Appendix E.4.** These evolved final QA pairs, approximately 40K in total, are subsequently used for supervised fine-tuning and can be used directly for reinforcement learning. **Additional data contamination experiments, reported in Appendix E.1, address potential concerns regarding data contamination.**

To validate the quality of WEBEXPLORER-QA and demonstrate the effectiveness of our iterative evolution process, we conduct a comprehensive comparative analysis across multiple web navigation datasets. Our evaluation encompasses our initial QA pairs, the final evolved QA pairs, and established datasets from prior work, including WebDancer (Wu et al., 2025a), SailorFog (Li et al., 2025a), WebShaper (Tao et al., 2025), and ASearcher (Gao et al., 2025). Using Claude-4-Sonnet as our evaluation model, we assess both accuracy and the average number of tool calling turns required to solve each QA pair. Additionally, we employ a more powerful model, OpenAI-o3, to analyze the tool calling turns distribution between our Initial QA and Evolved QA, as well as between Evolved QA and the challenging benchmark BrowseComp-en. **Furthermore, we conduct uniqueness and correctness validation to ensure data quality.** Both validation studies show that over 95% of our data maintains answer uniqueness and factual correctness, providing strong assurance of quality. Details are provided in Appendix E.2 and Appendix E.3.

324 The results in Table 1 demonstrate the critical importance of our iterative evolution process. Evolution
 325 significantly increases the complexity of the queries, with accuracy dropping from 86.6% to
 326 67.1% and average solution turns increasing from 7.9 to 9.9, indicating successful creation of
 327 complex multi-step reasoning tasks. Furthermore, our evolved WEBEXPLORER-QA achieves the highest
 328 average turn count compared to existing datasets, demonstrating superior complexity. Figure 3 (left)
 329 also shows that easy QA pairs solvable within 4 turns decrease significantly after evolution. While
 330 Figure 3 (right) shows a gap remains between our Evolved QA and BrowseComp-en in tool calling
 331 turns, BrowseComp-en presents excessive difficulty with below 20% accuracy on most open-source
 332 models. Therefore, completely mirroring BrowseComp-en’s difficulty level is unnecessary for training
 333 current open-source models.

335 3 COLD START AND REINFORCEMENT LEARNING

337 Our training methodology adopts the established post-training two-phase paradigm: supervised fine-
 338 tuning for cold start initialization, followed by reinforcement learning for advanced capability de-
 339 velopment (Guo et al., 2025; Zhipu AI, 2025). The initial supervised fine-tuning phase enables
 340 models to acquire proper invocation of search and browse functions while developing foundational
 341 long-horizon search capabilities. Subsequently, reinforcement learning further enhances reasoning
 342 abilities, extending model performance to longer contexts and increased maximum turn limits, ulti-
 343 mately achieving more advanced long-horizon problem-solving behaviors.

345 3.1 SUPERVISED FINE-TUNING FOR COLD START

347 After synthesizing the challenging QA pairs as described in §2.4, we leverage the commercial
 348 model to collect high-quality trajectories for supervised fine-tuning. We employ rejection sam-
 349 pling during data collection, ensuring that our fine-tuning process exclusively utilizes correct tra-
 350 jectories. We adopt the ReAct framework (Yao et al., 2023) as our foundational format, incor-
 351 porating search and browse as the two primary tools for actions α , enclosed by `<tool_call>`
 352 and `</tool_call>` tags. The framework includes explicit reasoning thoughts τ denoted by
 353 `<think>` tags, enabling transparent cognitive processes, and tool responses as observations o
 354 marked by `<tool_response>` tags. Our collected trajectories encompass multiple rounds of
 355 action α , thought τ , and observation o sequences, with an example demonstrated in Appendix A.

357 3.2 REINFORCEMENT LEARNING

359 Following the cold-start phase, which endows the model with fundamental search and browse action
 360 capabilities along with long-horizon reasoning ability, we conduct further reinforcement learning
 361 training to enhance reasoning performance and optimize decision-making strategies using GRPO
 362 algorithm (Shao et al., 2024). Notably, in the RL phase, we can directly utilize the synthesized QA
 363 pairs without requiring solving trajectories. For reward design, we implement a composite reward
 364 function that balances structural correctness with answer accuracy:

$$365 \quad R = 0.2 \cdot R_{\text{format}} + R_{\text{correct}} \quad (4)$$

368 Format rewards R_{format} evaluate the correctness of response formatting, primarily assessing whether
 369 tool calls and thought structures adhere to the specified format requirements. For accuracy rewards
 370 R_{correct} , we leverage the DeepSeek-V3 model (Liu et al., 2024) as an automated judge to evaluate
 371 whether the final responses are correct given the ground truth answers. This automated evaluation
 372 approach enables scalable assessment while maintaining high reliability.

373 To accommodate the model’s expanding reasoning capabilities during training, we implement a
 374 progressive context length expansion strategy. We begin with a maximum length of 64K tokens and a
 375 tool calling turn limit of 50. As the model generates increasingly complex trajectories, we gradually
 376 increase the maximum length to 96K tokens with a 75-turn limit, and finally to 128K tokens with
 377 a 100-turn limit. This progressive expansion allows the model to develop more sophisticated long-
 378 horizon reasoning patterns throughout the training process.

Model	BC-en	BC-zh	GAIA	WebWalkerQA	FRAMES	Xbench-DS	HLE
OpenAI-o3 [†]	50.9	58.1	70.5 [†]	71.7	84.0	66.7	20.2
Claude-4-Sonnet [†]	12.2	29.1	68.3 [†]	61.7	80.7	64.6	20.3
GLM-4.5	26.4	37.5	66.0 [†]	65.6 [†]	78.9 [†]	70.0 [†]	21.2 [†]
DeepSeek-V3.1	30.0	49.2	63.1 [†]	61.2 [†]	83.7	71.2	29.8
Kimi-K2 [†]	14.1	28.8	57.7	63.0	72.0	50.0	18.1
WebShaper-72B	-	-	60.0	52.2	-	-	-
WebShaper-32B (QwQ)	-	-	53.3	49.7	-	-	-
WebShaper-32B	-	-	52.4	51.4	-	-	-
WebSailor-72B	12.0	30.1	55.4	-	-	55.0	-
WebSailor-32B	10.5	25.5	53.2	-	-	53.3	-
WebSailor-7B	6.7	14.2	33.0	-	-	34.3	-
ASearcher-Web-QwQ	5.2	15.6	52.8	34.3	70.9	42.1	12.5
WebThinker-32B	2.8	-	48.5	46.5	-	-	15.8
MiroThinker-32B-DPO-v0.1	13.0	17.0	57.3	49.3	71.7	-	11.8
MiroThinker-8B-DPO-v0.1	8.7	13.6	46.6	45.7	64.4	-	-
WEBEXPLORER-8B (SFT)	7.9	21.3	43.7	59.8	72.6	47.5	16.0
WEBEXPLORER-8B (RL)	15.7	32.0	50.0	62.7	75.7	53.7	17.3

Table 2: Accuracy (%) of deep research agents on information-seeking benchmarks. BC-en and BC-zh denote BrowseComp-en and BrowseComp-zh respectively. XBench-DS refers to XBench-DeepSearch. **Bold** indicates the best performance among open-source models $< 100B$, while underlined values represent the best performance among models $< 10B$ parameters. All scores of WEBEXPLORER-8B are computed as Avg@4 using LLM-as-Judge. Entries marked with a dagger ([†]) were reproduced by us under our scaffold: on model name = entire row; on a number = that entry only.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUPS

Benchmarks We evaluate on several information-seeking benchmarks including BrowseComp-en (Wei et al., 2025) and BrowseComp-zh (Zhou et al., 2025), GAIA (Mialon et al., 2024) (103-sample text-only validation subset), WebWalkerQA (Wu et al., 2025b), FRAMES (Krishna et al., 2024), and XBench-DeepSearch (Xbench-Team, 2025). We also evaluate on the frontier academic benchmark HLE (Phan et al., 2025) to verify generalization capabilities beyond Wikipedia-alike knowledge QA. We report Avg@4 scores using LLM-as-Judge evaluation with DeepSeek-V3 (Liu et al., 2024) following previous work (Li et al., 2025a; Tao et al., 2025).

Models We compare our approach against both proprietary and open-source agents. For proprietary models, we primarily benchmark against OpenAI-o3 and Claude-4-Sonnet (Anthropic, 2025). Among open-source agents, we compare against advanced models including, GLM-4.5 (Zhipu AI, 2025), DeepSeek-V3.1 (Liu et al., 2024), Kimi-K2 (Team et al., 2025), WebShaper (Tao et al., 2025), WebSailor (Li et al., 2025a), ASearcher (Gao et al., 2025), WebThinker (Li et al., 2025b) and MiroThinker (MiroMind Team, 2025).

Scaffold Details Following previous work (Li et al., 2025a; Tao et al., 2025), our agent scaffold uses two tools: `search` returns top-10 Google results and `browse` retrieves URL content via Jina (Jina.ai, 2025) and answers queries using Gemini 2.5 Flash (Comanici et al., 2025). Further details about these two tools can be found in Appendix D. Using this unified scaffold, we evaluate OpenAI-o3, Claude-4-Sonnet, and Kimi-K2 on all benchmarks, and evaluate GLM-4.5 and DeepSeek-V3.1 on a subset (marked with [†] where the original reports lack results), ensuring a fair cross-model comparison.

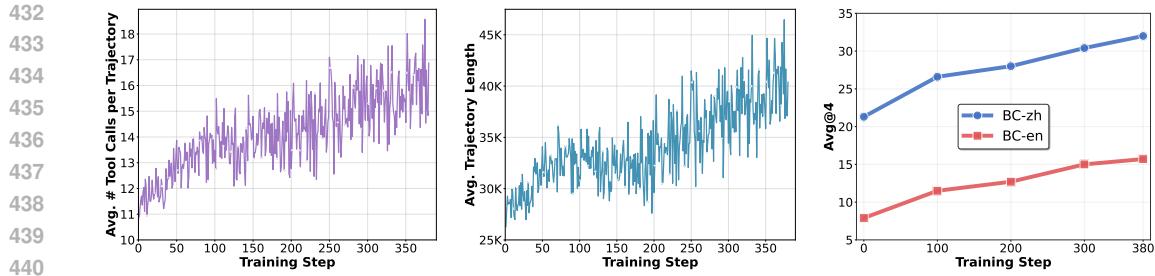


Figure 4: Training progress during RL. **Left:** Average # tool calls per trajectory **Middle:** Average trajectory length (# tokens) **Right:** Avg@4 scores on BrowseComp-en and BrowseComp-zh.

Training Details We use the Qwen3-8B model (Yang et al., 2025) to conduct SFT and RL training, resulting in the WEBEXPLORER-8B model. For supervised fine-tuning, we utilize approximately 13K training samples with a training batch size of 32 and a learning rate of 1e-5 for 4 epochs. For reinforcement learning, we train on approximately 12K samples using the GRPO algorithm, where each group consists of 8 rollouts with a batch size of 64 and a learning rate of 1e-6. **To further verify the critical contribution of the evolution step in our data synthesis framework, we conduct ablation experiments by training models on QA data from the exploration stage only (without the evolution step), with the detailed results presented in Appendix F.1.**

4.2 EXPERIMENTAL RESULTS

WEBEXPLORER-8B Establishes State-of-the-Art Performance at Its Scale As shown in Table 2, our WEBEXPLORER-8B demonstrates exceptional performance across diverse information-seeking benchmarks. Following supervised fine-tuning, WEBEXPLORER-8B (SFT) already exhibits competitive results, surpassing WebSailor-7B with scores of 7.9% on BrowseComp-en, 72.6% on FRAMES, and 47.5% on XBench-DS. The subsequent reinforcement learning phase yields substantial improvements, with our final model achieving 15.7% on BrowseComp-en and 32.0% on BrowseComp-zh. Notably, WEBEXPLORER-8B (RL) surpasses the previous best sub-10B model, MiroThinker-8B-DPO-v0.1, across all benchmarks, establishing new state-of-the-art performance at its parameter scale.

WEBEXPLORER-8B Outperforms Much Larger Models Despite having only 8B parameters, WEBEXPLORER-8B consistently outperforms much larger open-source models, demonstrating remarkable parameter efficiency. On BrowseComp-en, BrowseComp-zh, FRAMES, and WebWalkerQA, our model achieves the best performance among open-source sub-100B models. Specifically, with scores of 15.7% on BrowseComp-en, 32.0% on BrowseComp-zh, and 75.7% on FRAMES, our model surpasses these significantly larger models including WebSailor-72B and even larger model Kimi-K2. Similarly, on WebWalkerQA, we achieve 62.7%, outperforming WebShaper-72B by 10 absolute points. Our model also demonstrates competitive performance on additional benchmarks, achieving 50.0% on GAIA and 53.7% on XBench-DS, rivaling models with more parameters like WebThinker-32B and WebSailor-32B.

Strong Generalization from BrowseComp-Inspired Training Data Notably, while our QA pair synthesis methodology, particularly the evolution stage, is inspired by BrowseComp-en, and the exemplars provided in synthesis prompts are both derived from BrowseComp-en, we ensure diversity by collecting seed entities from Wikipedia across different domains. The resulting model also generalizes effectively across diverse benchmarks and domains beyond the BrowseComp-en/zh. As demonstrated previously, WEBEXPLORER-8B achieves strong performance across all information-seeking benchmarks, including GAIA, WebWalkerQA, FRAMES, and XBench-DS. This cross-benchmark success indicates substantial improvement in general information-seeking capabilities, demonstrating the generalization benefits of training on our challenging, complex queries. More remarkably, despite our training data not being STEM-focused, our model extends beyond information-seeking tasks. It achieves 17.3% on the HLE benchmark, which encompasses questions from diverse academic disciplines. This performance surpasses previous 32B models, like WebThinker-32B, further validating the robustness and transferability of our approach.

486 **RL Training Dynamics and Performance Analysis** To analyze the training dynamics, we
 487 recorded the average number of tool calls per trajectory and the average response length (number of
 488 tokens) during RL training, with results presented in Figure 4. Throughout the RL training process,
 489 the average number of tool calls increases steadily from approximately 11 to over 16, significantly
 490 exceeding the typical range of fewer than 10 tool calls observed in previous work (Gao et al., 2025).
 491 This substantial increase suggests that our model learns to execute more sophisticated multi-step rea-
 492 soning strategies. Concurrently, the average trajectory length grows during the RL process, scaling
 493 to over 40K tokens, as the search tool responses constitute the majority source of trajectory length.
 494 Importantly, we observe that performance on both BrowseComp-en and BrowseComp-zh improves
 495 consistently throughout this process, with BrowseComp-en increasing from 7.9% to 15.7%, demon-
 496 strating a strong correlation between increased tool usage complexity and task performance. Similar
 497 to the emergence of long chain-of-thought phenomena observed in previous single-turn RL studies
 498 on mathematical or logical tasks (Guo et al., 2025; Zeng et al., 2025; Yu et al., 2025; Liu et al.,
 499 2025), these findings validate the effectiveness of our RL training approach in developing advanced
 500 deep research agents that can tackle complex information-seeking tasks through extended reasoning
 501 chains.

5 RELATED WORK

504 Web agent research has evolved into two largely complementary directions: (1) interactive web
 505 agents that manipulate websites through low-level UI actions within real browser environments, and
 506 (2) deep research agents that perform long-horizon, multi-turn information seeking through search
 507 and browse APIs.

5.1 INTERACTIVE WEB AGENTS

510 Interactive web agents execute multi-step actions and task planning within dynamic web environ-
 511 ments through direct manipulation of browser interfaces. These systems support a rich action space
 512 including clicking, typing, form filling, scrolling, and cross-page navigation. Representative bench-
 513 marks include MiniWoB (Liu et al., 2018), WebShop (Yao et al., 2022), Mind2Web (Deng et al.,
 514 2023), WebArena (Zhou et al., 2023), WebVoyager (He et al., 2024), and VisualWebArena (Koh
 515 et al., 2024). Recent advances include self-evolving online reinforcement learning curricula (Qi
 516 et al., 2025) and human-inspired exploration techniques such as BrowserAgent (Zhang et al., 2025).
 517 These agents primarily target structured, goal-oriented interaction scenarios with clear success crite-
 518 ria, distinguishing them from open-ended information-seeking tasks that emphasize iterative search
 519 and deep reasoning over retrieved content.

5.2 DEEP RESEARCH AGENTS

522 Deep research agents, also referred to as information-seeking agents or search agents, specialize
 523 in multi-hop reasoning and long-horizon exploration to answer complex queries that require gath-
 524 ering information from multiple sources. This direction has emerged as a distinct and important
 525 area within web agent research. Early search-augmented systems such as Search-R1 (Jin et al.,
 526 2025) and ZeroSearch (Sun et al., 2025) primarily tackled relatively straightforward QA datasets
 527 including Natural Questions (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018). Recent
 528 advances (Wu et al., 2025a; Li et al., 2025a; Tao et al., 2025; Gao et al., 2025) have shifted toward
 529 synthesizing more challenging training data and developing sophisticated deep research models tar-
 530 geting demanding benchmarks like BrowseComp (Wei et al., 2025). Unlike interactive web agents,
 531 these systems typically operate through high-level search and browse APIs, emphasizing reasoning
 532 depth and information integration over direct web manipulation.

6 CONCLUSION

536 We present WEBEXPLORER, a simple framework for synthesizing high-quality information-seeking
 537 query-answer data for training deep research agents. Leveraging our data through SFT and RL,
 538 our WEBEXPLORER-8B model achieves state-of-the-art performance across multiple information-
 539 seeking benchmarks, even outperforming larger models like WebSailor-72B on BrowseComp-en/zh.
 This success highlights a practical approach for building strong long-horizon **deep research** agents.

540

REPRODUCIBILITY STATEMENT

541

542 For our scaffold, we provide details about the used tools in Appendix D. For data synthesis, our
 543 WEBEXPLORER-QA framework is entirely prompting-based and designed for easy reproduction.
 544 We provide detailed prompting strategies and implementation guidelines in Appendix B. For the
 545 model, we use the widely adopted Qwen3-8B (Yang et al., 2025) and employ the GRPO (Shao
 546 et al., 2024) algorithm for reinforcement learning, ensuring our experimental setup can be readily
 547 replicated using standard tools and methodologies.

548

549

REFERENCES

550

551 Anthropic. Introducing claude 4. Technical report, 2025. URL <https://www.anthropic.com/news/claude-4>.

552

553 Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green,
 554 Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifmoghaddam, Yanxi Li, Haoran Hong,
 555 Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhua Chen, and Jimmy Lin.
 556 Browsecamp-plus: A more fair and transparent evaluation benchmark of deep-research agent.
 557 *arXiv preprint arXiv:2508.06600*, 2025.

558

559 Yuxing Cheng, Yi Chang, and Yuan Wu. A survey on data contamination for large language models.
 560 *arXiv preprint arXiv:2502.14425*, 2025.

561

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

566

567 Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su.
 568 Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing
 569 Systems*, 36:28091–28114, 2023.

570

571 Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu.
 572 Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl, 2025.
 573 URL <https://arxiv.org/abs/2508.07976>.

574

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

578

579 Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan,
 580 and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models.
 581 *arXiv preprint arXiv:2401.13919*, 2024.

582

583 Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and
 584 Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement
 585 learning. *arXiv preprint arXiv:2503.09516*, 2025.

586

587 Jina.ai. Jina, 2025. URL <https://jina.ai>.

588

589 Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig,
 590 Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal
 591 agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association
 592 for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

593

594 Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler,
 595 Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-
 596 augmented generation. In *North American Chapter of the Association for Computational Lin-
 597 guistics*, 2024. URL <https://api.semanticscholar.org/CorpusID:272753636>.

598

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

594 Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. Platypus: Quick, cheap, and powerful refinement of
 595 llms. *arXiv preprint arXiv:2308.07317*, 2023.
 596

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

600 Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and
 601 Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability.
 602 *arXiv preprint arXiv:2504.21776*, 2025b.
 603

604 Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao,
 605 Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint*
 606 *arXiv:2412.19437*, 2024.

607 Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement
 608 learning on web interfaces using workflow-guided exploration. In *International Conference on*
 609 *Learning Representations (ICLR)*, 2018. URL <https://arxiv.org/abs/1802.08802>.
 610

611 Junteng Liu, Yuanxiang Fan, Zhuo Jiang, Han Ding, Yongyi Hu, Chi Zhang, Yiqi Shi, Shitong
 612 Weng, Aili Chen, Shiqi Chen, et al. Synlogic: Synthesizing verifiable reasoning data at scale for
 613 learning logical reasoning and beyond. *arXiv preprint arXiv:2505.19641*, 2025.

614 Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA:
 615 a benchmark for general AI assistants. In *The Twelfth International Conference on Learning*
 616 *Representations*, 2024. URL <https://openreview.net/forum?id=fibxvahvs3>.
 617

618 MiroMind Team. Miromind open deep research, 2025. URL <https://miromind.ai/blog/miromind-open-deep-research>. Blog post, August 2025.
 619

620 OpenAI. Introducing deep research. Technical report, 2025. URL <https://openai.com/index/introducing-deep-research/>.
 621

622 Vardaan Pahuja, Yadong Lu, Corby Rosset, Boyu Gou, Arindam Mitra, Spencer Whitehead,
 623 Yu Su, and Ahmed Hassan Awadallah. Explorer: Scaling exploration-driven web trajectory
 624 synthesis for multimodal web agents. In *Findings of the Association for Computational Lin-
 625 guistics: ACL 2025*, pp. 6300–6323, Vienna, Austria, July 2025. Association for Compu-
 626 tational Linguistics. ISBN 979-8-89176-256-5. URL <https://aclanthology.org/2025.findings-acl.326/>.
 627

628 Perplexity AI. Introducing perplexity deep research. <https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research>, February 2025. Accessed: 2025-
 629 07-25.

630 Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin
 631 Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity’s last exam. *arXiv preprint*
 632 *arXiv:2501.14249*, 2025.

633 Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang,
 634 Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. WebRL: Training LLM web agents via
 635 self-evolving online curriculum reinforcement learning. In *The Thirteenth International Confer-
 636 ence on Learning Representations*, 2025. URL <https://openreview.net/forum?id=0VKEAFjEqv>.
 637

638 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
 639 Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathemati-
 640 cal reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
 641

642 Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krim-
 643 mel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, et al. jina-embeddings-
 644 v3: Multilingual embeddings with task lora. *arXiv preprint arXiv:2409.10173*, 2024.

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

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

654 Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen,
 655 Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv
 656 preprint arXiv:2507.20534*, 2025.

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

660 Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang,
 661 Zekun Xi, Gang Fu, Yong Jiang, et al. Webdancer: Towards autonomous information seeking
 662 agency. *arXiv preprint arXiv:2505.22648*, 2025a.

663 Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Deyu Zhou,
 664 Pengjun Xie, and Fei Huang. Webwalker: Benchmarking llms in web traversal. *ArXiv,
 665 abs/2501.07572*, 2025b. URL <https://api.semanticscholar.org/CorpusID:275471576>.

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

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

670 Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov,
 671 and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question
 672 answering. In *Proceedings of the 2018 conference on empirical methods in natural language
 673 processing*, pp. 2369–2380, 2018.

674 Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable
 675 real-world web interaction with grounded language agents. *Advances in Neural Information Pro-
 676 cessing Systems*, 35:20744–20757, 2022.

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

680 Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian
 681 Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system
 682 at scale. *arXiv preprint arXiv:2503.14476*, 2025.

683 Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-
 684 zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv
 685 preprint arXiv:2503.18892*, 2025.

686 Zhengbo Zhang, Zhiheng Lyu, Junhao Gong, Hongzhu Yi, Xinming Wang, Yuxuan Zhou, Jiabing
 687 Yang, Ping Nie, Yan Huang, and Wenhui Chen. Browseragent: Building web agents with human-
 688 inspired web browsing actions. *arXiv preprint arXiv:2510.10666*, 2025.

689 Zhipu AI. GLM-4.5: Reasoning, Coding, and Agentic Abilities. <https://z.ai/blog/glm-4.5>, July 2025. Accessed: 2025-08-07.

690 Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling
 691 Jin, Chenxuan Xie, Meng Cao, et al. Browsecamp-zh: Benchmarking web browsing ability of
 692 large language models in chinese. *arXiv preprint arXiv:2504.19314*, 2025.

693 Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng,
 694 Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for build-
 695 ing autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

702 A TRAJECTORY EXAMPLE
703
704
705
706707 To illustrate how our agent framework operates in practice, we provide a detailed trajectory example
708 from the WEBEXPLORER supervised fine-tuning data. This example demonstrates the agent's multi-
709 step reasoning process, thought, tool call, and tool response when handling complex queries.
710
711
712
713
714715 **Question:** In a county adjacent to the nation's capital, a small press gaming company co-
716 founded by someone whose previous pulp adventure game had won a silver-level award in a
717 category focusing on game mechanics excellence. This person reached a specific age mile-
718 stone during an online funding campaign spanning consecutive years that occurred exactly
719 6 years after their game's award. What was the exact final funding amount raised by this
720 campaign, expressed in US dollars without the dollar sign or comma separators?
721722 STEP 1
723724 Thought
725726 <think>
727 This is a complex multi-part query that I need to break down systematically. Let me identify the
728 key components:
729 1. A county adjacent to the nation's capital (so near Washington DC)730 ...
731 Let me start by searching for information about gaming companies near Washington DC that had
732 crowdfunding campaigns, and awards for pulp adventure games.
</think>733 I need to work through this step-by-step to identify the specific gaming company, award, and
734 crowdfunding campaign. Let me start by searching for information about gaming companies near
735 Washington DC and awards for pulp adventure games.736 Action
737739 <tool_call>
740 {“name”: “search”, “arguments”: {“queries”: [“small press gaming company Washington DC
741 area”, “pulp adventure game silver award game mechanics ENnie”, “gaming company county
742 adjacent Washington DC crowdfunding”]}
743 </tool_call>744 Observation
745746 <tool_response>
747 — search result for [small press gaming company Washington DC area] —
748 <title>op Washington, DC Gaming Companies (10) - Built In </title>
749 <url><https://builtin.com/companies/location/washington-dc/type/gaming-companies> </url>
750 <snippet>Top Washington, DC Gaming Companies (10) · Edge Systems Technologies · Some-
751 thing Wicked Games · Polygon · Implicit Solution · BetMGM · Snaapiq · Microcoaching. </s-
752 nippet>
753 ...
754 </tool_response>
755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

STEP 2

Thought

<think>

From the search results, I can see some relevant information:

1. I found mention of “Spirit of the Century” which won a Silver ENnie for Best Rules in 2007, and it’s described as “A really well put together game of pulp adventure”

...

</think>

Let me search more specifically for information about Spirit of the Century, which appears to be a pulp adventure game that won a silver ENnie, and its publisher.

Action

<tool_call>

{“name”: “search”, “arguments”: {“queries”: [“Spirit of the Century silver ENnie 2007 Evil Hat Productions”, “Evil Hat Productions Maryland Virginia crowdfunding Kickstarter”, “Fred Hicks Evil Hat Productions age milestone funding campaign”]}}

</tool_call>

Observation

<tool_response>

— search result for [Spirit of the Century silver ENnie 2007 Evil Hat Productions] —

<title>Spirit of the Century - Wikipedia </title>

<url>https://en.wikipedia.org/wiki/Spirit_of_the_Century </url>

<snippet>Spirit of the Century is a pulp role-playing game published by Evil Hat Productions, and based on Evil Hat’s FATE system. ... 2007 Silver ENnie Award for Best ... </snippet>

...

</tool_response>

STEP 3

Thought

<think>

Great! Now I have some key information:

1. Spirit of the Century is a pulp adventure game by Evil Hat Productions

...

</think>

Perfect! I found that Spirit of the Century by Evil Hat Productions won a Silver ENnie for Best Rules in 2007, and Evil Hat is based in Silver Spring, Maryland (Montgomery County, adjacent to DC). Now let me search for their crowdfunding campaigns around 2013 (6 years after 2007) and information about Fred Hicks’ age milestones.

Action

<tool_call>

{“name”: “search”, “arguments”: {“queries”: [“Evil Hat Productions Kickstarter 2013 2014 Fred Hicks”, “Fate Core Kickstarter 2013 funding amount final total”, “Fred Hicks 40th birthday milestone age Fate Core campaign”]}}

</tool_call>

810
811
812
813
814
815
816
817
818
819
820
821
822
823

Observation

```
<tool_response>
— search result for [Evil Hat Productions Kickstarter 2013 2014 Fred Hicks] —
<title>Race to Adventure! A Spirit of the Century™ Board Game - Kickstarter </title>
<url>https://www.kickstarter.com/projects/evilhat/race-to-adventure-a-spirit-of-the-centurytm-
board/posts?ref=project_share </url>
<snippet>Evil Hat Productions is raising funds for Race to Adventure! A Spirit of the Century™
Board Game on Kickstarter! Race to Adventure!
</snippet>
...
</tool_response>
```

824
825
826
827
828
829
830
831
832
833
834
835
836
837
838

STEP 4

Thought

```
<think>
I found some crucial information! The Fate Core Kickstarter by Evil Hat Productions raised
exactly $433,365 and ended on January 30, 2013. This is 6 years after the 2007 ENnie award for
Spirit of the Century.
...
</think>
Excellent! I found that the Fate Core Kickstarter raised $433,365 and ended on January 30, 2013 -
which is exactly 6 years after the 2007 ENnie award. Let me get more details about this campaign
to confirm it spans consecutive years and involves an age milestone.
```

839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856

Action

```
<tool_call>
{"name": "browse", "arguments": {
"url": "https://www.kickstarter.com/projects/evilhat/fate-core",
"query": "When did this campaign start and end? Did it span 2012 and 2013? Any mention of
Fred Hicks birthday or age milestone during the campaign?"}}
```

857
858
859
860
861
862
863

Observation

```
<tool_response>
The campaign started on **December 3, 2012**, and ended on **January 29, 2013**, spanning
**56 days** and crossing over from **2012 to 2013**.
There is **no mention of Fred Hicks' birthday or age milestone** in the provided content.
</tool_response>
```

STEP 5

864
865
866
867
868
869
870
871
872
873
874

Thought

```
<think>
Perfect! This confirms that the Fate Core Kickstarter:
...
</think>
Perfect! The Fate Core campaign spanned consecutive years (December 3, 2012 to January 29, 2013) and occurred exactly 6 years after the 2007 ENnie. Now let me search for information about Fred Hicks' birthday and age milestones during this period.
```

875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904

Action

```
<tool_call>
{"name": "search", "arguments": {"queries": ["Fred Hicks birthday December January 40 years old", "Fred Hicks born 1972 1973 Evil Hat", "Fred Hicks 40th birthday December 2012 January 2013"]}}
</tool_call>
```

Observation

```
<tool_response>
— search result for [Fred Hicks birthday December January 40 years old] —
<title>Happy birthday to the first general manager of Forsythe Associates ... </title>
<url>https://www.facebook.com/groups/649530655242462/posts/1496564073872445 </url>
<snippet>Happy birthday to the first general manager of Forsythe Associates [1962-1971], Fred Hicks [1920-1971]. </snippet>
...
</tool_response>
```

STEP 6

Thought

```
<think>
Perfect! I found the crucial information. ...
</think>
Therefore, the exact final funding amount raised by this campaign, expressed in US dollars without the dollar sign or comma separators, is: **433365**
```

B DETAILS OF WEBEXPLORER-QA SYNTHESIS FRAMEWORK

Compared with previous works (Li et al., 2025a; Tao et al., 2025) involving complex synthesis processes, our approach presents a simpler prompting-based framework. Here, we provide details about the synthesis methodology to facilitate reproducibility.

Our framework operates through two stages, each guided by carefully designed prompting strategies that leverage large language models for autonomous information gathering and query evolution.

B.1 MODEL-BASED EXPLORATION

We first collect a large number of entities from Wikipedia. In the first stage, we provide a seed entity as the search entry point along with three exemplar QA pairs from BrowseComp-en to demonstrate the desired question characteristics. The prompting strategy encourages the model to conduct search

918 and browsing activities starting from the seed entity, then synthesize a challenging query-answer pair
 919 using the collected knowledge.
 920

921 The key instruction emphasizes creating challenging queries with subtle and obscured clues. We
 922 explicitly prompt the model to ensure that while the question should be challenging, the answer
 923 must remain unique and verifiable through the information space it constructs. This stage results in
 924 initial QA pairs that incorporate multi-website reasoning.

925 **Model-Based Exploration Prompt**

927 You need to create a challenging question for deep search based on
 928 real information.

929 You should start by collecting information from the internet, then
 930 select a truth, and create a question where the truth needs to be
 931 discovered through search.

932 You will start with a random "seed", then search and browse for
 933 whatever you want on the Internet, and create the question and
 934 truth from the information you gather.

935 You should provide several subtle and blurred clues to make the
 936 question challenging, while ensuring the truth is unique.

937 There are some examples:
 938 {examples}

939 Let's start, with the seed of "{seed}".

940 You need to provide the following information in the final
 941 <answer></answer> tag:

942 <question>
 943 {{The challenging question you created based on real information.}}
 944 </question>
 945 <truth>
 946 {{The one and only exact truth to the question.}}
 947 </truth>

948 **IMPORTANT:** You must include the <question> and <truth> tags in your
 949 final response for the system to parse your answer correctly. Do
 950 not provide any other response format.

951
 952 **B.2 ITERATIVE QUERY EVOLUTION**
 953

954 We provide the full trajectory from the first stage as input to the evolution process. The second
 955 stage systematically transforms the initial QA pair into a more challenging variant through strate-
 956 gic information reduction and obfuscation. The prompting strategy provides the model with the
 957 original question-answer pair and explicit instructions for three primary evolution mechanisms: (1)
 958 removing redundant or overly explicit descriptions that provide multiple pathways to the answer,
 959 (2) systematically obfuscating specific details such as dates, locations, and proper names with
 960 vaguer descriptors, and (3) searching for alternative terminologies to replace explicit entity mentions.
 961 The evolution process can iterate up to five times, with each cycle potentially increasing reasoning
 962 complexity.

963 This prompting-based approach eliminates the need for complex graph construction heuristics or
 964 predefined evolution rules, instead leveraging the model's natural language understanding and web
 965 exploration capabilities. The framework's simplicity enables easy and quick reproducibility for
 966 generating high-quality challenging QA pairs.

967 **Iterative Query Evolution Prompt**

969 You need to make the following question more challenging while
 970 keeping the truth unique.

```

972
973     Original question: {question}
974     Original truth: {answer}
975
976     You should make the question more challenging in the following
977     ways:
978     1. Remove some descriptions, especially when there are multiple
979     descriptions that can lead to the truth answer in the question
980     2. Make one description more vague (such as date, location, name,
981     etc.) in the question while keeping the truth answer unique
982     3. Search for new descriptions or alternative terms to replace
983     specific entities in the question
984
985     You can use search and browse tools in this process. Make sure the
986     improved question is more challenging but the truth remains unique.
987
988     You can iteratively make the question more and more challenging
989     using these approaches up to 5 times. For each iteration, provide
990     the evolved question within <question> and </question> tags.
991     Provide the final improved question and truth pair in the final
992     <answer></answer> tag:
993     <question>
994     {{The improved, more challenging and complex question.}}
995     </question>
996     <truth>
997     {{The same exact truth.}}
998     </truth>
999
1000
1001
1002     C ILLUSTRATION OF MODEL-BASED EXPLORATION
1003
1004
1005     Here, we provide an example to illustrate how the model-based exploration works in Figure 5. Starting
1006     from the seed ‘‘Brazil National Team’’, the model iteratively conducts search and browse
1007     actions to discover various interconnected facts. By focusing on one specific event—the ‘‘1976 FA
1008     Cup’’—the model explores diverse information including details about players and referees, ultimately
1009     constructing a Query-Answer pair that requires deep reasoning across multiple discovered
1010     connections.
1011
1012
1013
1014     D TOOL SCHEMAS
1015
1016
1017     Our system utilizes two primary tools for web-based information retrieval: a search tool for discovering
1018     relevant content and a browse tool for detailed content analysis. The following sections detail
1019     the schemas for each tool.
1020
1021
1022     D.1 SEARCH TOOL
1023
1024
1025     The search tool enables web search functionality through Google search API, allowing the model to
1026     discover relevant URLs and content snippets based on input queries.

```

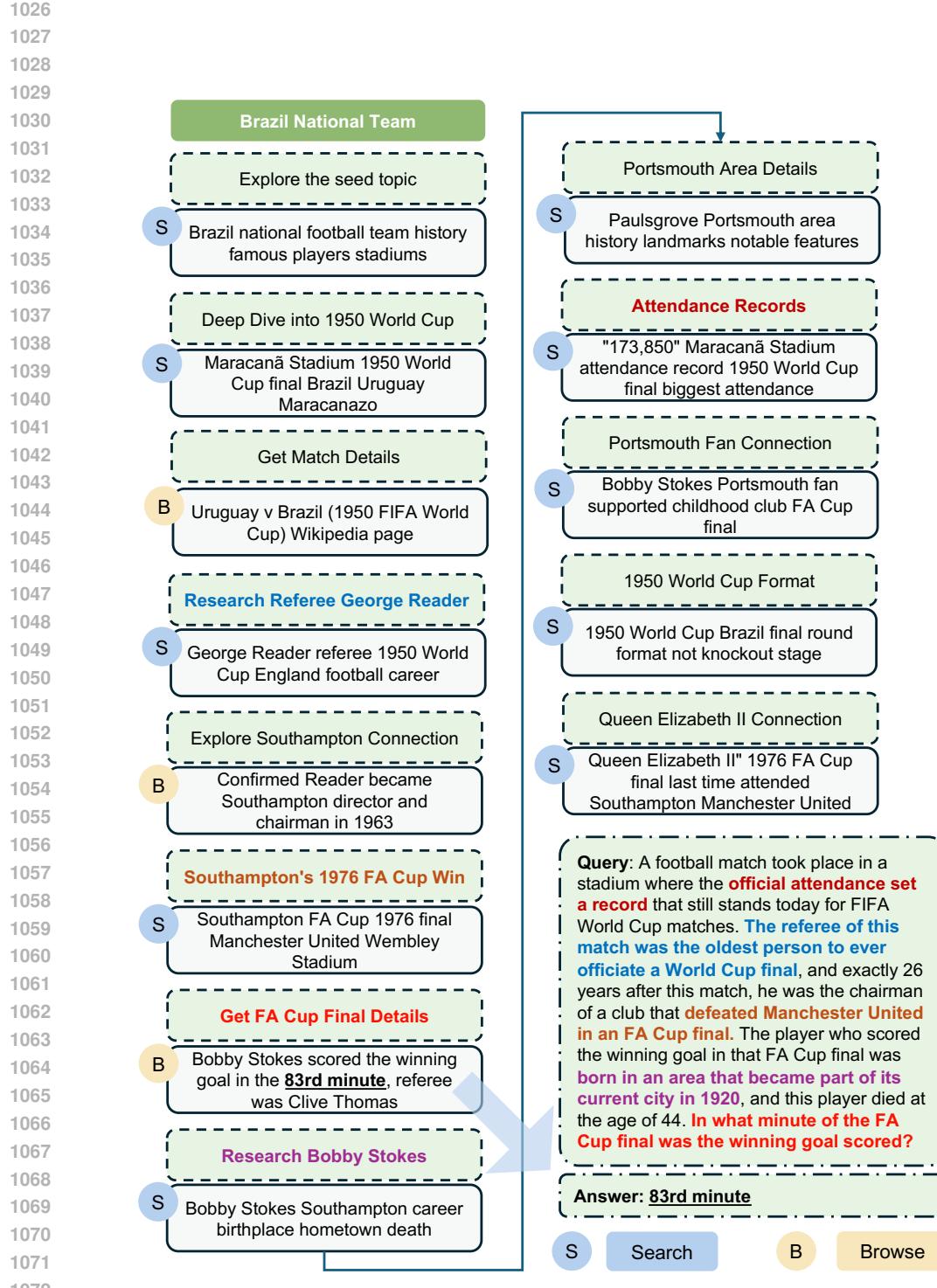


Figure 5: Illustration of model-based exploration and initial Query-Answer pair synthesis. Starting from the seed “Brazil National Team”, the model iteratively explores using **S** (Search) and **B** (Browse) actions to discover interconnected facts, then synthesizes a challenging query–answer pair that requires deep reasoning across multiple discovered connections.

```

1080
1081     Search Tool Schema
1082
1083     tool_schema:
1084         type: function
1085         function:
1086             name: search
1087             description: Web search.
1088             parameters:
1089                 type: object
1090                 properties:
1091                     queries:
1092                         type: array
1093                         description: The queries will be sent to Google. You will
1094                             get the brief search results with (title, url,
1095                             snippet)s for each query.
1096                         items:
1097                             type: string
1098                         required:
1099                             - queries

```

1100 The search tool accepts an array of query strings and returns search results containing titles, URLs,
 1101 and content snippets for each query. It is configured to return the top 10 results per query.

1104 D.2 BROWSE TOOL

1105
 1106 The browse tool provides detailed content analysis capabilities by fetching and processing content
 1107 from specific URLs using a combination of content extraction and language model analysis.

```

1109
1110     Browse Tool Schema
1111
1112     tool_schema:
1113         type: function
1114         function:
1115             name: browse
1116             description: Explore specific information in a url.
1117             parameters:
1118                 type: object
1119                 properties:
1120                     url:
1121                         type: string
1122                         description: The url will be browsed, and the content
1123                             will be sent to a Large Language Model (LLM)
1124                             as the based information to answer a query.
1125                     query:
1126                         type: string
1127                         description: The query to this url content. You will
1128                             get an answer by another LLM.
1129             required:
1130                 - url
1131                 - query

```

1132 The browse tool takes a URL and a specific query as input. It uses Jina (Jina.ai, 2025) for content
 1133 extraction and Gemini Flash (Comanici et al., 2025) as the generation engine to analyze the retrieved
 content and provide targeted answers to the input query.

1134
1135

E DATA ANALYSIS

1136
1137

E.1 CONTAMINATION ANALYSIS

1138
1139
1140
1141
1142
1143

To address potential concerns of data contamination from using three BrowseComp-en exemplars in our data synthesis, we conducted an embedding-based similarity analysis. We used Jina Embeddings v3 (1024-dimensional) Sturua et al. (2024) to compute embeddings for all training questions and BrowseComp validation questions. For each validation question, we calculated the cosine similarity with all training question embeddings and recorded the maximum similarity. The results are as follows:

1144
1145
1146
1147
1148

- Maximum cosine similarity: 0.738
- Mean similarity: 0.581
- 95th percentile: 0.668

1149
1150
1151
1152

These values fall below standard contamination thresholds (typically 0.80) used in the previous data contamination detection works Lee et al. (2023); Cheng et al. (2025), confirming negligible contamination risk.

1153
1154

E.2 ANSWER UNIQUENESS VALIDATION

1155
1156
1157
1158
1159

During the iterative query evolution process, we deliberately remove salient clues and obfuscate specifics (e.g., dates, names, locations) to increase task difficulty. A potential concern is whether this obfuscation might lead to queries with multiple valid answers. To address this, we conducted a systematic validation experiment.

1160
1161
1162
1163
1164

We randomly sampled 50 query-answer pairs from our synthesized WEBEXPLORER-QA dataset. For each question, we employed OpenAI-o3—a frontier model with strong deep-research capabilities—to independently solve the question 8 times, yielding a total of 400 solution trajectories. We then performed **manual verification** on all o3-proposed solutions that mismatched with our answers to determine whether they represented genuine alternative valid solutions.

1165
1166
1167
1168
1169
1170
1171

As shown in Table 3, out of 400 o3 solution trajectories, 42 trajectories were judged as mismatched by our automated judge, covering 13 distinct questions. Manual inspection revealed that 11 of these mismatched trajectories actually represented genuine alternative valid answers, corresponding to 4 distinct questions. This indicates that only 4 out of 50 questions (8%) exhibited multiple valid answers, confirming that 92% of evolved queries maintain answer uniqueness. This 8% rate is well within acceptable tolerance for large-scale training purposes, as minor label noise is known to have minimal impact on modern deep learning systems.

1172
1173

Table 3: Answer uniqueness validation results across 50 randomly sampled questions.

1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187

Validation Stage	Count	Percentage
Total sampled questions	50	100%
Total solution trajectories	400	—
<i>o3 Solving Results:</i>		
Solutions match our answers	358	89.5%
Solutions differ from our answers	42	10.5%
<i>Manual Verification (42 differing cases):</i>		
Trajectories with genuine alternative answers	11	26.2% (of 42)
Trajectories where o3 made errors	31	73.8% (of 42)
<i>Question-Level Uniqueness:</i>		
Distinct questions with mismatches	13	26.0% (of 50)
Questions with alternative valid answers	4	8.0% (of 50)
Questions maintaining answer uniqueness	46	92.0%

1188
1189

E.3 ANSWER CORRECTNESS VALIDATION

1190
1191
1192
1193
1194
1195
1196
1197

To ensure the quality of our synthesized data, we conducted factual correctness validation using OpenAI-o3 and manual verification. Our validation strategy leverages the asymmetric property of information-seeking data—hard to solve but easy to verify. We first use o3 to independently solve the questions; cases where o3’s solutions match our answers are considered validated, significantly reducing the need for manual verification. For the remaining cases where o3’s solutions differ from our answers, we perform manual verification to assess correctness. This approach is efficient because while these questions are challenging to solve, humans can easily verify answer correctness by directly searching for the provided answer.

1198
1199
1200
1201
1202
1203

We randomly sampled 400 QA pairs from our synthesized WEBEXPLORER-QA dataset. For each pair, we asked o3 to independently solve the question within our search framework without seeing our answer. We then compared o3’s solutions with our answers. For cases where o3’s solutions matched our answers, we consider our answers as validated correct. For cases where o3’s solutions differed from our answers, we conducted manual verification to determine the correctness of our answers.

1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218

Table 4: Answer correctness validation results on 400 randomly sampled QA pairs.

Validation Stage	Count	Percentage
Total sampled QA pairs	400	100%
<i>o3 Independent Solving:</i>		
Solutions match our answers	325	81.25%
Solutions differ from our answers	75	18.75%
<i>Manual Verification (75 differing cases):</i>		
Our answers incorrect	16	21.3% (of 75)
Our answers correct (o3 error)	59	78.7% (of 75)
Overall correct answers	384	96.0%
Overall incorrect answers	16	4.0%

As shown in Table 4, when o3 independently solved the 400 questions, its solutions matched our answers in 325 cases (81.25%), validating the correctness of those QA pairs. For the 75 cases where o3’s solutions differed from our answers, we conducted manual verification to assess the correctness of our answers. Among these 75 pairs, we found that 16 of our answers were indeed incorrect, while the remaining 59 were correct (o3 made solving errors in these cases). This yields an overall correctness rate of 96% ($(325 + 59) / 400$), with only 4% noise level well within acceptable tolerance for large-scale training purposes.

E.4 DATA GENERATION FRAMEWORK ANALYSIS

1226
1227
1228
1229
1230
1231

To provide deeper insights into our data synthesis pipeline, we analyze the tool calling patterns across the exploration and evolution stages. Table 5 presents the average number of tool calls broken down by tool type and stage.

1232
1233
1234
1235
1236
1237
1238
1239

Table 5: Average tool calling statistics across exploration and evolution stages.

Tool Type	Exploration	Evolution	Total
Search Tool Steps	7.11	7.33	14.45
Browse Tool Steps	3.04	0.15	3.19
Total Tool Calls	10.15	7.49	17.64

1240
1241

The exploration stage averages 10.15 tool calls per query, with a 7.11:3.04 ratio between search and browse operations, reflecting the information gathering process. The evolution stage requires 7.49 tool calls on average, predominantly search-based (7.33 search vs. 0.15 browse), as the model

verifies that evolved queries remain answerable while increasing difficulty. The overall synthesis process requires an average of 17.64 tool calls per QA pair.

Regarding the number of exploration steps, the model automatically determines when to stop searching. This adaptive termination ensures that exploration is neither prematurely truncated nor unnecessarily prolonged, allowing the synthesis process to naturally accommodate queries of varying complexity.

E.5 COMPARISON WITH SAILORFOG-QA

To provide a comprehensive comparison with other synthetic data generation approaches, we compare the tool-call distribution of our evolved WEBEXPLORER-QA with SailorFog-QA, another recent synthetic dataset for deep research agents. Figure 6 presents the tool-call distribution comparison between these two datasets.

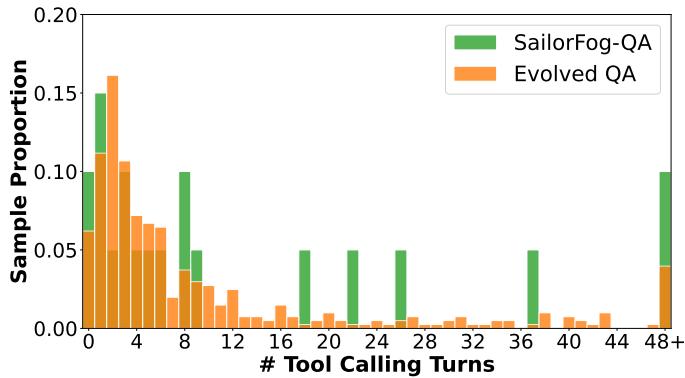


Figure 6: Tool-call distribution comparison between WEBEXPLORER-QA (evolved) and SailorFog-QA. Note that SailorFog-QA only contains 20 open-sourced samples, resulting in a sparse distribution.

As shown in Figure 6, we compare our evolved data with SailorFog-QA. However, we note an important limitation: SailorFog-QA only open-sourced 20 data samples, which is insufficient for drawing statistically reliable distributions. This extremely small sample size results in a very sparse distribution with many empty bins and high variance across different tool-call ranges. The sparsity makes it challenging to draw robust conclusions about the underlying distribution characteristics.

Nevertheless, despite the limited sample size and resulting sparsity in the SailorFog-QA distribution, we can still observe that our WEBEXPLORER-QA achieves a comparable complexity profile. Both datasets demonstrate long-horizon characteristics with multi-turn tool usage, confirming that our synthesis pipeline generates data of a similar difficulty level to other state-of-the-art synthetic approaches.

F EXPERIMENTAL DETAILS AND ANALYSIS

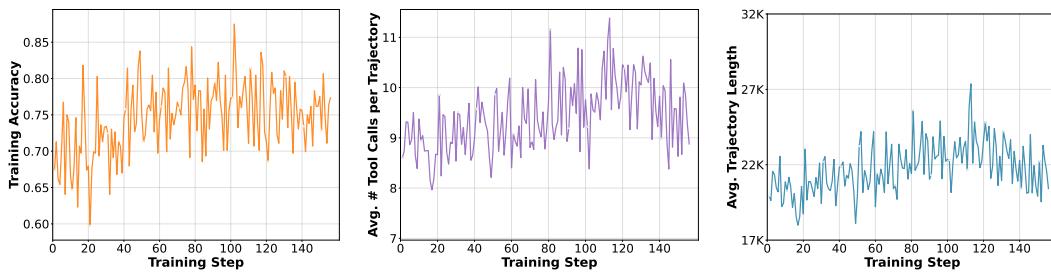
F.1 ABLATION STUDY ON EVOLUTION STAGE

To isolate the contribution of the evolution step in our data synthesis pipeline, we conducted an ablation study by training models with and without the evolution stage. Specifically, we trained a variant model using only the QA pairs from the exploration stage (without the subsequent evolution stage) through the same supervised fine-tuning (SFT) and reinforcement learning (RL) pipeline.

Results As shown in Table 6, models trained without the evolution step consistently show weaker performance across all benchmarks, with the SFT model achieving only 36.3% average accuracy compared to 38.4% with evolution, and the gap widening substantially after RL training (37.9% vs. 43.9%). More importantly, the evolution step proves critical for effective reinforcement learning:

1296 Table 6: Performance comparison of models trained with and without evolution step. “Full” refers
 1297 to QA pairs that went through both exploration and evolution stages.

Model	BC	BC-zh	GAIA	WebWalkerQA	FRAMES	XBench-DS	HLE	Avg
SFT w/o evolve	6.5	20.8	40.3	58.3	66.3	46.3	15.5	36.3
SFT (full)	7.9	21.3	43.7	59.8	72.6	47.5	16.0	38.4
RL w/o evolve	8.0	23.0	44.7	58.5	68.8	45.8	16.3	37.9
RL (full)	15.7	32.0	50.0	62.7	75.7	53.7	17.3	43.9
<i>RL Gain w/o</i>	<i>+1.5</i>	<i>+2.2</i>	<i>+4.4</i>	<i>+0.2</i>	<i>+2.5</i>	<i>-0.5</i>	<i>+0.8</i>	<i>+1.6</i>
<i>RL Gain (full)</i>	<i>+7.8</i>	<i>+10.7</i>	<i>+6.3</i>	<i>+2.9</i>	<i>+3.1</i>	<i>+6.2</i>	<i>+1.3</i>	<i>+5.5</i>



1308
 1309 Figure 7: Training progress during RL. **Left:** Training Accuracy **Middle:** Average # tool calls per
 1310 trajectory **Right:** Average trajectory length (# tokens).

1316 RL yields only a 1.6-point average improvement when training on data without evolution, compared
 1317 to substantial 5.5-point gains with evolved data difference.

1319
 1320 **Analysis** To understand this performance gap, we analyze the RL training dynamics in Figure 7. In
 1321 contrast to RL training with evolved data, which exhibits steadily increasing tool calls and response
 1322 length throughout training, RL training without evolution quickly saturates. The QA data without
 1323 evolution exhibits lower difficulty, causing the model to achieve over 75% training accuracy early in
 1324 the process with significantly fewer average tool calls compared to RL (full) in Figure 4. This early
 1325 saturation limits the model’s ability to learn more complex reasoning patterns, thereby constraining
 1326 further improvement potential.

1327 This stark contrast demonstrates that the evolution step generates sufficiently challenging training
 1328 data that enables effective RL optimization. This finding underscores that data quality, not just
 1329 training methodology, is essential for building capable deep research agents.

F.2 RL TRAINING DETAILS

1335 **Reward Design.** We implement a composite reward function that balances structural correctness
 1336 with answer accuracy:

$$R = 0.2 \cdot R_{\text{format}} + R_{\text{correct}} \quad (5)$$

1337 The format reward R_{format} evaluates the correctness of response formatting, primarily assessing
 1338 whether tool calls and reasoning structures adhere to the specified format requirements. We
 1339 find that tool call formatting is significantly more challenging for models to learn compared to
 1340 thought generation. As illustrated in Appendix A, tool calling requires the model to generate
 1341 a precise JSON structure: `{"name": "search", "arguments": {"queries": ["...", "..."]}}`. Any missing or misplaced character (e.g., quotation marks, brackets, or
 1342 braces) results in an invalid tool call, causing trajectory termination and yielding $R_{\text{format}} = 0$. When
 1343 trajectories are interrupted due to format errors, models are unlikely to provide correct answers,
 1344 resulting in $R_{\text{correct}} = 0$ as well.

1345 In our initial experiments, we observed that insufficient SFT training frequently led to malformed
 1346 tool calls. However, as the SFT training data volume increased, format failures nearly disappeared,
 1347 confirming that adequate supervised pre-training is essential for stable RL optimization.

1350
 1351 The accuracy reward R_{correct} leverages the DeepSeek-V3 model (Liu et al., 2024) as an automated
 1352 judge to evaluate whether final responses correctly answer the question given the ground truth. The
 1353 reward is binary: $R_{\text{correct}} = 1$ for correct answers and $R_{\text{correct}} = 0$ for incorrect answers.
 1354

1355 **Training Configuration.** We adopt GRPO (Shao et al., 2024), purely on-policy setting, for rein-
 1356 forcement learning. The training configuration is as follows:

- 1357 • Learning rate: 1e-6
- 1358 • Generation batch size: 64
- 1359 • Update batch size: 64
- 1360 • Group size: 8
- 1361 • KL loss: None
- 1362 • Initial maximum context length: 64K tokens
- 1363 • Initial sampling temperature: 1.0
- 1364
- 1365

1366 During training, we observed that tool calling frequency and trajectory length increased steadily, as
 1367 Figure 4 shows, necessitating adaptive adjustments to accommodate longer contexts. At training
 1368 step 200, we increased the maximum context length to 96K tokens and raised the sampling temper-
 1369 ature to 1.1 to encourage exploration. As trajectory lengths continued to grow, we further extended
 1370 the context length to 128K tokens at training step 320, maintaining the temperature at 1.1. These
 1371 progressive adjustments enabled the model to handle increasingly complex, long-horizon reasoning
 1372 trajectories while maintaining stable training dynamics.

1373

1374 G THE USE OF LARGE LANGUAGE MODELS

1375

1376 We used large language models only for text polishing to improve grammar and readability. All in-
 1377 tellectual contributions, including the experimental approach, analysis, and scientific insights, were
 1378 developed solely by the authors.

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403