WebResearcher: Unleashing unbounded reasoning capability in Long-Horizon Agents

Zile Qiao*(✉), Guoxin Chen*, Xuanzhong Chen*, Donglei Yu*, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, Rui Min, Minpeng Liao, Yong Jiang(✉), Pengjun Xie, Fei Huang, Jingren Zhou

Tongyi Lab, Alibaba Group

https://tongyi-agent.github.io/blog

https://github.com/Alibaba-NLP/DeepResearch

Abstract

Recent advances in deep-research systems have demonstrated the potential for AI agents to autonomously discover and synthesize knowledge from external sources. In this paper, we introduce **WebResearcher**, a novel framework for building such agents through two key components: (1) **IterResearch**, an iterative deep-research paradigm that reformulates deep research as a Markov Decision Process, where agents periodically consolidate findings into evolving reports while maintaining focused workspaces—overcoming the context suffocation and noise contamination that plague existing mono-contextual approaches; and (2) **WebFrontier**, a scalable data synthesis engine that generates high-quality training data through tool-augmented complexity escalation, enabling systematic creation of research tasks that bridge the gap between passive knowledge recall and active knowledge construction. Notably, we find that the training data from our paradigm significantly enhances tool-use capabilities even for traditional mono-contextual methods. Furthermore, our paradigm naturally scales through parallel thinking, enabling concurrent multi-agent exploration for more comprehensive conclusions. Extensive experiments across 6 challenging benchmarks demonstrate that WebResearcher achieves state-of-the-art performance, even surpassing frontier proprietary systems.

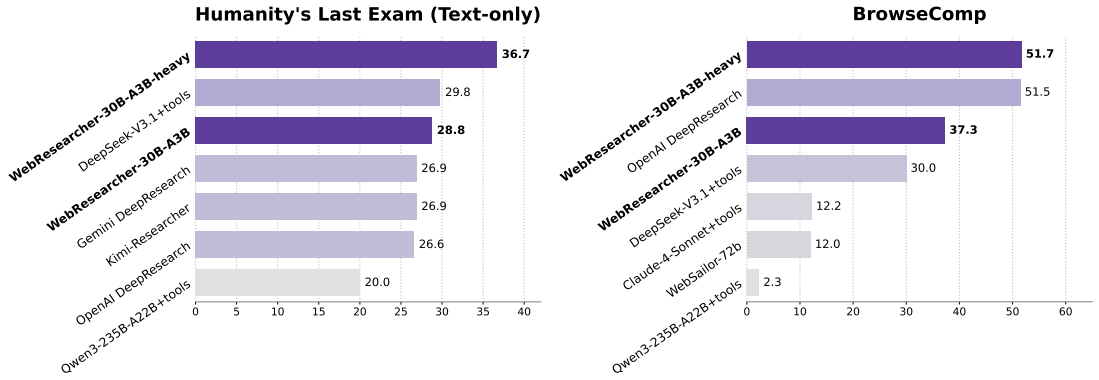


Figure 1: Performance comparison between WebResearcher and state-of-the-art deep-research agents.

*Equal Core Contributors.

✉Corresponding authors. {qiaozile.qzl, yongjiang.yj}@alibaba-inc.com

1 Introduction

The pursuit of artificial general intelligence (AGI) has historically focused on scaling models to acquire vast passive knowledge (Anthropic, 2024; Gemma et al., 2025; Google, 2025a; Meta, 2025; OpenAI, 2025c; Guo et al., 2025a; Yang et al., 2025). However, this knowledge-centric approach may reach a critical limitation: while models can memorize and recall information, they struggle to actively discover, verify, and synthesize new knowledge from external sources—a capability fundamental to human intelligence. This limitation has catalyzed a paradigm shift toward actively autonomous agent systems that emulate human research workflows. Rather than relying solely on pre-trained knowledge, these systems dynamically construct understanding by autonomously decomposing complex problems, orchestrating sophisticated tool use, and synthesizing disparate findings into coherent, evidence-grounded narratives. This emerging class of systems, often referred to as deep research (OpenAI, 2025a), represents a crucial step toward AGI by bridging the gap between passive knowledge repositories and active knowledge constructors.

Recent advances in deep-research systems, exemplified by pioneers like OpenAI's Deep Research (OpenAI, 2025a), Google's Gemini Deep Research (Google, 2025b), Grok DeepSearch (xAI, 2025), and Kimi-Researcher (MoonshotAI, 2025), have demonstrated breakthrough performance on challenging benchmarks including Humanity's Last Exam (HLE) (Phan et al., 2025) and BrowseComp (Wei et al., 2025). The success of these proprietary systems has spurred significant open-source development. Recent open-source efforts including WebThinker (Li et al., 2025c), WebShaper (Tao et al., 2025), and WebSailor (Li et al., 2025a) have shown competitive performance in deep-research tasks. Notably, these open-source implementations have converged on a remarkably similar architectural pattern: a *mono-contextual paradigm* that continuously accumulates all retrieved information and intermediate reasoning steps into a single, ever-expanding context window (Chen et al., 2025; Jin et al., 2025; Li et al., 2025b; Wu et al., 2025). While this linear accumulation strategy appears intuitive and has shown initial success, a deeper analysis reveals that it fundamentally constrains the potential of deep-research agents. Specifically, this prevalent paradigm suffers from two critical limitations that become increasingly severe as research complexity grows: **(1) Cognitive Workspace Suffocation**: The ever-expanding context progressively constrains the model's capacity for deep reasoning, as the fixed context window becomes dominated by accumulated data rather than active thinking space, forcing premature conclusions. **(2) Irreversible Noise Contamination**: Without mechanisms to filter or revise earlier content, irrelevant information and initial errors persist throughout the entire process, diluting signal quality and propagating biases that compound over time. These limitations reveal a paradox: as deep-research agents gather more information to solve complex problems, their mono-contextual architecture becomes increasingly ineffective at processing and reasoning over that very information.

In this work, we introduce **IterResearch**, an *Iterative Deep-Research Paradigm* that reformulates deep research as a Markov Decision Process (MDP) (Bellman, 1957; Puterman, 1990). Unlike the mono-contextual approach that suffers from unbounded state expansion and noise contamination, IterResearch periodically consolidates its findings into a synthesized report and reconstructs its workspace, maintaining both continuity of knowledge and clarity of reasoning at arbitrary depths of research. Specifically, IterResearch operates through discrete rounds where each state contains only essential components: the research question, an evolving report synthesizing all previous findings and current research progress, and the immediate context from the recent tool interaction. This evolving report serves as the agent's central memory—progressively refined through each round as new insights are integrated with existing knowledge. Between rounds, a state transition function preserves this updated report while discarding ephemeral information, ensuring the Markov property while preventing information loss. This periodic synthesis is the core of our paradigm: it not only preserves essential knowledge to guide subsequent reasoning but also maintains a focused cognitive workspace for each phase, effectively preventing both suffocation and noise propagation. Therefore, IterResearch achieves what mono-contextual systems cannot—sustained high-quality reasoning across the entire research process, enabling the agent to pursue arbitrarily complex

investigations through iterative refinement rather than exhaustive single-pass accumulation.

Furthermore, to address the critical bottleneck of data scarcity in training deep-research agents, we develop **WebFrontier**, a *Scalable Data Synthesis Engine* that leverages large language models augmented with diverse external tools to systematically generate high-quality training data for complex research tasks. WebFrontier tackles a fundamental challenge in agentic AI development: how to create high-quality and large-scale training data while maintaining factual accuracy and verifiability. Our approach employs a three-stage iterative workflow—seed generation from diverse corpora, tool-augmented complexity escalation, and rigorous quality control—to produce tasks that effectively span the capability gap between baseline models and their tool-augmented counterparts. The engine's core mechanism involves a self-bootstrapping process where tool-augmented agents progressively refine simple questions into research problems that require multi-source synthesis, cross-domain reasoning, and computational verification. This systematic approach enables the generation of a large-scale dataset that explore varying levels of complexity while ensuring factual grounding. The synthesized data serves as the foundation for training IterResearch through a multi-stage training, enabling the model to acquire both robust tool-use capabilities and sophisticated reasoning skills.

Finally, at inference time, we introduce the **Research-Synthesis Framework**, built upon our IterResearch paradigm. This framework consists of two phases: Parallel Research and Integrative Synthesis. In the Parallel Research phase, multiple Research Agents concurrently solve the target problem following the IterResearch method, with each agent deriving a final report and the predicted answer. Subsequently, in the Integrative Synthesis phase, a single Synthesis Agent integrates these findings to produce a more comprehensive and robust conclusion. By synthesizing from the final reports rather than the entire research trajectories, the Synthesis Agent can process a wider diversity of research paths within a constrained context. This approach effectively leverages test-time scaling, maximizing the benefits of divergent exploration in complex deep research scenarios.

We systematically evaluate WebResearcher on benchmarks spanning diverse domains and task types. Our experiments demonstrate that WebResearcher achieves state-of-the-art performance across 6 challenging benchmarks, even surpassing frontier proprietary systems. On Humanity's Last Exam (HLE), one of the most demanding tests for AI reasoning, WebResearcher-heavy achieves 36.7% accuracy, substantially outperforming all existing systems including DeepSeek-V3.1 (29.8%) and OpenAI Deep Research (26.6%). Similarly, on complex web navigation tasks like BrowseComp-en, our system reaches 51.7%, matching OpenAI's proprietary Deep Research system while exceeding the best open-source alternative by 21.7 percentage points. These results validate that our iterative synthesis paradigm fundamentally addresses the limitations of mono-contextual approaches, enabling sustained high-quality reasoning even in the most complex research scenarios. Furthermore, we demonstrate that the training data generated under the IterResearch paradigm **provides benefits beyond our own system**. When traditional mono-contextual methods are trained with our iterative paradigm data, they show substantial performance improvements. This highlights that our paradigm produces superior training signals that enhance agentic capabilities.

2 IterResearch: An Iterative Deep-Research Paradigm

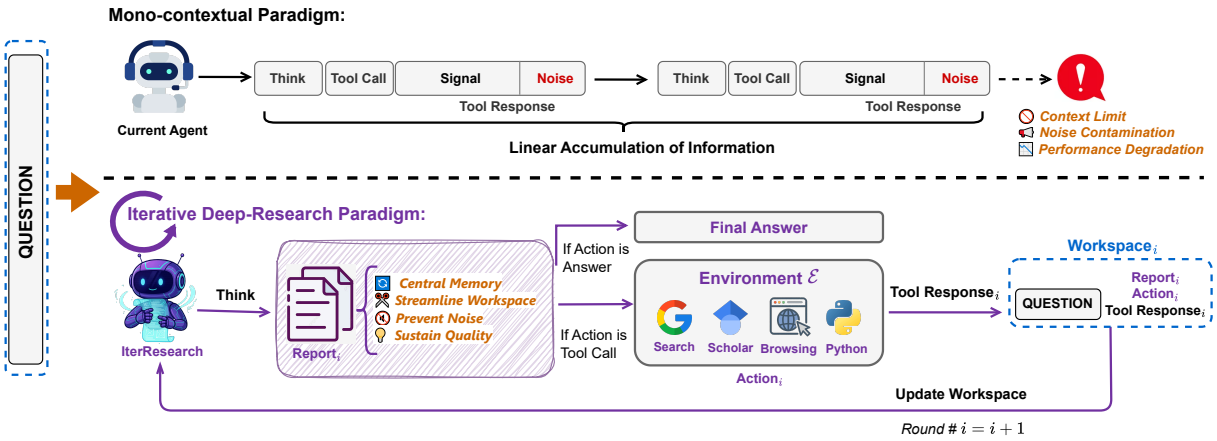


Figure 2: An illustration of our **Iterative Deep-Research Paradigm** in contrast to the prevalent **Mono-contextual Paradigm**. **(Top)** The mono-contextual approach linearly accumulates all information into a single, ever-expanding context, leading to cognitive suffocation and noise contamination. **(Bottom)** Our IterResearch paradigm deconstructs research into discrete rounds. In each round i, the agent operates on a lean, reconstructed `Workspace`. It first **Thinks**, then **synthesizes** new findings into an evolving summary **Report**$_i$, and finally decides on an **Action**$_i$. The crucial step is the reconstruction: the `Workspace` for the next round is rebuilt using only the essential outputs of the previous round (the updated `Report` and `Tool Response`), thus preventing context bloat and enabling sustained reasoning.

Deep-research agents aim to accomplish in minutes what would take human researchers hours. Such long-horizon tasks require navigating heterogeneous evidence sources, coordinating multiple rounds of tool use, and maintaining coherent chains of reasoning across an ever-expanding body of information. Yet, this very complexity directly challenges the mono-contextual, linear accumulation paradigm adopted by current research agents (Li et al., 2025c; Tao et al., 2025; Li et al., 2025a), which is fundamentally constrained by: (1) Cognitive Workspace Suffocation: As the context window fills with accumulated data, the model's capacity for active reasoning diminishes. The fixed context budget becomes dominated by historical information rather than providing space for deep thinking, forcing premature conclusions when the window approaches its limit. (2) Irreversible Noise Contamination: Without mechanisms to filter or revise earlier content, irrelevant information and initial errors persist throughout the entire process. This noise accumulation dilutes signal quality and propagates biases that compound over time, degrading the overall research quality. These limitations create a paradox: as agents gather more information to solve complex problems, their mono-contextual architecture becomes increasingly ineffective.

To overcome these limitations, we propose IterResearch, which reformulates deep research as a Markov Decision Process (MDP) with periodic state reconstruction. Instead of maintaining an ever-expanding context, IterResearch operates through discrete rounds where each state contains only essential components, as illustrated in Figure 2. The key insight of IterResearch is to replace linear accumulation with iterative synthesis and reconstruction. Each research round operates on a focused `Workspace` that maintains clarity while preserving continuity through an evolving report that serves as the agent's central memory. At each round i, the agent's state s_i consists of three components: (1) The original research `Question` q, (2) The evolving `Report`$_{i-1}$ from the previous round (empty for $i = 1$), and (3) The most recent `Action`$_{i-1}$ and its `Tool Response`$_{i-1}$ (if $i > 1$). This compact state representation ensures the Markov property while maintaining all essential information for decision-making.

To implement this iterative paradigm effectively, we define three structured meta-information categories— `Think` , `Report` , and `Action` —that guide the agent's decision in each round:

- `Think` : This component serves as the agent's cognitive scratchpad where it articulates its internal

reasoning process. The agent analyzes the current state (workspace), evaluates the outcome of its previous action, reflects on the research progress, and formulates a plan for its next action. This component ensures the agent's decision-making is transparent and interpretable for current state, and is not directly used in subsequent rounds to prevent clutter.

- `Report`: The centerpiece of our paradigm, this component represents the agent's evolving central memory. Rather than appending raw data, the agent synthesizes new findings with existing knowledge to produce a coherent, high-density summary. This updated report captures all critical information discovered to date and serves as the primary component for constructing the next round's workspace.
- `Action`: The agent's concrete action for the current round, which takes one of two forms:
  - *Tool Call*: A specific command to interact with the external `Environment`, such as invoking a search engine or a code interpreter, to gather new information.
  - *Final Answer*: A terminal action, generated when the agent determines it has sufficient evidence to resolve the initial `Question`. This concludes the research process.

Our IterResearch paradigm fundamentally reimagines deep research as an iterative synthesis process rather than linear accumulation. The complete research unfolds through discrete rounds: starting from just the research question, the agent generates its initial Think-Report-Action triplet; in subsequent rounds, it reconstructs a focused workspace from the question, previous report, and latest tool response, then produces an updated synthesis. This Report synthesis is the cornerstone of our approach—the agent doesn't merely append new findings but actively integrates them with existing knowledge, resolving conflicts and updating conclusions to maintain a coherent, high-density summary that captures all critical discoveries while filtering out noise. The process continues until the agent determines sufficient evidence has been gathered, producing a Final Answer.

This iterative paradigm provides structural advantages that compound over long-horizon research. By maintaining a constant-size workspace regardless of research depth, IterResearch preserves full reasoning capacity throughout the entire process—where mono-contextual systems suffer from diminishing returns as contexts bloat, our approach maintains consistent performance whether conducting ten or hundred rounds of investigation. The periodic synthesis acts as an intelligent filter, preserving signal while eliminating noise, enabling error recovery through report revision, and ensuring monotonic information gain. Through this disciplined state maintenance centered on evolving reports, IterResearch transforms deep research from exhaustive single-pass accumulation into iterative refinement, achieving theoretically unbounded research depth while maintaining both efficiency and quality—capabilities that are fundamentally impossible under the mono-contextual paradigm.

3 A Scalable Data Engine for Advancing Agentic Intelligence

The advancement of agentic intelligence, characterized by capabilities of complex reasoning and autonomous tool use, is fundamentally constrained by the quality and complexity of their training data. To address this limitation, we introduce a scalable data engine designed to synthesize a large-scale, high-quality dataset that systematically probes and extends the capabilities of current models. Our engine leverages a collaborative multi-agent framework organized into a three-stage iterative workflow: (1) seed data generation, (2) iterative complexity escalation, and (3) rigorous quality control. As depicted in Figure 3, this process orchestrates a team of specialized agents to generate progressively more challenging tasks.

3.1 Stage 1: Seed Data Generation

The process initiates with a diverse, multidisciplinary corpus of contemporary documents, including webpages, academic papers, and e-books. A Summary Agent preprocesses this corpus by paraphrasing

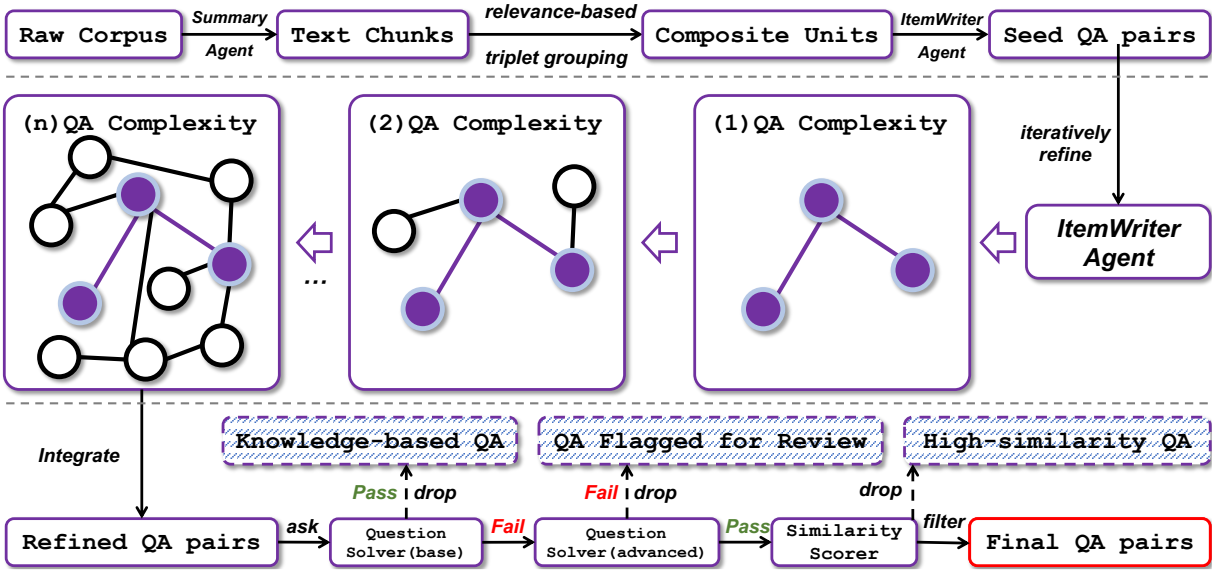


Figure 3: Overview of the three-stage data synthesis workflow, powered by a multi-agent system. The process begins with seed data generation from a curated corpus. It then enters an iterative loop where tool-augmented agents systematically increase task complexity. The workflow concludes with a multi-stage quality control process to calibrate difficulty and ensure factual correctness.

content, removing artifacts (e.g., HTML tags), and distilling the text into information-dense chunks.

To generate initial tasks that require non-trivial reasoning, we form composite units by combinatorially grouping these thematically related chunks. An ItemWriter Agent is then prompted with these composite units to generate seed question-answer (QA) pairs. These initial pairs are designed to require multi-source information synthesis, thereby providing the foundation for the subsequent complexity escalation stage.

3.2 Stage 2: Iterative Complexity Escalation

The core of the data engine is a self-bootstrapping refinement loop orchestrated by the ItemWriter Agent. At this stage, the agent is augmented with a suite of external tools: (i) general web search, (ii) academic literature search, (iii) webpage browswer, and (iv) Python code interpreter. For each seed QA pair, the tool-augmented agent iteratively evolves both the question and answer to increase their cognitive complexity and extend their scope beyond the original context. This iterative evolution is driven by four key operations. Initially, the agent performs knowledge expansion, querying external sources to broaden the question's scope. It then engages in conceptual abstraction, analyzing materials to distill higher-level principles and identify subtle cross-domain relationships. To ensure correctness, factual grounding is achieved through multi-source cross-validation, enhancing the answer's accuracy and depth. Finally, the agent leverages a Python environment for computational formulation, crafting problems that demand quantitative calculation or logical simulation.

This iterative process creates a virtuous cycle where a more sophisticated QA pair generated in one iteration becomes the seed for the next. This enables a controlled and systematic escalation in task complexity.

3.3 Stage 3: Rigorous Quality Control

To ensure the final dataset is of high quality and precisely calibrated to the target difficulty, all generated QA pairs undergo a rigorous validation process managed by specialized agents. First, a QuestionSolver Agent, operating in a baseline mode without access to tools, attempts to answer each question. Any pair answered correctly at this step is deemed too simple for our target complexity level and is filtered out.

Second, the remaining challenging pairs are passed to the same QuestionSolver Agent, now operating in an advanced, tool-augmented mode that mirrors the capabilities of our target model. Pairs that the agent successfully solves in this mode are designated as high-value, complex-reasoning instances and retained for the final dataset. Conversely, any pair that this advanced agent fails to solve is considered intractable or potentially flawed, and is consequently discarded or flagged for expert human review. Throughout this validation pipeline, a Judge Agent automatically assesses the correctness of the solver's output against the ground-truth answer. Concurrently, a SimilarityScorer Agent filters out newly generated pairs that are semantically redundant with existing data, thereby maintaining dataset diversity.

In summary, our data engine is designed to achieve three primary objectives: (1) efficiently generate a large volume of complex tasks situated within the "capability gap" between a baseline model and its tool-augmented counterpart; (2) ensure that all generated tasks maintain high complexity while being factually correct and verifiable; and (3) systematically map and expand the frontiers of reasoning and tool-use for advanced LLM agents.

4 Training and Test-Time Optimization

4.1 Rejection Sampling Fine-Tuning

To train IterResearch, we adopt a rejection sampling fine-tuning (RFT) approach that leverages well-formed trajectories generated by prompting large language models to follow our iterative paradigm's structured format.

Trajectory Generation and Filtering. For each training instance consisting of a research question $q^{(i)}$ and reference answer $a^{(i)}$, we employ prompt the LLMs to generate multiple research trajectories following the IterResearch paradigm. Each trajectory $\tau^{(i)} = \{(s_1^{(i)}, r_1^{(i)}, o_1^{(i)}), ..., (s_{T_i}^{(i)}, r_{T_i}^{(i)}, o_{T_i}^{(i)})\}$ consists of T_i rounds, where $s_j^{(i)}$ represents the state at round j, $r_j^{(i)}$ represents the structured response (Think-Report-Action) and $o_j^{(i)}$ denotes the corresponding tool observation. We apply strict rejection sampling, retaining only trajectories whose final answers exactly match the reference $a^{(i)}$, ensuring the training data embodies both correct reasoning processes and accurate conclusions.

Training Objective. The model learns to generate structured responses conditioned on the iterative research context. Specifically, at each round j, the model must produce $r_j^{(i)}$ given the current state $s_j^{(i)}$. The training objective maximizes the conditional log-likelihood across all accepted trajectories:

$$\mathcal{L}(\theta) = \sum_{i=1}^{K} \sum_{j=1}^{T_i} \log p_\theta \left(r_j^{(i)} \,\middle|\, s_{j-1}^{(i)} \right), \tag{1}$$

where K denotes the number of accepted trajectories and θ represents model parameters. Crucially, this objective enforces the Markov property of our paradigm—each round's generation depends only on the immediate previous state rather than the entire history. During training, we compute gradients only over the model-generated response tokens $r_j^{(i)}$, treating observations $o_j^{(i)}$ as given context. This ensures the model learns to reason and synthesize rather than to predict tool outputs, maintaining a clear separation between the reasoning agent and external tools.

4.2 Reinforcement Learning

To further enhance IterResearch's research capabilities, we employ reinforcement learning to optimize the model's ability to explore diverse reasoning paths while maintaining high-quality synthesis at each round. A key advantage of our iterative paradigm is that each trajectory naturally decomposes into multiple training samples—one for each research round—whereas mono-contextual approaches yield only a single sample per trajectory. Specifically, for each research question $q^{(i)}$ with G rollouts, trajectory

g unfolds over $T_g^{(i)}$ rounds, where each round j produces a training tuple $(s_{g,j}^{(i)}, r_{g,j}^{(i)}, o_{g,j}^{(i)})$ consisting of the state, response, and tool response. This decomposition yields a rich training corpus:

$$\mathcal{C}^{(i)} = \left\{ (s_{g,j}^{(i)}, r_{g,j}^{(i)}) : g \in [1, G], j \in [1, T_g^{(i)}] \right\}, \tag{2}$$

containing $\sum_{g=1}^{G} T_g^{(i)}$ samples per question. Aggregating across all N training questions, our iterative paradigm generates a total corpus:

$$\mathcal{C}_{\text{total}} = \bigcup_{i=1}^{N} \mathcal{C}^{(i)}, \quad |\mathcal{C}_{\text{total}}| = \sum_{i=1}^{N} \sum_{g=1}^{G} T_g^{(i)}. \tag{3}$$

This represents a substantial data amplification compared to mono-contextual approaches which would yield only $N \times G$ samples. However, the variable trajectory lengths introduce a practical challenge: the total sample count varies across batches due to different $T_g^{(i)}$ values, conflicting with distributed training requirements for fixed batch sizes. To address this while preserving data efficiency, we employ *minimal-loss downsampling*, reducing the entire training corpus to the largest multiple of the data parallel (DP) size that does not exceed the original count:

$$|\mathcal{C}_{\text{train}}| = \left\lfloor \frac{|\mathcal{C}_{\text{total}}|}{\text{DP}_{\text{size}}} \right\rfloor \times \text{DP}_{\text{size}}. \tag{4}$$

This approach ensures uniform distribution across devices while minimizing data loss (typically <1%), maintaining distributed training stability.

To optimize IterResearch over these multi-round trajectories, we adopt Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025). We optimize the following objective:

$$\mathcal{J}_{\text{GSPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{Q}, \mathcal{C}_{\text{train}}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{|\mathcal{C}_{\text{train}}|} \sum_{g=1}^{G} \sum_{j=1}^{T_g} \min(\rho_{g,j}(\theta)\hat{A}_{g,j}, \text{clip}(\rho_{g,j}(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_{g,j}) \right] \tag{5}$$

where $\mathcal{Q}$ is the training set, $\hat{A}_{g,j} = \frac{r_{g,j} - \mu_r}{\sigma_r}$ is the normalized advantage, with μ_r and σ_r computed across all (g, j) pairs in $\mathcal{C}_{\text{train}}$, and $\rho_{g,j}(\theta)$ is the importance ratio based on sequence likelihood. Notably, all $\sum_{g=1}^{G} T_g$ rounds from the G trajectories for question q form *one group*, enabling efficient batched training while respecting the variable-length nature of our iterative research process. This differs from traditional GSPO where each trajectory would be treated separately—our approach leverages the natural decomposition of trajectories into rounds, treating each round as an independent training sample while maintaining group-level advantage normalization across all rounds. This design maximizes data utilization and ensures balanced learning across different research depths.

4.3 Research-Synthesis: Harnessing Test-time Scaling with IterResearch

To further unlock the potential of IterResearch, we further investigate test-time scaling. Given that DeepResearch involves multi-round tool calls and intensive reasoning, directly aggregating the context from every complete trajectory is computationally infeasible. Therefore, effective context management during test-time scaling is crucial, enabling the use of minimal context to accurately represent the problem-solving logic of a trajectory.

To address this challenge, we introduce the **Research-Synthesis Framework** as illustrated in Figure 4. The framework consists of two distinct phases: Parallel Research and Integrative Synthesis. The former phase fosters the concurrent exploration of diverse problem-solving approaches, while the latter integrates these disparate perspectives into a single, unified solution.

Parallel Research In Parallel Research phase, we employ n Research Agents to independently solve the target problem. Each agent adheres to the IterResearch paradigm but carves out a unique solution trajectory by invoking distinct tools and generating different lines of reasoning. Ultimately, this phase

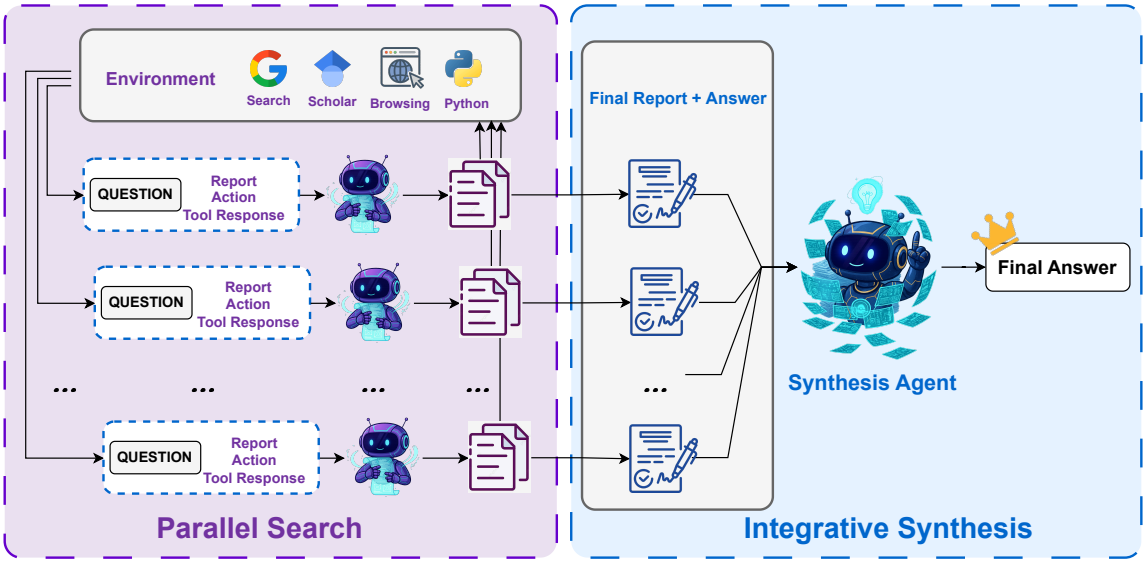


Figure 4: Illustration of Reason-Synthesis Framework

yields a set of final reports and their corresponding predicted answers, one from each agent. This collection can be formally expressed as:

$$\begin{aligned} \mathcal{M} &= \{(\texttt{Final\_Report}_u, \texttt{Answer}_u) : u \in [1, n]\} \\ (\texttt{Final\_Report}_u, \texttt{Answer}_u) &= \text{IterResearch}_u(q) \end{aligned} \tag{6}$$

Integrative Synthesis The Integrative Synthesis phase employs a single Synthesis Agent to consolidate the findings from all Research Agents and produce a final, reasoned conclusion. It takes the complete set of reports and answers as input to generate the final answer, represented as:

$$\texttt{Final\_Answer} = \text{Synthesis}(\mathcal{M}) \tag{7}$$

It is noted that each report from IterResearch concisely encapsulates its entire reasoning path. Consequently, the Synthesis Agent can assess a broader range of solution strategies under a limited context, fully harnessing the power of test-time scaling. In our experiments, we employ Qwen3-235B-A22B as our Synthesis Agent.

5 Experiments

5.1 Experimental Setup

Models and Benchmarks We implement our WebResearcher using Qwen3-30B-A3B (Yang et al., 2025) as the backbone model, considering both model performance and computational efficiency. The complete system integrates our iterative research paradigm (IterResearch) with web-scale training data constructed via WebFrontier. To comprehensively evaluate WebResearcher's capabilities, we conduct extensive experiments on 8 challenging benchmarks:

- **HLE** (Phan et al., 2025) - Humanity's Last Exam is an expert-curated benchmark of 2,500 highly challenging questions spanning a wide range of disciplines, designed to assess frontier-level academic competence. We use the 2,154 text-only questions.
- **GAIA** (Mialon et al., 2023) - A set of 466 real-world task questions evaluating general AI assistants under demanding conditions, emphasizing multi-step reasoning, multimodality, and tool use. We adopt 103 cases from the text-only validation subset (Li et al., 2025c; Wu et al., 2025).
- **BrowseComp-en** (Wei et al., 2025) - A benchmark of 1,266 questions probing agents' ability to locate and integrate hard-to-find, interrelated web information, with an emphasis on persistent browsing and factual reasoning.
- **BrowseComp-zh** (Zhou et al., 2025) - A Chinese web-browsing benchmark with 289 multi-hop questions highlighting retrieval and reasoning challenges specific to the Chinese information ecosystem.
- **Xbench-DeepSearch** (Xbench-Team, 2025) - A specialized deep-search benchmark evaluating agents' end-to-end capabilities in planning, searching, reasoning, and summarization. Featuring expert-curated questions with broad search spaces and deep reasoning requirements, it complements existing benchmarks with substantial Chinese-context coverage.
- **FRAMES** (Krishna et al., 2024) - A comprehensive RAG benchmark with 824 questions testing factuality, retrieval quality, and multi-hop reasoning. It evaluates models' ability to synthesize information from multiple retrieved sources while maintaining factual accuracy and reasoning coherence.

Baselines We compare our WebResearcher against the following baselines:

- **General LLMs with Tools**: Models equipped with external tools for complex reasoning. We evaluate Qwen3-30B-A3B, Qwen3-235B-A22B (Yang et al., 2025), Claude-4-Sonnet (Anthropic, 2025), OpenAI-o3 (OpenAI, 2025b), DeepSeek-V3.1 and DeepSeek-R1 (Guo et al., 2025b), GLM-4.5 (Zeng et al., 2025), and Kimi-K2 (Team et al., 2025).
- **Commercial Deep Research Agents**: We test OpenAI's DeepResearch (OpenAI, 2025a), Gemini Deep Research (Google, 2025b), Perplexity Deep Research (Perplexity, 2025), Grok-DeepResearch (**?**), and Kimi-Researcher (MoonshotAI, 2025). However, as not all of them are fully accessible via API, they were not tested across all benchmarks and experiments.
- **Open-source Deep Research Agents**: We compare our method with recent open-source web/search agents, including WebDancer (Wu et al., 2025), WebSailor (Li et al., 2025a), MiroThinker (MiroMindAI, 2025), WebExplorer (Liu et al., 2025). These represent the current state-of-the-art in open-source web research systems.

Tools Our framework equips agents with four essential tools that enable comprehensive research capabilities, from information discovery to computational analysis. Each tool is designed to handle batch operations efficiently and return structured outputs suitable for iterative research processes.

- **Search** enables web information retrieval via Google search engine. It accepts multiple queries simultaneously and returns top-10 results for each, with each result containing title, snippet, and URL for quick relevance assessment.
- **Scholar** provides access to academic literature through Google Scholar. Similar to Search, it supports batch queries and returns scholarly metadata including authors, venues, and citation counts, enabling efficient academic research.
- **Visit** extracts detailed content from specific web pages with goal-oriented summarization. The agent provides URLs along with extraction goals (e.g., "find experimental results"), and the tool first retrieves full content via Jina (Jina.ai, 2025), then uses Qwen3 (Yang et al., 2025) to produce focused summaries based on the specified goals.
- **Python** executes code in a sandboxed environment for computational tasks. It supports standard libraries for data analysis and visualization, with all outputs explicitly printed to ensure clear result communication.

Evaluation Metrics and Hyper-parameters We adopt the **pass@k** metric (Chen et al., 2021) to evaluate the model's performance. In our experiments, we primarily report **pass@1**, which represents the percentage of problems solved correctly in a single attempt. To determine the correctness of a generated solution, we employ an **LLM-as-a-Judge** approach (Liu et al., 2024; Wang et al., 2024). For all generation tasks, we use nucleus sampling with a **temperature** of 0.6 and a **top-p** of 0.95.

For a dataset with n problems, pass@1 is formally calculated as:

$$\text{pass@1} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(\text{problem } i \text{ is solved}), \tag{8}$$

where $\mathbb{I}(\cdot)$ is the indicator function. For pass@k where $k > 1$, we generate k independent samples per problem and consider it solved if at least one sample is correct.

5.2 Main Results

We present comprehensive evaluation results across 6 challenging benchmarks, categorized into complex goal-oriented web tasks (Table 2) and general web navigation and reasoning challenges (Table 1).

Overall Performance. WebResearcher demonstrates state-of-the-art performance across diverse deep-research benchmarks, significantly outperforming both larger models and existing deep-research systems. WebResearcher achieves remarkable results that surpass open-source deep-research agents and even proprietary deep-research systems. Across 6 challenging benchmarks spanning complex reasoning, web navigation, and long-horizon information-seeking tasks, WebResearcher consistently ranks among the top performers, validating the effectiveness of our iterative synthesis paradigm over the prevalent mono-contextual approach.

General Web Navigation and Reasoning Benchmarks. Table 1 demonstrates WebResearcher's superior performance on web-scale information synthesis tasks, where the advantages of our iterative paradigm become most pronounced. On Humanity's Last Exam (HLE), arguably one of the most challenging benchmarks for frontier AI systems, WebResearcher-heavy achieves 36.7% accuracy—dramatically outperforming all systems including DeepSeek-V3.1 (29.8%), OpenAI Deep Research (26.6%), and Gemini Deep Research (26.9%). This exceptional 6.9 percentage point improvement over the next best system on HLE, which requires deep academic knowledge synthesis across multiple disciplines, validates our paradigm's core strength: maintaining deep reasoning capabilities throughout extended research processes by ensuring each round operates with full cognitive capacity rather than diminishing workspace.

The performance gains are equally impressive on web navigation benchmarks. On BrowseComp-en, WebResearcher-heavy achieves 51.7% accuracy, matching OpenAI's Deep Research (51.5%) while vastly

Table 1: Results on General Web Navigation and Reasoning Benchmarks. † marks the result from the corresponding official reports.

Backbone	Humanity's Last Exam	BrowseComp	BrowseComp-ZH
General LLMs with tools			
Qwen3-30B-A3B	13.2	0.5	13.5
Qwen3-235B-A22B	20.0	2.3	29.4
DeepSeek-R1	24.8†	8.9†	35.7†
Claude-4-Sonnet	20.3†	12.2†	29.1†
Commercial Deep Research Agents			
Perplexity Deep Research	21.1†	-	-
Gemini Deep Research	26.9†	-	-
Kimi-Researcher	26.9†	-	-
OpenAI-o3	20.2†	49.7†	58.1†
OpenAI Deep Research	26.6†	51.5†	-
Open-source Deep Research Agents			
WebSailor-72B	-	12.0†	30.1†
WebShaper-72B	-	-	-
MiroThinker-32B	-	13.0†	17.0†
WebExplorer-8B	-	15.7†	32.0†
Kimi-K2	18.1†	14.1†	28.8†
GLM-4.5	21.2†	26.4†	37.5†
DeepSeek-V3.1	29.8†	30.0†	49.2†
Ours			
WebResearcher-30B-A3B	**28.8**	**37.3**	**45.2**
WebResearcher-30B-A3B-heavy	**36.7**	**51.7**	**56.8**

exceeding all open-source alternatives—DeepSeek-V3.1, the next best open-source system, achieves only 30.0%. This 21.7 percentage point improvement demonstrates the critical importance of our iterative synthesis approach when handling complex web navigation tasks that require maintaining coherent understanding across multiple information sources.

Similarly strong results are observed on the Chinese-language benchmark BrowseComp-zh, where WebResearcher-heavy achieves 56.8%, approaching o3's 58.1% while significantly outperforming DeepSeek-V3.1 (49.2%). These multilingual results highlight that our iterative paradigm effectively handles culturally-diverse information sources through its structured synthesis process—each round's report distills cross-lingual insights into a coherent narrative, preventing the confusion that often arises when mono-contextual systems accumulate mixed-language content without proper consolidation mechanisms.

Complex Goal-Oriented Web Tasks. Table 2 reveals WebResearcher's exceptional capability in handling complex, multi-step reasoning tasks. On GAIA, WebResearcher achieves 72.8% accuracy, surpassing all evaluated systems including Claude-4-Sonnet (68.3%) and OpenAI-o3 (70.5%), with a remarkable 9.7 percentage point improvement over DeepSeek-V3.1 (63.1%). This substantial gain demonstrates the superiority of iterative synthesis when tackling tasks that require sophisticated tool orchestration and cross-domain reasoning. The key advantage of our iterative paradigm becomes evident here: by periodically reconstructing the workspace and synthesizing findings, WebResearcher maintains consistent

Table 2: Results on Complex, Goal-Oriented Web Tasks Benchmarks. † marks the result from the corresponding official reports.

Backbone	GAIA	Xbench-DeepSearch	Frames
General LLMs with tools			
Qwen3-30B-A3B	35.9	32.0	56.4
Qwen3-235B-A22B	45.6	46.0	-
DeepSeek-R1	-	55.0†	82.0†
Claude-4-Sonnet	68.3†	64.6†	80.7†
Commercial Deep Research Agents			
Kimi-Researcher	-	69.0†	78.8†
OpenAI-o3	70.5†	66.7†	84.0†
OpenAI Deep Research	67.0†	-	-
Open-source Deep Research Agents			
WebSailor-72B	-	55.0†	-
WebExplorer-8B	-	53.7†	-
Kimi-K2	57.3†	50.0†	72.0†
GLM-4.5	66.0†	70.0†	78.9†
DeepSeek-V3.1	63.1†	71.2†	83.7†
Ours			
WebResearcher-30B-A3B	**72.8**	**71.0**	**84.8**
WebResearcher-30B-A3B-heavy	**75.7**	**73.0**	**85.1**

reasoning quality throughout extended research processes, whereas mono-contextual systems suffer from progressive degradation due to context bloat.

On Xbench-DeepSearch, our system reaches 71.0%, matching DeepSeek-V3.1 (71.2%) while vastly exceeding other open-source alternatives like WebSailor-72B (55.0%) and Kimi-K2 (50.0%). Similarly impressive results are observed on Frames (84.8%), where WebResearcher outperforms all systems including DeepSeek-V3.1 (83.7%) and OpenAI-o3 (84.0%). These consistent improvements across diverse task types reveal the fundamental advantage of iterative synthesis: by periodically consolidating findings and reconstructing focused workspaces, WebResearcher can pursue complex reasoning chains and adapt search strategies based on synthesized insights—capabilities that mono-contextual systems inherently lack due to their linear accumulation constraints.

6 Analysis

6.1 The Primacy of the Iterative Paradigm

To verify that the performance gains of our model stem from its core design rather than confounding factors, we conducted a targeted ablation study. The objective was to isolate and measure the direct impact of our Iterative Deep-Research Paradigm.

Experimental Setup We designed an ablation variant, herein referred to as Mono-Agent. This agent utilizes the same underlying model architecture as our full agent but is constrained to a linear, non-iterative inference strategy. Specifically, it accumulates all generated information—including thoughts, tool interactions, and observations—into a single, continuously expanding context window, lacking any

mechanism for synthesis or reset. We compare this against two other agents: Mono-Agent + Iter, which represents the Mono-Agent architecture enhanced with our iterative research training data but still using the linear inference strategy, and WebResearcher, our full model employing the iterative paradigm.

Results and Analysis The results, presented in Table 3, clearly demonstrate the paradigm's efficacy. The Mono-Agent + Iter consistently outperforms the base Mono-Agent across all benchmarks: HLE (25.4 vs. 18.7), BC-EN (30.1 vs. 25.4), and BC-ZH (40.4 vs. 34.6). This initial improvement highlights the benefit of our specialized training data.

Table 3: Main results comparing different agents on the HLE, BC-EN, and BC-ZH benchmarks.

Agent	HLE	BC-EN	BC-ZH
WebResearcher	**28.8**	**37.3**	**45.2**
Mono-Agent	18.7	25.4	34.6
Mono-Agent + Iter	25.4	30.1	40.4

However, the most significant finding is the performance gap between the non-iterative Mono-Agent + Iter and our full WebResearcher agent (e.g., 28.8 vs. 25.4 on HLE). This delta isolates the impact of the iterative paradigm itself. The inferior performance of the linear strategy is attributable to two critical failure modes: 1) Contextual Degradation, where the model's attention is saturated with an excess of low-value historical data, impairing its ability to identify salient information; and 2) Irreversible Error Propagation, where early mistakes or noisy observations remain in the context, progressively corrupting subsequent reasoning steps. This is particularly detrimental in long-horizon tasks that require numerous steps.

Conversely, our iterative paradigm directly mitigates these issues. By periodically synthesizing key findings and resetting the contextual workspace, our agent maintains a focused and refined context for each reasoning cycle. This mechanism is fundamental to sustaining high-level cognitive performance. This study provides compelling evidence that the iterative paradigm itself, not merely the training data or base model, is the critical driver of WebResearcher's success in complex, long-horizon research tasks.

6.2 Analysis of Tool-Use Behavior

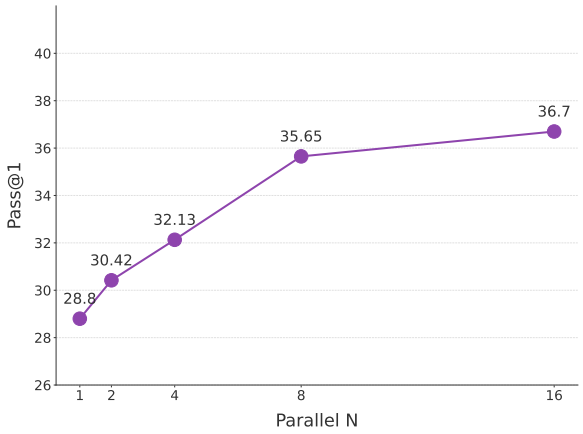
The core strength of IterResearch lies in its iterative paradigm, which facilitates longer and more complex reasoning chains. To substantiate this claim, we conduct an in-depth analysis of its tool-calling behavior across diverse benchmarks, demonstrating that IterResearch exhibits highly adaptive and efficient tool-use strategies tailored to task-specific demands.

Our analysis focuses on the frequency and length of tool invocation sequences—specifically involving Search (web search), Scholar (academic search), Visit (web page access), and Python (code execution)—on the HLE and BrowseComp benchmarks. The tool-use profile of IterResearch shifts dramatically based on the nature of the task.

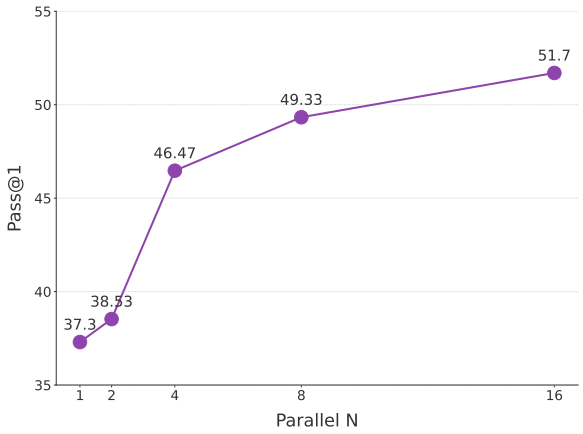
On the HLE benchmark, which primarily contains questions requiring academic and professional knowledge, the agent adopts a focused and concise strategy. The Scholar tool is prominently used, constituting 25.4% of all tool calls, reflecting the need for specialized literature search. The average reasoning chain is short, with tasks being resolved in an average of only 4.7 turns. This indicates efficient, targeted information retrieval for well-defined problems.

In stark contrast, on BrowseComp, where tasks necessitate extensive web navigation and information integration across multiple pages, the agent's behavior highlights its capacity for prolonged and complex reasoning. The Search (56.5%) and Visit (39.7%) tools become paramount, jointly accounting for over 96% of all tool invocations. This strategic shift is mirrored by a significant increase in reasoning complexity: the average number of turns skyrockets to 61.4 per task, with the most complex problems requiring over 200 interaction turns to solve.

This marked divergence in both tool selection and reasoning chain length underscores IterResearch's



(a) Effect of n on HLE



(b) Effect of n on BrowseComp

Figure 5: Effect of n in Reason-Synthesis Framework

sophisticated ability to dynamically adapt its problem-solving approach. It can execute brief, precise actions for knowledge-based queries (HLE) as well as sustain long, exploratory sequences for complex web-based tasks (BrowseComp), validating the effectiveness of its iterative reasoning architecture.

6.3 Analysis on Reasoning Trajectories in Reason-Synthesis Framework

In Section 4.3, we introduced Reason-synthesis Framework to enhance model performance by running n inference trajectories in parallel. To consolidate the findings from all Research Agents, a Synthesis Agent aggregates these reasoning paths to generate the final answer. This section presents a quantitative analysis of the impact of the hyperparameter n—the number of parallel research—on final performance.

Experimental Setup Our analysis is conducted on the HLE benchmark using the IterResearch-30B-A3B model. We systematically vary the number of parallel research, n, evaluating the model's performance (pass@1) for $n \in \{1, 2, 4, 8, 16\}$. The case of $n = 1$ serves as the baseline, representing the model's performance without any Test-Time Scaling.

Results and Insights The experimental results, illustrated in Figure 5, reveal a clear and positive correlation between the number of trajectories (n) and model performance.

As we increase n, there is a consistent and significant improvement in the pass@1 score. The most substantial gains are observed when scaling n from 1 to 8. This result underscores the benefits of the **ensemble effect** inherent in TTS. Each Research Agent explores a unique reasoning path, potentially uncovering distinct facets of the problem or overcoming specific intermediate obstacles. By fusing the final outcomes of these diverse explorations, the Synthesis Agent can produce a more robust and accurate final answer.

As expected, these performance improvements are accompanied by a linear increase in computational cost, as each trajectory is processed independently. Furthermore, the performance gains begin to exhibit **diminishing marginal returns** for $n > 8$, indicating a trade-off between accuracy and computational budget.

Our analysis demonstrates that employing multiple parallel research trajectories within the Reason-Synthesis framework is a highly effective technique for performance enhancement. The number of trajectories, n, serves as a direct and controllable parameter to balance performance gains against computational cost. Based on our findings, a configuration of $n = 8$ offers a compelling trade-off, delivering substantial performance improvements over the baseline while maintaining manageable computational overhead.

7 Related Work

Deep Research The development of autonomous deep-research agents has witnessed significant progress from both proprietary and open-source efforts. These proprietary systems (OpenAI, 2025a; Google, 2025b; xAI, 2025; Anthropic, 2025; Perplexity, 2025; MoonshotAI, 2025) have established benchmarks for deep-research capabilities but remain opaque. In contrast, open-source efforts (Chen et al., 2025; Jin et al., 2025; Li et al., 2025b;c; Tao et al., 2025; Li et al., 2025a; Fang et al., 2025; Su et al., 2025) have advanced the field through transparent architectures and training methodologies. However, these works predominantly adopt the mono-contextual paradigm, accumulating all retrieved information into a single, ever-expanding context. While this linear approach has shown initial success, it fundamentally constrains reasoning capacity as contexts bloat and allows noise contamination to persist throughout the research process. WebResearcher departs from this prevalent paradigm by introducing an iterative synthesis framework that maintains focused cognitive workspaces through periodic consolidation and reconstruction. Our approach draws inspiration from human research workflows (Bellman, 1957; Puterman, 1990), where researchers iteratively refine their understanding through cycles of exploration, synthesis, and focused investigation. Unlike mono-contextual systems that suffer from irreversible degradation, WebResearcher sustains high-quality reasoning at arbitrary research depths through its Markov Decision Process formulation, enabling complex multi-hop reasoning and cross-domain synthesis that existing architectures struggle to achieve.

8 Conclusion

In this paper, we presented WebResearcher, a novel framework that fundamentally rethinks deep-research agents through three key contributions: (1) IterResearch, an iterative paradigm that reformulates deep research as a Markov Decision Process with periodic consolidation, overcoming the context suffocation and noise contamination of mono-contextual approaches; (2) WebFrontier, a scalable data synthesis engine that addresses training data scarcity through tool-augmented complexity escalation; and (3) a Research-Synthesis Framework that enables effective test-time scaling through parallel multi-agent exploration. Extensive experiments across 6 challenging benchmarks demonstrate that WebResearcher achieves state-of-the-art performance, surpassing even frontier proprietary systems. These results validate our core insight that effective deep research requires structured iteration with periodic synthesis rather than unbounded accumulation.

References

Anthropic. Introducing computer use, a new claude 3.5 sonnet, and claude 3.5 haiku. October 2024.

Anthropic. Claude takes research to new places. https://www.anthropic.com/news/research, April 2025.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957.

Guoxin Chen, Minpeng Liao, Peiying Yu, Dingmin Wang, Zile Qiao, Chao Yang, Xin Zhao, and Kai Fan. C-3PO: Compact plug-and-play proxy optimization to achieve human-like retrieval-augmented generation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=hlpwAmQ4wr.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, Shibin Wu, Zhengwei Tao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Towards general agentic intelligence via environment scaling, 2025.

Team Gemma, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Google. Gemini 2.5 pro. https://deepmind.google/technologies/gemini/pro/, April 2025a.

Google. Deep research is now available on gemini 2.5 pro experimental., 2025b. URL https://blog.google/products/gemini/deep-research-gemini-2-5-pro-experimental/.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025b.

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

Jina.ai. Jina, 2025. URL https://jina.ai/.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. *arXiv preprint arXiv:2409.12941*, 2024.

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

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

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *CoRR*, abs/2504.21776, 2025c. doi: 10.48550/ARXIV.2504.21776. URL https://doi.org/10.48550/arXiv.2504.21776.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, et al. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*, 2025.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. Calibrating llm-based evaluator. In Nicoletta Calzolari, Min-Yen Kan, Véronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pp. 2638–2656. ELRA and ICCL, 2024. URL https://aclanthology.org/2024.lrec-main.237.

Team Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. https://ai.meta.com/blog/llama-4-multimodal-intelligence/, April 2025.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

MiroMindAI. Mirothinker, 2025. URL https://github.com/MiroMindAI/MiroThinker.

MoonshotAI. Kimi-researcher, 2025. URL https://moonshotai.github.io/Kimi-Researcher/.

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

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

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

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

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

Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2: 331–434, 1990.

Liangcai Su, Zhen Zhang, Guangyu Li, Zhuo Chen, Chenxi Wang, Maojia Song, Xinyu Wang, Kuan Li, Jialong Wu, Xuanzhong Chen, Zile Qiao, Zhongwang Zhang, Huifeng Yin, Shihao Cai, Runnan Fang, Zhengwei Tao, Wenbiao Yin, et al. Scaling agents via continual pre-training, 2025.

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

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

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. Leave no document behind: Benchmarking long-context llms with extended multi-doc QA. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 5627–5646. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.emnlp-main.322.

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

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

xAI. Grok 3 beta — the age of reasoning agents, 2025. URL https://x.ai/news/grok-3.

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

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

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

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

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