

Deep Researcher with Test-Time Diffusion

Rujun Han^{*1}, Yanfei Chen^{*1}, Zoey CuiZhu², Lesly Miculicich¹, Guan Sun², Yuanjun Bi², Weiming Wen², Hui Wan², Chunfeng Wen², Solène Maître², George Lee¹, Vishy Tirumalashetty², Emily Xue², Zizhao Zhang², Salem Haykal², Burak Gokturk¹, Tomas Pfister¹ and Chen-Yu Lee¹

¹Google Cloud AI Research, ²Google Cloud

Deep research agents, powered by Large Language Models (LLMs), are rapidly advancing; yet, their performance often plateaus when generating complex, long-form research reports using generic test-time scaling algorithms. Drawing inspiration from the iterative nature of human research, which involves cycles of searching, reasoning, and revision, we propose the Test-Time Diffusion Deep Researcher (TTD-DR). This novel framework conceptualizes research report generation as a diffusion process. TTD-DR initiates this process with a preliminary draft, an updatable skeleton that serves as an evolving foundation to guide the research direction. The draft is then iteratively refined through a "denoising" process, which is dynamically informed by a retrieval mechanism that incorporates external information at each step. The core process is further enhanced by a self-evolutionary algorithm applied to each component of the agentic workflow, ensuring the generation of high-quality context for the diffusion process. This draft-centric design makes the report writing process more timely and coherent while reducing information loss during the iterative search process. We demonstrate that our TTD-DR achieves state-of-the-art results on a wide array of benchmarks that require intensive search and multi-hop reasoning, significantly outperforming existing deep research agents.

1. Introduction

Enabled by the recent advanced LLMs, building Deep Research (DR) agents has rapidly gained traction within both research and industry communities. These agents demonstrate remarkable capabilities, including the generation of novel ideas (Hu et al., 2024; Si et al., 2024), effective information gathering through search tools (Jin et al., 2025; Li et al., 2025a), and the execution of analyses or experiments prior to drafting research reports or papers (Yamada et al., 2025; Zheng et al., 2024). Existing DR agents primarily leverage test-time scaling approaches such as Chain-of-Thought (CoT) (Wei et al., 2022), best-of-n sampling (Ichihara et al., 2025), Monte Carlo Tree Search (Świechowski et al., 2022), debate mechanisms (Liang et al., 2023), and self-refinement loops (Madaan et al., 2023). Despite the impressive progress, most popular public DR agents (Alzubi et al., 2025; Researcher, 2025; Roucher et al., 2025) compile these test-time algorithms and various tools without a deliberate design driven by human cognitive behavior in writing, and commonly lack a principled draft, search, and feedback mechanism that empowers human researchers. This indicates a fundamental limitation in current DR agent work and highlights the need for a more cohesive, purpose-built framework for DR agents that imitates or surpasses human research capabilities.



Figure 1 | Our method is inspired by the natural human writing process, which includes planning, drafting, and multiple revisions to the draft.

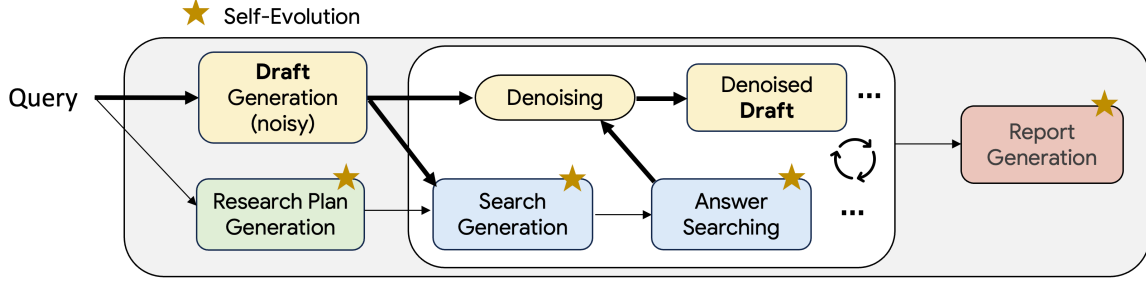


Figure 2 | Illustration of our **Test-Time Diffusion Deep Researcher (TTD-DR)** framework, designed to mimic the iterative nature of human research through a *draft*. A user query initiates both a preliminary draft and a research plan. This evolving draft, along with the research plan, dynamically informs the generation of search questions and subsequent information retrieval to be timely and coherent, while reducing information loss. The retrieved information is then leveraged to **denoise** and refine the initial draft in a continuous feedback loop. The entire workflow is further optimized by a self-evolutionary algorithm to enhance the quality of the research plan, generated questions, answers, and the final report, demonstrating the synergistic power of diffusion and self-evolution in achieving superior research outcomes.

Previous cognitive studies indicate that when human write about complex topics, they do not follow a linear progression, writing from the first word to the last. As Fig. 1 (Chitwood, 2022) illustrates, people typically first establish a high-level *plan*, then *draft* the research report based on the plan, and subsequently engage in multiple *revision* cycles (Flower and Hayes, 1981). Crucially, during the revision phase, writers often seek out literature or search tools to gather supplementary information that refines and strengthens their arguments (Catalano, 2013).

We observe a striking resemblance between this human writing pattern and the sampling process in a *diffusion model* augmented by *retrieval* (Zhang et al., 2023). In this analogy, a trained diffusion model initially generates a noisy draft, and the denoising module, aided by retrieval tools, revises this draft into higher-quality (or higher-resolution) outputs. Inspired by this diffusion sampling paradigm (Shen et al., 2025; Yang et al., 2022), we propose **Test-Time Diffusion (TTD)** for deep research agents. Our framework meticulously models the entire research report generation as an iterative diffusion process, mirroring human cognitive patterns. As vanilla diffusion sampling can be ineffective for generating high quality outputs for complex research tasks, we specifically design our TTD Deep Researcher consisting of two mechanisms as illustrated by Fig. 2 and detailed below.

(a) Denoising with Retrieval (Zhang et al., 2023): An initial research report, drafted primarily from the LLM’s internal knowledge, undergoes iterative refinement. The denoised draft, along with the research plan (Stage 1), guide the downstream research direction. Each denoising step is augmented by targeted retrieval of external information (Stage 2), significantly enhancing accuracy and comprehensiveness. **(b) Self-Evolution** (Lee et al., 2025; Novikov et al., 2025): Beyond the report-level diffusion through a draft, each individual component within the agentic workflow (e.g., plan, question, answer and report generation) undergoes its own optimization process. This encourages the exploration of diverse knowledge, mitigates the information loss for each unit agent throughout the long agentic trajectories, and thus provides more conducive context for report diffusion. The intricate interplay and synergistic combination of these two algorithms are crucial for achieving high quality research outcomes.

Prior work primarily centers on scientific paper writing agents (Chen et al., 2025; Gottweis et al., 2025; Lu et al., 2024; Tang et al., 2025; Yamada et al., 2025), with a specific emphasis on generating

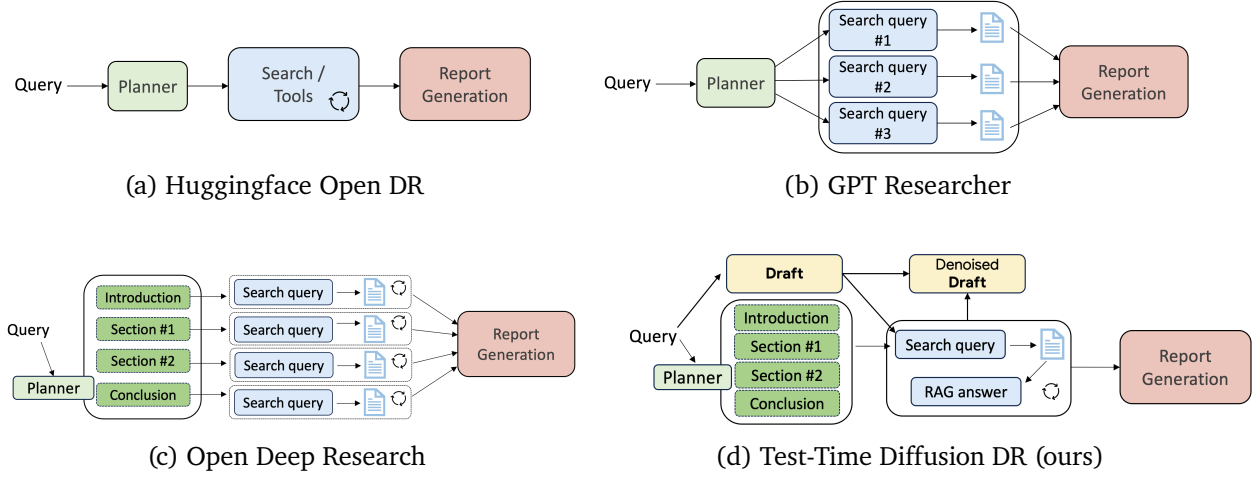


Figure 3 | A comparison of our method with other open-source deep researchers. (a) Huggingface Open DR (Roucher et al., 2025) utilizes a lightweight planner to determine subsequent actions, such as calling search or browse tools, and repeats these actions until an answer is found. (b) GPT Researcher (Researcher, 2025) also employs a lightweight planner to generate and execute multiple search queries in parallel before a generator synthesizes the retrieved documents into a report. (c) Open Deep Research (Research, 2025) uses a planner to outline the final report’s structure and then conducts iterative research for each section individually before combining them. (d) Our **TTD-DR** introduces a draft denoising mechanism. Unlike Open Deep Research, TTD-DR avoids separated searches for each section to maintain global context and uses a RAG-based answer generator to process retrieved documents before saving them for the final report generation.

academic publications. More recently, the scope has broadened to general research agents (Li et al., 2025b; Zheng et al., 2025) designed for broader information-seeking and reasoning use cases. In contrast to these existing efforts, our work introduce a deep research agent engineered for significantly broader applications. Specifically, we develop a research companion capable of generating helpful and comprehensive reports for complex research questions across diverse industry domains, including finance, biomedical, recreation, and technology (Han et al., 2024), similar to deep research products offered by OpenAI (2025), Perplexity (2025) and Grok (2025). Our framework targets search and reasoning-intensive user queries that current state-of-the-art LLMs cannot fully address using their internal knowledge or with conventional search tools. We summarize our key contributions below:

- We propose a **Test-Time Diffusion Deep Researcher (TTD-DR)**, a novel test-time diffusion framework that enables the iterative drafting and revision of research reports, leading to more timely and coherent information integration while reducing information loss throughout the research process.
- We stress test our **TTD-DR** using only search tools that are easily accessible to most agentic systems, eliminating the need to integrate additional proprietary tools (e.g., multimodal, web browsing).
- We establish a rigorous evaluation methodology for deep research agents, employing comprehensive metrics and expert evaluators. Our experiments demonstrate that **TTD-DR** substantially outperforms various leading research agents for tasks either require writing a long and comprehensive research report or need multi-hop search and reasoning to identify concise answers.
- We conduct a comprehensive ablation study and in-depth analysis to elucidate the individual contributions of **TTD-DR**’s components and demonstrate its effectiveness in surpassing leading DR agents.

2. Test-Time Diffusion Deep Researcher (TTD-DR)

Our approach, the Test-Time Diffusion Deep Researcher (TTD-DR), is inspired by the iterative nature of human research, which involves cycles of planning, drafting, searching for information, and revision. We conceptualize the generation of a complex research report as a *diffusion* process where an initial, noisy draft is progressively refined into a high-quality final output. This is achieved through two core mechanisms operating in synergy: (1) Report-Level Refinement via **Denoising with Retrieval**, where the entire report draft evolves, and (2) Component-wise Optimization via **Self-Evolution**, which enhances the quality of each step in the research workflow.

The TTD-DR framework is designed to address the limitations of existing DR agents. As illustrated in Figure 3, many public agents like Huggingface Open DR (Roucher et al., 2025), GPT (Researcher, 2025) Researcher, and Open Deep Research (Alzubi et al., 2025) employ a linear or parallelized process of planning, searching, and generation. This can lead to a loss of global context and miss critical dependencies during the research process. Our draft-centric, iterative approach maintains coherence and provides a dynamic guide for the research direction, mitigating information loss. Proprietary DRs from OpenAI (2025), Perplexity (2025) and Grok (2025) remain largely black box.

2.1. Backbone Deep Research Agent

Fig. 4 illustrates our backbone deep research agent consisting of 3 major stages with several key components for an agentic framework: unit LLM agent, workflows and agent states. We explain them in details below.

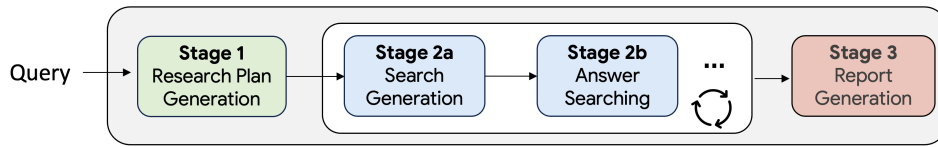


Figure 4 | Our backbone DR agent operates in three stages, as illustrated above. **Stage 1** generates a detailed research plan that outlines the final report’s structure and guides the information search. **Stage 2** iteratively generates search questions (2a) and then uses a RAG-like system to synthesize precise answers from retrieved documents (2b), rather than saving raw data. Finally, **Stage 3** synthesizes all gathered information to produce the final report. Each stage can be individually optimized using a self-evolutionary algorithm detailed in Sec. 2.2.

Stage 1: Research Plan Generation is a dedicated unit LLM agent which generates a structured research plan upon receiving a user query. This plan outlines a list of key areas needed for the final report, serving as an initial scaffold to guide the subsequent information-gathering process. Once a research plan is generated, it will be saved in agent stages and then transferred to its sub-agent.

Stage 2: Iterative Search and Synthesis is a loop workflow nested in its parent sequential workflow. It contains of two sub-agents: Search Question Generation (Stage 2a) formulates a search query based on the research plan, the user query, and the context from previous search iterations (i.e., past questions and answers). Answer Searching (Stage 2b) searches the available sources (such as Google search) to find relevant documents and returns a summarized answer. This loop (Stage 2a → Stage 2b) continues until the research plan is adequately covered or a maximum number of iterations is reached.

Stage 3: Final Report Generation is a unit LLM agent in its parent sequential workflow (Stage 2 → Stage 3), which generates a comprehensive and coherent final report by synthesizing all the

structured information gathered – the plan from Stage 1 and the series of question-answer pairs from Stage 2.

2.2. Component-wise Self-Evolution

The backbone DR agent introduced above determines the overall research directions (Stage 1), and supplies the context and information (Stage 2) for the final report writing (Stage 3). We enhance the performance of each stage’s agents in order to *find* and *preserve* the high quality context. To accomplish this goal, we leverage a self-evolutionary algorithm to improve each stage’s agents. Figure 5 illustrates our proposed algorithm inspired by recent self-evolution work (Lee et al., 2025; Novikov et al., 2025). Here we use the search answer generation as an example, but this algorithm can be applied to all stage agents such as plan generation, search question and even the final report generation to improve their output quality. This algorithm is implemented in a parallel workflow with the following sequential and loop workflows.

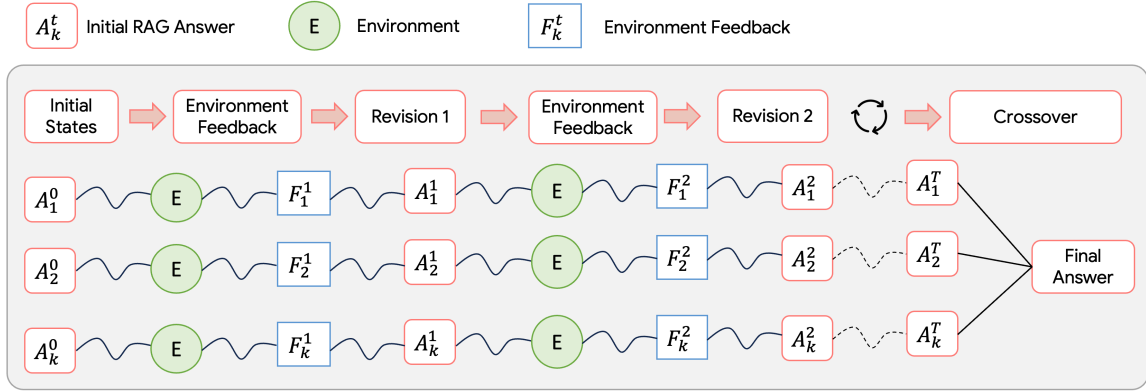


Figure 5 | Illustration of the component-wise **Self-Evolution** applied to Search Answer (Stage 2b in Figure 4). The process starts with multiple variants of initial answers. Each variant then undergoes a self-evolving episode where it first interacts with the environment to obtain a fitness score and feedback. It is then revised based on the feedback. This process repeats until the maximum number of iterations is reached. Finally, multiple revised variants from all episodes are merged to produce the final answer.

1. **Initial States.** The leftmost blocks produce multiple diverse variants of an output (e.g., several possible answers to a search query) conditioned on the output of previous stages. Each block is implemented with a unit LLM Agent, allowing for the sampling of multiple answers using varied parameters (e.g., temperature, top_k) to explore a larger search space. This ideally leads to discovery of potentially more valuable information.
2. **Environmental Feedback.** Each answer variant is assessed by an LLM-as-a-judge, utilizing auto-raters for metrics such as Helpfulness and Comprehensiveness. These raters not only provide fitness scores but also generate textual critiques that help improve the answer.
3. **Revision Step.** With the scores and feedback from the previous step, each variant undergoes a revision step to adapt toward better fitness scores. The “Environmental Feedback” and “Revision” steps repeat until a stopping criterion is met, forming a loop workflow.
4. **Cross-over.** Finally, multiple revised variants are merged into a single, high-quality output. This merging process consolidates the best information from all evolutionary paths, producing superior context for the main report generation process. The merging prompt can be found in Appendix A.5.

While self-evolution improves the quality of each component’s output, this information is not incorporated into the final report until the search process is complete. This delay motivates our second mechanism, Denoising with Retrieval, which integrates the agent’s findings in a more timely and coherent manner to guide the research direction effectively.

2.3. Report-level Denoising with Retrieval

Inspired by the sampling process in diffusion models, where a noisy image is iteratively refined, we prompt an LLM to generate an initial draft report based on the user’s query. This draft serves as a “noisy” starting point, as illustrated in Figure 2. However, as noted in prior work, having a model denoise its own output without external context can lead to slow convergence and sub-optimal results (Shen et al., 2025; Yoon et al., 2025; Zhang et al., 2023). This is particularly true for complex research queries where external information from search tools is essential for improving the draft. This observation motivates us to design a retrieval-augmented denoising process connected directly to our backbone DR workflow introduced in Sec. 2.1.

Specifically, as shown in Algorithm 1, we feed the current draft report into Stage 2a of the backbone DR workflow to inform the generation of the next search query (Line 2). After obtaining a synthesized answer in Stage 2b (Line 4), the new information is used to revise the report draft, either by adding new details or by verifying existing information (Line 6). This process—feeding the denoised report back to generate the next search query—is repeated in a continuous loop. The draft is progressively “denoised” until the search process concludes, at which point a final agent writes the final report based on all historical search answers and revisions (Stage 3).

Algorithm 1 Denoising with Retrieval

Input: $q, \mathcal{M}, \mathcal{P}, \mathcal{R}_0, Q, \mathcal{A}$ ▷ query, all agents, plan, initial noisy draft, history of search questions and answers
1: **for** $t \in \{1, \dots, N\}$ **do** ▷ N : max number of revision steps
2: $Q_t = \mathcal{M}_Q(q, \mathcal{P}, \mathcal{R}_{t-1}, Q, \mathcal{A})$ ▷ generate the next question to address gaps in \mathcal{R}_t
3: $Q_t \rightarrow Q$
4: $A_t = \mathcal{M}_A(Q_t)$ ▷ retrieve external information to provide concrete delta for denoising
5: $A_t \rightarrow \mathcal{A}$
6: $\mathcal{R}_t = \mathcal{M}_R(q, \mathcal{R}_{t-1}, Q, \mathcal{A})$ ▷ remove “noise” (imprecision, incompleteness) from the previous draft
7: **if** `exit_loop` **then**
8: **break** ▷ if `exit_loop` is called, stop revision
9: **end if**
10: **end for**

In summary, this continuous feedback loop, where the evolving draft guides the search and the search refines the draft, ensures the report remains coherent and the research stays on track. The final, “denoised” report is generated after the search process concludes, based on the full history of revisions and retrieved answers. The synergy between the component-wise self-evolution and the report-level diffusion process is critical, allowing TTD-DR to achieve state-of-the-art results.

3. Experimental Setup

To rigorously evaluate our Test-Time Diffusion Deep Researcher (TTD-DR), we established a comprehensive experimental framework. This section details the evaluation metrics, the datasets used for benchmarking, and the specifics of our implementation.

3.1. Evaluation Metrics

Our DR agent is inherently a complicated multi-agent system. Each stage of this system generates long responses that the final agent combine coherently to produce a comprehensive report for users.

Evaluating long-form LLM responses and complex agentic trajectories presents significant challenges due to the vast number of facts to verify, intricate long-term logical dependencies, and the inherent subjectivity of both LLM and human judges (Han et al., 2024; Li et al., 2024; Si et al., 2024). To ensure quality and efficiency of our evaluators, we collect high-quality human judgment annotations, calibrate LLM-as-a-judge calibrated with human preferences, and use the calibrated LLM-as-a-judge as the final evaluator. We provide more details of evaluation metrics below.

- **Helpfulness** and **Comprehensiveness** are the two most commonly used metrics for evaluating long-form LLM responses, particularly for research outputs (Coelho et al., 2025; Lim et al., 2025; Schmidgall et al., 2025). We therefore adopt these two metrics and construct a new side-by-side quality comparison framework based on them. **Helpfulness** is defined by four criteria: 1) satisfying user intent, 2) ease of understanding (fluency and coherence), 3) accuracy, and 4) appropriate language. **Comprehensiveness** is defined as the absence of missing key information. Web search is permitted to better understand the query if needed. Guidelines for determining the Helpfulness and Comprehensiveness levels of a report can be found in Appendix A.1.
- **Side-by-side quality comparison** (also known as pairwise evaluation), a widely adopted method for assessing long-form LLM responses (Han et al., 2024; Li et al., 2024; Liu et al., 2024; Si et al., 2024). Raters were asked to express their preference between two reports (A and B) considering both Helpfulness and Comprehensiveness, using the following scale: 1) **Much Better** If A is both more helpful and more comprehensive than B; 2) **Better** If A is more helpful than B and equally comprehensive as B, or if A is more comprehensive than B and equally helpful as B; 3) **Slightly Better** If A is more helpful but less comprehensive than B; Otherwise, select 4) **About The Same** If none of the above conditions are met. The same logic applies when B is better than A. Our custom-built human annotation interface can be found in Appendix A.2. Each pair is scored twice to compute agreement among human raters. We then deploy an LLM-as-a-judge with the same human instructions to align with human ratings. We discuss more calibration details in the next subsection.
- **Correctness** is used for our multi-hop short-form QA tasks (Phan et al., 2025). For such tasks, we can simply prompt LLMs to compare the long-form answers produced by our agents with the given ground-truths. We follow the standard evaluation prompt¹ to first extract a single answer from LLMs’ responses and then compare the extracted answers with ground-truths.

3.2. LLM-as-a-judge Calibration

Given the absence of ground truth for long-form responses in the LONGFORM RESEARCH and DEEP CONSULT benchmarks, a common practice for scalable evaluation is to leverage LLM-as-a-judge (Coelho et al., 2025; Han et al., 2024; Lim et al., 2025; Schmidgall et al., 2025; Si et al., 2024). However, most prior work in DR agents has not specifically calibrated LLM-as-a-judge’s quality with human raters, raising questions regarding the reliability of auto-evaluators.

In contrast, we align our LLM-as-a-judge with human ratings by comparing 200 reports from our DR agents with those from OpenAI Deep Research. We then utilize an evaluator prompt similar to the one used in our human evaluation for side-by-side comparisons and then calculate the alignment scores between the auto-raters and human raters. Table 3 in Appendix A.3 provides details and results regarding our selection of Gemini-1.5-pro as our LLM-as-a-judge.

For the Correctness auto-rater used to assess the HLE and GAIA dataset, we do not calibrate it with human ratings. This is because an official evaluation prompt exists for these tasks, and we

¹https://scale.com/leaderboard/humanitys_last_exam_text_only

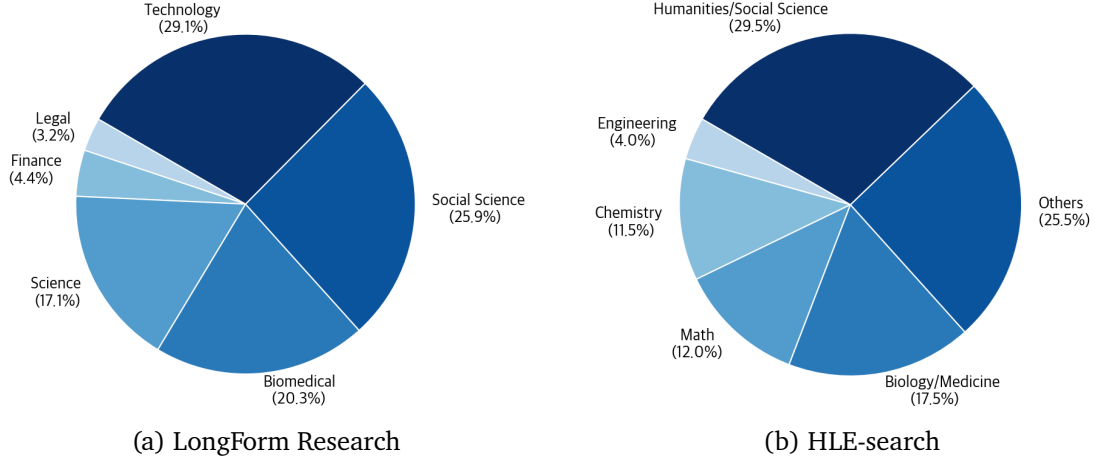


Figure 6 | Query domain distribution of the evaluation sets: LONGFORM RESEARCH (left) and HLE-SEARCH (right), both demonstrating diverse domain coverage.

maintain consistency with the research community by adhering to the original prompt. Furthermore, all answers in these two benchmarks have a straightforward ground-truth answer, simplifying the judgment of LLM response correctness. Therefore, we use Gemini-1.5-pro as the evaluator model without further human calibration for these specific tasks.

3.3. Data

Our chosen benchmarks focus on two broad tasks. 1) Complex queries that require research agents to produce a long-form comprehensive report (LongForm Research and DeepConsult) 2) multi-hop queries that require extensive search and reasoning to answer (HLE and GAIA). Both categories fit into our objective of building a general-purpose, real-world research companion, similar to OPENAI DEEP RESEARCH (OpenAI, 2025) and PERPLEXITY DEEP RESEARCH (Perplexity, 2025). Notably, both tasks may require up to 20 search steps (hops) to fully address user queries, as show in Figure 7a and 12a in the appendix. Other datasets are outside the scope of this work if they do not require extensive search (e.g., only need a few search steps), such as long-form RAG-QA (Han et al., 2024; Stelmakh et al., 2022) and short-form multi-hop QA (Trivedi et al., 2022; Yang et al., 2018). This also applies to datasets not targeting general-purpose research report generation, such as AI-Researcher (Tang et al., 2025). Additionally, we focus on search tool usage, deferring the incorporating of other tools such as browsing and coding to future work.

LongForm Research. To benchmark our DR agent system against other baselines, we first curate a set of licensed real-world queries that demand search and complex reasoning. This dataset best represents our target use cases where users require deep research to create helpful and comprehensive reports. This evaluation set comprises 205 queries covering multiple industry domains, as demonstrated in Figure. 6.

DeepConsult (Lim et al., 2025) is a collection of business and consulting-related prompts designed for deep research. The query set spans a wide range of topics, including marketing, finance, technology trend and business planning.

Humanity’s Last Exam (HLE) (Phan et al., 2025) is a benchmark of 2,500 extremely challenging questions across dozens of subject areas, intended as the final closed-ended benchmark for broad academic capabilities. We focus on the text-only subset, reserving the multi-modality for future research. We name this dataset HLE-FULL.

Table 1 | In this table, we show our **TTD-DR**’s performances against different baseline systems for **LONGFORM RESEARCH**, **DEEPCONSULT**, **HLE** and **GAIA** datasets. Win rate (%) are computed based on OpenAI Deep Research. Correctness is computed as matching between system predicted and reference answers. For Grok DeeperSearch on HLE-FULL, there is no public number provided, and we are not able to scrape the full 2K queries due to research budget and Grok DeeperSearch’s daily scrape limits.

	LONGFORM RESEARCH	DEEPCONSULT	HLE-SEARCH	HLE-FULL	GAIA
	Win Rate	Win Rate	Correctness	Correctness	Correctness
OPENAI DEEP RESEARCH	-	-	29.1	26.6	67.4
PERPLEXITY DEEP RESEARCH	21.8	32.0	14.5	21.1	54.5
GROK DEEPERSEARCH	16.1	16.0	19.3	-	47.9
GPT-RESEARCHER	18.3	9.4	2.0	4.1	37.7
OPEN DEEP SEARCH	2.6	2.2	3.0	0.4	20.9
TTD-DR (ours)	69.1	74.5	33.9	34.3	69.1

HLE-search. A significant number of queries in the HLE dataset do not require extensive searching to resolve. To better benchmark our target use cases of search with reasoning, we identify queries from HLE that demand the most search capabilities. Specifically, we prompt the Gemini-1.5-pro model to categorize all queries into either [a] pure reasoning and [b] requiring search. The prompt used can be found in the Appendix A.4. Finally, we randomly sample 200 queries from categories [b]. As shown in Table 2, the LLM’s own performances on this curated subset is considerably lower compared with the full set. Its question domain distribution can also be found in Figure 6. Therefore, we believe HLE-SEARCH serves as a more suitable benchmark for our research focus.

GAIA (Mialon et al., 2023) is another public benchmark that evaluates AI on real-world questions, encompassing questions across three levels of difficulty. Successful completion of these tasks requires abilities such as reasoning, multi-modal fluency, web browsing, and tool-use proficiency. We use the evaluation set to compare with other baselines.

3.4. Implementation Details

Agentic Framework. To implement our **TTD-DR**, we require a modular and easily extensible agent system capable of leveraging leading LLMs, such as Gemini-2.5-pro, to seamlessly orchestrate workflows, call tools, and execute tasks. Google Agent Development Kit (ADK)² is a recently released agent development platform that satisfies all these requirements. All components described in Sec. 2 can be easily implemented with ADK. We thus chose to build our deep researcher based on ADK.

We fix maximum denoising with retrieval steps to 20. Other hyper-parameters for SELF-EVOLUTION algorithm can be found in Appendix A.6. We use grounding with Google search³ to implement the RAG system in Stage 2b.

3.5. Compared Systems

We compare our RA systems with the leading RA agents in the market: OPENAI DEEP RESEARCH (OpenAI, 2025), PERPLEXITY DEEP RESEARCH (Perplexity, 2025), GROK DEEPSEARCH (Grok, 2025), OPEN DEEP SEARCH (Alzubi et al., 2025) and GPT-RESEARCHER (Researcher, 2025). For DR agents not supported by an API, we manually scraped and saved their raw outputs.

²<https://google.github.io/adk-docs/>

³<https://cloud.google.com/vertex-ai/generative-ai/docs/grounding/overview>

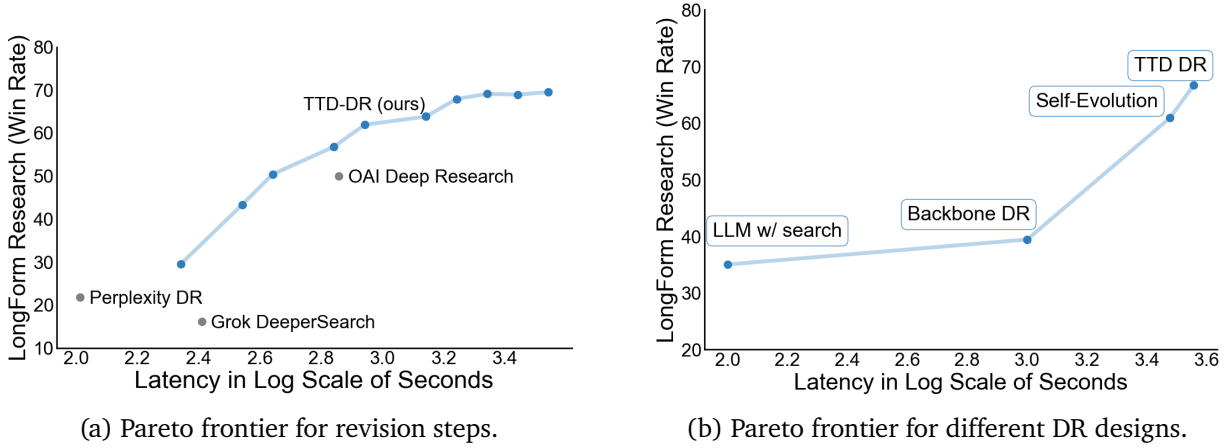


Figure 7 | Pareto frontier between DR agent performances and latency for LONGFORM RESEARCH. **Left:** the dots from left to right represent adding more search/revision steps up to 20, which shows with similar latency, we achieve better or on-par compared with other DR agents. **Right:** the dots from left to right represent 1) GEMINI-2.5-PRO w/ SEARCH TOOL, 2) BACKBONE DR AGENT, 3) + SELF-EVOLUTION and 4) + DIFFUSION WITH RETRIEVAL, which shows our final algorithm is most efficient in terms of test-time scaling (steepest slope).

For ablation study, we compare with baseline LLMs Gemini-2.5-pro and Gemini-2.5-flash, along with their variants that include a simple search tool (simple RAG). For our DR Agent, we compare the following. 1) BACKBONE DR AGENT is our backbone DR Agent without any test-time scaling algorithms. 2) + SELF-EVOLUTION and 3) + DENOISING WITH RETRIEVAL are two DR agent variants enhanced by our proposed test-time scaling algorithms. Our DR agents use Gemini-2.5-pro as the base model. All other baselines agents use their default LLMs (e.g. o3 for OpenAI DR).

4. Results and Analysis

4.1. Main Results

Table 1 presents the performance comparisons between our **TTD-DR** and other DR systems. Our **TTD-DR** consistently achieves superior results across all benchmarks. Specifically, when compared to OPENAI DEEP RESEARCH, our method achieves 69.1% and 74.5% win rate in side-by-side comparisons for the two *long-form* research report generation tasks. Additionally, it outperforms OPENAI DEEP RESEARCH by 4.8%, 7.7% and 1.7% on the three extensive research datasets with *short-form* ground-truth answers. Figure 8 further illustrates the Helpfulness and Comprehensiveness auto-rater scores for the two *long-form* research tasks, where our **TTD-DR** also surpasses OPENAI DEEP RESEARCH, particularly for the LONGFORM RESEARCH dataset.

Table 2 shows the ablation study for our DR agents. It's evident that even the most advanced LLMs with strong reasoning capabilities, such as GEMINI-2.5-FLASH and GEMINI-2.5-PRO, perform poorly without any search tools. For instance, on the curated HLE-SEARCH dataset, GEMINI-2.5-PRO, despite showing relatively good results on the full HLE set (20.9%), achieves only 8.6% accuracy. The performance of both base LLMs significantly improves when augmented with search tools, though their results remain considerably lower than OPENAI DEEP RESEARCH.

Now, examining the three agentic DR agents, the basic DR agent shows significant improvement over LLMs with search tool but still underperforms OPENAI DEEP RESEARCH. With the addition of the proposed SELF-EVOLUTION algorithm, we observe that for LONGFORM RESEARCH and

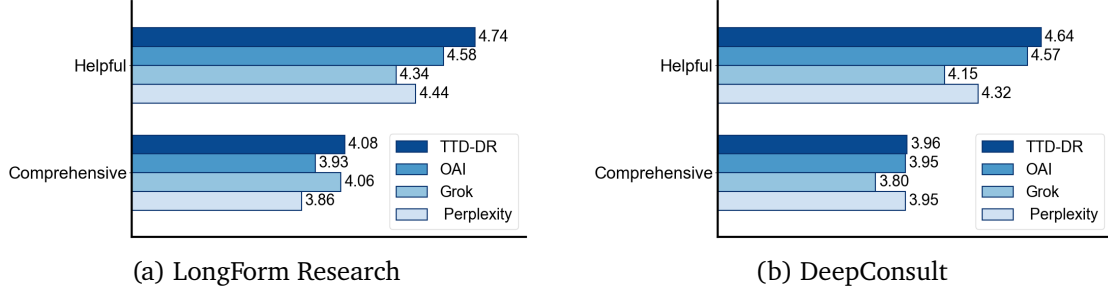


Figure 8 | Single-sided evaluation metrics comparisons between OpenAI Deep Research and our **TTD-DR** for **LONGFORM RESEARCH** (left) and **DEEPCONSULT** (right) benchmarks. **TTD-DR**'s reports tends to be more helpful and comprehensive than other DR agents.

Table 2 | In this Table, we show the ablation study of our DR Agent's performances across all benchmark datasets. All Win rate (%) are computed against OpenAI Deep Research. Correctness (%) uses LLM-as-a-judge with the standard evaluation prompt.

	LONGFORM RESEARCH	DEEPCONSULT	HLE-SEARCH	HLE-FULL	GAIA
	Win Rate	Win Rate	Correctness	Correctness	Correctness
OPENAI DEEP RESEARCH	-	-	29.1	26.6	67.4
LLM w/o agentic workflow					
GEMINI-2.5-FLASH	21.0	16.7	2.8	11.6	31.5
GEMINI-2.5-FLASH W/ SEARCH TOOL	27.8	17.6	14.6	14.6	57.6
GEMINI-2.5-PRO	31.0	17.6	8.6	20.9	57.0
GEMINI-2.5-PRO W/ SEARCH TOOL	35.0	19.6	20.0	21.6	61.8
Test-Time Diffusion Deep Researcher (ours)					
BACKBONE DR AGENT	39.4	24.5	26.8	28.6	61.8
+ SELF-EVOLUTION	60.9	59.8	30.6	29.4	63.0
+ DIFFUSION WITH RETRIEVAL	69.1	74.5	33.9	34.3	69.1

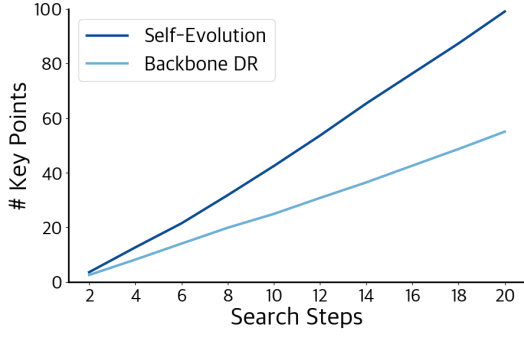
DEEPCONSULT, our system outperforms **OPENAI DEEP RESEARCH** with 60.9% and 59.8% win rates, respectively. The Correctness scores on the two HLE datasets also show an improvement of 1.5% and 2.8% against OpenAI DR, respectively, although we still underperform on **GAIA** by 4.4%. Finally, incorporating **DIFFUSION WITH RETRIEVAL** leads to substantial gains over **OPENAI DEEP RESEARCH** across all benchmarks.

Furthermore, we plot the Pareto frontier of our systems to study the trade-off between latency and performances. In Figure 7b, the x-axis represents the \log_{10} of seconds. The left y-axis shows our **TTD-DR**'s win rate over OpenAI DR on **LONGFORM RESEARCH**. The data points, from left to right, represent **GEMINI-2.5-PRO W/ SEARCH TOOL**, **DR-AGENT-BASE**, **+ SELF-EVOLUTION** and **+ DIFFUSION WITH RETRIEVAL** with increasing latency. The convex shape, particularly the upward trending slope of the last two points, indicates that our two proposed algorithms provide more performance gains per unit increase in latency. This demonstrates that both denoising with retrieval and self-evolution are efficient algorithms for test-time scaling.

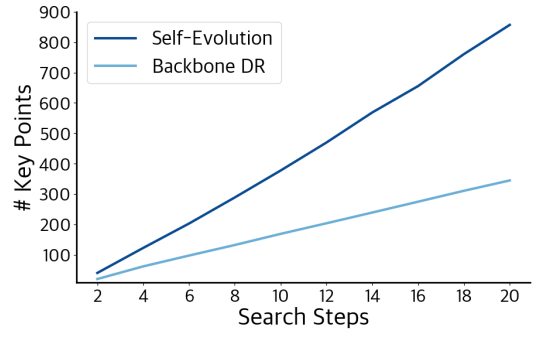
4.2. Analysis

This section provides a deeper understanding of how our two proposed methods contribute to the improvements in DR agents.

Improvement of self-evolution over backbone DR. Figure 9 shows the cumulative complexity comparisons for search queries and answers on **DEEPCONSULT**. Complexity is measured by key



(a) Query complexity comparison.



(b) Answer complexity comparison.

Figure 9 | Stage 2 generated search question (left) and answer (right) complexity by number of key points extracted by LLM using Prompt A.7 and A.8 in the appendix. Self-evolution encourages both search question and answer diversity, which enhance the richness of information available, and thus explains the final quality improvements.

points extracted by an LLM (Gemini-2.5-pro). We observe that self-evolution significantly increases the complexity of the search process, which enriches the information gathered and, consequently, lead to better final report quality.

Our final diffusion algorithm allows for the revision and saving of intermediate reports, enabling us to assess the step-by-step report quality, as illustrated by Figure 7a. As we increase computing resources by adding more search and revision steps, we achieve increasingly significant gains against OpenAI Deep Research. Results for HLE-SEARCH can be found in Appendix A.11. We next aim to understand the contributions of the denoising with retrieval algorithm to these improvements, building upon the self-evolution algorithm.

Improvement of denoising with retrieval over self-evolution. Figure 10a displays the cumulative search query novelty comparisons on DEEPCONSULT. Novelty is measured by the percentage of cumulative new points generated (extracted by Gemini-2.5-pro using Prompt A.9). We can observe that denoising with retrieval increases query novelty by more than **12 percentage points** throughout the search and revision process by feeding the revised report to guide the exploration of new queries. In Figure 10b, we present the report attribution in answers (computed using Gemini-2.5-pro with Prompt A.10) during early search and revision steps. Notably, at Step 9, denoising w/ retrieval already incorporates **51.2% of the final report information**, and outperforms self-evolution (with 20 search steps) by **4.2% in win ratio** (last point in Figure 10c). These results indicate that denoising with retrieval effectively leverage information in early stages, leading to timely preservation of knowledge when agents are learning most efficiently, as shown in Figure 7a.

5. Related Work

We review related work that motivates our deep research agents.

Test-time compute scaling. Baek et al. (2024); Lu et al. (2024); Zheng et al. (2024) are earlier efforts to build research assistant/scientist agents with search tools and iterative refinement algorithms during test time. More recently, Gottweis et al. (2025) proposes an AI Co-scientist agent for biomedical research integrating test-time algorithms such as debates mechanism to generate novel ideas, tournaments to compare and rank research hypothesis and self-critique to refine research proposals. Schmidgall et al. (2025) builds an end-to-end scientific paper writing agent with self-reflection at

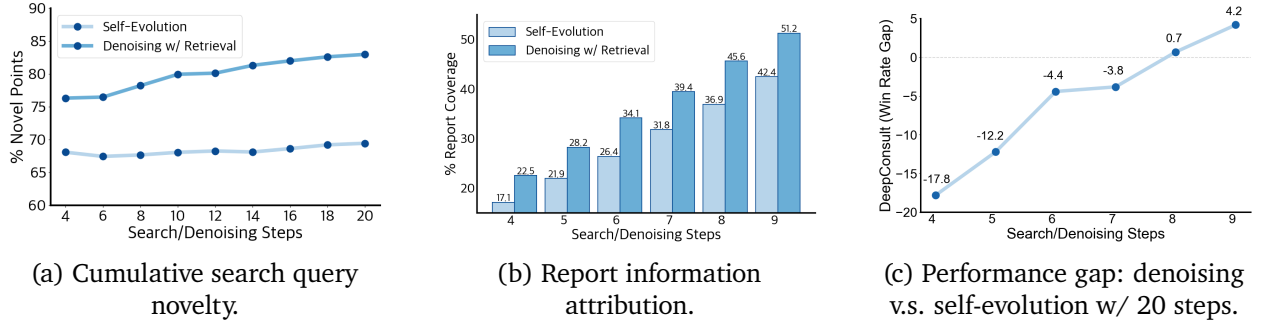


Figure 10 | Comparisons between denoising with retrieval and self-evolution algorithms. **(a)**: percentage of cumulative novel points (Prompt A.9) in Stage 2 generated search queries, which shows denoising with retrieval algorithm guides the generation of more unexplored search queries. **(b)**: cumulative information attribution of the final report in Stage 2 search answers (Prompt A.10), which demonstrates that our final method incorporates information timely in early search stages. **(c)** shows the performance gap between early steps of denoising v.s. self-evolution with full 20 search steps. With only 9 steps, denoising w/ retrieval already incorporates 51.2% of the final report information, and outperforms self-evolution with 20 steps by 4.2% per win ratio.

each stage of their agent workflow. Notably, they enable a co-pilot mode where a human can step in and provide feedback, which is shown to improve overall paper quality. Yamada et al. (2025) designs a machine learning research agent by incorporating a tree-search inference algorithm that is able to write a full research paper accepted by ICLR workshop. Tang et al. (2025) proposes a multi-agent system that is able to review literature, generate new ideas, invent new algorithms, conduct experiments and draft a publication-ready paper. Similarly, DeerFlow (2025) leverages a multi-agent system with planner, coordinator, researcher and reporter to produce comprehensive responses to general user queries.

Amongst test-time algorithms, self-evolving (Lee et al., 2025; Novikov et al., 2025; Qiu et al., 2025) emerges recently as a popular framework to design various agentic systems including DR. Our **self-evolution** algorithm shares common spirit with this method, particularly in its capability to conduct multiple self-critique and self-refinements. However, **TTD-DR** differs from self-evolving in that 1) our framework is fundamentally driven by human cognitive behavior, and we draw the commonality between retrieval augmented diffusion process and human writing process to develop our test-time diffusion DR; 2) Self-evolution improves individual agents to provide high quality contextual information to assist the main denoising algorithm. Both human cognitive behavior and the interplay of self-evolution and denoising with retrieval are not explicitly modeled in prior work.

Agent Tuning. A few recent works explored improving deep research agent via training. Earlier work focuses on building an agentic RAG system that is able to conduct deep search and reasoning. Guan et al. (2024) proposes a multitask learning objective with both component-wise SFT data and model feedback to jointly train each module in its agentic RAG system. Jin et al. (2025) converts search actions and LLM final responses into a single sequence input, and train the RAG system end-to-end with final response reward. More recently, Li et al. (2025b), Zheng et al. (2025), Shi et al. (2025), and Kimi-Researcher (2025) leverage reinforcement learning to training a research assistant agent that is able to leverage search and browsing tools to collect information and write reports. In our work, we focus on test-time compute, and leave agent tuning for future work.

LLM diffusion models. Traditional LLM training paradigm leverages autoregressive objective to train models and sample outputs. LLM Diffusion models attempt to improve the scalability of state-of-the-art LLMs by breaking the assumption of sampling from first to the last tokens. LLM diffusion

models are trained to first generate a complete "noisy" draft, and they iteratively denoise multiple tokens into a full high quality draft (Gemini, 2025; Nie et al., 2025; Yang et al., 2022). Due to highly parallelizable generation processing, this line of work has the potential to achieve higher efficiency while preserving quality. Our work is inspired by LLM Diffusion models by introducing the denoising mechanism during test-time report writing, but differ from them in that we do not train our agents; instead we assume LLM agents are well crafted to perform denoising tasks.

6. Conclusions

The Deep Researcher with Test-Time Diffusion (TTD-DR) agent is a novel framework for generating research reports, inspired by the iterative nature of human research. This agent addresses the limitations of existing DR agents by conceptualizing report generation as a *diffusion process*. TTD-DR initiates with a preliminary draft, an updatable skeleton that guides the research direction. This draft is then refined iteratively through a “denoising”, dynamically informed by a retrieval mechanism that incorporates external information at each step. The core process is further enhanced by a self-evolutionary algorithm applied to each component of the agentic workflow, ensuring the generation of high-quality context for the diffusion process.

The TTD-DR framework achieves state-of-the-art results across various benchmarks requiring intensive search and multi-hop reasoning, significantly outperforming existing DR agents. It demonstrates superior performance in generating comprehensive long-form research reports and identifying concise answers for multi-hop search and reasoning tasks. The framework’s draft-centric design guides the report writing process to be more timely and coherent while reducing information loss during the iterative search process.

Limitations

While TTD-DR shows significant advancements, the current work primarily focuses on search tool usage and does not incorporate other tools such as browse and coding. Future work could explore integrating these additional tools to further enhance the DR agents’ performance and broaden their application. Additionally, agent tuning for improving deep research agents is left for future work, as the current focus is on test-time scaling.

References

- S. Alzubi, C. Brooks, P. Chiniya, E. Contente, C. von Gerlach, L. Irwin, Y. Jiang, A. Kaz, W. Nguyen, S. Oh, H. Tyagi, and P. Viswanath. Open deep search: Democratizing search with open-source reasoning agents. 03 2025. URL <https://arxiv.org/abs/2503.20201>.
- J. Baek, S. K. Jauhar, S. Cucerzan, and S. J. Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. April 2024.
- A. Catalano. Patterns of graduate students’ information seeking behavior: a meta-synthesis of the literature", journal of documentation. *Patterns of graduate students’ information seeking behavior: a meta-synthesis of the literature*, 69(2):243–274, 2013. URL <https://doi.org/10.1108/00220411311300066>.
- Q. Chen, M. Yang, L. Qin, J. Liu, Z. Yan, J. Guan, D. Peng, Y. Ji, H. Li, M. Hu, Y. Zhang, Y. Liang, Y. Zhou, J. Wang, Z. Chen, and W. Che. Ai4research: A survey of artificial intelligence for scientific research. 07 2025. URL <https://arxiv.org/pdf/2507.01903>.

- M. S. Chitwood. Do you know the steps of the writing process?, 2022. URL <https://melanieschitwood.com/do-you-know-the-steps-of-the-writing-process/>.
- J. Coelho, J. Ning, J. He, K. Mao, A. Paladugu, P. Setlur, J. Jin, J. Callan, J. Magalhães, B. Martins, and C. Xiong. Deepresearchgym: A free, transparent, and reproducible evaluation sandbox for deep research. 05 2025. doi: 10.48550/arXiv.2505.19253.
- DeerFlow. Deerflow, 2025. URL <https://github.com/bytedance/deer-flow>.
- L. Flower and J. R. Hayes. A cognitive process theory of writing. *College Composition and Communication*, 32(4):365–387, 1981. ISSN 0010096X. URL <http://www.jstor.org/stable/356600>.
- Gemini. Gemini diffusion, 2025. URL <https://deepmind.google/models/gemini-diffusion/>.
- J. Gottweis, W.-H. Weng, A. Daryin, T. Tu, A. Palepu, P. Sirkovic, A. Myaskovsky, F. Weissenberger, K. Rong, R. Tanno, K. Saab, D. Popovici, J. Blum, F. Zhang, K. Chou, A. Hassidim, B. Gokturk, A. Vahdat, P. Kohli, and V. Natarajan. Towards an ai co-scientist. 02 2025. doi: 10.48550/arXiv.2502.18864.
- Grok. Grok, 2025. URL <https://grok.com/>.
- J. Guan, W. Wu, Z. Wen, P. Xu, H. Wang, and M. Huang. Amor: A recipe for building adaptable modular knowledge agents through process feedback. 2024. URL <https://arxiv.org/abs/2402.01469>.
- R. Han, Y. Zhang, P. Qi, Y. Xu, J. Wang, L. Liu, W. Y. Wang, B. Min, and V. Castelli. RAG-QA arena: Evaluating domain robustness for long-form retrieval augmented question answering. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4354–4374, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.249. URL <https://aclanthology.org/2024.emnlp-main.249/>.
- X. Hu, H. Fu, J. Wang, Y. Wang, Z. Li, R. Xu, Y. Lu, Y. Jin, L. Pan, and Z. Lan. Nova: An iterative planning and search approach to enhance novelty and diversity of llm generated ideas. 2024. URL <https://arxiv.org/abs/2410.14255>.
- Y. Ichihara, Y. Jinnai, T. Morimura, K. Abe, K. Ariu, M. Sakamoto, and E. Uchibe. Evaluation of best-of-n sampling strategies for language model alignment. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL <https://openreview.net/forum?id=H4S4ETc8c9>.
- B. Jin, H. Zeng, Z. Yue, D. Wang, H. Zamani, and J. Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.
- Kimi-Researcher. Kimi-researcher end-to-end rl training for emerging agentic capabilities, 2025. URL <https://moonshotai.github.io/Kimi-Researcher/>.
- K.-H. Lee, I. Fischer, Y.-H. Wu, S. B. Dave Marwood, D. Schuurmans, and X. Chen. Evolving deeper llm thinking. 2025. URL <https://arxiv.org/abs/2501.09891>.
- D. Li, B. Jiang, L. Huang, A. Beigi, C. Zhao, Z. Tan, A. Bhattacharjee, Y. Jiang, C. Chen, T. Wu, K. Shu, L. Cheng, and H. Liu. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv: 2411.16594*, 2024.
- X. Li, G. Dong, J. Jin, Y. Zhang, Y. Zhou, Y. Zhu, P. Zhang, and Z. Dou. Search-o1: Agentic search-enhanced large reasoning models. *CoRR*, abs/2501.05366, 2025a. doi: 10.48550/ARXIV.2501.05366. URL <https://doi.org/10.48550/arXiv.2501.05366>.

- X. Li, J. Jin, G. Dong, H. Qian, Y. Zhu, Y. Wu, J.-R. Wen, and Z. Dou. Webthinker: Empowering large reasoning models with deep research capability. 2025b. URL <https://arxiv.org/abs/2504.21776>.
- T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, Z. Tu, and S. Shi. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.
- A. Lim, S. Jain, and V. Seng. Deepconsult: A deep research benchmark for consulting / business queries, 2025. URL <https://github.com/Su-Sea/ydc-deep-research-evals>.
- Y. Liu, H. Zhou, Z. Guo, E. Shareghi, I. Vulic, A. Korhonen, and N. Collier. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *arXiv preprint arXiv:2403.16950*, 2024.
- C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.
- A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhunoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck, A. Yazdanbakhsh, and P. Clark. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=S37h0erQLB>.
- G. Mialon, C. Fourier, C. Swift, T. Wolf, Y. LeCun, and T. Scialom. Gaia: a benchmark for general ai assistants. 11 2023. URL <https://arxiv.org/abs/2311.12983>.
- S. Nie, F. Z. 1, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li. Large language diffusion models. 2025. URL <https://arxiv.org/abs/2502.09992>.
- A. Novikov, N. Vu, M. Eisenberger, E. Dupont, P.-S. Huang, A. Z. Wagner, S. Shirobokov, B. Kozlovskii, F. J. R. Ruiz, M. P. K. Abbas Mehrabian, A. See, S. Chaudhuri, G. Holland, A. Davies, S. Nowozin, P. Kohli, and M. Balog. Alphaevolve: A coding agent for scientific and algorithmic discovery. 2025. URL <https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/alphaevolve-a-gemini-powered-coding-agent-for-designing-advanced-algorithms/AlphaEvolve.pdf>.
- OpenAI. Introducing deep research, 2025. URL <https://openai.com/index/introducing-deep-research/>.
- Perplexity. Introducing perplexity deep research, 2025. URL <https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research>.
- L. Phan, A. Gatti, Z. Han, N. Li, J. Hu, H. Zhang, C. B. C. Zhang, M. Shaaban, J. Ling, S. Shi, M. Choi, A. Agrawal, A. Chopra, A. Khoja, R. Kim, R. Ren, J. Hausenloy, O. Zhang, M. Mazeika, T. Nguyen, D. Anderson, I. A. Shah, M. Doroshenko, A. C. Stokes, M. Mahmood, J. Lee, O. Pokutnyi, O. Iskra, J. P. Wang, R. Gerbicz, J.-C. Levin, S. Popov, F. Feng, S. Y. Feng, H. Zhao, M. Yu, V. Gangal, C. Zou, Z. Wang, M. Kazakov, G. Galgon, J. Schmitt, A. Sanchez, Y. Lee, W. Yeadon, S. Sauers, M. Roth, C. Agu, S. Riis, F. Giska, S. Utpala, A. Cheatom, Z. Giboney, G. M. Goshu, S.-J. Crowson, M. M. Naiya, N. Burns, L. Finke, Z. Cheng, H. Park, F. Fournier-Facio, J. Zampese, J. Wydallis, J. B. Wydallis, R. G. Hoerr, M. Nandor, T. Gehringer, J. Cai, B. McCarty, J. Nam, E. Taylor, J. Jin, G. A. Loume, H. Cao, A. C. Garretson, D. Sileo, Q. Ren, D. Cojoc, P. Arkhipov, U. Qazi, A. Bacho, L. Li, S. Motwani, C. S. de Witt, A. Kopylov, J. Veith, E. Singer, P. Rissone, J. Jin, J. W. L. Shi, C. G. Willcocks, A. Prabhu, L. Tang, K. Zhou, E. de Oliveira Santos, A. P. Maksimov, E. Vendrow, K. Zenitani, J. Robinson, A. Mikov, J. Guillod, Y. Li, B. Pageler, J. Vendrow, V. Kuchkin, P. Marion, D. Efremov, J. Lynch,

K. Liang, A. Gritsevskiy, D. Martinez, N. Crispino, D. Zvonkine, N. W. Fraga, S. Soori, O. Press, H. Tang, J. Salazar, S. R. Green, L. Brüssel, M. Twayana, A. Dieuleveut, T. R. Rogers, W. Zhang, R. Finocchio, B. Li, J. Yang, A. Rao, G. Loiseau, M. Kalinin, M. Lukas, C. Manolescu, N. Stambaugh, S. Mishra, A. G. K. Kamdoun, T. Hogg, A. Jin, C. Bosio, G. Sun, B. P. Coppola, H. Heidinger, R. Sayous, S. Ivanov, J. M. Cavanagh, J. Shen, J. M. Imperial, P. Schwaller, S. Senthilkuma, A. M. Bran, A. Algaba, B. Verbeken, K. V. den Houte, L. V. D. Sypt, D. Noever, L. Schut, I. Sucholutsky, E. Zheltonozhskii, Q. Yuan, D. Lim, R. Stanley, S. Sivarajan, T. Yang, J. Maar, J. Wykowski, M. Oller, J. Sandlin, A. Sahu, C. G. Ardito, Y. Hu, F. M. Dias, T. Kreiman, K. Rawal, T. G. Vilchis, Y. Zu, M. Lackner, J. Koppel, J. Nguyen, D. S. Antonenko, S. Chern, B. Zhao, P. Arsene, S. Ivanov, R. Poświata, C. Wang, D. Li, D. Crisostomi, A. Dehghan, A. Achilleos, J. A. Ambay, B. Myklebust, A. Sen, D. Perrella, N. Kaparov, M. H. Inlow, A. Zang, K. Ramakrishnan, D. Orel, V. Poritski, S. Ben-David, Z. Berger, P. Whitfill, M. Foster, D. Munro, L. Ho, D. B. Hava, A. Kuchkin, R. Lauff, D. Holmes, F. Sommerhage, A. Zhang, R. Moat, K. Schneider, D. Pyda, Z. Kazibwe, M. Singh, D. Clarke, D. H. Kim, S. Fish, V. Elser, V. E. G. Vilchis, I. Klose, C. Demian, U. Anantheswaran, A. Zweiger, G. Albani, J. Li, N. Daans, M. Radionov, V. Rozhoň, V. Ginis, Z. Ma, C. Stump, J. Platnick, V. Nevirkovets, L. Basler, M. Piccardo, N. Cohen, V. Singh, J. Tkadlec, P. Rosu, A. Goldfarb, P. Padlewski, S. Barzowski, K. Montgomery, A. Menezes, A. Patel, Z. Wang, J. Tucker-Foltz, J. Stade, D. Grabb, T. Goertzen, F. Kazemi, J. Milbauer, A. Shukla, H. Elgnainy, Y. C. L. Labrador, H. He, L. Zhang, A. Givré, H. Wolff, G. Demir, M. F. Aziz, Y. Kaddar, I. Ångquist, Y. Chen, E. Thornley, R. Zhang, J. Pan, A. Terpin, N. Muennighoff, H. Schoelkopf, E. Zheng, A. Carmi, J. Shah, E. D. L. Brown, K. Zhu, M. Bartolo, R. Wheeler, A. Ho, S. Barkan, J. Wang, M. Stehberger, E. Kretov, P. Bradshaw, J. Heimonen, K. Sridhar, Z. Hossain, I. Akov, Y. Makarychev, J. Tam, H. Hoang, D. M. Cunningham, V. Goryachev, D. Patramanis, M. Krause, A. Redenti, D. Aldous, J. Lai, S. Coleman, J. Xu, S. Lee, I. Magoulas, S. Zhao, N. Tang, M. K. Cohen, M. Carroll, O. Paradise, J. H. Kirchner, S. Steinerberger, M. Ovchynnikov, J. O. Matos, A. Shenoy, M. Wang, Y. Nie, P. Giordano, P. Petersen, A. Szyber-Betley, P. Faraboschi, R. Riblet, J. Crozier, S. Halasyamani, A. Pinto, S. Verma, P. Joshi, E. Meril, Z.-X. Yong, A. Tee, J. Andréoletti, O. Weller, R. Singhal, G. Zhang, A. Ivanov, S. Khoury, N. Gustafsson, H. Mostaghimi, K. Thaman, Q. Chen, T. Q. Khánh, J. Loader, S. Cavalleri, H. Szlyk, Z. Brown, H. Narayan, J. Roberts, W. Alley, K. Sun, R. Stendall, M. Lamparth, A. Reuel, T. Wang, H. Xu, P. Hernández-Cámara, F. Martin, T. Preu, T. Korbak, M. Abramovitch, D. Williamson, I. Bosio, Z. Chen, B. Bálint, E. J. Y. Lo, M. I. S. Nunes, Y. Jiang, M. S. Bari, P. Kassani, Z. Wang, B. Ansarinejad, Y. Sun, S. Durand, G. Douville, D. Tordera, G. Balabanian, E. Anderson, L. Kvistad, A. J. Moyano, H. Milliron, A. Sakor, M. Eron, I. C. McAlister, A. F. D. O., S. Shah, X. Zhou, F. Kamalov, R. Clark, S. Abdoli, T. Santens, H. K. Wang, E. Chen, A. Tomasiello, G. B. D. Luca, S.-Z. Looi, V.-K. Le, N. Kolt, N. Mündler, A. Semler, E. Rodman, J. Drori, C. J. Fossum, L. Gloor, M. Jagota, R. Pradeep, H. Fan, T. Shah, J. Eicher, M. Chen, K. Thaman, W. Merrill, M. Firsching, C. Harris, S. Ciobăcă, J. Gross, R. Pandey, I. Gusev, A. Jones, S. Agnihotri, P. Zhelnov, S. Usawasutsakorn, M. Mofayezi, A. Piperski, M. Carauleanu, D. K. Zhang, K. Dobarskyi, D. Ler, R. Leventov, I. Soroko, T. Jansen, S. Creighton, P. Lauer, J. Duersch, V. Taamazyan, D. Bezzi, W. Morak, W. Ma, W. Held, T. Duc Huy, R. Xian, A. R. Zebaze, M. Mohamed, J. N. Leser, M. X. Yuan, L. Yacar, J. Lengler, K. Olszewska, H. Shahrtash, E. Oliveira, J. W. Jackson, D. E. Gonzalez, A. Zou, M. Chidambaram, T. Manik, H. Haffenden, D. Stander, A. Dasouqi, A. Shen, E. Duc, B. Golshani, D. Stap, M. Uzhou, A. B. Zhidkovskaya, L. Lewark, M. O. Rodriguez, M. Vincze, D. Wehr, C. Tang, S. Phillips, F. Samuele, J. Muzhen, F. Ekström, A. Hammon, O. Patel, F. Farhidi, G. Medley, F. Mohammadzadeh, M. Peñaflor, H. Kassahun, A. Friedrich, C. Sparrow, R. H. Perez, T. Sakal, O. Dhamane, A. K. Mirabadi, E. Hallman, K. Okutsu, M. Battaglia, M. Maghsoudimehrabani, A. Amit, D. Hulbert, R. Pereira, S. Weber, Handoko, A. Peristyy, S. Malina, S. Albanie, W. Cai, M. Mehkary, R. Aly, F. Reidegeld, A.-K. Dick, C. Friday, J. Sidhu, H. Shapourian, W. Kim, M. Costa, H. Gurdogan, B. Weber, H. Kumar, T. Jiang, A. Agarwal, C. Ceconello, W. S. Vaz, C. Zhuang, H. Park, A. R. Tawfeek, D. Aggarwal, M. Kirchhof, L. Dai, E. Kim, J. Ferret, Y. Wang, M. Yan,

- K. Burdzy, L. Zhang, A. Franca, D. T. Pham, K. Y. Loh, J. Robinson, A. Jackson, S. Gul, G. Chhablani, Z. Du, A. Cosma, J. Colino, C. White, J. Votava, V. Vinnikov, E. Delaney, P. Spelda, V. Stritecky, S. M. Shahid, J.-C. Mourrat, L. Vetoshkin, K. Sponselee, R. Bacho, F. de la Rosa, X. Li, G. Malod, L. Lang, J. Laurendeau, D. Kazakov, F. Adesanya, J. Portier, L. Hollom, V. Souza, Y. A. Zhou, J. Degorre, Y. Yalin, G. D. Obikoya, L. Arnaboldi, Rai, F. Bigi, M. C. Boscá, O. Shumar, K. Bacho, P. Clavier, G. Recchia, M. Popescu, N. Shulga, N. M. Tanwie, D. Peskoff, T. C. H. Lux, B. Rank, C. Ni, M. Brooks, A. Yakimchyk, Huanxu, Liu, O. Häggström, E. Verkama, H. Gundlach, L. Brito-Santana, B. Amaro, V. Vajipey, R. Grover, Y. Fan, G. P. R. e Silva, L. Xin, Y. Kratish, J. Łucki, W.-D. Li, S. Gopi, A. Caciolai, J. Xu, K. J. Scaria, F. Vargus, F. Habibi, Long, Lian, E. Rodolà, J. Robins, V. Cheng, T. Fruhauff, B. Raynor, H. Qi, X. Jiang, B. Segev, J. Fan, S. Martinson, E. Y. Wang, K. Hausknecht, M. P. Brenner, M. Mao, X. Zhang, D. Avagian, E. J. Scipio, A. Ragoler, J. Tan, B. Sims, R. Plecnik, A. Kirtland, O. F. Bodur, D. P. Shinde, Z. Adoul, M. Zekry, A. Karakoc, T. C. B. Santos, S. Shamseldeen, L. Karim, A. Liakhovitskaia, N. Resman, N. Farina, J. C. Gonzalez, G. Maayan, S. Hoback, R. D. O. Pena, G. Sherman, E. Kelley, H. Mariji, R. Pouriamanesh, W. Wu, S. Mendoza, I. Alarab, J. Cole, D. Ferreira, B. Johnson, M. Safdari, L. Dai, S. Arthornthurasuk, A. Pronin, J. Fan, A. Ramirez-Trinidad, A. Cartwright, D. Pottmaier, O. Taheri, D. Outevsky, S. Stepanic, S. Perry, L. Askew, R. A. H. Rodríguez, A. M. R. Minissi, S. Ali, R. Lorena, K. Iyer, A. A. Fasiludeen, S. M. Salauddin, M. Islam, J. Gonzalez, J. Ducey, M. Somrak, V. Mavroudis, E. Vergo, J. Qin, B. Borbás, E. Chu, J. Lindsey, A. Radhakrishnan, A. Jallon, I. M. J. McInnis, P. Kumar, L. P. Goswami, D. Bugas, N. Heydari, F. Jeanplong, A. Apronti, A. Galal, N. Ze-An, A. Singh, J. of Arc Xavier, K. P. Agarwal, M. Berkani, B. A. de Oliveira Junior, D. Malishev, N. Remy, T. D. Hartman, T. Tarver, S. Mensah, J. Gimenez, R. G. Montecillo, R. Campbell, A. Sharma, K. Meer, X. Alapont, D. Patil, R. Maheshwari, A. Dendane, P. Shukla, S. Bogdanov, S. Möller, M. R. Siddiqi, P. Saxena, H. Gupta, I. Enyekwe, R. P. V, Z. EL-Wasif, A. Maksapetyan, V. Rossbach, C. Harjadi, M. Bahaloohoreh, S. Bian, J. Lai, J. L. Uro, G. Bateman, M. Sayed, A. Menshawy, D. Duclosel, Y. Jain, A. Aaron, M. Tiryakioglu, S. Siddh, K. Krenek, A. Hoover, J. McGowan, T. Patwardhan, S. Yue, A. Wang, and D. Hendrycks. Humanity's last exam, 2025. URL <https://arxiv.org/abs/2501.14249>.
- J. Qiu, X. Qi, T. Zhang, X. Juan, J. Guo, Y. Lu, Y. Wang, Z. Yao, Q. Ren, X. Jiang, X. Zhou, D. Liu, L. Yang, Y. Wu, K. Huang, S. Liu, H. Wang, and M. Wang. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*, 2025.
- O. D. Research. Open deep research, 2025. URL https://github.com/langchain-ai/open_deep_research.
- G. Researcher. Gpt researcher, 2025. URL <https://github.com/assafelovic/gpt-researcher>.
- A. Roucher, A. V. del Moral, merve, T. Wolf, and C. Fourrier. Open-source deepresearch – freeing our search agents, 2025. URL <https://huggingface.co/blog/open-deep-research>.
- S. Schmidgall, Y. Su, Z. Wang, X. Sun, J. Wu, X. Yu, J. Liu, Z. Liu, and E. Barsoum. Agent laboratory: Using llm agents as research assistants. 2025. URL <https://arxiv.org/abs/2501.04227>.
- H. Shen, J. Zhang, B. Xiong, R. Hu, S. Chen, Z. Wan, X. Wang, Y. Zhang, Z. Gong, G. Bao, et al. Efficient diffusion models: A survey. *Transactions on Machine Learning Research (TMLR)*, 2025.
- W. Shi, H. Tan, C. Kuang, X. Li, X. Ren, C. Zhang, H. Chen, Y. Wang, L. Shang, F. Yu, and Y. Wang. Pangu deepdive: Adaptive search intensity scaling via open-web reinforcement learning. *arXiv preprint arXiv:2505.24332*, 2025.
- C. Si, D. Yang, and T. Hashimoto. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. 2024. URL <https://arxiv.org/abs/2409.04109>.

- I. Stelmakh, Y. Luan, B. Dhingra, and M.-W. Chang. ASQA: Factoid questions meet long-form answers. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.566. URL <https://aclanthology.org/2022.emnlp-main.566/>.
- J. Tang, L. Xia, Z. Li, and C. Huang. Ai-researcher: Autonomous scientific innovation. 2025. URL <https://arxiv.org/abs/2505.18705>.
- H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL <https://aclanthology.org/2022.tacl-1.31/>.
- J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.
- Y. Yamada, R. T. Lange, C. Lu, S. Hu, C. Lu, J. Foerster, J. Clune, and D. Ha. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. 2025. URL <https://arxiv.org/abs/2504.08066>.
- L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang. Diffusion models: A comprehensive survey of methods and applications. 2022. URL <https://arxiv.org/abs/2209.00796>.
- Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL <https://aclanthology.org/D18-1259/>.
- J. Yoon, H. Cho, Y. Bengio, and S. Ahn. Fast monte carlo tree diffusion: 100x speedup via parallel sparse planning. 06 2025. URL <https://arxiv.org/abs/2506.09498>.
- K. Zhang, X. Yang, W. Y. Wang, and L. Li. Redi: efficient learning-free diffusion inference via trajectory retrieval. In *Proceedings of the 40th International Conference on Machine Learning*, ICML’23. JMLR.org, 2023.
- Y. Zheng, S. Sun, L. Qiu, D. Ru, C. Jiayang, X. Li, J. Lin, B. Wang, Y. Luo, R. Pan, Y. Xu, Q. Min, Z. Zhang, Y. Wang, W. Li, and P. Liu. OpenResearcher: Unleashing AI for accelerated scientific research. In D. I. Hernandez Farias, T. Hope, and M. Li, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 209–218, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-demo.22. URL <https://aclanthology.org/2024.emnlp-demo.22/>.
- Y. Zheng, D. Fu, X. Hu, X. Cai, L. Ye, P. Lu, and P. Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. 2025. URL <https://arxiv.org/abs/2504.03160>.
- M. Świechowski, K. Godlewski, B. Sawicki, and J. Mańdziuk. Monte carlo tree search: a review of recent modifications and applications. *Artificial Intelligence Review*, 56, 07 2022. doi: 10.1007/s10462-022-10228-y.

Task

Prompt: Write an article about Gmail Proxy Over Thunderbird

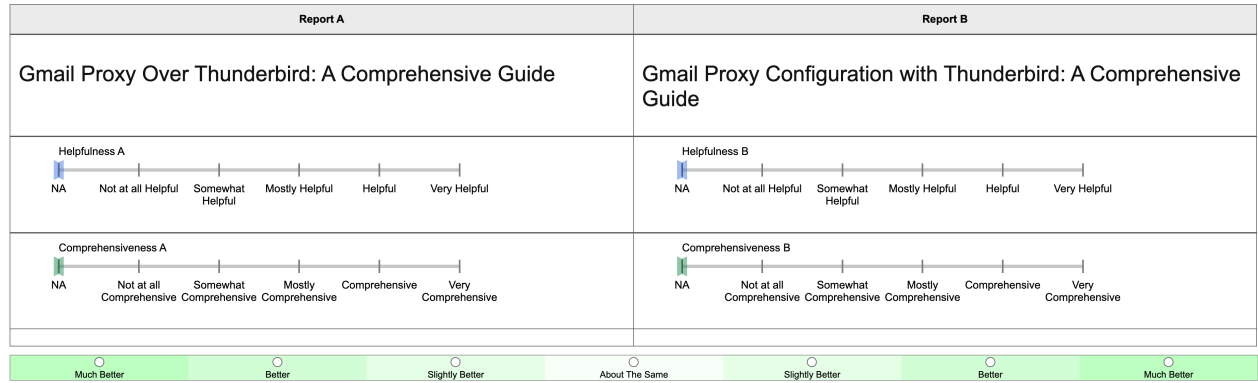


Figure 11 | Helpfulness, Comprehensiveness, and side-by-side rating between Report A and B. Report are simplified for clarify purpose.

A. Appendix

A.1. Evaluation Guidelines

Helpfulness categories can be found below.

- Very Helpful: all statements are helpful based on the guideline above.
- Helpful: Most statements are helpful except for 1-2 statements with minor issues according to the guideline above.
- Mostly Helpful: 1-2 statements seriously fail the guideline above, or 3-5 statements have minor issues.
- Somewhat Helpful: > 2 statements with serious issues, or > 5 statements with minor issues.
- Not at all Helpful: None statements are helpful.

Comprehensiveness categories can be found below.

- Very Comprehensive: it is hard to identify any points that could be added to the report to make it more comprehensive.
- Comprehensive: it is hard to identify any major points that could be added to the report to make it more comprehensive. It would be nicer to add some minor points, but not necessary.
- Mostly Comprehensive: There 1-2 major points that should be added to the report.
- Somewhat Comprehensive: There are more than 2 major points that should be added.
- Not at all Comprehensive: There are more than 5 major points that should be added.

A.2. Human Annotation Interface

Figure 11 shows our human annotation interface.

Table 3 | In this Table, we show the alignments between our auto-rater and human raters. Human accuracy is computed comparing two raters' scores by treating one as ground-truth and taking average.

Evaluator Models	Correlation	Accuracy
GEMINI-1.5-PRO-002	0.22	60.8
GEMINI-2.0-FLASH-001	0.07	51.1
GEMINI-2.5-PRO-PREVIEW-03-25	0.12	47.8
HUMAN	-	69.0

A.3. Human and LLM-as-a-judge Alignment

A.4. HLE Query Categorization

We use the following prompt to categorize HLE queries into 1) reasoning only and 2) reasoning+search.

HLE Query Categorization Prompt

You are an expert categorizing a query from a user. Your task is to assign the query to one of the following 2 categories:
 * "Reasoning": The query can be answered with pure logical reasoning without any external world knowledge.
 * "Search": The query can NOT be answered with pure logical reasoning, but requires additional information that can be obtained through searching the web.

The query is in the <query></query> tags and the answer to the query is in <reference></reference>.
 We also provide rational in <rational></rational> that explains the answer.

First, follow the instructions in the <instructions></instructions> tags below to assess the Correctness of the answer.

<rubrics>

Please output using the scale below:

- * 1: Reasoning: The query can be answered with pure logical reasoning without any external world knowledge.
- * 2: Search: The query can NOT be answered with pure logical reasoning, but requires additional information that can be obtained through searching the web.

</rubrics>

Here is the query:

<query>

{query}

</query>

Here is the answer:

<reference>

{answer}

</reference>

Here is the rational that leads to the reference answer:

<rational>

{rational}

</rational>

Review the rubrics in the <rubrics></rubrics> tags above to rate the answer.

First, think step by step, put your thinking in <thinking></thinking> tags. Your thinking must be shorter than 200 words. Then, provide your category inside <rating></rating> tags. Remember your output must be either 1 or 2 in <rating></rating> tags.

A.5. Answer Merging.

We use the following prompt to merge multiple answer into one for the parallel denoising algorithm.

Answer Merging Prompt

Your task is to research a topic and try to fulfill the user query in the `<user>` tags.

`<instructions>`

You are given a list of candidate answers in `<answer_list>` tags below. Combine them into a single answer so that,
+ it best fulfills the initial user query in the `<user>` tags.

+ If there are conflicting information, try to reconcile them in a logically sound way.

`</instructions>`

Here is the user query.

`<user>`

{query}

`</user>`

Here is the list of candidate answers you need to merge.

`<answer_list>`

{answer_list}

`</answer_list>`

Only output a combined answer from the answers in `<answer_list>`. Do NOT use other information.

A.6. Hyper-parameters

We list a few key hyper-parameters for our self-evolution algorithm shown in Fig. 5. To recap, this algorithm generates multiple initial states, each undergoes self-evolving steps before being merged into a final one. So it introduces two sets of hyper-parameters: n number of initial states and s number of evolving steps.

Hyper-parameters	Description	LONGFORM RESEARCH	DEEPCONSULT	HLE	GAIA
n_p	No. of initial plan states	1	1	1	1
n_q	No. of initial search query states	5	5	5	5
n_a	No. of initial answer states	3	3	3	3
n_r	No. of initial report states	1	1	5	5
s_p	No. of plan self-evolving steps	1	1	1	1
s_q	No. of search query self-evolving steps	0	0	0	0
s_a	No. of answer self-evolving steps	0	0	0	0
s_r	No. of report self-evolving steps	1	1	0	0

Table 4 | We show hyper-parameter description and best settings in this table.

A.7. Question Complexity

Unique Question Key Points Extraction

You are provided with a question in `<question>` tag. Analyze the complexity of the question.

`<question>`

{question}

`</question>`

Breakdown the question into unique key points, and then calculate the number of key points in the question.

First, put your thinking in `<thinking>` `</thinking>` tags, and then put the number in `<number>` `</number>` tags.
Return an integer.

A.8. Answer Complexity

Unique Answer Key Points Extraction

You are provided with an answer in `<answer>` tag. Analyze the complexity of the answer.

```
<answer>
{answer}
</answer>
```

Breakdown the answer into unique key points, and then calculate the number of key points in the answer.

First, put your thinking in `<thinking>` tags, and then put the number in `<number>` tags.
Return an integer.

A.9. Query Novelty

Search Question Novelty

You are provided with a list of used questions in `<question_list>` tags and a new question in `<new_question>` tags. You need to judge how novel the new question is given the used questions.

```
<question_list>
{question_list}
</question_list>
```

```
<new_question>
{new_question}
</new_question>
```

Breakdown the new question into unique key points, and then calculate the number of key points that are NOT semantically covered in any of the used questions.

First, put your thinking in `<thinking>` tags, and then put the number in `<number>` tags.
Return an integer.

A.10. Report Coverage

Report Coverage

Given a context in `<context>` tags, you need to judge how much content in this context is included in the response in `<response>` tags.

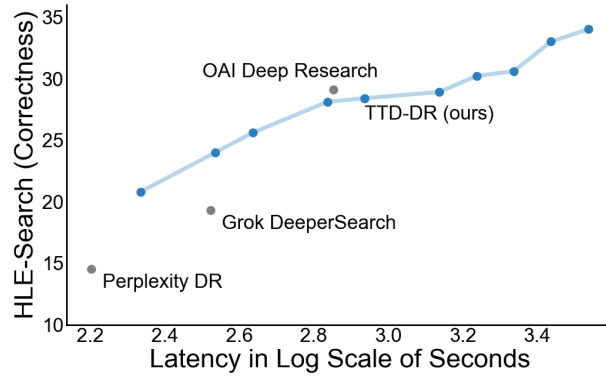
```
<context>
{context}
</context>
```

```
<response>
{response}
</response>
```

Breakdown the context into sentences, and then calculate the ratio of sentences that are semantically covered in response.

First, put your thinking in `<thinking>` tags, and then put the ratio in `<ratio>` tags. Round the ratio to 2 decimal places.

A.11. Additional Analysis Results



(a) Pareto frontier for different DR designs.

Figure 12 | Pareto frontier between DR agent performances and latency for HLE-SEARCH. The dots from left to right represent adding more search/revision steps up to 20, which shows with similar latency, we achieve on-par or better results compared with competing DR agents. Note that HLE dataset only requires identify short-form answer, which does not align perfectly well with our primary tasks of writing real-world long-form reports.

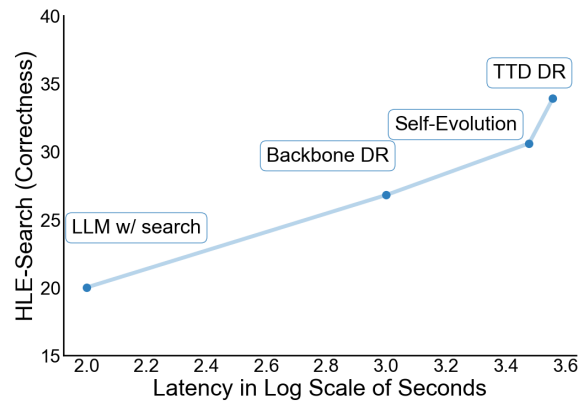


Figure 13 | Pareto frontier between DR agent performances and latency for HLE-search. The dots from left to right represent 1) GEMINI-2.5-PRO w/ SEARCH TOOL, 2) BACKBONE DR AGENT, 3) + SELF-EVOLUTION and 4) + DIFFUSION WITH RETRIEVAL, which shows our final algorithm is most efficient in terms of test-time scaling (steepest slope).