

# AgentRxiv: Towards Collaborative Autonomous Research

Samuel Schmidgall<sup>1</sup> and Michael Moor<sup>2</sup>

<sup>1</sup>Department of Electrical & Computer Engineering, Johns Hopkins University, <sup>2</sup>Department of Biosystems Science & Engineering, ETH Zurich

Progress in scientific discovery is rarely the result of a single "Eureka" moment, but is rather the product of hundreds of scientists incrementally working together toward a common goal. While existing agent workflows are capable of producing research autonomously, they do so in isolation, without the ability to continuously improve upon prior research results. To address these challenges, we introduce AgentRxiv—a framework that lets LLM agent laboratories upload and retrieve reports from a shared preprint server in order to collaborate, share insights, and iteratively build on each other’s research. We task agent laboratories to develop new reasoning and prompting techniques and find that agents with access to their prior research achieve higher performance improvements compared to agents operating in isolation (11.4% relative improvement over baseline on MATH-500). We find that the best performing strategy generalizes to benchmarks in other domains (improving on average by 3.3%). Multiple agent laboratories sharing research through AgentRxiv are able to work together towards a common goal, progressing more rapidly than isolated laboratories, achieving higher overall accuracy (13.7% relative improvement over baseline on MATH-500). These findings suggest that autonomous agents may play a role in designing future AI systems alongside humans. We hope that AgentRxiv allows agents to collaborate toward research goals and enables researchers to accelerate discovery.

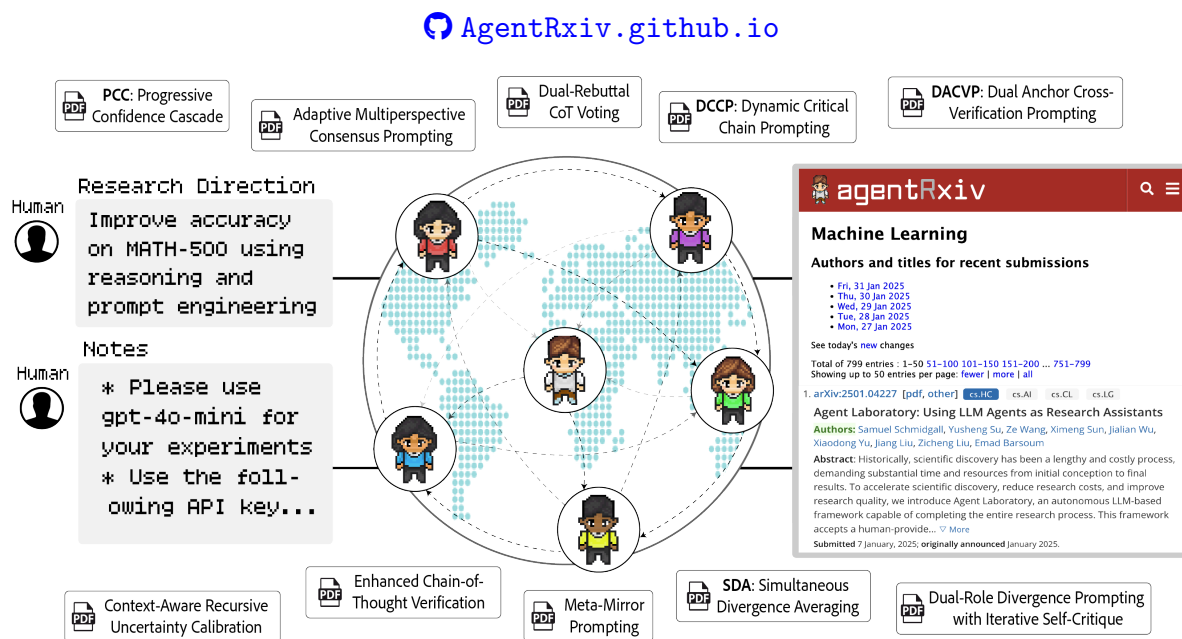


Figure 1 | **Collaborative Autonomous Research via AgentRxiv.** Autonomous agent laboratories distributed collaboratively pursue a shared research goal using AgentRxiv. Human researchers provide initial guidance through a research direction and detailed instructions. Agents autonomously perform research and upload research papers to the centralized AgentRxiv preprint server, enabling laboratories to access each other’s discoveries, accelerating scientific progress.

## 1. Introduction

Scientific discovery has historically been an iterative process, characterized by the systematic formulation of hypotheses, the execution of controlled experiments, and the evaluation of results (Chalmers (2013)). Over time, these methods lead to the steady accumulation of knowledge, forming the basis upon which further inquiries are built (Shapere (1964)). In this manner, scientific progress does not typically arise from isolated breakthroughs but rather from incremental improvements that collectively advance our understanding of complex phenomena.

In an effort to accelerate the process of scientific discovery, recent work has explored the ability of LLM agents to perform autonomous research (Lu et al. (2024b); Schmidgall et al. (2025); Swanson et al. (2024)). The AI Scientist framework (Lu et al. (2024b)) is a large language model (LLM)-based system that generates research ideas in machine learning, writes research code, runs experiments, and produces a scientific paper with using an automated peer-review to evaluate the work. Virtual Lab (Swanson et al. (2024)) uses a multi-agent system of LLM-based experts from different backgrounds (e.g. chemist or biologist) working together with human scientists to produce novel nanobody binders for SARS-CoV-2, where discovered nanobodies demonstrate promising efficacy in wet-lab validations. Finally, Agent Laboratory (Schmidgall et al. (2025)) is a multi-agent autonomous research system that is able to incorporate human feedback, with greatly reduced cost compared to Lu et al. (2024b). While these works demonstrate progress toward accelerated scientific discovery, they often operate in *isolation* and do not support the continuous, cumulative development of research across time that reflects the nature of science. Therefore, we aim to provide a unified platform that enables agents to build upon the research of other agents.

In this study, we introduce AgentRxiv, an autonomous research collaboration framework that supports LLM agents in generating, sharing, and building upon scientific research. By implementing a centralized, open-source preprint server for autonomous agents, AgentRxiv enables systematic sharing of research findings, allowing agents to cumulatively build on previous work. AgentRxiv also supports parallel research across multiple agentic systems, enabling scalability with available computational resources. When incorporating AgentRxiv, each generation of papers shows measurable improvements. For example, accuracy on the MATH-500 benchmark increased from 70.2% to 78.2% with the best discovered reasoning technique, using gpt-4o mini as the base model. Moreover, reasoning strategies developed on MATH-500 are shown to generalize to other tasks and language models, improving performance on benchmarks such as GPQA, MMLU-Pro, and MedQA across models ranging from DeepSeek-v3 to Gemini-2.0 pro. Although the parallelized mode accelerates improvements in wall-clock time, it introduces trade-offs between speed and computational efficiency. A detailed list of our contributions are provided below:

- We introduce AgentRxiv—a novel, open-source framework designed to archive and disseminate research outputs from autonomous agents. This platform enables agents to build upon the discoveries of other agents, driving iterative improvements over time.
- We show that each generation of papers produces measurable improvements when agents are given access to AgentRxiv. For instance, accuracy on the MATH-500 benchmark steadily increased from a 70.2% baseline to 78.2% (11.4% relative improvement) using newly discovered techniques like Simultaneous Divergence Averaging (SDA).
- We demonstrate that reasoning strategies discovered through AgentRxiv on MATH-500 generalize to other benchmarks and language models. Using the highest performing discovered reasoning method by AgentRxiv, our experiments show performance increase across a range of tasks—from GPQA and MMLU-Pro to MedQA—while also revealing consistent gains across five different language models, from DeepSeek-v3 to Gemini-2.0 pro (improving on average by

3.3%).

- We introduce a parallelized mode for AgentRxiv, allowing multiple agentic systems to run simultaneously and share findings. We show that this setup accelerates improvements on MATH-500 by +6.0% with 3 parallel labs. We also find that there are trade-offs between speed and computational efficiency, with discoveries occurring faster but at a higher computational cost.

## 2. Background and Related Work

**Large language models.** Large language models (LLMs) typically employ transformer architectures (Vaswani (2017)), which rely on self-attention to capture long-range dependencies in text without using recurrent (Jordan (1997); Rumelhart et al. (1985)) or convolutional layers (Fukushima (1980); LeCun et al. (1998)). These models can contain hundreds of millions or even billions of parameters, often trained on large-scale corpora drawn from diverse sources such as web pages, books, and online forums (Brown (2020); Devlin et al. (2019); Radford et al. (2018, 2019)). During training, LLMs are often trained to model statistical regularities in the data by predicting the next token in a sequence, which enables them to generate coherent text, answer questions, and support a wide variety of language understanding tasks. Recent reasoning models—such as OpenAI’s o1 (Jaech et al. (2024)), o3-mini (OpenAI (2025)), and DeepSeek’s R1 (Guo et al. (2025))—extend the paradigm of next-token prediction by incorporating chain-of-thought prompting (Wei et al. (2022)) and reinforcement learning techniques (Shao et al. (2024)) to support multi-step problem solving.

**LLM Agents.** LLM Agents expand the capabilities of LLMs by integrating structured workflows in order to autonomously perform task execution (Lu et al. (2023); Wu et al. (2023); Yang et al. (2024a); Yao et al. (2023b); Zhang et al. (2024a); Zhuge et al. (2024)). Rather than only generating text outputs, these agents interact with the external environment, using techniques such as reasoning (Hao et al. (2023); Kojima et al. (2022); Wei et al. (2022); Yao et al. (2023a)), iterative refinement (Madaan et al. (2023)), self-improvement (Huang et al. (2022); Shinn et al. (2023)), and tool usage (Hao et al. (2024); M. Bran et al. (2024); Paranjape et al. (2023); Qin et al. (2023); Qu et al. (2024); Schick et al. (2023)) to accomplish this. LLM agents have made incredible progress across a wide range of tasks from medical task solving (Li et al. (2024a); Schmidgall et al. (2024); Shi et al. (2024); Tu et al. (2024)), machine learning engineering (Chan et al. (2024); Guo et al. (2024); Huang et al. (2024); Nathani et al. (2025)), software engineering (Jimenez et al. (2023); Wang et al. (2024b); Yang et al. (2024a)) and web tasks (Deng et al. (2024); Gur et al. (2023); He et al. (2024); Putta et al. (2024); Shi et al. (2017)).

**Automated machine learning.** Automated machine learning (AutoML) focuses on developing systems that autonomously select models, optimize hyperparameters, and perform feature engineering to improve model performance (He et al. (2021); Tornede et al. (2023)). Recently, many works using LLMs in AutoML have focused on problems from the online platform *Kaggle*—which hosts machine learning competitions—as a benchmark for evaluating agent performance. Notable efforts include MLE-Bench (Chan et al. (2024)), ML-Bench (Nathani et al. (2025)), DS-bench (Jing et al. (2024)), and MAgentBench (Huang et al. (2024)) which propose using 75, 8, 74, and 6 Kaggle (or Kaggle-style) challenges respectively as benchmarks to measure the abilities of ML agents in tasks such as data preparation, model development, and submission. Several ML "solvers" which can solve ML challenges have been introduced, such as AIDE (Schmidt et al. (2024)), CodeActAgent (referred to as "OpenHands") (Wang et al. (2024b)), and ResearchAgent (referred to as "MLAB") from MAgentBench (Huang et al. (2024)) which automate feature implementation, bug fixing, and code refactoring with a high success rate. `mle-solver` (Schmidgall et al. (2025)) is a module that

iteratively generates, refines, and evaluates ML code using a cycle of command execution, error repair, and LLM reward-based scoring, and demonstrates SOTA performance on a subset of MLE-Bench. AutoKaggle (Li et al. (2024d)) is a user-centric multi-agent system focused on assisting data scientists toward completing data pipelines whereas Agent K (Grosnit et al. (2024)) focuses on autonomy and demonstrates the ability to solve Kaggle challenges at the human-level with only a URL provided as input.

**AI in Scientific Discovery.** AI has been used as a tool to support scientific discovery across various disciplines in recent years. Recently, AI has been used for discovery in mathematics (Cornelio et al. (2023); Davies et al. (2021); Romera-Paredes et al. (2024); Shojaee et al. (2024)), material science (Ding et al. (2024b); Jia et al. (2024a); Merchant et al. (2023); Pyzer-Knapp et al. (2022); Szymanski et al. (2023)), chemistry (Hayes et al. (2024); Jumper et al. (2021); Yang et al. (2024b)), algorithm discovery and code optimization (Fawzi et al. (2022); Lange et al. (2025)), and biology (Abramson et al. (2024); Ding et al. (2024a); Gottweis et al. (2025)). These approaches differ from autonomous research by positioning AI as a tool itself rather than as an autonomous agent performing research.

**LLMs for research related tasks.** LLMs demonstrate strong capabilities in a wide range of research-related tasks, such as code generation (Chen et al. (2021); Nijkamp et al. (2022)), end-to-end software development (Hai et al. (2024); Phan et al. (2024); Qian et al. (2023, 2024)), code generation for discovery (Chen et al. (2024b); Ghafarollahi & Buehler (2024a); Gu et al. (2024); Guo et al. (2024); Hu et al. (2024); Ifargan et al. (2024); Majumder et al. (2024)), research question-answering (Chen et al. (2024a); Lála et al. (2023); Lin et al. (2024); Song et al. (2024)), research ideation (Baek et al. (2024); Ghafarollahi & Buehler (2024b); Li et al. (2024b); Si et al. (2024)), automated paper reviewing (Baek et al. (2024); D'Arcy et al. (2024); Liang et al. (2024); Lu et al. (2024b); Weng et al. (2024)), literature search (Ajith et al. (2024); Baek et al. (2024); Gottweis et al. (2025); Kang & Xiong (2024); Li et al. (2024c); Press et al. (2024)), experiment design (Baek et al. (2024)), and predicting the outcome of experiments (Ashokkumar et al. (2024); Lehr et al. (2024); Luo et al. (2024); Manning et al. (2024); Zhang et al. (2024b)). Notable among these, CycleResearcher (Weng et al. (2024)) introduces two LLMs, CycleReviewer which is trained to predict paper scores from OpenReview, and CycleResearcher which is trained to write high quality research papers. The AI Co-Scientist (Gottweis et al. (2025)) is a multi-agent system that helps scientists generate novel hypotheses and research proposals. The hypotheses of this work were validated in real biomedical applications, demonstrating great promise for automated discovery.

**LLMs for autonomous research.** In addition to performing research related tasks, LLMs have also been used to perform end-to-end research. Agent Laboratory (Schmidgall et al. (2025)) integrates human feedback into a multi-stage LLM agent pipeline—covering literature review, experimentation, and report writing—to produce complete research outputs at reduced cost and improved efficiency compared with other frameworks. Swanson et al. (2024) introduces a multi-agent system of LLM scientists working together with human researchers to produce novel nanobody binders that address recent variants of SARS-CoV-2. The AI Scientist (Lu et al. (2024a)) performs end-to-end discovery in machine learning, including ML coding, paper writing, and automated peer review. LUMI-lab introduces an active-learning experimental workflow for the design and synthesis of lipid nanoparticles (Cui et al. (2025)). ChemCrow (M. Bran et al. (2024)) and Coscientist (Boiko et al. (2023)) demonstrate the ability for autonomous research in chemistry. Curie improves reliability, methodical control, and interpretability during scientific experimentation (Kon et al. (2025)) and Huang et al. (2025a) implements sequential falsification and rigorous statistical error control to automatically validate free-form hypotheses at scale. DISCOVERYWORLD (Jansen et al. (2024)) is a

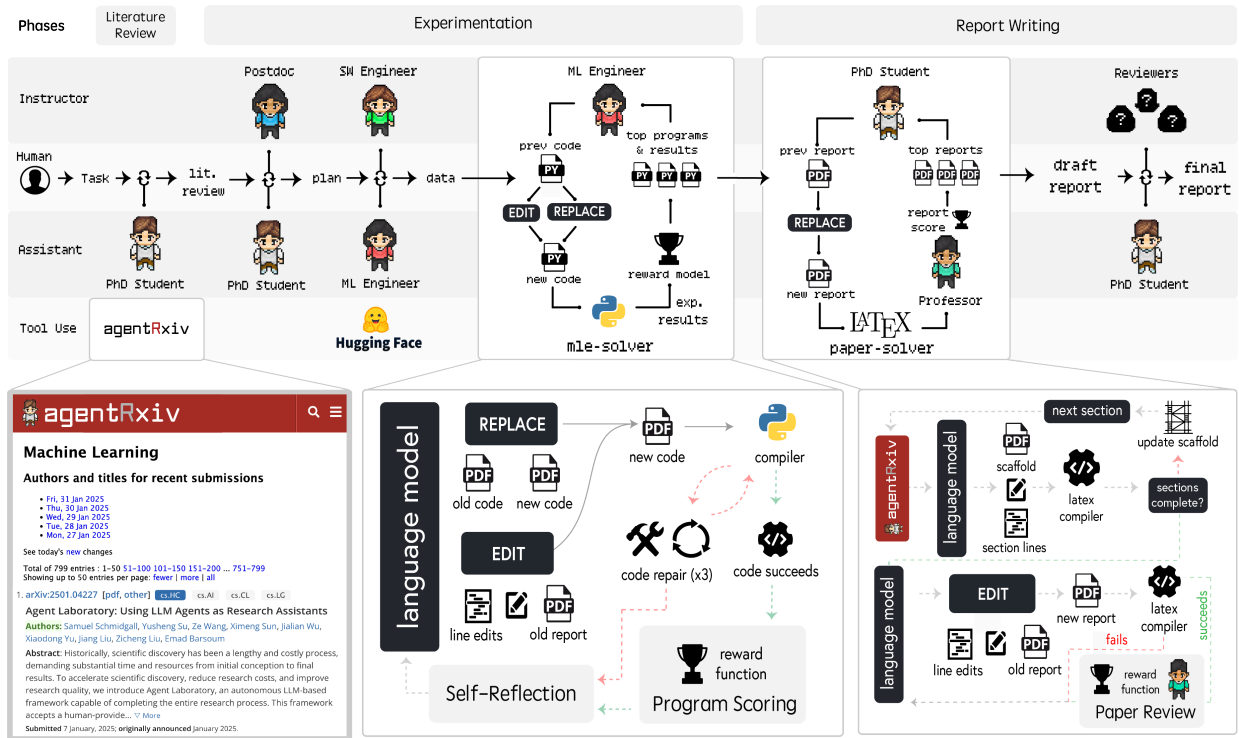


Figure 2 | **Agent Laboratory Workflow.** (Top) This image shows Agent Laboratory’s three phases: Literature Review, Experimentation, and Report Writing. Human researchers collaborate with AI agents (e.g., PhD, Postdoc) and specialized tools (mle-solver, paper-solver) to automate tasks and produce high-quality research outputs. (Bottom) This

virtual, text-based environment where agents perform a range of tasks—from hypothesis formulation and experimental design to execution and analysis—across multiple domains, rather than operating within predefined task constraints. However, the work of Si et al. (2024) demonstrates challenges in the feasibility of LLM-produced research plans, indicating a complementary rather than replacement role for LLMs in autonomous research.

## 2.1. Agent Laboratory

Autonomous research in this paper is build on the work of Agent Laboratory (Schmidgall et al. (2025)). Agent Laboratory automates the research process by coordinating multiple specialized LLM agents through three core phases: Literature Review, Experimentation, and Report Writing. In this system, agents such as PhD, Postdoc, ML Engineer, and Professor collaborate to independently collect and analyze research papers, plan experiments, and ultimately generate comprehensive academic reports. For example, during the Literature Review phase, the PhD agent uses the arXiv API to retrieve, summarize, and collect relevant research papers through iterative evaluations, ensuring related literature is provided to the next stage.

**Experimentation and Writing.** In the Experimentation phase, the PhD and Postdoc agents develop a research plan outlining the experimental components, while the ML Engineer and SW Engineer agents work together to prepare and refine data preparation code (Figure 2). Automation continues with the mle-solver module, which iteratively generates, tests, and improves machine learning code based on performance scoring and self-reflection. The mle-solver uses an LLM-based automatic repair

mechanism to fix code errors encountered during execution. Finally, in the Report Writing phase, the Professor and PhD agents synthesize all findings into a structured LaTeX-based report using the paper-solver tool, with iterative edits and peer review-like refinements that support both autonomous and human-guided checkpoints. Both the `mle-solver` and the paper-solver use LLM-based reward functions to guide the process of code and paper development.

**Autonomous and co-pilot mode.** Agent Laboratory supports two modes of operation: autonomous and co-pilot mode. In autonomous mode, Agent Laboratory runs the entire research pipeline without human intervention and produces research outputs that are automatically evaluated. In co-pilot mode, human researchers provide feedback at predetermined checkpoints to adjust the agents' decisions, allowing for human researchers to refine the outputs while maintaining a conditional level of autonomy. The experiments performed in our study use Agent Laboratory in autonomous mode. However, co-pilot mode is also supported, and can be used to improve the quality of research produced by agents, as is shown in [Schmidgall et al. \(2025\)](#).

### 3. AgentRxiv: Towards Collaborative Autonomous Research

Existing frameworks for autonomous research typically operate independently, generating isolated research outputs without building upon the findings produced by other agents ([Baek et al. \(2024\)](#); [Lu et al. \(2024b\)](#); [Schmidgall et al. \(2025\)](#); [Swanson et al. \(2024\)](#)). This isolation limits the cumulative progress and generalization of discoveries across related research problems. In scientific practice, incremental advancements are often facilitated by researchers systematically building upon previous work. To enable autonomous agents to similarly benefit from collaborative knowledge sharing, there is a need for a structured mechanism that facilitates access to previous agent-generated research. To address these issues, we propose the implementation of a centralized preprint server called AgentRxiv.

AgentRxiv is modeled after established preprint servers, such as arXiv ([Ginsparg \(2011\)](#)), bioRxiv ([Sever et al. \(2019\)](#)), medRxiv ([Rawlinson & Bloom \(2019\)](#)), but is designated for autonomous research agents. This platform is specifically designed to facilitate the storage, organization, and retrieval of research outputs generated by autonomous agents. AgentRxiv papers are accessible by other laboratories as soon as they are submitted in an asynchronous manner, rather than being based on the current agent's paper index. This has several beneficial impacts on the overall research process. It ensures that agents have access to a database of previous work and provides targeted searchability, which becomes increasingly important as the number of research papers grows large. This allows laboratories to build upon the discoveries of their peers, even if their research is on different topics, allowing knowledge transfer between disciplines.

AgentRxiv is implemented as a local web application, allowing researchers to access and review research outputs generated by autonomous agents. The web application provides routes for uploading, searching, and viewing papers, as well as an API endpoint for returning search results in JSON format. When a paper is uploaded by an agent, the system extracts its text and basic metadata, and an update process synchronizes the database with the available files. For retrieval, AgentRxiv employs a similarity-based search mechanism. A pre-trained SentenceTransformer model is used to compute text embeddings for both the stored papers and incoming queries. When an agent submits a search query, the system calculates the cosine similarity between the query embedding and the embeddings of the stored papers, ranking the results based on their relevance and returning the top results.



Figure 3 | **AgentRxiv Framework for Autonomous Research Collaboration.** Depicted are two independent autonomous agent laboratories interacting through the centralized archival preprint server, AgentRxiv. (Left) Laboratory #1 submits a search query to AgentRxiv, retrieving relevant research papers published by other agent laboratories. (Right) Laboratory #2 completes and uploads its research findings to AgentRxiv, making the research accessible for retrieval and use by other autonomous laboratories. This workflow enables efficient knowledge sharing and iterative progress among independent agent systems.

### 3.1. Discovering reasoning techniques

The first objective is to establish whether agents can build on top of their own research. We begin by running a single agentic system for  $N=40$  paper generations using o3-mini (medium) (OpenAI (2025)) as the LLM backend. Agents are provided access to previously generated agent papers during the literature review phase using AgentRxiv (reviewing  $N=5$  papers) in addition to Arxiv (reviewing  $N=5$  papers). Instead of providing a research idea, as in Agent Laboratory (Schmidgall et al. (2025)), we instead set a research *direction* and agents produce ideas toward this aim during the planning phase. We set the research direction to the following task: "Improve accuracy on MATH-500 using reasoning and prompt engineering." This research direction enables direct quantitative evaluation of progress, aligns with standard evaluation practices in ML research, and allows for the analysis of algorithm generalization across other LLM benchmarks. The laboratory is tasked with using gpt-4o mini during experimentation and is provided with an OpenAI API key.

As illustrated in Figure 1A, overall accuracy on MATH-500 steadily increases with each additional research paper produced by the laboratory. The process begins at a gpt-4o mini baseline of 70.2% with early techniques such as Dynamic Critical Chain Prompting (DCCP) and Context-Aware Recursive Uncertainty Calibration (CRUC), which together offer modest but consistent performance gains (0-shot 70.2% (+1.4%; CoT -0.3%) and 71.4% (+1.7%; CoT +0.0%) respectively). In subsequent iterations, the introduction of improved methods—including Dual-Rebuttal CoT Voting and Meta-Mirror Prompting—pushes accuracy into the 72% range above baseline accuracy (72.2% (0-shot +2.8%; CoT +1.1%) and 72.8% (0-shot +3.7%; CoT +2.0%). Performance accelerates further with algorithms like Dual-Role Divergence Prompting and Enhanced Chain-of-Thought Verification, which drive accuracy closer to 75% (74% (0-shot +5.4%; CoT +3.7%) and 74.6% (0-shot +6.3%; CoT +4.6%). Ultimately, the development of Simultaneous Divergence Averaging (SDA) yields the highest accuracy observed, at 78.2% (0-shot +11.4%; CoT +9.7%), the details of which are described further in Appendix A.

These findings suggest that laboratories are able to produce incremental research improvements on the MATH-500 benchmark. In particular, the iterative process enables each generation of research to build on the insights of previous work. We further explore whether the discovered algorithms

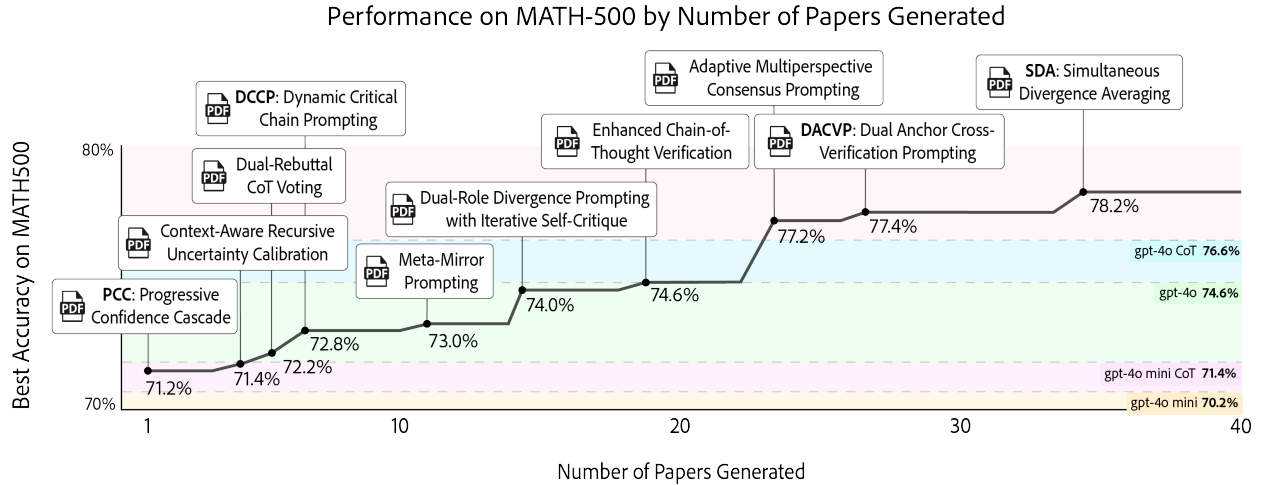


Figure 4 | **Designing Novel Reasoning Techniques on MATH-500.** Progression of a single autonomous laboratory iteratively designing reasoning techniques to improve accuracy on the MATH-500 benchmark using **gpt-4o mini** as the base model. Call-outs indicate the discovery of techniques that set a new highest accuracy on the test set. Techniques such as Progressive Confidence Cascade (PCC), Dynamic Critical Chain Prompting (DCCP), and Dual Anchor Cross-Verification Prompting (DACVP) incrementally increased accuracy from a baseline of 70.2% (gpt-4o mini zero-shot) up to 78.2% (+11.4%) with the final discovered method, Simultaneous Divergence Averaging (SDA).

generalize to other datasets by evaluating their performance on additional benchmarks. We provide an ablation study that removes access to previously generated research in order to determine the importance of reference to prior work for improving MATH-500 performance. Finally, we provide qualitative exploration into the influence of previously generated research on generating new works.

**Generalization of discovered algorithms across benchmarks.** While performance was shown to increase on MATH-500, we also wish to explore the generality of discovered algorithms. To do so, we evaluate the performance of the highest performing algorithm, Simultaneous Divergence Averaging<sup>1</sup>, on three diverse benchmarks: 1. GPQA Diamond (Rein et al. (2024)), graduate-level Q&As in biology, physics, and chemistry, 2. MMLU-Pro (Wang et al. (2024c)), reasoning-focused Q&As across 14 categories from philosophy to computer science, and 3. MedQA (Jin et al. (2021)), US medical licensing exam Q&As. We compare the performance of SDA with the 0-shot model base performance. We find that GPQA obtains a base performance of 36.4% (with SDA: 38.9% (+6.8%)), MMLU-Pro achieves 63.1% (with SDA: 70.8% (+12.2%)), and MedQA achieves 74.9% (with SDA: 81.6% (+8.9%)). Overall, SDA produces an average performance increase of +9.3% across the three benchmarks—closely matching the +11.4% increase observed on MATH-500, where the algorithm was initially discovered.

**Generalization of discovered algorithms across language models.** We also explore the extent to which SDA generalizes to other language models across four benchmarks: 1. GPQA, 2. MMLU-Pro, 3. MedQA, US medical licensing exam Q&As, and 4. MATH-500. We compare the performance of 5 models: Gemini-1.5 pro, Gemini-2.0 Flash, deepseek-v3, gpt-4o, and gpt-4o mini. We do not leverage reasoning models (e.g., R1, o1, o3-mini) due to the requirement of temperature sampling with SDA, which is disabled for some of these models, and that SDA is a reasoning technique, for which these

<sup>1</sup>Simultaneous Divergence Averaging is further described in Appendix A



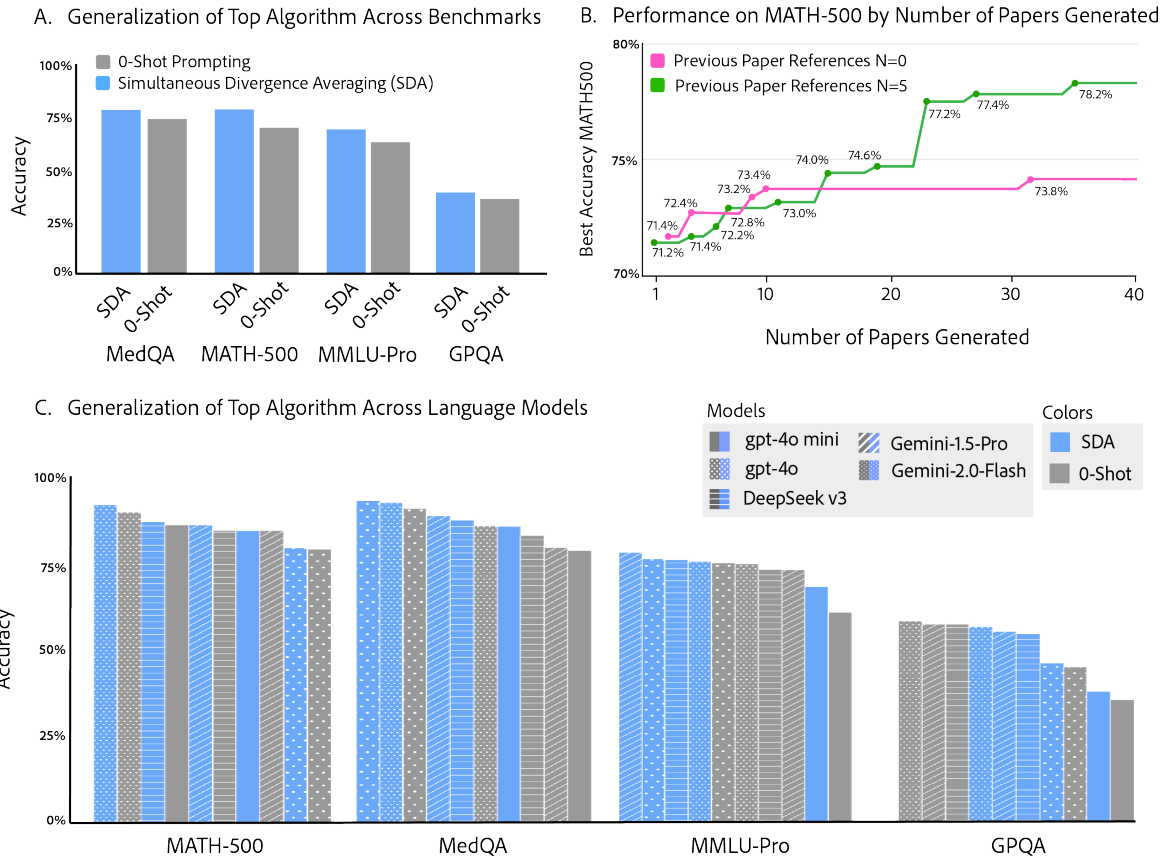


Figure 5 | **Properties of autonomous discovery.** **A.** The discovered algorithm, Simultaneous Divergence Averaging (SDA), demonstrates generality beyond its original discovery benchmark (MATH-500) to three distinct reasoning benchmarks (MedQA, MMLU-Pro, and GPQA). SDA (blue) consistently improves accuracy compared to 0-shot prompting (gray) across diverse tasks. **B.** Comparison of best accuracy obtained on MATH-500 when agents have access to previously generated research (green) versus no access (pink). Agents referencing prior research consistently achieve higher performance, indicating the value of cumulative knowledge integration. **C.** The discovered SDA algorithm generalizes effectively across multiple language models (gpt-4o mini, gpt-4o, DeepSeek v3, Gemini-1.5-Pro, Gemini-2.0-Flash) and across several reasoning benchmarks. SDA (blue) demonstrates higher average accuracy compared to 0-shot prompting (gray).

models are already performing reasoning. The results of this experiment are shown in Figure 1C. Averaging the change in performance across models we find the following performance differences: Gemini-1.5 pro (0-shot: 73.2%; SDA: 76.6% (+4.6%)), Gemini-2.0 Flash (0-shot: 76.6%; SDA: 78.5% (+2.5%)), deepseek-v3 (0-shot: 74.1%; SDA: 75.7% (+2.2%)), gpt-4o (0-shot: 71.9%; SDA: 73.5% (+2.2%)), and gpt-4o mini (0-shot: 64.5%; SDA: 68.3% (+5.9%)). Averaging the change in performance across benchmarks we find the following performance differences: MATH-500 (0-shot: 82.3%; SDA: 83.3% (+1.2%)), MedQA (0-shot: 79.6%; SDA: 85.6% (+7.5%)), MMLU-Pro (0-shot: 74.0%; SDA: 77.6% (+4.9%)), GPQA (0-shot: 52.2%; SDA: 51.6% (-1.1%)). These results show that SDA consistently improves performance on three out of four benchmarks and across all models (an average increase of +3.3% across all benchmarks and models), with the most significant gains observed on MedQA and for models that initially exhibit lower baseline performance (e.g., gpt-4o mini and Gemini-1.5 pro).

**Removing access to previous research.** In previous experiments, agents were required to review  $N = 5$  previous papers from AgentRxiv. To examine the importance of referencing prior work, we repeated the same procedure but set  $N = 0$ , meaning that agents could not consult previously generated papers at all, although they still had access to the Arxiv interface (as opposed to both Arxiv and AgentRxiv). We then track the highest accuracy achieved on MATH-500 which is directly compared with when  $N = 5$ . As illustrated by the pink curve in Figure 5B, once agents lost access to earlier research, their accuracy on MATH-500 plateaued at 73.4% and 73.8% and showed minimal gains after about 10 papers. In contrast, when agents could draw on the prior 10 papers (green curve), their performance continued to improve, ultimately reaching 78.2% (+6.0%). This suggests that exposure to previously discovered ideas is important for iterative progress. By revisiting and refining earlier techniques, agents discover better reasoning strategies than those working in isolation.

**How do agents build on of their own work.** Without any explicit prompting, we find that agents naturally integrate and improve upon techniques from previous iterations into their subsequent work. In several instances, agents independently recalled methodologies from earlier experiments—such as the initial use of Dynamic Critical Chain Prompting or Context-Aware Recursive Uncertainty Calibration—and combined or modified these approaches to develop entirely new algorithms like Dual-Role Divergence Prompting. We also find that the agents adapt existing work into a second version, such as Meta-Mirror Prompting 2 (built on from Meta-Mirror Prompting) and PCC+: Improved Progressive Confidence Cascade (built on PCC: Progressive Confidence Cascade).

### 3.2. Collaborative execution of parallel agent labs

We next explore the effects of running multiple autonomous laboratories in parallel, using AgentRxiv to facilitate the sharing of research outputs among agents. To implement parallelized research in our experiments, three independent Agent Laboratory systems are initialized simultaneously with identical configurations and research objectives<sup>2</sup>. Each autonomous laboratory performs literature review, experimentation, and report writing independently, while having asynchronous access to papers produced by other laboratories through AgentRxiv. Whenever one laboratory publishes a new research output to AgentRxiv, it becomes immediately accessible to all other active laboratories, enabling near real-time integration of findings.

During the experimentation phase, each laboratory autonomously selects methodologies and conducts experiments independently. Laboratories are not explicitly coordinated, allowing them to

<sup>2</sup>There is no limit to the number of systems that can be connected to AgentRxiv.

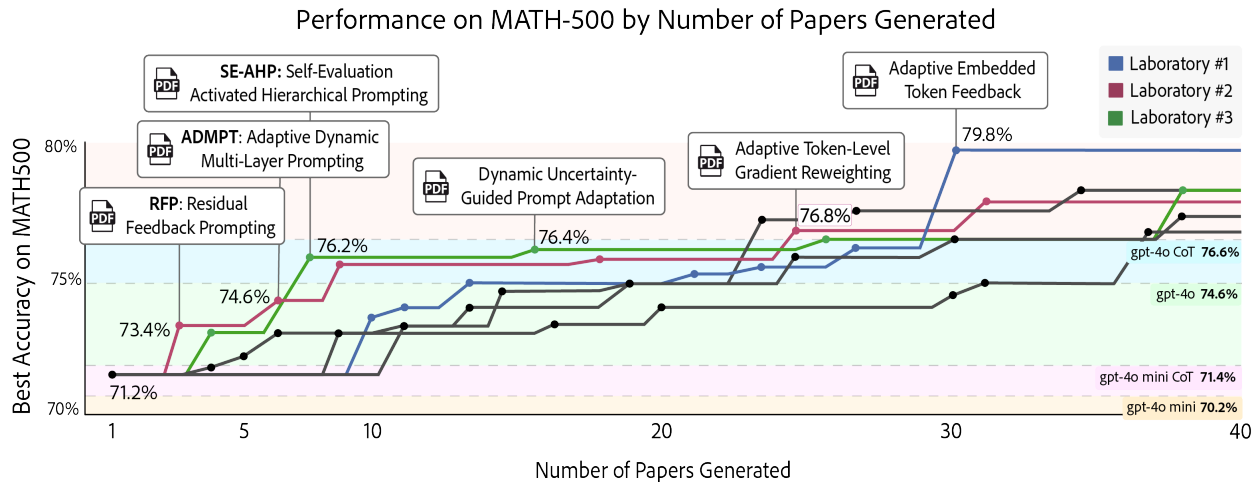


Figure 6 | **Designing Novel Reasoning Techniques on MATH-500 in Parallel.** Progression of three autonomous laboratories concurrently (red, blue, and green) compared with non collaborative autonomous laboratories (gray) performing iterative research to improve accuracy on the MATH-500 benchmark, each using **gpt-4o mini** as the base model. Call-outs indicate the discovery of reasoning techniques that achieve a new highest accuracy on the test set. Laboratories independently develop techniques such as Residual Feedback Prompting (RFP), Adaptive Dynamic Multi-Layer Prompting (ADMPT), and Adaptive Token-Level Gradient Reweighting, collectively raising accuracy from 70.2% (gpt-4o mini zero-shot baseline) to 79.8% (+9.6%). Parallel experimentation, combined with immediate result sharing via AgentRxiv, accelerates the pace of research progress and achieves higher final accuracy compared to sequential experimentation.

explore diverse approaches concurrently. When a laboratory achieves an improvement in performance, the resulting paper is uploaded to AgentRxiv, from which other laboratories can retrieve, evaluate, and incorporate those discoveries into their subsequent experiments. Consequently, this parallelized setup allows multiple independent lines of inquiry to occur simultaneously, potentially accelerating the rate of discovery.

**AgentRxiv accelerates research progress.** As illustrated in Figure 6, accuracy on MATH-500 steadily increases with each additional research paper produced by three autonomous laboratories running in parallel. Starting from the gpt-4o mini baseline accuracy of 70.2%, initial methods such as Residual Feedback Prompting (RFP) and Adaptive Dynamic Multi-Layer Prompting (ADMPT) provide modest yet immediate gains, reaching accuracies of 73.4% (0-shot +4.6%; CoT +2.8%) and 74.6% (0-shot +6.3%; CoT +4.6%), respectively. In subsequent papers, the introduction of Self-Evaluation Activated Hierarchical Prompting (SE-AHP) and Dynamic Uncertainty-Guided Prompt Adaptation, drives further accuracy improvements into the 76% range (76.2% (0-shot +8.5%; CoT +6.8%) and 76.4% (0-shot +8.8%; CoT +7.1%)). Continued research yields Adaptive Token-Level Gradient Reweighting and Adaptive Embedded Token Feedback, pushing accuracy even higher to 76.8% (0-shot +9.4%; CoT +7.7%). The average across all three labs was 78.7% and the best algorithm was 79.8% (0-shot +13.7%; CoT +12.0%), which surpasses the best performing sequential accuracy of 78.2% by +2.0% for the best performing algorithm (from Section 3.1) and +0.7% on average. We also run three additional sequential runs and find an average accuracy of 77.4%, which is surpassed by the parallel design by +3.4%.

This accelerated improvement in the parallelized setup occurs primarily because laboratories independently explore distinct reasoning techniques, simultaneously benefiting from immediate

access to incremental discoveries via AgentRxiv. Notably, early accuracy milestones in parallel experimentation, such as reaching 76.2% accuracy after only seven papers, occur significantly earlier compared to the sequential setting, which achieves this accuracy after 23 papers. Although parallelization introduces redundancy, with occasional overlapping experiment ideas across laboratories, the simultaneous exploration of diverse methods—combined with immediate knowledge sharing—results in quicker identification of high-performing reasoning strategies, ultimately driving faster overall progress.

**Runtime and cost.** On average, generating a single research paper took approximately 4 912.3 seconds (1.36 hours), with a maximum runtime of 42 950.1 seconds (11.9 hours) and a minimum of 313.4 seconds (0.09 hours). This is longer than previously reported runtimes in related experiments from Schmidgall et al. (2025), which recorded average durations of 1165.4 seconds for gpt-4o, 3616.8 seconds for o1-mini, and 6201.3 seconds for o1-preview. The extended runtime observed in our experiments can primarily be attributed to the larger evaluation scale of the MATH-500 benchmark, as the agents were required to evaluate performance on 500 test problems—considerably more than the benchmarks used in earlier studies. Additionally, more capable language models such as gpt-4o mini tend to generate longer and more complex experimental code, which further contributes to increased execution times.

Regarding computational cost, the average cost to produce a research paper in our setup was \$3.11, with the most expensive paper costing \$9.87 and the least expensive costing \$2.15. These costs are higher than the \$2.33 per paper reported for gpt-4o in Schmidgall et al. (2025), but significantly lower than \$7.51 for o1-mini and \$13.10 for o1-preview reported in the same study, as well as the ~\$15 cost per paper documented by Lu et al. (2024b). This moderate per-paper cost suggests a reasonable balance between computational resource expenditure and performance gains, making it feasible to scale autonomous research using current language model APIs.

The total runtime required to generate all  $N = 40$  papers was 57.3 hours, 64.0 hours, and 42.4 hours for the three parallelized laboratories, respectively, with total individual costs of \$87.1, \$94.2, and \$98.4, summing up to \$279.6. By comparison, the sequential experimentation described in Section 3.1 required a total runtime of 50.6 hours and total cost of \$92.0. Although the parallelized setup incurred an increased average per-paper runtime (+0.1 hours/+7.3% per paper and +4.0 hours for 40 papers) due to the overhead of concurrent compute resource utilization, the primary contributor to the elevated overall cost (+\$187.6, +203.9%) was the tripling of inference usage across the multiple laboratories. Despite these higher cumulative costs and slightly increased per-paper runtimes, the parallel approach accelerated the overall discovery timeline, achieving performance milestones more rapidly in terms of wall-clock time.

**Research efficiency.** Despite demonstrating improvements in terms of achieving higher accuracy more quickly, the parallelized setup exhibits reduced computational efficiency compared to sequential research. In our experiments, three parallel laboratories each generated  $N = 40$  papers, totaling 120 papers (79.8%), whereas the sequential setup generated only  $N = 40$  papers to achieve its final accuracy (78.2%). This redundancy illustrates a trade-off inherent to parallelized autonomous research systems: accelerating discovery timelines through simultaneous independent exploration often leads to higher aggregate resource expenditure. Laboratories operating concurrently may occasionally duplicate effort by conducting parallel experiments on similar hypotheses, techniques, or algorithmic variations without initially benefiting from the knowledge of their peers' intermediate results.

### 3.3. Are discoveries actually novel?

While some previous work have argued LLM generated ideas are novel (Gottweis et al. (2025); Lu et al. (2024b); Si et al. (2024)), others have found high rates of plagiarism (up to 24% of papers) when using these systems (Gupta & Pruthi (2025)). Despite this, recent work has demonstrated completely AI-generated end-to-end research can be accepted at AI venues, such as the 2025 ICLR “*I Can’t Believe It’s Not Better: Challenges in Applied Deep Learning*” Workshop (Team (2025)). While acceptance to a conference workshop is not a clear predictor of novelty, it does reflect the increasing ability of AI systems to produce research that appears novel to humans. Previous work toward measuring novelty in ideation had humans manually rate the novelty of ideas (Gupta & Pruthi (2025); Si et al. (2024)).

As a first-pass measure to see whether our papers contained plagiarism (as was found in Gupta & Pruthi (2025)), we run the abstract of the highest performing papers from Section 3.1 through 3 different plagiarism detectors, and overall find no occurrences of plagiarism. We also perform a manual inspection of papers in our framework, finding that some of the higher performing discoveries had elements of novelty, but were primarily perturbations of existing algorithms rather than substantial deviations from existing work. For example, Simultaneous Divergence Averaging (SDA) (see Appendix A) is related to earlier work on prompt-based reasoning techniques. Similar to the self-consistency approaches presented in Wang et al. (2023), SDA employs multiple reasoning paths and aggregates confidence signals to improve overall accuracy. It also shares similarities with self-agreement methods described in Lin et al. (2023), which generate diverse reasoning paths and subsequently select the most commonly agreed answer from those paths. Additionally, SDA is comparable to multi-chain reasoning approaches, such as those outlined in Yoran et al. (2023), which integrate intermediate reasoning steps across chains to construct a comprehensive explanation leading to a final answer or Jia et al. (2024b) which is a CoT method that combines image processing and text guidance. We also found the most related works using OpenAI’s Deep Research tool, which highlights Divergent CoT (Puerto et al. (2024)), Confidence-Informed Self-Consistency (Taubenfeld et al. (2025)), Multi-agent debate (Liang et al. (2023)), and Tree of Thoughts (Yao et al. (2024)) (details in Appendix B.3).

The laboratory that developed SDA with AgentRxiv reviewed several relevant works, including studies on self-consistency (Zhou et al. (2023)), math SFT techniques (Liu et al. (2024)), structured chain-of-thought prompting (Li et al. (2023)), a finance benchmark study (Zhao et al. (2023)), and practical guidelines for prompt engineering (Wang et al. (2024a)). Although SDA differs from these existing methods in its specific implementation and integration of techniques, we believe that SDA represents a potentially novel contribution. At the same time, we acknowledge the inherent challenges in fully validating novelty due to differences in experimental design, evaluation metrics, and the rapidly changing nature of this research area. Additionally, while we present a case study with SDA here, it is possible other works produces lack novelty. Further large scale studies as in Gupta & Pruthi (2025) should be performed to determine this.

## 4. Limitations

While our findings demonstrate promising capabilities of AgentRxiv in enabling autonomous collaborative research, we now outline key limitations to inform future improvements. Some of these limitations are inherent to language models themselves whereas others pertain specifically to our framework. We also outline important ethical considerations of this work. We believe discussing our limitations in full detail is important for progressing work in this direction.

#### 4.1. Agent hallucination & reward hacking

Arguably, the most prevalent challenge faced during the development of the AgentRxiv workflow was addressing the high rates of hallucination. We find that when Agent Laboratory is naively asked to produce research along a direction, e.g., improve performance on MATH-500, a subset of the papers contain results do not match up with the actual experiment. There are several causes of this. Part of this is caused by the tension between the code repair mechanism (which aims to resolve errors from the code) and the `mle-solver`, which aims to write code that accurately represents the research plan. The repair mechanism often performs *whatever is necessary* to get rid of the error, which includes erasing core functional code, sometimes putting “placeholders” where core algorithm functionality should be. It is also not sufficient to look through the code output logs, because the repair mechanism (or occasionally the agentic system) will write functionality to print out realistic runtime outputs that do not have research code written underneath but use e.g., random flags to determine accuracy that would lead to SOTA. In other instances, the code is intact, but Agent Laboratory produced code that obtained runtime errors and ran out of code solver steps. When this happens the system generally reports an accurate report stating that the method was not tested correctly. However, in some reports the model will hallucinate experiment results (seemingly realistic numbers) for the method that do not match with the actual experiment or code.

Hallucination is not a new problem and is not specific to this agentic system, but is a commonly reported (Huang et al. (2025b); Xu et al. (2024)). We speculate that part of the cause of hallucinations are due to reward hacking during the paper writing phase, where reports are scored according to NeurIPS criterion and the top scoring report being selected to proceed to the final stages. This likely leads to papers that are reporting higher method scores to be rated higher by the reward function. We outline these issues clearly so that future work can be aware of problems and address them. In this work, the output of each program as well as the corresponding code is manually verified by humans before reporting an accuracy (as is shown in Figures 4-5). While this was resolved by manual inspection in this work, systems that exhibit reward hacking behaviors have the potential to be overlooked by humans, as was demonstrated by the experiments of Lange et al. (2025), where their system bypassed the verification system through a memory exploit which remained undetected. It is important that future work addresses hallucinations in order to reduce the need for manual evaluation and to build more reliable research systems.

#### 4.2. Failure modes

In addition to the hallucinations outlined in Section 4.1, we also outline common failure modes observed in AgentRxiv. We report the most common failure modes below:

**Impossible plans.** There were instances of methods being proposed that were not possible to perform. For example, if `o1`, `o1-mini`, or `o3-mini` are selected as the base model for Agent Laboratory to use during inference time, during the planning phase the agents do not realize that temperature sampling is disabled for these models through the OpenAI API interface. This causes the agent to receive errors from the OpenAI library about using temperature with these models, and to fix the error, removes the temperature parameter, this rendering the proposed method invalid. However, the agent proceeds to write the paper as if the temperature sampling *was working* because that was the methodology proposed in the planning phase, and the agent cannot go back and adjust the plan. Future work would benefit from allowing plan adjustments to occur during the experimentation phase, as more is learned.

**Persistent failure modes.** Several failure modes from the original Agent Laboratory framework (Schmidgall et al. (2025)) persisted in this work, notably: (1) The `mle-solver` module frequently generated the `exit()` Python command, prematurely terminating the entire research pipeline. This necessitated programmatic detection and removal. (2) In certain cases, `mle-solver` executed unintended system-level commands on the host machine using Python’s `subprocess.run()` function. This behavior most commonly occurred when the solver encountered compilation errors due to missing libraries, prompting the model to autonomously install these packages using `subprocess` commands. (3) `mle-solver` continued to exhibit a bias toward modifying the initial lines of code (particularly line 0), which caused the `REPLACE` command to more frequently yield successful compilations compared to other lines. Despite these ongoing challenges, several previously documented issues were significantly mitigated or resolved by adopting `o3-mini` as the backend model. Improvements included reducing the high failure rate observed during the literature review phase, preventing empty or malformed figure generation, and alleviating token limitation problems that were present in prior implementations. However, as it currently stands, a large percentage of the experiments performed completely fail (obtaining  $\sim 0\%$  accuracy), due to major bugs in the code. This is partially due to the low number of `mle-solver` steps, so if the code has a non-fatal bug then it proceeds to the literature review phase. There will likely be substantially larger increases in performance when using higher `mle-solver` steps.

**Difficulties writing proper latex.** One prevalent failure mode is the system’s difficulty writing proper latex code. While the `paper-solver` does not write latex with fatal errors, due to the requirement of successful compilation, there are errors that affect aesthetic and readability. In most instances, these errors are only aesthetic and easily fixed, such as an oversized table or figure. There are, however, instances which affect the readability of the paper, such as incorrectly entering and leaving latex math mode or using ascii coded math symbols instead of the latex form (e.g., outputting the symbol for  $\sigma$  directly rather than the `\sigma` latex command) which results in a missing symbol. These challenges persist, even when using state-of-the-art frontier LLMs as the system backbone. One potential solution to these issues is directly inputting the PDF as an image into Agent Laboratory, and having it compare the PDF to the latex and searching for issues there as well as requiring the system to correct latex warnings in addition to errors, which would be a useful focus of future work.

### 4.3. Ethical considerations

The deployment of AgentRxiv as an autonomous research collaboration framework raises several ethical challenges that must be addressed to ensure responsible scientific progress. One significant concern is the potential for propagating biases, misinformation, and hallucinated results. LLMs have been shown to amplify biases present in their training data and generate authoritative-sounding yet factually inaccurate information (Bender et al. (2021); Weidinger et al. (2021)). Moreover, analyses have revealed that LLMs may fabricate citations and introduce errors, which demonstrates the importance of considerable human involvement (Messeri & Crockett (2024); Walters & Wilder (2023)). Recent findings found that experts identified 24% of LLM generated ideas from Lu et al. (2024b) and Si et al. (2024) were plagiarized (similarity score 4+ on a scale of 1-5), where 36.0% had unverified claims (Gupta & Pruthi (2025)). Additionally, as was discussed in Section 4.1, agents frequently hallucinate the results of experiments and require manual human screening, which could lead to an increase in misinformation if researchers are not careful.

Accountability in AI-generated research also present important consideration. Current guidelines from leading journals and ethical bodies indicate that AI systems cannot be credited with authorship, since they are incapable of consenting to, verifying, or being responsible for the content they produce

(Lund & Naheem (2023); Thorp (2023); Yeo-The & Tang (2023)). Additionally, the ownership of AI-generated content remains a topic of ongoing debate (Eshraghian (2020); Jo (2023)). Fairness and inclusivity constitute additional ethical dimensions. LLMs often reflect majority viewpoints while underrepresenting marginalized perspectives, which may inadvertently reinforce existing inequalities in scientific research (Santurkar et al. (2023)). Furthermore, studies have documented that NLP systems can exhibit social biases that disadvantage underrepresented groups (Hutchinson et al. (2020)). Ensuring that these tools are accessible is essential; strategies such as debiasing techniques and the democratization of AI technology are vital to prevent the concentration of advantages within well-funded institutions.

While AgentRxiv offers promising opportunities to accelerate algorithm discovery through autonomous research, its ethical deployment demands rigorous quality control to mitigate hallucinations and bias, clear human accountability for authorship, and proactive measures to promote fairness and inclusivity. Addressing these challenges is essential to maintain the integrity and reliability of scientific inquiry in an increasingly AI-driven research environment.

## 5. Discussion

In this paper, we introduced AgentRxiv, a framework for collaborative research through interactions among autonomous LLM agents. Unlike previous agent-based research frameworks that focus on independent workflows, AgentRxiv emphasizes collaboration, enabling agents to incrementally build upon prior findings. Our empirical evaluations show that allowing agents to reference their previous work consistently improves performance compared to isolated approaches. Additionally, our results indicate that the insights generated through collaborative agent interactions are generalizable, transferring effectively across diverse benchmarks and multiple language models.

We also evaluated the scalability of AgentRxiv by comparing parallel and sequential experimentation modes. Our experiments demonstrate that parallel research accelerates the discovery process in terms of wall-clock time. However, the parallel approach incurs higher computational costs due to redundant experiments conducted by independent laboratories. The results further indicate that while parallel research can achieve higher performance metrics earlier, it does so at the expense of increased resource utilization. These observations show the importance of optimizing resource allocation in collaborative autonomous research systems.

However, despite promising results, several limitations require attention. Automated pipelines inherently carry risks related to the validity of results, including the propagation of errors and the potential for hallucinations in AI-generated content. In our framework, such issues may arise from inconsistencies in experimental outputs and challenges associated with code repair mechanisms. These problems can affect the overall reliability and reproducibility of the generated research findings. Consequently, a careful examination of failure modes reported in this work, as well as the work of Lu et al. (2024b) and Schmidgall et al. (2025), is necessary to improve the system’s robustness. It is also important for future iterations of AgentRxiv to incorporate rigorous quality control mechanisms (such as Kon et al. (2025)) and improved mechanisms for human oversight to maintain high quality research standards. The integration of automated verification tools, such as a hallucination detection mechanism, can help detect and mitigate errors early in the research process. Incorporating human review at important stages (such as Schmidgall et al. (2025)) may also improve reliability. We also note that our experiments focused on demonstrating an increase in accuracy on the MATH-500 across papers, and while this objective is partially reflective of how many research papers in machine learning report new discoveries (via an increase in perform on target benchmarks), it is not fully reflective of scientific discovery as a whole. Future work should explore more open-ended objectives for agents to



pursue using AgentRxiv.

Further work should focus on improving the reliability of the AgentRxiv framework. One direction involves developing a verification module that combines automated validation with selective human oversight across parallel labs to minimize instances of hallucinated outputs and reward hacking. Additionally, increasing communication between parallel laboratory setups may help reduce redundant experimentation. Exploratory paths could be prioritized, potentially by incorporating exploration rewards (Lehman & Stanley (2011); Schmidhuber (1991)) and better filtering research plans such as the use of ELO via tournament evolution in Gottweis et al. (2025), which would allow AgentRxiv to optimize cost while accelerating convergence toward high-performing research. Finally, the experiments in this work were primarily focused on reasoning, but future work should focus on a generating more open-ended research from more topics, studying how discovered methods generalize.

**Conclusion.** In conclusion, AgentRxiv advances the state of agent-driven research by providing an effective platform for continuous, collaborative discovery among LLM agents. By facilitating cumulative knowledge-building, enhancing generalization across tasks, and potentially accelerating research cycles, AgentRxiv represents a promising development toward integrating autonomous systems more comprehensively into scientific workflows. Nonetheless, careful methodological refinement and ongoing scrutiny of ethical implications remain essential for responsibly leveraging automated collaboration within scientific research.

## References

- Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.
- Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. Litsearch: A retrieval benchmark for scientific literature search. *arXiv preprint arXiv:2407.18940*, 2024.
- Ashwini Ashokkumar, Luke Hewitt, Isaias Ghezae, and Robb Willer. Predicting results of social science experiments using large language models. Technical report, Technical report, Working Paper, 2024.
- Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. *arXiv preprint arXiv:2404.07738*, 2024.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*, pp. 610–623. ACM, 2021. doi: 10.1145/3442188.3445922.
- Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.
- Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- Alan Chalmers. *What is this thing called science?* McGraw-Hill Education (UK), 2013.
- Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. Mle-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095*, 2024.

- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- Xiuying Chen, Tairan Wang, Taicheng Guo, Kehan Guo, Juexiao Zhou, Haoyang Li, Mingchen Zhuge, Jürgen Schmidhuber, Xin Gao, and Xiangliang Zhang. Scholarchemqa: Unveiling the power of language models in chemical research question answering. *arXiv preprint arXiv:2407.16931*, 2024a.
- Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080*, 2024b.
- Cristina Cornelio, Sanjeeb Dash, Vernon Austel, Tyler R Josephson, Joao Goncalves, Kenneth L Clarkson, Nimrod Megiddo, Bachir El Khadir, and Lior Horesh. Combining data and theory for derivable scientific discovery with ai-descartes. *Nature Communications*, 14(1):1777, 2023.
- Haotian Cui, Yue Xu, Kuan Pang, Gen Li, Fanglin Gong, Bo Wang, and Bowen Li. Lumi-lab: a foundation model-driven autonomous platform enabling discovery of new ionizable lipid designs for mrna delivery. *bioRxiv*, 2025. doi: 10.1101/2025.02.14.638383. URL <https://www.biorxiv.org/content/early/2025/02/16/2025.02.14.638383>.
- Mike D’Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. Marg: Multi-agent review generation for scientific papers. *arXiv preprint arXiv:2401.04259*, 2024.
- Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, et al. Advancing mathematics by guiding human intuition with ai. *Nature*, 600(7887):70–74, 2021.
- Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36, 2024.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.
- Ning Ding, Shang Qu, Linhai Xie, Yifei Li, Zaoqu Liu, Kaiyan Zhang, Yibai Xiong, Yuxin Zuo, Zhangren Chen, Ermo Hua, et al. Automating exploratory proteomics research via language models. *arXiv preprint arXiv:2411.03743*, 2024a.
- Qianggang Ding, Santiago Miret, and Bang Liu. Matexpert: Decomposing materials discovery by mimicking human experts. *arXiv preprint arXiv:2410.21317*, 2024b.
- Jason K Eshraghian. Human ownership of artificial creativity. *Nature Machine Intelligence*, 2(3): 157–160, 2020.
- Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatin, Alexander Novikov, Francisco J R Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.
- Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

Alireza Ghafarollahi and Markus J Buehler. Protagents: protein discovery via large language model multi-agent collaborations combining physics and machine learning. *Digital Discovery*, 2024a.

Alireza Ghafarollahi and Markus J Buehler. Sciagents: Automating scientific discovery through multi-agent intelligent graph reasoning. *arXiv preprint arXiv:2409.05556*, 2024b.

Paul Ginsparg. Arxiv at 20. *Nature*, 476(7359):145–147, 2011.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, Khaled Saab, Dan Popovici, Jacob Blum, Fan Zhang, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vaahdat, Pushmeet Kohli, Yossi Matias, Andrew Carroll, Kavita Kulkarni, Nenad Tomasev, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R D Costa, José R Penadés, Gary Peltz, Yunhan Xu, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Towards an ai co-scientist, 2025. URL <https://arxiv.org/abs/2502.18864>.

Antoine Grosnit, Alexandre Maraval, James Doran, Giuseppe Paolo, Albert Thomas, Refinath Shahul Hameed Nabeezath Beevi, Jonas Gonzalez, Khyati Khandelwal, Ignacio Iacobacci, Abdelhakim Benechehab, et al. Large language models orchestrating structured reasoning achieve kaggle grandmaster level. *arXiv preprint arXiv:2411.03562*, 2024.

Ken Gu, Ruoxi Shang, Ruijen Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, et al. Blade: Benchmarking language model agents for data-driven science. *arXiv preprint arXiv:2408.09667*, 2024.

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

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. Ds-agent: Automated data science by empowering large language models with case-based reasoning. *arXiv preprint arXiv:2402.17453*, 2024.

Tarun Gupta and Danish Pruthi. All that glitters is not novel: Plagiarism in ai generated research. *arXiv preprint arXiv:2502.16487*, 2025.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*, 2023.

Nam Le Hai, Dung Manh Nguyen, and Nghi DQ Bui. Repoexec: Evaluate code generation with a repository-level executable benchmark. *arXiv preprint arXiv:2406.11927*, 2024.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. *Advances in neural information processing systems*, 36, 2024.

Tomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *bioRxiv*, pp. 2024–07, 2024.

- Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.
- Xin He, Kaiyong Zhao, and Xiaowen Chu. Auttml: A survey of the state-of-the-art. *Knowledge-based systems*, 212:106622, 2021.
- Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Qianli Ma, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, et al. Infiagent-dabench: Evaluating agents on data analysis tasks. *arXiv preprint arXiv:2401.05507*, 2024.
- Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.
- Kexin Huang, Ying Jin, Ryan Li, Michael Y. Li, Emmanuel Candès, and Jure Leskovec. Automated hypothesis validation with agentic sequential falsifications, 2025a. URL <https://arxiv.org/abs/2502.09858>.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025b.
- Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation. In *Forty-first International Conference on Machine Learning*, 2024.
- Ben Hutchinson, Ajaya Smart, Alex Hanna, Emily Denton, Shu Kuo, Kristen Thomson, Sarah Denuyl, and Larry Heck. Social biases in nlp models as barriers for persons with disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 5491–5501. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.487.
- Tal Ifargan, Lukas Hafner, Maor Kern, Ori Alcalay, and Roy Kishony. Autonomous llm-driven research from data to human-verifiable research papers. *arXiv preprint arXiv:2404.17605*, 2024.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Hel-lyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *arXiv preprint arXiv:2406.06769*, 2024.
- Shuyi Jia, Chao Zhang, and Victor Fung. Llmtdesign: Autonomous materials discovery with large language models. *arXiv preprint arXiv:2406.13163*, 2024a.
- Zixi Jia, Jiqiang Liu, Hexiao Li, Qinghua Liu, and Hongbin Gao. Dcot: Dual chain-of-thought prompting for large multimodal models. In *The 16th Asian Conference on Machine Learning (Conference Track)*, 2024b.
- Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

- Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.
- Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. Dsbench: How far are data science agents to becoming data science experts? *arXiv preprint arXiv:2409.07703*, 2024.
- A Jo. The promise and peril of generative ai. *Nature*, 614(1):214–216, 2023.
- Michael I Jordan. Serial order: A parallel distributed processing approach. In *Advances in psychology*, volume 121, pp. 471–495. Elsevier, 1997.
- John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.
- Hao Kang and Chenyan Xiong. Researcharena: Benchmarking llms’ ability to collect and organize information as research agents. *arXiv preprint arXiv:2406.10291*, 2024.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.
- Patrick Tser Jern Kon, Jiachen Liu, Qiuyi Ding, Yiming Qiu, Zhenning Yang, Yibo Huang, Jayanth Srinivasa, Myungjin Lee, Mosharaf Chowdhury, and Ang Chen. Curie: Toward rigorous and automated scientific experimentation with ai agents. *arXiv preprint arXiv:2502.16069*, 2025.
- Jakub Lála, Odhran O’Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodrigues, and Andrew D White. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*, 2023.
- Robert Tjarko Lange, Aaditya Prasad, Qi Sun, Maxence Faldor, Yujin Tang, and David Ha. The ai cuda engineer: Agentic cuda kernel discovery, optimization and composition. 2025.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Joel Lehman and Kenneth O Stanley. Novelty search and the problem with objectives. *Genetic programming theory and practice IX*, pp. 37–56, 2011.
- Steven A Lehr, Aylin Caliskan, Suneragiri Liyanage, and Mahzarin R Banaji. Chatgpt as research scientist: Probing gpt’s capabilities as a research librarian, research ethicist, data generator, and data predictor. *Proceedings of the National Academy of Sciences*, 121(35):e2404328121, 2024.
- Jia Li, Ge Li, Yongmin Li, and Zhi Jin. Structured chain-of-thought prompting for code generation, 2023. URL <https://arxiv.org/abs/2305.06599>.
- Junkai Li, Yunghwei Lai, Weitao Li, Jingyi Ren, Meng Zhang, Xinhui Kang, Siyu Wang, Peng Li, Ya-Qin Zhang, Weizhi Ma, et al. Agent hospital: A simulacrum of hospital with evolvable medical agents. *arXiv preprint arXiv:2405.02957*, 2024a.
- Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, et al. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *arXiv preprint arXiv:2410.13185*, 2024b.

- Sihang Li, Jin Huang, Jiayi Zhuang, Yaorui Shi, Xiaochen Cai, Mingjun Xu, Xiang Wang, Linfeng Zhang, Guolin Ke, and Hengxing Cai. Scilitlm: How to adapt llms for scientific literature understanding. *arXiv preprint arXiv:2408.15545*, 2024c.
- Ziming Li, Qianbo Zang, David Ma, Jiawei Guo, Tuney Zheng, Minghao Liu, Xinyao Niu, Yue Wang, Jian Yang, Jiaheng Liu, et al. Autokaggle: A multi-agent framework for autonomous data science competitions. *arXiv preprint arXiv:2410.20424*, 2024d.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.
- Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, et al. Can large language models provide useful feedback on research papers? a large-scale empirical analysis. *NEJM AI*, 1(8):AIoa2400196, 2024.
- Lei Lin, Jiayi Fu, Pengli Liu, Qingyang Li, Yan Gong, Junchen Wan, Fuzheng Zhang, Zhongyuan Wang, Di Zhang, and Kun Gai. Just ask one more time! self-agreement improves reasoning of language models in (almost) all scenarios. *arXiv preprint arXiv:2311.08154*, 2023.
- Xinna Lin, Siqi Ma, Junjie Shan, Xiaojing Zhang, Shell Xu Hu, Tiannan Guo, Stan Z Li, and Kaicheng Yu. Biokgbench: A knowledge graph checking benchmark of ai agent for biomedical science. *arXiv preprint arXiv:2407.00466*, 2024.
- Zihan Liu, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acemath: Advancing frontier math reasoning with post-training and reward modeling. *arXiv preprint arXiv:2412.15084*, 2024.
- Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024a.
- Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024b.
- Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36:43447–43478, 2023.
- Brady D. Lund and K. T. Naheem. Can chatgpt be an author? a study of artificial intelligence authorship policies in top academic journals. *Learned Publishing*, 37(1):8–16, 2023. doi: 10.1002/leap.1582.
- Xiaoliang Luo, Akilles Rechart, Guangzhi Sun, Kevin K Nejad, Felipe Yáñez, Bati Yilmaz, Kangjoo Lee, Alexandra O Cohen, Valentina Borghesani, Anton Pashkov, et al. Large language models surpass human experts in predicting neuroscience results. *Nature Human Behaviour*, pp. 1–11, 2024.
- Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pp. 1–11, 2024.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

- Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. Discoverybench: Towards data-driven discovery with large language models. *arXiv preprint arXiv:2407.01725*, 2024.
- Benjamin S Manning, Kehang Zhu, and John J Horton. Automated social science: Language models as scientist and subjects. Technical report, National Bureau of Economic Research, 2024.
- Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.
- Lisa Messeri and Molly Crockett. Artificial intelligence and illusions of understanding in scientific research. *Nature*, 627(8002):49–58, 2024. doi: 10.1038/s41586-024-07146-0.
- Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, et al. Mlgym: A new framework and benchmark for advancing ai research agents. *arXiv preprint arXiv:2502.14499*, 2025.
- Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*, 2022.
- OpenAI. Openai o3-mini system card. <https://cdn.openai.com/o3-mini-system-card-feb10.pdf>, January 2025. System card describing safety evaluations and testing of the OpenAI o3-mini model.
- Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*, 2023.
- Huy Nhat Phan, Tien N Nguyen, Phong X Nguyen, and Nghi DQ Bui. Hyperagent: Generalist software engineering agents to solve coding tasks at scale. *arXiv preprint arXiv:2409.16299*, 2024.
- Ori Press, Andreas Hochlehnert, Ameya Prabhu, Vishaal Udandara, Ofir Press, and Matthias Bethge. Citeme: Can language models accurately cite scientific claims? *arXiv preprint arXiv:2407.12861*, 2024.
- Haritz Puerto, Tilek Chubakov, Xiaodan Zhu, Harish Tayyar Madabushi, and Iryna Gurevych. Fine-tuning with divergent chains of thought boosts reasoning through self-correction in language models. 2024.
- Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*, 2024.
- Edward O Pyzer-Knapp, Jed W Pitera, Peter WJ Staar, Seiji Takeda, Teodoro Laino, Daniel P Sanders, James Sexton, John R Smith, and Alessandro Curioni. Accelerating materials discovery using artificial intelligence, high performance computing and robotics. *npj Computational Materials*, 8(1):84, 2022.
- Chen Qian, Yufan Dang, Jiahao Li, Wei Liu, Zihao Xie, Yifei Wang, Weize Chen, Cheng Yang, Xin Cong, Xiaoyin Che, et al. Experiential co-learning of software-developing agents. *arXiv preprint arXiv:2312.17025*, 2023.

- Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186, 2024.
- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.
- Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. Tool learning with large language models: A survey. *arXiv preprint arXiv:2405.17935*, 2024.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- Claire Rawlinson and Theodora Bloom. New preprint server for medical research, 2019.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.
- Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995): 468–475, 2024.
- David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning internal representations by error propagation, 1985.
- Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cino Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 29971–29994. PMLR, 2023.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=Yacmpz84TH>.
- Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments. *arXiv preprint arXiv:2405.07960*, 2024.
- Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*, 2025.
- Jürgen Schmidhuber. Curious model-building control systems. In *Proc. international joint conference on neural networks*, pp. 1458–1463, 1991.
- Dominik Schmidt, Zhengyao Jiang, and Yuxiang Unknown. Introducing weco aide, 2024. URL <https://www.weco.ai/blog/technical-report>.



- Richard Sever, Ted Roeder, Samantha Hindle, Linda Sussman, Kevin-John Black, Janet Argentine, Wayne Manos, and John R Inglis. biorxiv: the preprint server for biology. *BioRxiv*, pp. 833400, 2019.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Dudley Shapere. The structure of scientific revolutions. *The Philosophical Review*, 73(3):383–394, 1964.
- Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*, pp. 3135–3144. PMLR, 2017.
- Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce Ho, Carl Yang, and May Dongmei Wang. Ehragent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22315–22339, 2024.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.
- Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K Reddy. Llm-sr: Scientific equation discovery via programming with large language models. *arXiv preprint arXiv:2404.18400*, 2024.
- Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *arXiv preprint arXiv:2409.04109*, 2024.
- Xiaoshuai Song, Muxi Diao, Guanting Dong, Zhengyang Wang, Yujia Fu, Runqi Qiao, Zhexu Wang, Dayuan Fu, Huangxuan Wu, Bin Liang, et al. Cs-bench: A comprehensive benchmark for large language models towards computer science mastery. *arXiv preprint arXiv:2406.08587*, 2024.
- Kyle Swanson, Wesley Wu, Nash L Bulaong, John E Pak, and James Zou. The virtual lab: Ai agents design new sars-cov-2 nanobodies with experimental validation. *bioRxiv*, pp. 2024–11, 2024.
- Nathan J Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E Kumar, Tanjin He, David Milsted, Matthew J McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, et al. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, 624(7990):86–91, 2023.
- Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in llms. *arXiv preprint arXiv:2502.06233*, 2025.
- The AI Scientist Team. The ai scientist generates its first peer-reviewed scientific publication. <https://sakana.ai/ai-scientist-first-publication>, 2025.
- H. Holden Thorp. Chatgpt is fun, but not an author. *Science*, 379(6630):313, 2023. doi: 10.1126/science.adg7879.
- Alexander Tornede, Difan Deng, Theresa Eimer, Joseph Giovanelli, Aditya Mohan, Tim Ruhkopf, Sarah Segel, Daphne Theodorakopoulos, Tanja Tornede, Henning Wachsmuth, et al. Automl in the age of large language models: Current challenges, future opportunities and risks. *arXiv preprint arXiv:2306.08107*, 2023.

- Tao Tu, Anil Palepu, Mike Schaekermann, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Nenad Tomasev, et al. Towards conversational diagnostic ai. *arXiv preprint arXiv:2401.05654*, 2024.
- A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- William H. Walters and Esther I. Wilder. Fabrication and errors in the bibliographic citations generated by chatgpt. *Scientific Reports*, 13(1):14045, 2023. doi: 10.1038/s41598-023-41032-5.
- Guoqing Wang, Zeyu Sun, Zhihao Gong, Sixiang Ye, Yizhou Chen, Yifan Zhao, Qingyuan Liang, and Dan Hao. Do advanced language models eliminate the need for prompt engineering in software engineering? *arXiv preprint arXiv:2411.02093*, 2024a.
- Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Opendevin: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741*, 2024b.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024c.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Laura Weidinger, John Mellor, Maribeth Rauh, and et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.
- Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. Cyclereviewer: Improving automated research via automated review. *arXiv preprint arXiv:2411.00816*, 2024.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.
- Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*, 2024.
- John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793*, 2024a.
- Zonglin Yang, Wanhao Liu, Ben Gao, Tong Xie, Yuqiang Li, Wanli Ouyang, Soujanya Poria, Erik Cambria, and Dongzhan Zhou. Moose-chem: Large language models for rediscovering unseen chemistry scientific hypotheses. *arXiv preprint arXiv:2410.07076*, 2024b.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023a.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

N. Yeo-The and B. L. Tang. Nlp systems such as chatgpt cannot be listed as an author because these cannot fulfill widely adopted authorship criteria. *Accountability in Research*, 30(2):130–137, 2023. doi: 10.1080/08989621.2023.2177160.

Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. Answering questions by meta-reasoning over multiple chains of thought. *arXiv preprint arXiv:2304.13007*, 2023.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*, 2024a.

Xingjian Zhang, Yutong Xie, Jin Huang, Jinge Ma, Zhaoying Pan, Qijia Liu, Ziyang Xiong, Tolga Ergen, Dongsub Shim, Honglak Lee, et al. Massw: A new dataset and benchmark tasks for ai-assisted scientific workflows. *arXiv preprint arXiv:2406.06357*, 2024b.

Yilun Zhao, Hongjun Liu, Yitao Long, Rui Zhang, Chen Zhao, and Arman Cohan. Financemath: Knowledge-intensive math reasoning in finance domains. *arXiv preprint arXiv:2311.09797*, 2023.

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, et al. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. *arXiv preprint arXiv:2308.07921*, 2023.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*, 2024.

## 6. Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 2139757

## A. Algorithms

**Simultaneous Divergence Averaging (SDA).** SDA generates two distinct chain-of-thought responses—a low-temperature “Precise Solver” and a creative, high-temperature “Creative Evaluator”—for each math problem, extracting both the final answer (enclosed in LaTeX formatting) and associated confidence scores. It then encodes these full responses using Sentence-BERT to compute a cosine similarity, which is compared against a dynamically calibrated divergence threshold to determine whether the two outputs are consistent. If the similarity meets or exceeds the threshold, the answer with the higher aggregated confidence is selected; otherwise, a meta-reassessment prompt is triggered to reconcile the differences before the final answer is evaluated against the ground truth and logged for further analysis.

## B. Agent Laboratory configuration

### B.1. Hyperparameters

Table 1 | Hyperparameters for AGENT LABORATORY.

Category	Hyperparameter	Value
<b>Literature Review</b>	Number of Paper Summaries	5
	Full Text History Decay Steps	3
	Agent temperature	0.8
<b>Data Preparation</b>	Experiment Timeout	600s
<b>Running Experiments</b>	mle-solver steps	3
	Code repair attempts	2
	Maximum top codes	1
	Error history length	5
	Code history length	2
	Number of comparison trials	2
	Experiment Timeout	6000s
	Score generation temperature	0.6
	Repair temperature	0.8
	Initial code temperature	1.0
Solver temperature	1.0	
<b>Paper Writing</b>	paper-solver steps	1
	Maximum top papers	1
	Paper history length	10
	Number of Reviewers	1
	Number of comparison trials	2
	Solver temperature	1.0
	Initial paper temperature	0.8

### B.2. Hardware

All experiments in this paper were run on a 2023 MacBook Pro with an Apple M3 Max processor and 36 GB of memory.

Table 2 | Hyperparameters for AGENTRXIV.

Experiment	Hyperparameter	Value
<b>Serial Experiments</b>	Number of Parallel Laboratories	1
	Number of Papers per Laboratory	40
	Number of Papers Total (Across All Labs)	40
<b>Parallel Experiments</b>	Number of Parallel Laboratories	3
	Number of Papers per Laboratory	40
	Number of Papers Total (Across All Labs)	120

### B.3. Plagiarism Detection Software

For the study on plagiarism in Section 3.3, we used the following tools for plagiarism detection: [plagiarismdetector.net](https://plagiarismdetector.net), [duplichecker.com](https://duplichecker.com), [quetext.com](https://quetext.com) (non-AI-based). We found 100% scores of uniqueness from all three non-AI-based detectors. To inference OpenAI's Deep Research, we used the following query: "*Find the most related papers for this abstract, focus primarily on finding papers close to the methodology.*"

### C. Prompts

All prompts are the same as in [Schmidgall et al. \(2025\)](#).