ResearStudio: A Human-Intervenable Framework for Building Controllable Deep-Research Agents

Linyi Yang
Southern University of Science and Technology
Resear AI
yanglinyiucd@gmail.com

Yixuan Weng✉
Resear AI
✉ wengsyx@gmail.com

Abstract

Current deep-research agents run in a "fire-and-forget" mode: once started, they give users no way to fix errors or add expert knowledge during execution. We present RESEARSTUDIO, the first open-source framework that places real-time human control at its core. The system follows a *Collaborative Workshop* design. A hierarchical Planner–Executor writes every step to a live "plan-as-document," and a fast communication layer streams each action, file change, and tool call to a web interface. At any moment, the user can pause the run, edit the plan or code, run custom commands, and resume – switching smoothly between *AI-led, human-assisted* and *human-led, AI-assisted* modes. In fully autonomous mode, ResearStudio achieves state-of-the-art results on the GAIA benchmark, surpassing systems like OpenAI's DeepResearch and Manus. These results show that strong automated performance and fine-grained human control can coexist. The full code, protocol, and evaluation scripts are available at https://github.com/ResearAI/ResearStudio. We will continue to update the repository to encourage further work on safe and controllable research agents.[1]

1 Introduction

The advent of Large Language Models (LLMs) (Ouyang et al., 2022; Achiam et al., 2023; Yang et al., 2025) has catalyzed a new era in artificial intelligence, providing powerful engines for reasoning (Weng et al., 2022; Besta et al., 2024), comprehension (Rein et al., 2024; Weng et al., 2024), and generation (Zhu et al., 2024; Li et al., 2024). This has naturally led to the development of LLM-based autonomous agents (Roy et al., 2024; Huang et al., 2025), which leverage these models as a cognitive core to tackle complex (Yao et al., 2023). These long-horizon tasks were previously intractable (Zhang et al., 2025a). Most recently, a new class of advanced autonomous systems, termed Deep Research (DR) agents, have emerged, exemplified by industry-leading solutions such as OpenAI DR (OpenAI, 2025), Gemini DR (Team et al., 2023), and Memento (Zhou et al., 2025).

Yet prevailing agent frameworks (Chen et al., 2024; Zhu et al., 2025; Zheng et al., 2025; OpenAI, 2025) offer only a rigid, one-directional pipeline: once a task is issued, the user is reduced to a passive observer. When the agent misinterprets goals or pursues flawed strategies, there is no channel for timely human intervention, leading to errors, wasted compute, and diminished trust. Current Deep Research agents, therefore, fall short of the collaborative interface envisioned for AI.

We address this gap with the *Collaborative Workshop*, a shared, persistent, and interactive digital interface characterized by three key properties: (1) Transparency – all plans, intermediate artifacts, and actions are visible; (2) Symmetrical Control – humans and AI possess equivalent authority to modify any element; and (3) Dynamic Role Fluidity – control can seamlessly shift between AI-led and human-led workflows.

To this end, we present RESEARSTUDIO, which is the first open-source realisation of this paradigm. Its layered architecture and custom protocol let users pause or resume execution, edit any plan or file, execute their terminal commands, and export the full workspace at any time. Through the human-intervenable interface, users are able to pause and resume the agent's execution, directly edit not only the plan (TODO.md) but also any code or data file, take control of the terminal to run commands, and download the complete state of the workspace. This capability enables two complementary collaboration modes: **AI-led, Human-assisted**: the agent drives the workflow while the user audits, refines, and injects domain knowledge. **Human-led, AI-**

[1]Our live demo is publicly accessible at http://ai-researcher.net:3000/.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 896–905
November 4-9, 2025

Deep Research Agent	Online Search	OpenSource Framework	Pre-research Intervention	Real-time Content Adjustment
OpenAI DeepResearch	✓	×	×	×
OpenAI/Google Canvas	×	×	×	Text editing Only
Google DeepResearch	✓	×	✓	×
Kimi-Researcher	✓	×	×	×
Grok DeepSearch	✓	×	✓	×
Skywork Agent	✓	×	✓	Slide/Doc Only
ResearStudio (Ours)	✓	✓	✓	✓

Table 1: Comparative analysis of Deep Research Agent features. Symbol guide: ✓ indicates explicit support for the feature; × indicates no native support found in the provided materials.

assisted: the user orchestrates high-level strategy and delegates well-defined subtasks to the agent. Our contributions are three-fold:

1. We formalise the *Collaborative Workshop*, unifying transparency, symmetric control, and role fluidity for human-intervenable deep research interface.

2. We release RESEARSTUDIO, a fully open-source deep research agent that enables real-time bidirectional collaboration and live plan editing with the help of a search agent.

3. We empirically show that our design achieves state-of-the-art performance on the GAIA benchmark among existing Deep Research agents from both industry and academia, demonstrating that collaboration enhances, rather than sacrifices, capability.

2 Related Work

The development of autonomous agents has been accelerated by foundational LLM advancements in reasoning, such as Chain-of-Thought (Wei et al., 2022) and self-reflection (Shinn et al., 2023), and in action, through tool-use integration (Schick et al., 2023). Building on this, agent architectures have explored multi-agent coordination, as in AutoGen (Wu et al.), LangGraph and MetaGPT (Hong et al.), or hierarchical task decomposition, such as in AgentOrchestra (Zhang et al., 2025b) and OWL (Hu et al., 2025). While these frameworks significantly advance agent-to-agent and agent-to-environment interactions, they largely overlook the paradigm of direct, real-time human-agent collaboration. ResearStudio, in contrast, applies these foundational reasoning and tool-use capabilities within an architecture designed specifically to place the human user at the center of the workflow.

This collaborative gap is highlighted by two recent, divergent trends in AI systems, as summarized in our comparative analysis in Table 1. On one hand, highly capable autonomous agents like OpenAI's DeepResearch and Grok's DeepSearch demonstrate impressive performance on complex tasks. However, they operate with limited interactivity, offering minimal support for the kind of **Real-time Content Adjustment** necessary for course correction, thus relegating the user to a passive role. On the other hand, collaborative interfaces like OpenAI's and Google's Canvas offer rich editing environments but are typically confined to single-file manipulation ("Text editing Only") and lack the ability to execute complex, multi-tool tasks across an entire project. ResearStudio bridges this divide. It embeds a powerful autonomous agent capable of complex task execution on par with leading systems, but within a fully interactive and multi-file workshop environment. As shown in Table 1, ResearStudio is unique in providing an open-source framework that supports intervention at all stages, uniting state-of-the-art autonomy with genuine human control.

3 The ResearStudio Framework

The "Collaborative Workshop" paradigm is realized through a three-layer architecture, as depicted in Figure 1. Together, these tools form an intervenable substrate: every transformation is transparent, every action reversible, and every intermediate product editable, **turning the agent from an opaque executor into a controllable and reliable research partner.**

3.1 Tool Usage

To make deep-research agents *human-intervenable*, each tool is exposed through the same live interface that streams its inputs and outputs, allowing users to override or refine any step in real time.

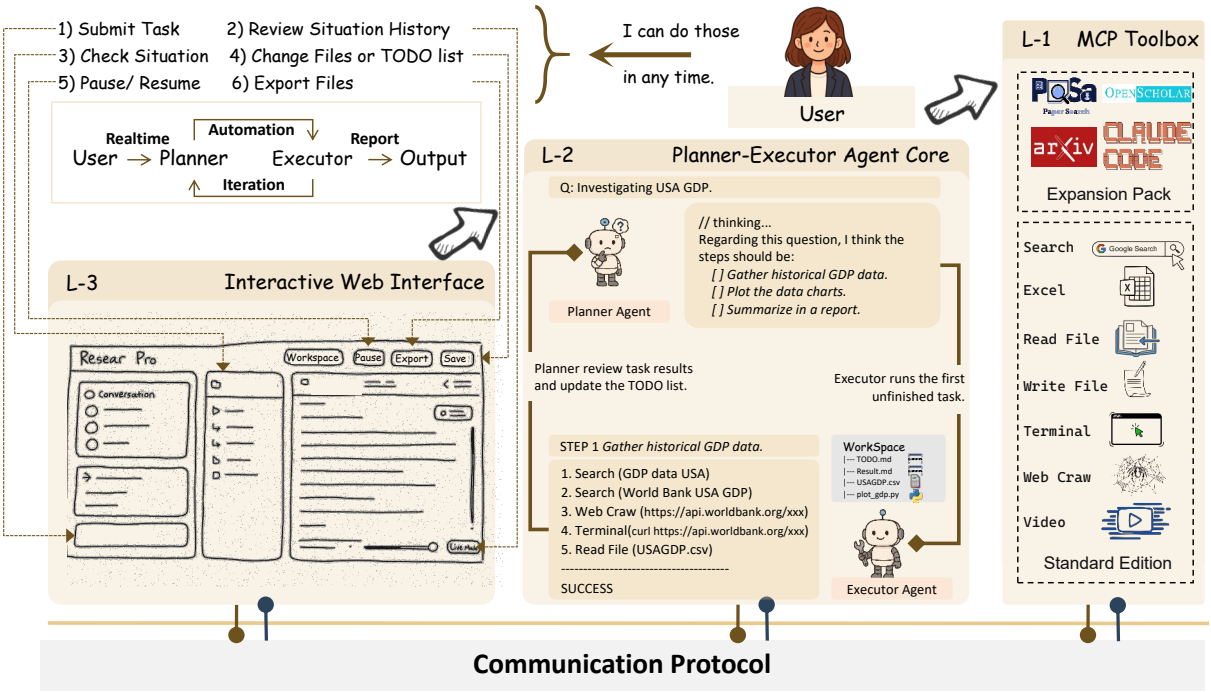

Figure 1: The overall architecture of the ResearStudio framework. This diagram illustrates the three core layers (L-1, L-2, L-3) and the primary workflow. The L-2 Agent Core, composed of a Planner and an Executor, processes a user's request. The Executor carries out steps by using tools from the L-1 MCP Toolbox. The entire process is accessible to the user via the L-3 WebPage, linked by a central Communication Protocol.

Document-Processing Toolkit. Given a file $f \in \mathcal{F}$ with

$$\mathcal{F} = \{\text{jpg}, \text{png}, \text{mp3}, \text{pptx}, \text{xlsx}, \text{csv}, \text{zip}, \text{txt}, \text{json}, \text{xml}, \text{docx}, \text{mov}, \text{pdf}, \ldots\},$$

the system invokes a modality-specific extractor $D(f)$ that yields text, captions, or structured objects (e.g., VLM captions for images, ASR transcripts for audio, slide-wise markdown for `.pptx`, row-wise CSV for spreadsheets). Extracted artefacts immediately appear in the UI, where the user can edit, annotate, or discard them before the agent continues – ensuring that downstream reasoning is always grounded in vetted content.

Search Toolkit. In terms of the search agent, we combine a self-hosted SEARXNG metasearch with CRAWL4AI page fetches. Results are re-ranked by contextual similarity, then streamed to the interface. The researcher can (i) accept, (ii) reject, or (iii) request deeper crawling of any hit. This tight human-in-the-loop filter cuts noise and steers the agent toward authoritative sources.

Deliberate Browser Omission. Although full browser automation offers rich interactions, we observed that LLM planners overuse it, incurring latency and extracting little structured data. By default we exclude browser control; users can re-enable it with a single toggle if a page truly requires dynamic rendering. This design keeps the interface fast and the provenance chain clean.

Code Toolkit. A sandboxed workspace lets the agent (or the user) create files, run shell or Python commands, and inspect outputs with state preserved across iterations. Every script is displayed pre-execution; the researcher can modify code, inject assertions, or roll back to a prior snapshot. Safe-exec guards whitelist common packages (*numpy, pandas, torch, etc.*) to balance flexibility and security. Every script or shell command is first rendered in the UI; the user may edit, comment, or disable the snippet before execution. Standard output, error streams, and rich artefacts (tables, figures) stream back in real time and are logged as immutable cells. A built-in diff viewer records successive file changes, enabling one-click rollback or branch creation for "what-if" explorations.

Interactive Web Interface. We provide a comprehensive view of the agent's conversation, the workspace files, and global controls, enabling the user to monitor progress and intervene at any time.

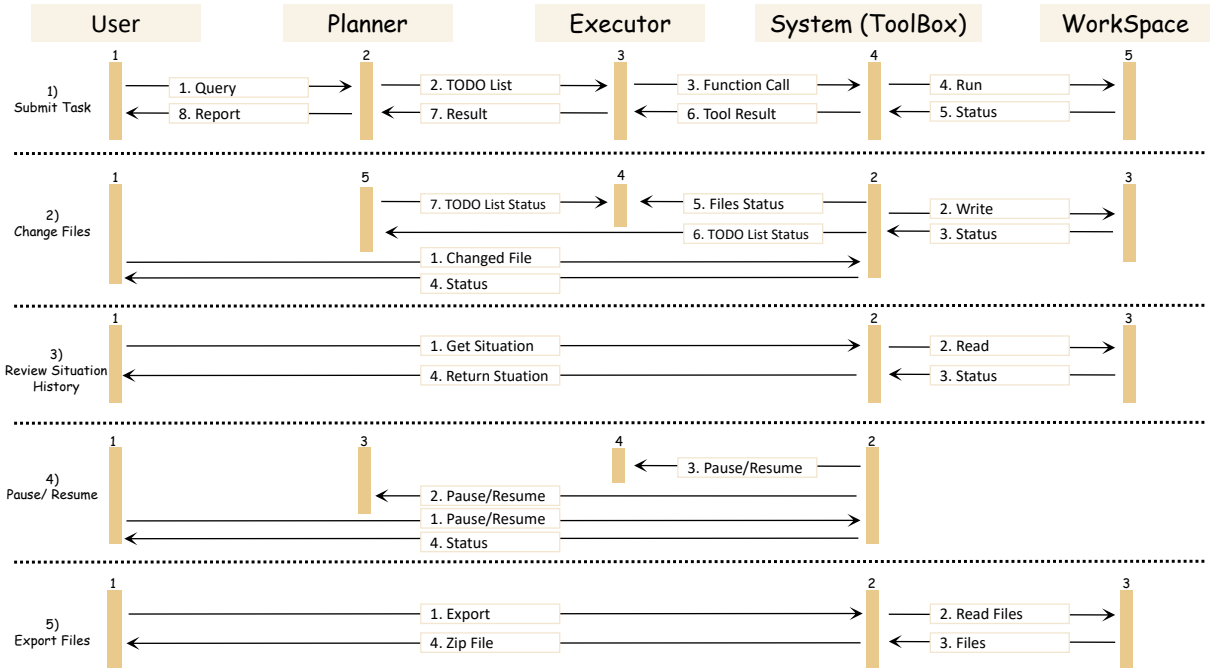

Figure 2: Core interaction workflows of the ResearStudio communication protocol. This diagram details the sequence of messages between the User, Planner, Executor, System (Toolbox), and Workspace for key operations, illustrating the bidirectional flow of information that enables real-time collaboration.

3.2 Bidirectional Protocols

The framework's collaborative capabilities are powered by a dual-layered communication system, whose core workflows are detailed in the sequence diagrams of Figure 2. At the machine level, the Model-Context Protocol (MCP), implemented with 'fastmcp', standardizes the Executor's tool calls into reliable, JSON-based functions. This corresponds to the agent's autonomous execution loop shown in Figure 2 ("Submit Task" workflow). The entire system is orchestrated by a central communication protocol that enables seamless interaction between all components.

More central to our contribution is the event-driven protocol governing the human-agent partnership. This protocol provides the technical foundation for the direct manipulation and control workflows. Upon starting a task, a long-lived connection is established between the frontend and backend. User actions are then translated into specific API calls tagged with a unique task ID. For instance, the "Change Files" workflow is enabled by a 'POST' request containing the new file content, which updates the Workspace and notifies the Executor. Similarly, the "Pause/Resume" workflow is implemented by a request that stalls all backend LLM calls, effectively freezing the agent's cognitive state until resumed. All real-time updates from the agent are streamed back to the user through this persistent connection, with large file contents being lazy-loaded on click to maintain UI responsiveness. This protocol transforms the system into a truly interactive and auditable workshop, where the user is a continuous collaborator rather than a passive observer.

3.3 Backend Models

The Planner is powered by `gpt-4.1`, the Executor by `o4-mini` for datasets excluding GAIA, the image processing by `gpt-4o`, the video agent by `gemini-2.5-pro` and the audio agent by `Assembly AI`. We select the `o3` as the executor in the GAIA benchmark. Each module is chosen to balance efficiency and task-specific capability, ensuring reliable multimodal perception, robust planning, and precise execution.

4 Experiments

To validate the effectiveness and capabilities of the ResearStudio framework, we conducted a rigorous evaluation on the GAIA benchmark (Mialon et al., 2023), a standard for testing general-purpose agentic systems on complex reasoning and multi-step tool use. **All experiments were performed in a fully autonomous mode, with no human intervention from task initiation to completion**,

Agent	Level-1	Level-2	Level-3	Average
ODR-smolagents	67.92	53.49	34.62	55.15
AutoAgent	71.70	53.49	26.92	55.15
OWL	84.91	67.44	42.31	69.09
A-World	**86.79**	69.77	34.62	69.70
OpenAI-DeepResearch	74.29	69.06	47.60	67.36
ResearStudio (Pass@1)	**77.36**	**69.77**	**61.54**	**70.91**

Table 2: Performance comparison of our agent against baseline methods ODR-smolagents (Roucher et al., 2025), AutoAgent (Chen et al., 2024), OWL (Hu et al., 2025), A-World (at Ant Group, 2025), and OpenAI-DeepResearch (OpenAI, 2025) on GAIA benchmark (Mialon et al., 2023). Average Task Runtime across all GAIA levels is approximately 20 minutes, while simpler tasks or specific successful cases (as discussed in Section 5) can be completed in under 10 minutes.

Agent	Level-1	Level-2	Level-3	Average
OWL (Hu et al., 2025)	75.27	61.01	32.65	60.80
A-World (at Ant Group, 2025)	80.65	64.78	24.49	63.12
ResearStudio (Ours)	**84.95**	**72.33**	**59.18**	**74.09**

Table 3: Performance comparison of our agent against open-source frameworks on the test set of the GAIA benchmark.

thereby assessing the core performance of our architecture. For our agent's configuration, the **Planner** is powered by `gpt-4.1`, while the **Executor** utilizes `o4-mini` for its tactical operations. To handle multimodal tasks, the framework is equipped with specialized models: `gpt-4o` for image-related tasks, `gemini-2.5-pro` for video processing, and Assembly AI for audio tasks. For evaluation, we report the **Exact Match (EM)** metric, where a prediction is considered correct only if it exactly matches the reference answer after normalizing for case, punctuation, and articles. The EM score is defined as the percentage of answers that achieve a perfect match.

Overall Results. Our experimental results demonstrate that ResearStudio achieves state-of-the-art performance on the GAIA benchmark across both validation and test sets. As shown in Table 2, on the GAIA validation set, our framework achieves a leading average score of **70.91%**, outperforming other established agents such as A-World (69.70%) and OpenAI-DeepResearch (67.36%). Notably, ResearStudio shows exceptional capability on the more complex tasks, achieving the highest scores on both Level-2 (**69.77%**) and the highly challenging Level-3 (**61.54%**) tasks. To further validate these findings on unseen data, we evaluated our agent on the GAIA test set. The results, presented in Table 3, solidify ResearStudio's superior performance. Our agent achieves an overall average score of **74.09%**, surpassing all listed baselines across every difficulty level, with scores of **84.95%** on Level-1, **72.33%** on Level-2, and **59.18%** on Level-3. These robust results, obtained in a fully autonomous setting, validate the efficacy of our Planner-Executor architecture and modular tool integration, proving that a framework designed for human-collaboration can also deliver superior performance on its own.

5 Discussion

The architecture of ResearStudio materializes into a practical and effective collaborative workflow, as illustrated by the user interface in Figure 3 and Table 4. The multi-panel design provides a user with complete situational awareness. For instance, a user can monitor the agent's detailed execution steps in the "Conversation & Activities" log while simultaneously inspecting the files it generates, such as "analysis_penguin.py" and "todo.md", in the "File Explorer". This transparent process allows for timely and precise intervention. If the user observes the agent writing flawed code into the "File Editor", they are not forced to wait for an error. Instead, using the "Global Controls", they can pause the agent, directly correct the script, and then resume the task. This action of switching from passive observation to active contribution demonstrates the fluid transition between an AI-led and a human-assisted workflow.

Operational Parameter / Metric	Typical Value
Average Task Runtime (GAIA)	~20 minutes
Max Concurrent Workers	50
Max Interaction Rounds	30
Average Steps per Task	~25
Typical Final Workspace Size	~100 MB

Table 4: Key operational parameters and metrics for a typical ResearStudio run on a complex task. The system is designed for both efficiency and the capacity to handle long-horizon problems.

Beyond enabling a more collaborative workflow, the ResearStudio architecture is also engineered for practical efficiency and scalability, with typical operational parameters summarized in Table 4. The

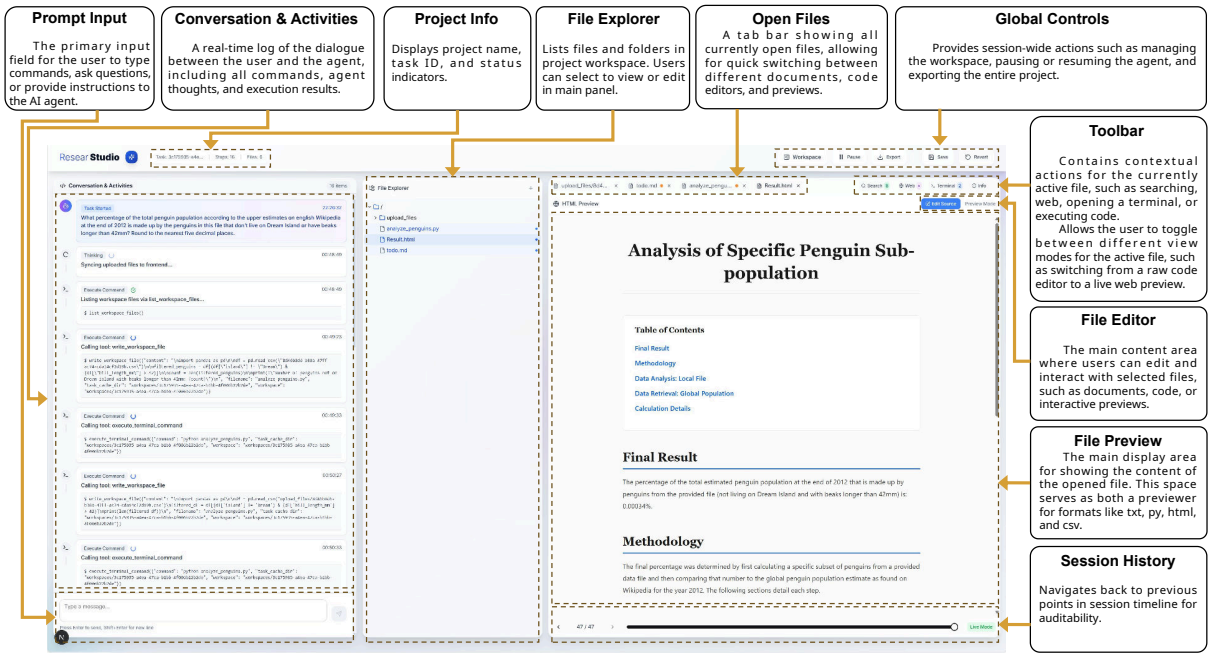

Figure 3: An overview of the ResearStudio user interface, designed as an integrated workspace for human-agent partnership. The layout provides a comprehensive view by juxtaposing the agent's real-time execution trace in the activity log with the tangible project artifacts in the file explorer and editor. This design allows a user to seamlessly monitor autonomous operations while retaining the ability to directly interact with and manage all project files and system states through dedicated controls.

framework is capable of managing long-horizon tasks, supporting up to 50 interaction rounds between the Planner and Executor, which allows for deep, iterative problem-solving. This process is highly efficient; a complex GAIA Level-3 task, for example, completes in approximately 10 minutes, a result of offloading heavy computation to dedicated tools rather than relying on pure model reasoning. Our system architecture supports a high degree of concurrency, capable of handling up to 50 concurrent workers, which facilitates scalable deployment for complex, multi-faceted projects. This efficiency makes the human-in-the-loop paradigm practical and appealing. A user is far more likely to engage and collaborate with a system that produces tangible results in minutes, not hours. The fully auditable record of all actions in the "Conversation & Activities" panel, combined with the "Session History" feature, not only enhances trustworthiness but also allows the completed workspace to serve as a detailed, reusable template for future tasks, further amplifying the long-term value of each collaborative session.

6 Conclusion

This work presented RESEARSTUDIO, an open-source framework that makes human–AI collaboration a foundational element rather than an afterthought. We contribute (i) a practical architecture that combines a hierarchical Planner–Executor with a live, bidirectional protocol to expose the agent's reasoning as an editable document, and (ii) an extensive empirical study showing that this transparency and control do not diminish autonomous performance. On the GAIA benchmark, RESEARSTUDIO attains state-of-the-art results while allowing users to intervene, correct, and guide the system at any stage. These findings demonstrate that accountability and efficiency can coexist in deep research agents. Ultimately, ResearStudio provides a concrete path forward for building the next generation of human-intervenable deep research agent. By releasing the full codebase, we aim to spur further work on AI systems that remain powerful yet verifiable, fostering safer and more trustworthy deployment in high-stakes domains.

Limitations

We acknowledge several limitations in the current version of ResearStudio. First, the framework's effectiveness in its collaborative mode is highly dependent on the user's domain expertise to identify subtle errors, and the requisite continuous monitoring can be cognitively demanding. This positions

the system more as a powerful tool for experts than a universally accessible partner for novices. Furthermore, while our experiments validate the architecture's strong autonomous performance, they do not yet formally quantify the practical benefits of the human-in-the-loop features central to our design. Finally, our current safety measures are primarily architectural, focusing on operational containment through sandboxing, but they have not yet been rigorously stress-tested against active adversarial attacks.

These limitations define clear avenues for our future work. To mitigate cognitive load and broaden the framework's accessibility, we plan to develop semi-autonomous intervention mechanisms, such as AI-powered alerts that flag potential errors or logical inconsistencies for human review. To empirically validate the "Collaborative Workshop" paradigm, a critical next step is to conduct formal Human-Computer Interaction (HCI) studies. These studies will measure key metrics—such as task completion times with and without intervention, error correction rates, and user satisfaction scores—to provide direct evidence of collaborative utility. Lastly, to bolster the system's robustness, we will undertake dedicated red-teaming efforts to assess its resilience against threats like prompt injection, data exfiltration, and the generation of harmful content, using the findings to engineer more sophisticated, dynamic safety protocols.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Agent Team at Ant Group. 2025. Aworld: A unified agent playground for computer and phone use tasks.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje Karlsson, Jie Fu, and Yemin Shi. 2024. Autoagents: a framework for automatic agent generation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 22–30.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, and 1 others. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, and 1 others. 2025. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *arXiv preprint arXiv:2505.23885*.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, and 1 others. 2025. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, and 1 others. 2024. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv:2411.16594*.

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

OpenAI. 2025. Deep research system card.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Aymeric Roucher, A Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. 2025. smolagents: A smol library to build great agentic systems.

Devjeet Roy, Xuchao Zhang, Rashi Bhave, Chetan Bansal, Pedro Las-Casas, Rodrigo Fonseca, and Saravan Rajmohan. 2024. Exploring llm-based agents for root cause analysis. In *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, pages 208–219.

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Kang Liu, and Jun Zhao. 2024. Mastering symbolic operations: Augmenting language models with compiled neural networks. In *The Twelfth International Conference on Learning Representations*.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2022. Large language models are better reasoners with self-verification. *arXiv preprint arXiv:2212.09561*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. 2025a. Darwin godel machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22954*.

Wentao Zhang, Ce Cui, Yilei Zhao, Yang Liu, and Bo An. 2025b. Agentorchestra: A hierarchical multi-agent framework for general-purpose task solving. *arXiv preprint arXiv:2506.12508*.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*.

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and Jun Wang. 2025. Memento: Fine-tuning llm agents without fine-tuning llms. *arXiv preprint arXiv: 2508.16153*.

Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. 2025. DeepReview: Improving LLM-based paper review with human-like deep thinking process. pages 29330–29355, Vienna, Austria. Association for Computational Linguistics.

Yuqi Zhu, Shuofei Qiao, Yixin Ou, Shumin Deng, Shiwei Lyu, Yue Shen, Lei Liang, Jinjie Gu, Huajun Chen, and Ningyu Zhang. 2024. Knowagent: Knowledge-augmented planning for llm-based agents. *arXiv preprint arXiv:2403.03101*.

A Discussion and Case Studies

While quantitative benchmarks validate the high autonomous performance of ResearStudio, a deeper understanding of its practical strengths and limitations can be gained through qualitative analysis of its operational workflow. This section presents two contrasting case studies from the challenging GAIA Level-3 benchmark. The first case illustrates the framework's proficiency in autonomously solving complex, engineering-style computational tasks. The second, a failure case, reveals a common vulnerability in autonomous agents and, in doing so, highlights the profound utility and core design philosophy of our Collaborative Workshop paradigm.

Our framework demonstrates remarkable efficiency on tasks that require algorithmic thinking and precise calculation. In one such GAIA Level-3 task, the agent was presented with a complex computational puzzle involving two 12-digit numbers, column transpositions, and a weighted-sum checksum validation. ResearStudio solved this intricate problem in approximately four minutes over 16 discrete steps. An analysis of its execution trace, presented in Figure 4, reveals a highly effective strategy. The **Planner** correctly identified that a brute-force search was the optimal approach and created a plan to first write a Python script to codify the logic. The **Executor** then successfully implemented this plan, generating and running a 'solve_puzzle.py' script. By intelligently offloading the complex computation to code rather than attempting to reason through it in-prompt, the agent demonstrated efficient and robust problem-solving. This case underscores ResearStudio's strength in

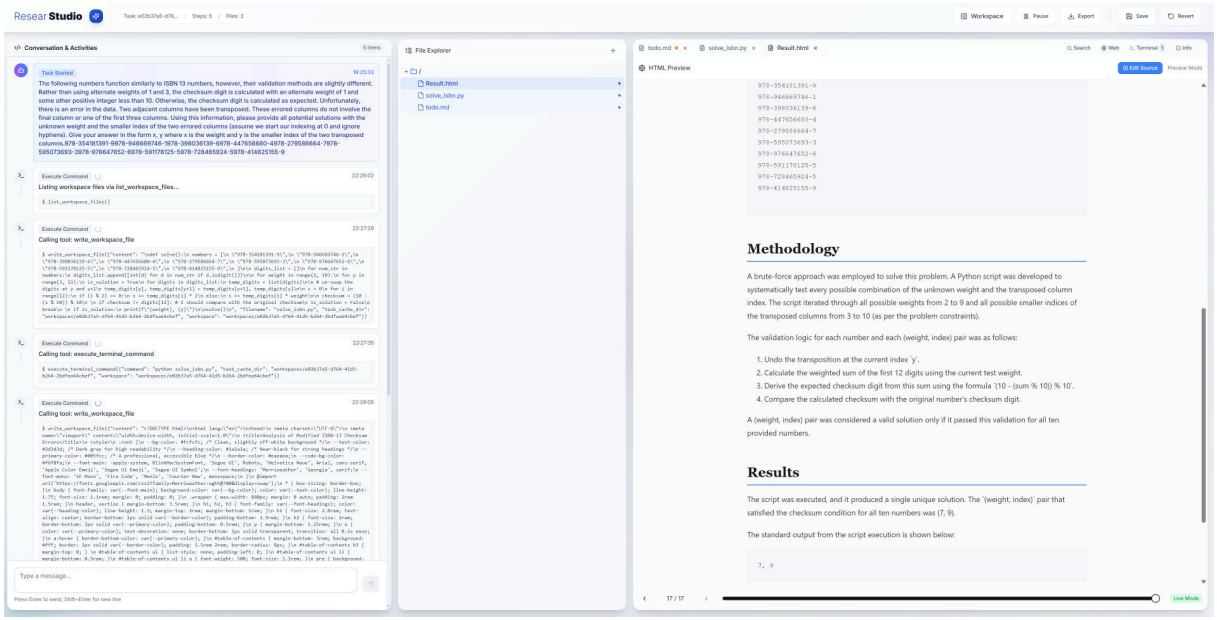

Figure 4: A successful execution trace of a GAIA Level-3 computational puzzle within the ResearStudio UI, showcasing the agent's ability to write and execute code to find a correct solution.

handling well-defined engineering and computational challenges where logic can be explicitly codified and executed.

However, when faced with tasks requiring a nuanced interpretation of ambiguous real-world data, the limitations of pure autonomy become apparent. For GAIA Level-3, the agent was asked to calculate the volume of a Freon-12 container at the bottom of the Mariana Trench. As shown in Figure 5, the agent's workflow was initially logical, correctly identifying the need to find the temperature and pressure to determine the substance's density. The failure occurred when its web search returned conflicting information: the near-freezing ambient temperature of the trench (1-4°C) and the extreme temperature of hydrothermal vents located within it (400°C). Lacking the human-like common sense to disambiguate this context, the agent incorrectly selected the extreme 400°C value. This single error created a cascading failure, leading it to select a completely wrong density value and produce a final answer that was orders of magnitude incorrect. This case highlights a critical vulnerability in many autonomous systems: a conflict between the model's internal knowledge and conflicting information retrieved from the open web can lead to catastrophic, yet logically consistent, failures.

While this failure reveals a limitation, it simultaneously provides the most potent demonstration of ResearStudio's core value. In a traditional "black-box" system, this task would have simply failed, leaving the user with a useless result and no recourse but to restart. Within the **Collaborative Workshop**, however, this catastrophic failure becomes a manageable, correctable mistake. A human collaborator, observing the agent's real-time activity log, would immediately recognize the 400°C temperature as nonsensical. At that precise moment, they could use the interface to **'Pause'** the agent, directly edit the agent's 'TODO.md' plan or a note file in the shared workspace to specify "use ambient temperature of 4°C," and then **'Resume'** the task. This simple, intuitive intervention would have guided the agent back to the correct path, leveraging human expertise to resolve the exact kind of contextual ambiguity that AI struggles with. This ability to seamlessly transfer control and inject human judgment into the loop is the ultimate utility of ResearStudio. It transforms the agent from an opaque, brittle tool into a resilient, trustworthy partner, providing the essential safeguard needed to deploy autonomous systems on complex, real-world problems.

B Safety Considerations

The introduction of autonomous agents that can interact with file systems and external web content necessitates a robust safety framework. In ResearStudio, safety is addressed through a combination of architectural design choices and interactive oversight capabilities. This section discusses the key

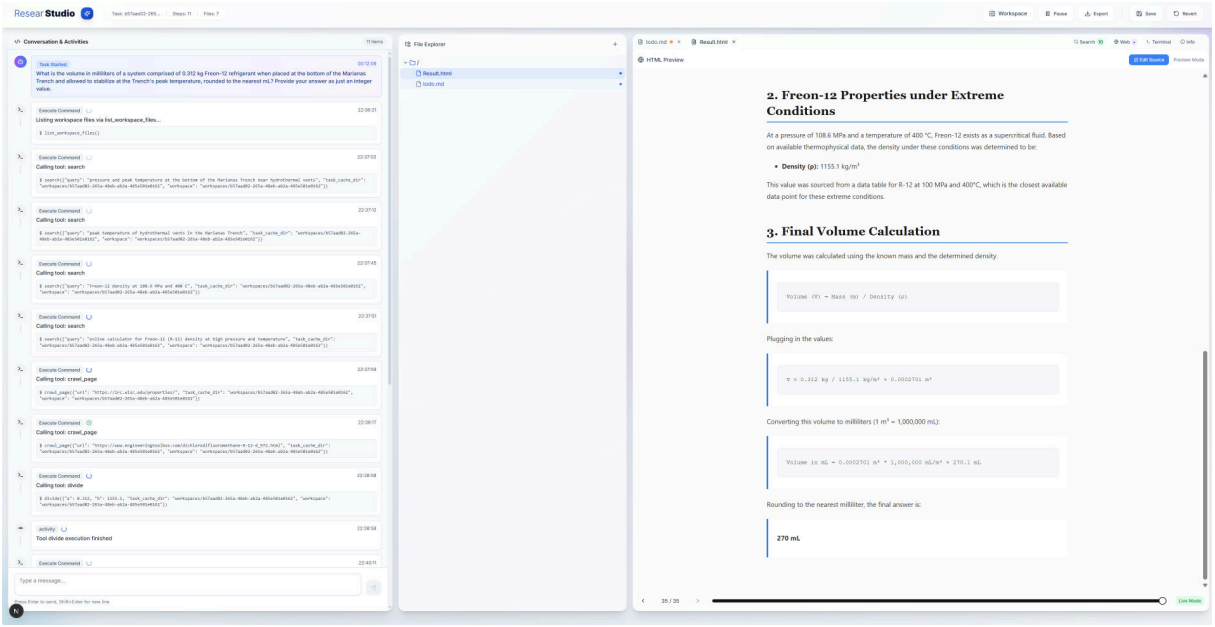

Figure 5: An execution trace of a failed GAIA Level-3 physics problem. The agent correctly structures the problem but fails by selecting the wrong temperature (400°C), leading to a cascade of incorrect calculations.

safety mechanisms, focusing on operational containment and the safeguards against both external and user-initiated threats.

B.1 Operational Safety and Sandboxing

A foundational element of ResearStudio's safety posture is the strict isolation of each task. Upon initiation, every task is assigned a unique workspace that is fully sandboxed from the host system and from other, concurrent tasks. This architectural choice is critical for preventing risks such as data exfiltration or unauthorized access to external resources. All tool operations, whether writing a file or executing a command, are confined within the boundaries of this isolated directory. This containment ensures that even if an agent were to exhibit unintended behavior, the potential impact would be restricted to the immediate task environment.

Furthermore, the execution of tools is managed through a layer of abstraction provided by the MCP. The agent does not directly execute system calls; instead, it formulates structured requests to independent, sandboxed tool services. For instance, the Python tool, as used in systems like Deep Research, runs within its own constrained environment without direct internet access, mitigating cybersecurity risks associated with arbitrary code execution. Similarly, while the agent can request actions via the *Terminal* tool, it does so through this mediated protocol. This prevents the agent from gaining unrestricted shell access and limits the scope of its command-line capabilities to operations relevant to the sandboxed workspace.

B.2 Interactive Oversight and Safeguards

While sandboxing provides technical containment, an additional layer of safety is established through interactive oversight and system-level safeguards. The "Plan-as-Document" principle is a key component, as it externalizes the agent's intentions into a human-readable "TODO.md" file before execution. This provides a critical checkpoint for a user to review the agent's proposed actions and intervene to prevent the pursuit of flawed or unsafe strategies. This is particularly relevant for mitigating risks from external content, such as prompt injection, where a user can spot anomalous changes in the agent's plan or activity log and use the "Pause" control to halt execution.

Crucially, these safeguards are not limited to protecting against external threats; they also address potential misuse by the user. Even within this collaborative framework, the system is designed to prevent the execution of malicious instructions. All user prompts are evaluated by input classifiers against safety policies to filter requests for disallowed content. This layered approach ensures that while the user is an empowered collaborator, the system maintains its own independent capacity to enforce safety protocols and prevent misuse.