DeepResearcher: Scaling Deep Research via Reinforcement Learning in Real-world Environments

Yuxiang Zheng[1,2,3*] **Dayuan Fu**[2,3*] **Xiangkun Hu**[2*]

Xiaojie Cai[1,3] **Lyumanshan Ye**[1,3] **Pengrui Lu**[1,3] **Pengfei Liu**[1,2,3†]

[1]SJTU [2]SII [3]GAIR

Abstract

Large Language Models (LLMs) equipped with web search capabilities have demonstrated impressive potential for deep research tasks. However, current approaches predominantly rely on either manually engineered prompts (*prompt engineering-based*) with brittle performance or reinforcement learning within controlled Retrieval-Augmented Generation (RAG) environments (*RAG-based*) that fail to capture the complexities of real-world interaction. In this paper, we introduce **DeepResearcher**, the first comprehensive framework for **end-to-end training** of LLM-based deep research agents through **scaling reinforcement learning (RL) in real-world environments** with authentic web search interactions. Unlike RAG-based approaches that assume all necessary information exists within a fixed corpus, our method trains agents to navigate the noisy, unstructured, and dynamic nature of the open web. We implement a specialized multi-agent architecture where browsing agents extract relevant information from various webpage structures and overcoming significant technical challenges. Extensive experiments on open-domain research tasks demonstrate that DeepResearcher achieves substantial improvements of up to **28.9** points over prompt engineering-based baselines and up to **7.2** points over RAG-based RL agents. Our qualitative analysis reveals emergent **cognitive behaviors** from end-to-end RL training, including the ability to formulate plans, cross-validate information from multiple sources, engage in self-reflection to redirect research, and maintain honesty when unable to find definitive answers. Our results highlight that end-to-end training in real-world web environments is not merely an implementation detail but a fundamental requirement for developing robust research capabilities aligned with real-world applications. We release DeepResearcher at https://github.com/GAIR-NLP/DeepResearcher.

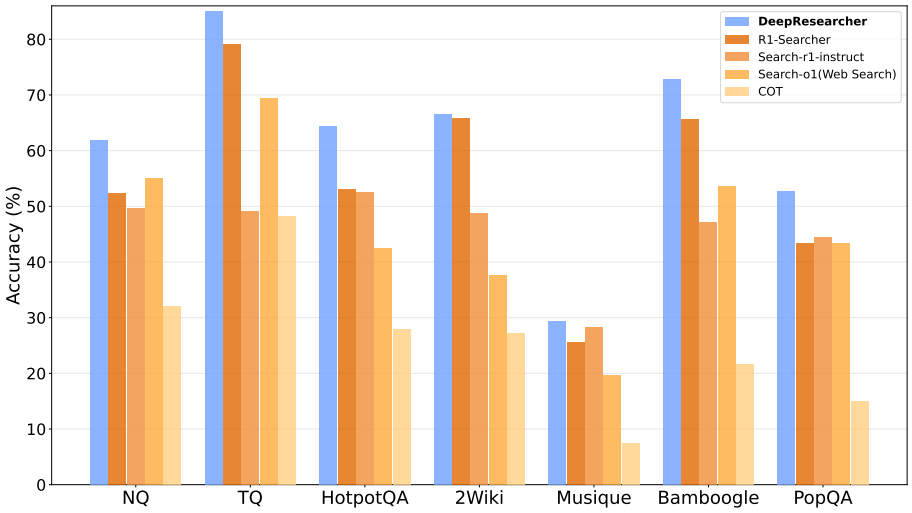


Figure 1: DeepResearcher performs the best on all 7 datasets measured by reliable model-based evaluation.

*Co-first authors
†Corresponding author

Emergent Planning and Cross Validation Behaviors

Planning

Question: *What is the name of the famous bridge in the birthplace of Gloria in D Major's composer?*
Ground Truth: Rialto Bridge

Response Trajectory from DeepResearcher:

<think>
To answer this question, I need to figure out who Gloria in D Major's composer was and their birthplace. Then I can find the famous bridge in that city.

Step 1: Identify the composer of Gloria in D Major.
Step 2: Find the place of birth of that composer.
Step 3: Look for a famous bridge in that city.
</think>
<search>
composer of Gloria in D Major
birthplace of the composer of Gloria in D Major
</search>
<tool_response>
{
'title': 'Gloria (Vivaldi) - Wikipedia',
'snippets': ' Antonio Vivaldi wrote at least three Gloria compositions, settings of the hymn Gloria in excelsis Deo, with words probably dating back to the 4th century.',
'url':'https://en.wikipedia.org/wiki/Gloria_(Vivaldi)'
}
{
'title': 'Antonio Vivaldi: Gloria in D Major, RV 589 - Boston Baroque',
'snippets': ' The Gloria, part of that Turin collection, received its twentieth-century premiere in 1930 and has remained the most popular of all Vivaldi's vocal works ever ...',
'url':'https://baroque.boston/vivaldi-gloria'
}
...(more results are omitted to save space)
</tool_response>
<think>
From the search results, **I can see that Antonio Vivaldi is the composer of Gloria in D Major** and **he was born in Venice**. Now I need to find the famous bridge in Venice.
</think>
<search>
famous bridge in Venice
</search>
<tool_response>
{
'title': 'THE 15 BEST Venice Bridges to Visit (2025) - Tripadvisor',
'snippets': 'Ponte della Paglia is a bridge that bridges the Rio di Palazzo canal on the border of the Sestiere San Marco and Cast',
'url':'https://www.tripadvisor.com/Attractions-g187870-Activities-c47-t5-Venice_Veneto.html'
}
{
'title': 'Rialto Bridge - Wikipedia',
'snippets': 'The Rialto Bridge is the oldest of the four bridges spanning the Grand Canal in Venice, Italy. Connecting the sestieri (districts) of San Marco and San Polo ...',
'url':'https://en.wikipedia.org/wiki/Rialto_Bridge'
}
...(more results are omitted to save space)
</tool_response>
<think>
The Rialto Bridge is the oldest and most famous bridge in Venice. I can now answer the question.
</think>
<answer>
Rialto Bridge
</answer>

Make a plan first
Step 1
Step 2
Step 3

Cross Validation

Question: *Which filmmaker is considered a pioneer, Herschell Gordon Lewis or Leah Meyerhoff?*
Ground Truth: *Herschell Gordon Lewis was a pioneer*

Response Trajectory from DeepResearcher:

<think>
To answer this question, I need to investigate the contributions of both Herschell Gordon Lewis and Leah Meyerhoff to filmmaking. Specifically, I need to understand if either of them is considered a pioneer. First, I will perform a web search to find out more about the filmmakers and what they are known for.
</think>
<search>
Herschell Gordon Lewis filmmaker pioneer
Leah Meyerhoff filmmaker pioneer
</search>
<tool_response>
{
'title': 'Herschell Gordon Lewis - Wikipedia',
'snippets': 'Herschell Gordon Lewis (June 15, 1926 September 26, 2016) was an American filmmaker, best known for creating the 'splatter' subgenre of horror films.',
'url':'https://en.wikipedia.org/wiki/Herschell_Gordon_Lewis'
}
...(more results are omitted to save space)
</tool_response>
<think>
The web search results show that Herschell Gordon Lewis is frequently referred to as a pioneer in the genre of horror films, often being called the "Godfather of Gore" and "Pioneer in the arena of gore or splatter flicks." On the other hand, Leah Meyerhoff is described as an award-winning independent filmmaker, particularly known for her narrative feature film "I Believe In Unicorns". There is no mention of her being a pioneer in the filmmaking industry.

Based on the information retrieved, it seems that Herschell Gordon Lewis is the filmmaker considered a pioneer.

The next step is to narrow down the exact reason why he is considered a pioneer. I will conduct another web search with more specific keywords to **pinpoint the characteristics of his work that set him apart as a pioneer**.
</think>
<search>
Herschell Gordon Lewis why considered a pioneer
Herschell Gordon Lewis pioneer features
</search>
<tool_response>
{
'title': 'Herschell Gordon Lewis, a Pioneer of Gore Cinema, Dies at 90',
'snippets': 'Mr. Lewis set the bar for low-budget horror in movies like "Blood Feast" that went on to inspire filmmakers like Wes Craven and Quentin ...',
'url':'https://www.nytimes.com/2016/09/28/movies/herschell-gordon-lewis-a-pioneer-of-gore-cinema-dies-at-90.html'
}
...(more results are omitted to save space)
</tool_response>
<think>
From the refined search, it is clear that Herschell Gordon Lewis is considered a pioneer due to his pioneering the "splatter" subgenre of horror films. His films introduced the extreme visuals and violent content that set them apart, even though they were often considered an "insult to the art of filmmaking". Therefore, Herschell Gordon Lewis is the filmmaker considered a pioneer.
</think>
<answer>
Herschell Gordon Lewis
</answer>

Validate result from difference sources

Figure 2: Planning and Cross Validation Behaviors. **Left**: DeepResearcher is capable of making plans and dynamically adjusting it throughout the reasoning process. Notably, the model can merge steps when appropriate. **Right**: DeepResearcher identifies the correct answer during its first tool call. However, rather than immediately committing to this result, it proceeds to verify its accuracy through subsequent steps. This rigorous behavior enhances the reliability of the model's responses, ensuring greater robustness in its final answers.

1 Introduction

The emergence of Large Language Models (LLMs) has fundamentally transformed the landscape of artificial intelligence, enabling increasingly autonomous problem-solving capabilities. When equipped with external tools such as web search and code execution (Li et al., 2025c), these models can tackle complex research tasks that previously required significant human workload and expertise. Notable examples include Gemini and OpenAI Deep Research (Google, 2024; OpenAI, 2025), Grok3's DeeperSearch (xAI, 2025), and open-source projects like MetaGPT (Hong et al., 2024), OpenManus (Liang et al., 2025), and OWL agents (CAMEL-AI.org, 2025). These systems demonstrate promising capabilities in synthesizing information, writing and executing code, and conducting iterative investigations across diverse domains. Despite their potential, most current agents are prompt-engineered LLM agents that face significant limitations, while the technical details of commercial systems like OpenAI Deep Research remain completely opaque. Specifically, prompt-engineered agents follow pre-defined workflows designed by developers (Anthropic, 2024), resulting in strict behavioral patterns. Consequently, they often struggle with instruction following, consistent reasoning, exhibit poor generalization to novel tasks, and require extensive manual prompt engineering to achieve reliable performance (Pan et al., 2025). The brittle nature of these systems becomes particularly evident when confronted with complex, multi-step research scenarios requiring adaptive information gathering through web search. While impressive commercial products exist, reproducible frameworks for creating robust research agents remain elusive.

Recent advances suggest that reinforcement learning (RL) offers a promising path forward for improving LLM capabilities. Studies by Guo et al. (2025) and Team et al. (2025) demonstrate that scaling reinforcement learning for LLMs on math and coding tasks (Li et al., 2025b) substantially improves their reasoning abilities. For open-domain tasks, OpenAI has acknowledged using reinforcement learning techniques to enhance their Deep Research agent's capabilities, but detailed methodologies remain proprietary and undisclosed, creating a significant gap in open research. Current open-source efforts to integrate RL with information retrieval, such as Search-R1 (Jin et al., 2025), R1-Searcher (Song et al., 2025), and ReSearch (Chen et al., 2025), have primarily focused on Retrieval-Augmented Generation (RAG) using *static*, *local* text corpora. While these approaches provide valuable insights, they **fundamentally fail to capture the dynamic, unpredictable nature of real-world web search environments**. The controlled RAG setting operates in a highly sanitized environment with a critical limiting assumption: *that all necessary information already exists within their fixed knowledge base. This assumption breaks down in real-world scenarios* where information might be absent, outdated, or require synthesis across domains not covered in the initial knowledge base. Beyond this fundamental limitation, RAG systems also fail to account for the substantial noise, variability in search quality, and the challenges of navigating diverse web content formats and structures.

In this work, we present the first comprehensive study of RL scaling for LLM agents operating with real-world web search capabilities. Our approach, **DeepResearcher**, trains agents to **interact directly with live search engines**, thereby learning to handle the inherent variability and complexity of the open web. By **training in genuine web environments rather than controlled simulations**, our system develops robust capabilities for handling the unpredictable nature of real-world information retrieval and synthesis.

DeepResearcher diverges significantly from prompt-based and RAG-based methods by applying several critical techniques absent from previous work:

- **Scaling RL for Deep Research**: Unlike prompt and SFT-based methods, we directly scale RL training for deep research with only outcome rewards.
- **Real-world Environment**: Unlike controlled RAG environments, real web search presents noisy, unstructured, and heterogeneous information sources that require sophisticated filtering and relevance assessment capabilities.
- **End-to-end Training**: We train the model end-to-end without human priors, enabling the agent to discover its own problem-solving strategies. This end-to-end approach significantly departs from human-designed workflows.
- **Addressing Implementation Challenges**: Training with real web search introduces unique challenges absent in RAG settings, including managing search API rate limits, handling network latency, addressing anti-crawling mechanisms, and processing diverse webpage structures.
- **Multi-agent Framework**: Our approach employs a specialized multi-agent architecture where dedicated browsing agents extract relevant information from entire webpages—a stark contrast to RAG-based systems that simply retrieve and present pre-processed text passages.

Our results show that DeepResearcher achieves up to 28.9 points of improvement in research task completion compared to prompt-engineered agents. When compared to RAG-based RL agents, DeepResearcher demonstrates an improvement of up to 7.2 points. These findings suggest that direct interaction with real search environments is not merely an implementation detail but a crucial component for developing robust research capabilities in autonomous systems that can perform effectively in real-world applications.

Furthermore, our qualitative analysis revealed several important cognitive behaviors that emerge from DeepResearcher's end-to-end RL scaling. During problem-solving, DeepResearcher demonstrates abilities to **make plans**

initially, **cross-validate answers** from multiple sources, engage in **reflection** to redirect research, and **maintain honesty** when unable to find exact answers. These capabilities represent important characteristics for deep research agents and mirror skills valued in human researchers.

To conclude, we make the following contributions:

- We introduce DeepResearcher, a novel RL framework specifically designed for training LLM agents in real web environments, enabling iterative reasoning and search, and synthesizing diverse web information to answer open-domain questions.
- We overcome numerous technical challenges inherent to RL scaling with real-world web search, including API rate limitations, webpage parsing variability, anti-crawling mechanisms, and network latency issues, making this the first successful implementation of reinforcement learning at scale in genuine web environments.
- We conduct extensive experiments across open-domain tasks, demonstrating significant improvements over prompt-engineered baselines and RAG-based RL approaches.
- We perform detailed analysis examining emergent behaviors from DeepResearcher's end-to-end RL scaling, finding that the system can formulate plans, cross-validate answers, reflect on its process, and maintain honesty about limitations.
- We open-source our complete training framework to the research community, fostering transparency and enabling further advancements in deep research systems.

2 Related Work

In this section, we review existing approaches to enhance large language models' (LLMs) ability to access external knowledge with search. We categorize these methods into prompt-based and training-based search agents. We also examine the environments in which these methods operate—either local retrieval-augmented generation (RAG) or real-world web search—and position our work in this landscape.

2.1 Prompt-Based Search Agents

Many current approaches rely on manually crafted workflows that specify how LLMs should interact with external knowledge sources (Wang et al., 2024a). Recent works such as OpenResearcher (Zheng et al., 2024), AirRAG (Feng et al., 2025), IterDRAG (Yue et al., 2024b), Plan*RAG (Verma et al.), Search-o1 (Li et al., 2025a) and Open Deep Search (Alzubi et al., 2025) have demonstrated significant progress in search capabilities through carefully designed workflows. However, these methods face inherent limitations due to their reliance on human-engineered prompts and interaction patterns, resulting in strict behavior patterns that limit adaptability.

2.2 Training-Based Search Agents

Recent developments have moved beyond manually crafted prompts toward training-based approaches that enable more flexible and adaptive search behaviors.

Supervised Fine-Tuning (SFT) SFT for RAG have become an enhanced alternative to manual optimization of RAG workflows (Yu et al., 2024; Wang et al., 2024b). For example, CoRAG (Wang et al., 2024b) utilizes Monte Carlo Tree Search (MCTS) to dynamically select the best document blocks under budget constraints. However, it faces limitations including high computational overhead due to MCTS and weak generalization to unknown scenarios due to the dependence on supervised signals.

Reinforcement Learning (RL) End-to-end reinforcement learning offers a promising alternative that effectively unlocks LLMs' inherent capabilities. By late 2024, large language models achieved remarkable breakthroughs in reasoning capability enhancement through RL (Guo et al., 2025; OpenAI, 2024; Team et al., 2025). Recent research has explored applying RL to external knowledge retrieval, with systems such as Search-R1 (Jin et al., 2025), ReSearch (Chen et al., 2025), and R1-Searcher (Song et al., 2025) abandoning manually specified cues in favor of models that autonomously develop reasoning during the retrieval process. While OpenAI has acknowledged using RL techniques to enhance their research agent's capabilities, detailed methodologies remain proprietary and undisclosed, creating a significant gap in open research.

2.3 Training Environments

Training environments for search agents can be broadly categorized into two types:

Local RAG Environments Current mainstream local RAG frameworks (Gao et al., 2023; Yu et al., 2024) rely on pre-built fixed knowledge repositories, resulting in three critical issues: information timeliness decay, poor domain adaptability, and storage efficiency bottlenecks. While RAG-based RL approaches like Search-R1 (Jin et al., 2025), ReSearch (Chen et al., 2025), and R1-Searcher (Song et al., 2025) have made progress, their experimental

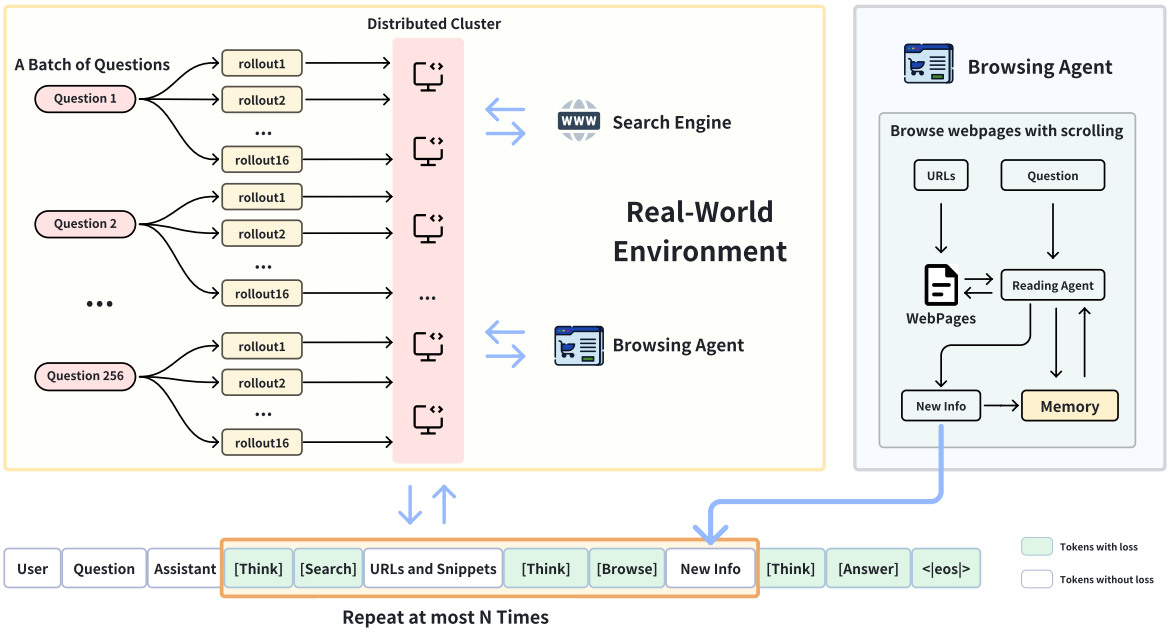


Figure 3: The trajectory of a single sample from a batch of questions processed in parallel by a distributed cluster. Each question undergoes multiple independent rollouts with distinct memory. Upper-left: Displays the batch of questions and their concurrent rollout paths. Upper-right: Shows the browsing agent retrieving web pages via URLs, processing them sequentially to incrementally extract relevant information. Bottom: Details the iterative decision-making steps, from initial query formulation and search to snippet retrieval, further reasoning, browsing, information extraction, and answer generation.

validation remains primarily confined to predefined knowledge bases and similarity-based search, restricting the search space and potentially limiting generalizability to real-world applications.

Real-World Web Search Environments Web search-based methods (Schick et al., 2023; Qin et al., 2023) integrate open search engines with LLMs to access real-time information. These approaches face unique challenges including managing search API rate limits, handling network latency, and processing diverse webpage structures. Despite these challenges, real-world environments offer unstructured and heterogeneous information sources that better reflect the complexity of actual research tasks. However, search-based methods requiring external system participation are seldom trained end-to-end, with research often gravitating toward optimization through manually crafted workflows (Wang et al., 2024a).

In contrast to previous work, our approach uniquely combines reinforcement learning with training in genuine web environments. Unlike existing RL methods that primarily focus on static, local text corpora, our method trains agents to interact directly with live search engines. This enables them to handle the inherent variability and complexity of the open web, developing robust capabilities for real-world information retrieval and synthesis. Our approach addresses the limitations of both prompt-based and RAG-confined methods by learning adaptive search strategies through direct interaction with the unpredictable nature of web environments.

3 Methodology

In this section, we describe the methodology used to train an agent capable of solving problems with web search in dynamic real-world environments.

3.1 Deep Research Trajectory

In a DeepResearcher's trajectory, it conducts reasoning and tool selection based on the user question and observations iteratively as illustrated in Figure 3.

Reasoning We restrict DeepResearcher to do reasoning before taking actions. Each reasoning process is wrapped in a `<think>` tag following the setting in DeepSeek-R1 (Guo et al., 2025).

Web Search Tool DeepResearcher invokes the web search tool by generating a JSON-formatted request with the tool name `web_search` and the search queries as arguments. Search results are returned in a structured format

comprising title, URL, and snippet for each webpage. The current implementation employs a fixed top-k (e.g., 10) value for search results retrieval. Future work could explore LLM-driven dynamic parameter optimization for enhanced search efficacy.

Web Browsing Agent The web browsing agent provides reliable, question-relevant, and incrementally updated information in to the DeepResearcher system. Specifically, the agent maintains a short-term memory repository for each query. Upon receiving a `web_browse` request, it processes the first page segment of the URL in the request. Subsequently, the web browsing agent takes two actions based on the query, historical memory, and the newly acquired webpage content: (1) determining whether to continue reading the next URL/segment or stop, and (2) appending relevant information to the short-term memory. Once the agent decides to discontinue further browsing, it compiles all newly added information from the short-term memory and returns it to the DeepResearcher system.

Answering When the model determines it has sufficient information to answer the question, it generates a final response within <answer></answer>as the answer to return to the user.

3.2 Addressing Challenges in Dynamic Real-World Web Environments

In our open, real-world web setting, several unique challenges arise that necessitate specialized solutions. The following sections detail our strategies for managing these issues effectively.

Challenge I: High-concurrency requests at a single moment The implementation of GRPO results in a large number of sampling iterations, leading to a significant volume of search queries and webpage crawling operations (e.g., 4096), causing long delays. To resolve this issue, we created a distributed CPU server cluster with 50 nodes, specifically designed to manage the Tool requests generated during the RL rollout process. Each server is tasked with handling a portion of these requests, processing search results, and crawling webpages based on the URLs identified by the language model for further reading.

Challenge II: Managing Web Crawling and API Limitations During the crawling phase, the system frequently encounters anti-crawl measures deployed by web servers, which may return irrelevant content or fail to respond entirely. Similarly, when interfacing with search engines or LLM APIs, restrictions such as provider rate limits (e.g. 200 per second) can arise. To mitigate these issues, we implemented a robust retry mechanism that effectively addresses exceptions encountered during API calls or webpage crawling. In addition, we introduced a caching strategy for search results: if an identical search query is made within a predetermined period (e.g., 7 days), the system retrieves the results from the cache. This approach not only reduces the API call frequency but also helps manage the associated costs, particularly for expensive services like the Google Search API.

Challenge III: Optimizing Information Extraction via a Multi-Agent Approach We employ a multi-agent framework wherein a dedicated reading agent is tasked with extracting pertinent information from crawled webpages. Given that many webpages are lengthy and may contain limited relevant content, these pages are partitioned into smaller segments. The reading agent mimics human behavior by processing content sequentially from the first page onward. Under the assumption that if the initial segments of a URL predominantly contain irrelevant information, the webpage is likely unproductive and can be skipped, this method enables more efficient resource allocation and improves overall information extraction accuracy.

3.3 RL Training Framework

Our approach utilizes Reinforcement Learning (RL) to train the agent. This section outlines how we employ the RL framework to train the agent and the tools used within it.

GRPO In this work, we adopt the **Group Relative Policy Optimization (GRPO)** algorithm. GRPO optimizes the current policy π_θ by leveraging a reference policy $\pi_{\theta_{\text{ref}}}$ along with a set of rollouts generated by an existing policy $\pi_{\theta_{\text{old}}}$. Specifically, given G rollouts

$$\tau = \{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|x)$$

(with each input $x \sim D$, where D is the experience distribution), GRPO estimates the baseline using these trajectories instead of training a separate critic. The current policy is then optimized by maximizing the following objective function:

$$\begin{aligned}\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \\ \frac{1}{G}\sum_{i=1}^{G}\left[\min\left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)}A_i, \text{clip}\left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)}, 1-\epsilon, 1+\epsilon\right)A_i\right) - \beta\mathbb{D}_{\text{KL}}\left(\pi_\theta||\pi_{\theta_{\text{ref}}}\right)\right],\end{aligned} \tag{1}$$

Masking Observations The output of the tool is an observation, not the desired result that the model is expected to produce. Therefore, we apply masking to prevent the observation from being involved in training, allowing only the model's responses to contribute to the training process.

3.4 Reward

Rewards play a crucial role during the training process, guiding the agent to continuously improve its performance. This section defines the reward structure and describes how the agent's behavior is rewarded.

We employ F1 score as our primary reward metric due to our utilization of open-domain QA datasets with short-answer ground truth. For future work involving long-form answers, more sophisticated reward mechanisms may be necessary, as noted in the Deep Research system card (OpenAI, 2025). The reward is determined by the following conditions:

$$\text{reward} = \begin{cases} -1 & \text{if format is incorrect} \\ \text{F1 score} & \text{if format is correct} \end{cases}$$

- **Format Penalty**: If the format is incorrect (e.g., missing tags or structural errors), the agent receives a penalty of -1.
- **F1 Reward**: If the format is correct, the reward is based on the word-level F1 score, which measures the accuracy of the generated answer compared to the reference answer. A higher F1 score results in a higher reward.

4 Beyond Memorization: Curating Search-Dependent Training Data

4.1 Leveraging Open Domain QA Data

Despite the growing interest in deep research capabilities for LLM agents, there currently exists no open-source training dataset specifically designed for this purpose. To address this gap, we leverage existing open domain question-answering datasets, which contain single-hop to multi-hop questions that inherently require online search to find accurate answers.

Our training corpus comprises a diverse collection of QA datasets that require varying degrees of retrieval complexity. Specifically, we utilize NaturalQuestions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (TQ) (Joshi et al., 2017) for single-hop scenarios, where answers can typically be found within a single web document. For more complex multi-hop scenarios, which require integrating information across multiple sources, we incorporate examples from HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), both of which were specifically designed to evaluate multi-step reasoning capabilities.

4.2 The Issue of Data Contamination

For training models that genuinely learn to leverage web search tools—rather than simply recalling memorized information—it is critical to address the problem of data contamination. Large language models have been pretrained on vast internet corpora, which likely include many of the QA pairs in standard benchmarks. Without proper contamination detection, the model might appear to successfully complete research tasks while actually using its parametric knowledge, defeating the purpose of learning web search strategies.

This contamination issue is particularly problematic in the context of our work, as it could lead to:

- Models that falsely appear to benefit from web search when actually using memorized knowledge.
- Failure to develop genuine search strategies when deployed on truly novel questions.
- Inability to generalize to real-world research scenarios where answers cannot be found in the model's training data.

4.3 Data Cleaning and Contamination Detection

To ensure the integrity of our training process, we implemented a comprehensive two-stage filtering methodology:

Low-Quality Question Filtering We exclude questions that could yield unreliable or problematic search results. Specifically, we eliminate: 1) Time-sensitive questions (e.g., "Who is the current CEO of Apple?"); 2) Highly subjective queries (e.g., "What is the best smartphone?"); and 3) Potentially harmful or policy-violating content. This filtering was implemented using DeepSeek-R1 (Guo et al., 2025) with a carefully designed evaluation prompt to systematically identify and mark problematic questions.

Contamination Detection To ensure the model genuinely learns to use search tools rather than memorizing answers, we employed a robust contamination detection procedure. For each candidate question, we randomly sample 10 responses from the base model we will use in training, and check if any response contains the ground truth answer (i.e., pass@10). Questions where the model demonstrated prior knowledge (by producing the correct answer without search) were excluded from the training set. This contamination screening is critical for preventing the model from developing a false reliance on parametric knowledge when search-based knowledge is required.

The prompts used for data cleaning and contamination detection are listed in Appendix A.1. After applying these quality control measures, we constructed a final training dataset of 80,000 examples with a distribution ratio of 1:1:3:3 for NQ:TQ:HotpotQA:2Wiki. This proportion deliberately emphasizes multi-hop scenarios (75% of examples), as these better reflect the complex information-seeking behaviors required for deep research questions.

5 Experiments

5.1 Experimental Setups

5.1.1 Model and Hyperparameters

We adopt Qwen2.5-7B-Instruct[1] (Yang et al., 2024a) as the backbone model for our training pipeline. The training is conducted using the verl framework[2]. At each training step, we sample 256 prompts, and sample 16 rollouts for each prompt. Each rollout consists of up to 10 tool calls followed by a final answer step. The training is performed with a mini-batch size of 4,096, which means one rollout stage will backprop for one time.

5.2 Evaluation and Results

5.2.1 Benchmarks

To thoroughly evaluate model performance across both in-domain (ID) and out-of-domain (OOD) settings, we construct a diverse benchmark suite spanning a range of open-domain QA challenges. For in-domain evaluation, we include the dev sets of NQ (Kwiatkowski et al., 2019), TQ (Joshi et al., 2017), HotpotQA (Yang et al., 2018), and 2Wiki (Ho et al., 2020) as mentioned in Section 4.

For out-of-domain evaluation, we introduce three datasets that differ significantly in question style and information distribution: MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2022), and PopQA (Mallen et al., 2022). These datasets test the model's generalization ability beyond the training domain.

To ensure a fair and balanced evaluation, we randomly sample 512 examples from the development sets of NQ, TQ, HotpotQA, 2Wiki, MuSiQue, and PopQA as well as all 125 samples from Bamboogle's development set. This sampling strategy allows us to assess model robustness across a broad range of topics and reasoning requirements.

5.2.2 Baselines

To evaluate the effectiveness of DeepResearcher, we compare it against the following baseline methods:

- **CoT Only**: This baseline employs Chain-of-Thought (CoT) () reasoning to generate answers without access to any external reference context.
- **RAG**: This approach combines Chain-of-Thought reasoning with retrieved reference context to guide the answer generation process.
- **Search-o1**: A multi-step reasoning baseline in which the model generates search queries or intermediate answers. For each query, the context is limited to a snippet retrieved by a retriever, rather than full documents.[3]
- **Search-o1 + Web Search**: In contrast to Search-o1, this setting lets the model access the open web. It can send real-time search queries through APIs like Serper and visit URLs to browse webpages. This supports richer, more dynamic information gathering and forms the basis of deep research.
- **Search-r1**: A reinforcement learning method for question answering. During the training and inference stages, it searches Wikipedia information with the help of a retriever. There are two versions: Search-r1-base and Search-r1-instruct, where the initial actor model is either the base model or the instruct model.
- **R1-Searcher**: Unlike Search-r1, when given a search query, it appends `"site:en.wikipedia.org"` to the query, search it via Bing, and summarizes the first three pages of the search results. DeepResearcher differs from this approach in three key aspects: (1) DeepResearcher is also trained with real-world environment; (2) DeepResearcher does not restrict the search space to a specific domain, such as Wikipedia; and (3) Our

[1] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
[2] https://github.com/volcengine/verl
[3] To ensure consistency with other results, we reimplemented search-o1 using our own prompt.

Method	Inference Environment	NQ		TQ		HotpotQA		2Wiki	
		F1	MBE	F1	MBE	F1	MBE	F1	MBE
Prompt Based									
CoT	Local RAG	19.8	32.0	45.6	48.2	24.4	27.9	26.4	27.3
CoT + RAG	Local RAG	42.0	59.6	68.9	75.8	37.1	43.8	24.4	24.8
Search-o1*	Local RAG	34.5	57.4	52.6	61.1	31.6	40.8	28.6	32.8
Search-o1	Web Search	32.4	55.1	58.9	69.5	33.0	42.4	30.9	37.7
Training Based									
Search-r1-base	Local RAG	**45.4**	60.0	71.9	76.2	**55.9**	63.0	44.6	47.9
Search-r1-instruct	Local RAG	33.1	49.6	44.7	49.2	45.7	52.5	43.4	48.8
R1-Searcher	Web Search	35.4	52.3	73.1	79.1	44.8	53.1	59.4	65.8
DeepResearcher	Web Search	39.6	**61.9**	**78.4**	**85.0**	52.8	**64.3**	**59.7**	**66.6**

Table 1: In-domain results on four datasets (NQ, TQ, HotpotQA, 2Wiki), evaluated by F1 and MBE metrics. DeepResearcher outperforms all baseline methods in MBE and shows competitive performance in F1, particularly excelling on TQ and 2Wiki. It is worth noting that Search-r1-base was trained and evaluated in a local RAG environment with direct access to the relevant Wikipedia corpus, while DeepResearcher must navigate the entire Internet to find information, achieving excellent results despite facing a more realistic and challenging scenario.

method allows the model to autonomously select URLs rather than compulsorily summarizing the top three search results.

5.2.3 Evaluation Metrics

Rule-based Metrics We evaluate model performance using F1 score aligning with the reward for training. Both ground-truth and predicted answers are normalized by converting to lowercase and removing all punctuation before computing the metrics.

Model-based Evaluation Rule-based evaluation doesn't suit long-form responses, so we adopt a model-based evaluation (MBE) approach using LLM-as-a-Judge (Zheng et al., 2023). Specifically, we prompt GPT-4o-mini (Hurst et al., 2024) to assess the model's answer against the question and ground truth answer, and label it as either "correct" or "incorrect." The MBE score is then computed as the accuracy of these judgments.(Zheng et al., 2023) The full prompt is provided in Appendix A.2.

5.2.4 Main Results

Table 1 and Table 2 present the main results of DeepResearcher and the baselines in-domain and out-of-domain, respectively. From these results, we draw the following observations:

DeepResearcher outperforms the baselines within training domains. As shown in Table 1, DeepResearcher achieves the highest performance across the four datasets when measured by the more reliable MBE metric, outperforming baselines by a substantial margin on TQ and 2Wiki. While Search-r1-base shows comparable MBE results on NQ and HotpotQA, it's important to note that Search-r1-base was specifically trained and evaluated using a local RAG system with direct access to the relevant Wikipedia corpus. In contrast, DeepResearcher must navigate the entire Internet to find relevant information, representing a more realistic and significantly more challenging scenario even though the answers ultimately come from Wikipedia.

DeepResearcher demonstrates exceptional generalization to novel domains. As revealed in Table 2, DeepResearcher consistently outperforms all other baselines across three OOD datasets. This indicates that the model successfully learns generalizable skills for reasoning, searching, and synthesizing information from different sources through RL scaling, rather than merely adapting to specific training distributions.

Importance of Real-World Environment in Training Questions in Bamboogle specifically require knowledge beyond Wikipedia's coverage. Consequently, DeepResearcher significantly outperforms local RAG-based methods on this benchmark. Furthermore, even when we enable R1-Searcher (which was trained using local RAG) to search the real-world web, it still performs substantially worse than DeepResearcher. These results demonstrate the critical advantage of using real-world environments during RL scaling training, as this exposure develops robust information retrieval and synthesis capabilities that cannot be achieved in controlled, static environments.

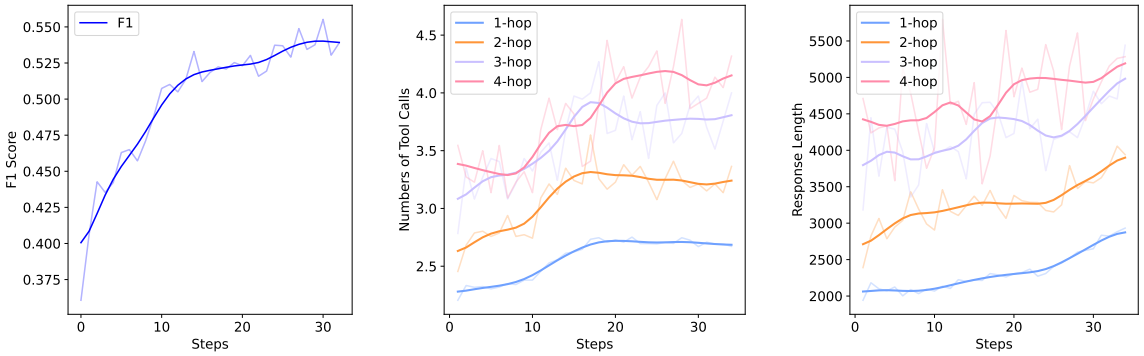
Method	Inference Environment	Musique		Bamboogle		PopQA	
		F1	MBE	F1	MBE	F1	MBE
Prompt Based							
CoT	Local RAG	8.5	7.4	22.1	21.6	17.0	15.0
CoT + RAG	Local RAG	10.0	10.0	25.4	27.2	46.9	48.8
Search-o1*	Local RAG	16.8	21.3	35.8	38.4	36.9	42.4
Search-o1	Web Search	14.7	19.7	46.6	53.6	38.3	43.4
Training Based							
Search-r1-base	Local RAG	26.7	27.5	56.5	57.6	43.2	47.0
Search-r1-instruct	Local RAG	26.5	28.3	45.0	47.2	43.0	44.5
R1-Searcher	Web Search	22.8	25.6	64.8	65.6	42.7	43.4
DeepResearcher	Web Search	**27.1**	**29.3**	**71.0**	**72.8**	**48.5**	**52.7**

Table 2: This table shows the performance of different methods on three out-of-domain datasets (Musique, Bamboogle, PopQA), evaluated by F1 and MBE metrics. DeepResearcher leads in both F1 and MBE on all datasets, demonstrating strong generalization capabilities compared to other methods. Notably, unlike the other datasets, Bamboogle's corpus is not entirely derived from Wikipedia pages.

6 Analysis

6.1 Training Dynamics

- **Performance gradually scaling with reinforcement learning:** Figure 4 (a) present the evaluation of F1 scores, across different training steps. The F1 score 0.375, and gradually increases to around 0.55 demonstrating a consistent upward trend. This result indicates the progressive improvement of the model's performance in reinforcement learning.
- **Training leads to increased reasoning steps in hard question:** Figure 4 (b) illustrates the average number of turns required for different reasoning hops. The general trend indicates that as the training progresses, the required number of tool calls also increases across different difficulty levels. Unlike the other three settings, the 4-hop setting continues to exhibit an increasing trend even after 34 steps. This suggests that the model is still learning to retrieve more information when dealing with more difficult questions.
- **Continuous learning makes long response without saturation:** Figure 4 (c) presents the length of responses for different reasoning hops. The response lengths also increase with reasoning complexity. However, all four settings show a sustained upward trend, indicating that the model continues to expand its reasoning processes during training. This further supports the idea that the model adapts to increasingly complex queries by generating more detailed outputs like double-check, refinement, planning, etc.



(a) (b) (c)

Figure 4: Training dynamics of F1, turns and response length.

6.2 Case Study

Figures 2 and 6 present four cases illustrating the model's behavior after reinforcement learning. From these examples, we identify several key behavioral patterns:

Planning

Question: *What is the name of the famous bridge in the birthplace of Gloria in D Major's composer?*
Ground Truth: Rialto Bridge

Response Trajectory from DeepResearcher:

<think>
To answer this question, I need to figure out who Gloria in D Major's composer was and their birthplace. Then I can find the famous bridge in that city.

Step 1: Identify the composer of Gloria in D Major.
Step 2: Find the place of birth of that composer.
Step 3: Look for a famous bridge in that city.
</think>
<search>
composer of Gloria in D Major
birthplace of the composer of Gloria in D Major
</search>

Make a plan first

<tool_response>
{
'title': 'Gloria (Vivaldi) - Wikipedia',
'snippets': ' Antonio Vivaldi wrote at least three Gloria compositions, settings of the hymn Gloria in excelsis Deo, with words probably dating back to the 4th century.',
'url':'https://en.wikipedia.org/wiki/Gloria_(Vivaldi)'
}
{
'title': 'Antonio Vivaldi: Gloria in D Major, RV 589 - Boston Baroque',
'snippets': ' The Gloria, part of that Turin collection, received its twentieth-century premiere in 1930 and has remained the most popular of all Vivaldi's vocal works ever ...',
'url':'https://baroque.boston/vivaldi-gloria'
}
...(more results are omitted to save space)
</tool_response>
<think>

Step 1

From the search results, **I can see that Antonio Vivaldi is the composer of Gloria in D Major** and **he was born in Venice**. Now I need to find the famous bridge in Venice.
</think>
<search>
famous bridge in Venice
</search>

Step 2

<tool_response>
{
'title': 'THE 15 BEST Venice Bridges to Visit (2025) - Tripadvisor',
'snippets': 'Ponte della Paglia is a bridge that bridges the Rio di Palazzo canal on the border of the Sestiere San Marco and Cast',
'url':'https://www.tripadvisor.com/Attractions-g187870-Activities-c47-t5-Venice_Veneto.html'
}
{
'title': 'Rialto Bridge - Wikipedia',
'snippets': 'The Rialto Bridge is the oldest of the four bridges spanning the Grand Canal in Venice, Italy. Connecting the sestieri (districts) of San Marco and San Polo ...',
'url':'https://en.wikipedia.org/wiki/Rialto_Bridge'
}
...(more results are omitted to save space)
</tool_response>
<think>
The Rialto Bridge is the oldest and most famous bridge in Venice. I can now answer the question.
</think>

Step 3

<answer>
Rialto Bridge
</answer>

Cross Validation

Question: *Which filmmaker is considered a pioneer, Herschell Gordon Lewis or Leah Meyerhoff?*
Ground Truth: *Herschell Gordon Lewis was a pioneer*

Response Trajectory from DeepResearcher:

<think>
To answer this question, I need to investigate the contributions of both Herschell Gordon Lewis and Leah Meyerhoff to filmmaking. Specifically, I need to understand if either of them is considered a pioneer. First, I will perform a web search to find out more about the filmmakers and what they are known for.
</think>
<search>
Herschell Gordon Lewis filmmaker pioneer
Leah Meyerhoff filmmaker pioneer
</search>
<tool_response>
{
'title': 'Herschell Gordon Lewis - Wikipedia',
'snippets': 'Herschell Gordon Lewis (June 15, 1926 September 26, 2016) was an American filmmaker, best known for creating the 'splatter' subgenre of horror films.',
'url':'https://en.wikipedia.org/wiki/Herschell_Gordon_Lewis'
}
...(more results are omitted to save space)
</tool_response>
<think>
The web search results show that Herschell Gordon Lewis is frequently referred to as a pioneer in the genre of horror films, often being called the "Godfather of Gore" and "Pioneer in the arena of gore or splatter flicks." On the other hand, Leah Meyerhoff is described as an award-winning independent filmmaker, particularly known for her narrative feature film "I Believe In Unicorns". There is no mention of her being a pioneer in the filmmaking industry.

Based on the information retrieved, it seems that Herschell Gordon Lewis is the filmmaker considered a pioneer.

The next step is to narrow down the exact reason why he is considered a pioneer. I will conduct another web search with more specific keywords to **pinpoint the characteristics of his work that set him apart as a pioneer**.
</think>
<search>
Herschell Gordon Lewis why considered a pioneer
Herschell Gordon Lewis pioneer features
</search>

Validate result from difference sources

<tool_response>
{
'title': 'Herschell Gordon Lewis, a Pioneer of Gore Cinema, Dies at 90',
'snippets': 'Mr. Lewis set the bar for low-budget horror in movies like "Blood Feast" that went on to inspire filmmakers like Wes Craven and Quentin ...',
'url':'https://www.nytimes.com/2016/09/28/movies/herschell-gordon-lewis-a-pioneer-of-gore-cinema-dies-at-90.html'
}
...(more results are omitted to save space)
</tool_response>
<think>
From the refined search, it is clear that Herschell Gordon Lewis is considered a pioneer due to his pioneering the "splatter" subgenre of horror films. His films introduced the extreme visuals and violent content that set them apart, even though they were often considered an "insult to the art of filmmaking". Therefore, Herschell Gordon Lewis is the filmmaker considered a pioneer.
</think>
<answer>
Herschell Gordon Lewis
</answer>

Figure 5: Planning and Cross Validation Behaviors. **Left**: DeepResearcher is capable of making plans and dynamically adjusting it throughout the reasoning process. Notably, the model can merge steps when appropriate. **Right**: DeepResearcher identifies the correct answer during its first tool call. However, rather than immediately committing to this result, it proceeds to verify its accuracy through subsequent steps. This rigorous behavior enhances the reliability of the model's responses, ensuring greater robustness in its final answers.

Reflection

Question: *Where was the place of burial of Djedefhor's father?*
Ground Truth: *Great Pyramid of Giza*

Response Trajectory from DeepResearcher:
<think>
To answer this question, I need to find out information about Djedefhor's father and the place of his burial. I will start with a web search.
</think>
<search>
Herschell Gordon Lewis filmmaker pioneer
Leah Meyerhoff filmmaker pioneer
place of burial of Djedefhor's father
Djedefhor father burial site
who was Djedefhor's father and where was he buried
</search>
<tool_response>
...(omitted to save space)
</tool_response>
<think>
The web search results provide clues about Djedefhor's father. **However, the search results are mostly about Boromir's father Denethor from the Lord of the Rings, not Djedefhor's father. I need to refine my search to find information about Djedefhor specifically.**
</think>
<search>
who was Djedefhor's father and where was he buried
Djedefhor father name
Djedefhor father identity
</search>
<tool_response>
...(omitted to save space)
</tool_response>
<think>
The new search results provide information about Djedefhor's father, Khufu, and also his burial place. **However, the search results for his burial need to be more specific.** I will use the web browser tool to look at the pages where the information may be more relevant.
</think>
<browse_webpage>
https://en.wikipedia.org/wiki/Djedefre
https://www.ancient-egypt.org/history/old-kingdom/4th-dynasty/djedefre/biography-of-djedefre.html
https://ancientegyptonline.co.uk/djedefre/
</browse_webpage>
<tool_response>
...(omitted to save space)
</tool_response>
<think>
From the browsing pages, it is evident that Djedefhor's father is Khufu. The pages specifically mention that Khufu was the father of Djedefhor. However, I need to find the place of burial of Djedefhor's father, Khufu.
</think>
<search>
place of burial of Khufu
Khufu burial site
where was Khufu buried
</search>
<tool_response>
...(omitted to save space)
</tool_response>
<think>
The search results clearly indicate that Khufu's burial place was in the Great Pyramid of Giza. This pyramid served as the tomb for pharaoh Khufu.
</think>
<answer>
Great Pyramid of Giza
</answer>

Reflection and adjust research direction

Honesty

Question: *In the seventh-largest country in Africa, what is the oil prodcution of the city at the mouth of the Congo River?*
Ground Truth: *1200000 oilbbl/d*

Response Trajectory from DeepResearcher:
<think>
To answer this question, I need to identify the city at the mouth of the Congo River in the seventh-largest country in Africa, and then find its current oil production.
</think>
<search>
seventh largest country in africa
city at the mouth of the Congo River in the seventh largest country in africa
</search>
<tool_response>
...(omitted to save space)
</tool_response>
<think>
From the search results, I found that Angola is the seventh largest country in Africa, and its capital Luanda is located near the mouth of the Congo River. Now I need to find the oil production of Luanda.
</think>
<search>
oil production of Luanda
oil production in Angola
</search>
<tool_response>
...(omitted to save space)
</tool_response>
<think>
From the search results, I can see that Angola currently produces around 1.16 million barrels of oil per day. However, I should find the specific production of the city at the mouth of the Congo River, which is Luanda.
</think>
<search>
oil production of Luanda
oil production in Luanda
</search>
<tool_response>
...(omitted to save space)
</tool_response>
<think>
After a series of web searches, I couldn't find specific information about the oil production of Luanda. The results are mostly about Angola's total oil production. **Since Luanda is the capital and the largest city of Angola, it's likely that a significant portion of Angola's oil production happens there.** For a more exact figure, we might need to explore specialized databases or recent reports about the oil industry in Luanda.
</think>
<browse_webpage>
https://en.wikipedia.org/wiki/Luanda
https://www.worldometers.info/country-data/angola/oil-production
</browse_webpage>
<think>
It seems the information about the oil production of Luanda is not directly available in the search results or in the previously browsed webpages. Given the information from previous searches, Angola is a major oil producer, and as the capital and largest city, Luanda likely plays a key role in the country's oil production. However, to answer the question precisely, we may need a more detailed report on the city's oil production, which might be included in specialized databases or recent reports. **Since the exact oil production of Luanda is not available from the searches and browsing**, and given the context and the previously found information that Angola is a major oil producer, **it's reasonable to infer that the oil production in the city at the mouth of the Congo River (Luanda) is a significant part** of Angola's total oil production. **However, without a precise figure, I will not provide a number.**
</think>
<answer>
significant portion
</answer>

Be honest when didn't find the exact answer

Figure 6: Reflection and Honesty Behavior. The search and browse are 2 apis in json format in the real inference stage. Left: When the retrieved information does not fully align with the question, DeepResearcher recognizes this discrepancy based on environmental feedback and refines its search query in subsequent tool calls. This proves its reflection ability. Right: DeepResearcher is capable of recognizing when it has not found the correct answer and appropriately declines to provide a response to be honesty.

- **Behavior I: Planning when addressing multi-hop questions:** As demonstrated on the left side of Figure 2, DeepResearcher is capable of making plans and dynamically adjusting it throughout the reasoning process. Notably, the model can merge steps when appropriate, indicating that planning abilities can emerge naturally without the necessity of SFT on explicit planning data (Yue et al., 2024a).
- **Behavior II: Cross-validation before finalizing its answers:** As observed on the right side of Figure 2, DeepResearcher identifies the correct answer during its first tool call. However, rather than immediately committing to this result, it proceeds to verify its accuracy through subsequent steps. This cautious approach enhances the reliability of the model's responses, ensuring greater robustness in its final predictions.
- **Behavior III: Reflection when observations deviate from expectations:** The left side of Figure 6 illustrates the model's ability to reflect on its search process. When the retrieved information does not fully align with the question, DeepResearcher recognizes this discrepancy based on environmental feedback and refines its search query in subsequent tool calls. This reflective capability is essential for preventing the model from getting stuck (Fu et al., 2025) in reasoning, enabling it to enhance overall problem-solving efficiency.
- **Behavior IV: Honesty by acknowledging its limitations:** A reliable model should minimize hallucinations and provide honest responses when it lacks the necessary knowledge (Yang et al., 2024b). We observe that DeepResearcher is capable of recognizing when it has not found the correct answer and appropriately declines to provide a response. This behavior is beneficial, however, current question-answering evaluation metrics do not yet account for this aspect of model reliability.

7 Conclusion

In conclusion, we presents DeepResearcher, a groundbreaking approach for scaling reinforcement learning in LLMs to operate effectively in real-world web search environments. Unlike existing methods that rely on static knowledge bases or controlled retrieval settings, DeepResearcher trains agents to interact directly with live search engines, allowing them to navigate the inherent complexity and variability of the open web. This direct engagement with dynamic search environments leads to substantial improvements in task completion and research capabilities compared to both prompt-engineered and RAG-based RL agents.

By adopting an end-to-end training framework, DeepResearcher moves beyond human-engineered workflows, empowering the agent to autonomously develop problem-solving strategies. Our approach not only addresses the unique challenges of real-world web search, such as network latency and anti-crawling mechanisms, but also provides a robust multi-agent architecture that enhances the agent's ability to collect diverse information from the web. The resulting system demonstrates notable cognitive behaviors such as planning, cross-validation, reflection, and maintaining honesty, which are crucial for autonomous agents conducting deep research.

The success of DeepResearcher marks a significant milestone in the evolution of LLM agents, showing that scaling reinforcement learning in real-world environments can unlock substantial improvements in research performance. This approach offers a promising path forward for building more adaptive, intelligent systems capable of solving complex, open-domain problems that are relevant to real-world applications.

References

[1] Salaheddin Alzubi, Creston Brooks, Purva Chiniya, Edoardo Contente, Chiara von Gerlach, Lucas Irwin, Yihan Jiang, Arda Kaz, Windsor Nguyen, Sewoong Oh, et al. 2025. Open deep search: Democratizing search with open-source reasoning agents. *arXiv preprint arXiv:2503.20201*.

[2] Anthropic. 2024. Building effective agents.

[3] CAMEL-AI.org. 2025. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. `https://github.com/camel-ai/owl`. Accessed: 2025-03-07.

[4] Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. Research: Learning to reason with search for llms via reinforcement learning.

[5] Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Jingyi Song, and Hao Wang. 2025. Airrag: Activating intrinsic reasoning for retrieval augmented generation via tree-based search. *arXiv preprint arXiv:2501.10053*.

[6] Dayuan Fu, Keqing He, Yejie Wang, Wentao Hong, Zhuoma Gongque, Weihao Zeng, Wei Wang, Jingang Wang, Xunliang Cai, and Weiran Xu. 2025. Agentrefine: Enhancing agent generalization through refinement tuning. *arXiv preprint arXiv:2501.01702*.

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

[8] Google. 2024. Gemini deep research.

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

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

[11] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

[12] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

[13] Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

[14] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

[15] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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

[17] Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025b. Limr: Less is more for rl scaling.

[18] Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025c. Torl: Scaling tool-integrated rl.

[19] Xinbin Liang, Jinyu Xiang, Zhaoyang Yu, Jiayi Zhang, and Sirui Hong. 2025. Openmanus: An open-source framework for building general ai agents. `https://github.com/mannaandpoem/OpenManus`.

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

[21] OpenAI. 2024. Learning to reason with llms, september 2024.

[22] OpenAI. 2025. Deep research system card. Technical report, OpenAI.

[23] Melissa Z Pan, Mert Cemri, Lakshya A Agrawal, Shuyi Yang, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Kannan Ramchandran, Dan Klein, Joseph E. Gonzalez, Matei Zaharia, and Ion Stoica. 2025. Why do multiagent systems fail? In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

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

[25] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

[26] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

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

[28] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

[29] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*.

[30] Prakhar Verma, Sukruta Prakash Midigeshi, Gaurav Sinha, Arno Solin, Nagarajan Natarajan, and Amit Sharma. Plan $\backslash$ast rag : $Efficienttest-timeplanningforretrievalaugmentedgeneration.$ $InWorkshop$ *on Reasoning and Planning for Large Language Models*.

[31] Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, et al. 2024a. Searching for best practices in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17716–17736.

[32] Ziting Wang, Haitao Yuan, Wei Dong, Gao Cong, and Feifei Li. 2024b. Corag: A cost-constrained retrieval optimization system for retrieval-augmented generation. *arXiv preprint arXiv:2411.00744*.

[33] xAI. 2025. Grok 3.

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

[35] Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2024b. Alignment for honesty. *Advances in Neural Information Processing Systems*, 37:63565–63598.

[36] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[37] Tian Yu, Shaolei Zhang, and Yang Feng. 2024. Auto-rag: Autonomous retrieval-augmented generation for large language models. *arXiv preprint arXiv:2411.19443*.

[38] Murong Yue, Wenlin Yao, Haitao Mi, Dian Yu, Ziyu Yao, and Dong Yu. 2024a. Dots: Learning to reason dynamically in llms via optimal reasoning trajectories search. *arXiv preprint arXiv:2410.03864*.

[39] Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2024b. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*.

[40] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

[41] Yuxiang Zheng, Shichao Sun, Lin Qiu, Dongyu Ru, Cheng Jiayang, Xuefeng Li, Jifan Lin, Binjie Wang, Yun Luo, Renjie Pan, Yang Xu, Qingkai Min, Zizhao Zhang, Yiwen Wang, Wenjie Li, and Pengfei Liu. 2024. OpenResearcher: Unleashing AI for accelerated scientific research. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 209–218, Miami, Florida, USA. Association for Computational Linguistics.

A Prompts

A.1 Prompt for Question Quality Level Evaluation

The prompt below displays two templates. Identifies if questions are time-sensitive, subjective, or potentially harmful. Includes classification guidelines, question placeholder, and required answer tag format.

Prompt for training data quality checking

Please identify whether the given question is time-sensitive, subjective, or may cause harmful answers.

- Time-sensitive: The answer to the question may change over time.
- Harmful: The answer to the question may be harmful or offensive.
- Subjective: The answer to the question may be subjective and not based on facts.

Here is the question:
<question>
{question}
</question>

Wrap your answer in <answer> tags with one of the following values:
- time_sensitive: if the question is time-sensitive
- harmful: if the question may cause harmful answers
- subjective: if the question is subjective
- good: if the question is none of the above

The prompt below shows the template prompt for contamination detection. To tests if AI responses are influenced by training data contamination.

Prompt for contamination detection

Give a short answer to the following question. The answer should be in English.

Question: {question}

Your answer:

A.2 Prompt for Model's Answer Quality Level Evaluation

The prompt below provides instructions for evaluating the correctness of AI-generated answers (pred answer) against a list of ground truth answers. To judge if a predicted answer correctly answers a question by comparing it to ground truth answers.

Prompt for Model-based Evaluation

You will be given a question and its ground truth answer list where each item can be a ground truth answer. Provided a pred_answer, you need to judge if the pred_answer correctly answers the question based on the ground truth answer list.
You should first give your rationale for the judgement, and then give your judgement result (i.e., correct or incorrect).

Here is the criteria for the judgement:
1. The pred_answer doesn't need to be exactly the same as any of the ground truth answers, but should be semantically same for the question.
2. Each item in the ground truth answer list can be viewed as a ground truth answer for the question, and the pred_answer should be semantically same to at least one of them.

question: {question}
ground truth answers: {gt_answer}
pred_answer: {pred_answer}

The output should in the following json format:
"'json
{
"rationale": "your rationale for the judgement, as a text",
"judgement": "your judgement result, can only be 'correct' or 'incorrect'"
}
"'

Your output:

B Training Scaling Result

Figure 7 presents F1 score in 7 benchmarks. We sampled 125 cases from each benchmarks' development set. DeepResearcher can scaling in all benchmarks especially in OOD benchmarks.

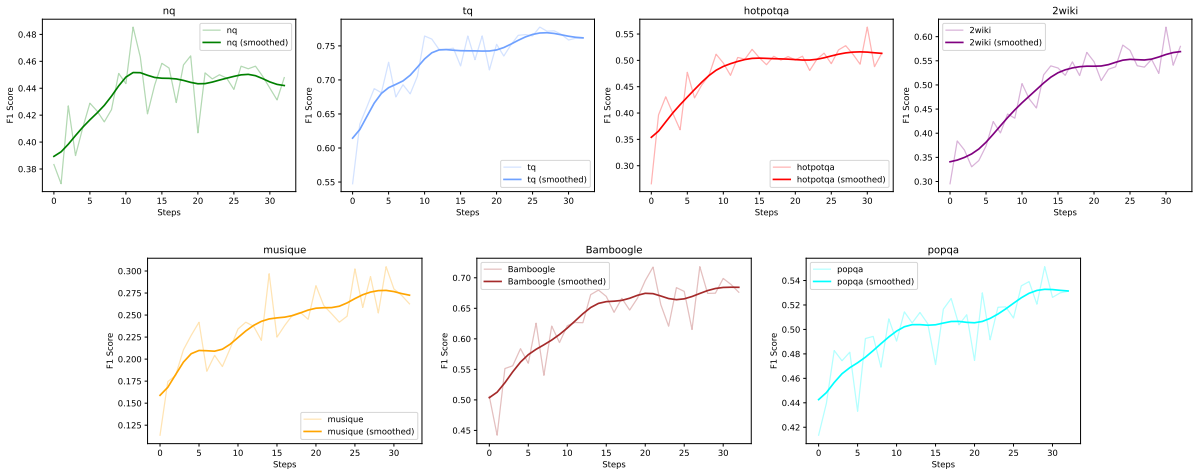


Figure 7: F1 score during training