Agent Laboratory: Using LLM Agents as Research Assistants

Samuel Schmidgall[1,2], Yusheng Su[1], Ze Wang[1], Ximeng Sun[1], Jialian Wu[1], Xiaodong Yu[1], Jiang Liu[1], Michael Moor[3], Zicheng Liu[1], Emad Barsoum[1],
[1]AMD, [2]Johns Hopkins University, [3]ETH Zurich
Correspondence: schmi46@jhu.edu; jiang.liu@amd.com

Abstract

Historically, scientific discovery has been a lengthy and costly process, demanding substantial time and resources from initial conception to final results. To accelerate scientific discovery, reduce research costs, and improve research quality, we introduce `Agent Laboratory`, an autonomous LLM-based framework capable of completing the entire research process. This framework accepts a human-provided research idea and progresses through three stages–literature review, experimentation, and report writing–in order to produce research, including a code repository and a research report, while enabling users to provide feedback and guidance at each stage. We deploy `Agent Laboratory` with various state-of-the-art LLMs and invite multiple researchers to assess its quality by participating in a survey, providing human feedback to guide the research process, and then evaluate the final paper. We found that: (1) `Agent Laboratory` driven by o1-preview generates the best research outcomes; (2) The generated machine learning code is able to achieve state-of-the-art performance compared to existing methods; (3) Incorporating human involvement improves the overall quality of research; (4) `Agent Laboratory` reduces research expenses, achieving an 84% decrease compared to previous autonomous research methods. We hope `Agent Laboratory` enables researchers to allocate more effort toward creative ideation rather than low-level coding and writing, ultimately accelerating scientific discovery.

1 Introduction

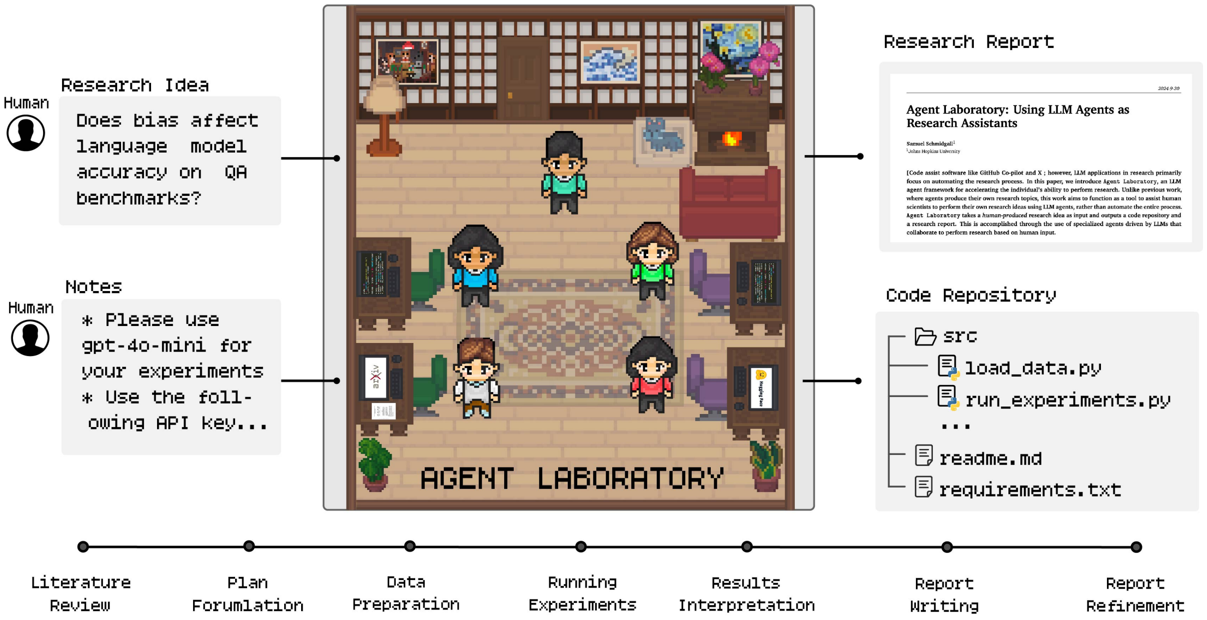
Scientists frequently face constraints that limit the number of research ideas they can explore at any given time, resulting in ideas being prioritized based on predicted impact. While this process helps determine which concepts are worth investing time in and how best to allocate limited resources effectively, many high quality ideas remain unexplored. If the process of exploring ideas had less limitations, researchers would be able to investigate multiple concepts simultaneously, increasing the likelihood of scientific discovery.

In an effort to achieve this, recent work has explored the capability of LLMs to perform research ideation and automated paper generation, where LLM agents perform the role of human scientists (Schmidgall and Moor, 2025; Lu et al., 2024a; Yamada et al., 2025). These frameworks generate novel research ideas, write code, conduct experiments, and create scientific papers with automated peer-review systems to evaluate the work. However, while these works demonstrate that current LLMs can generate ideas judged to be more novel than those produced by human experts, (Si et al., 2024) indicates that LLMs still exhibit weaknesses in feasibility and implementation details, suggesting a complementary rather than replacement role for LLMs in research. Therefore, we aim to design an autonomous agent pipeline that can assist humans toward implementing their own research ideas.

In this work, we introduce `Agent Laboratory`, an autonomous pipeline for accelerating the individual's ability to perform machine learning research. Unlike previous approaches, where agents participate in their own research ideation independent of human input (Lu et al., 2024b; Baek et al., 2024), `Agent Laboratory` is designed to assist human scientists in executing their own research ideas using language agents. `Agent Laboratory` takes as input a human research idea and outputs a research report and code repository produced by autonomous language agents, allowing various levels of human involvement, where feedback can be provided at a frequency based on user preference.

We hope that this work takes a step toward accelerating scientific discovery in machine learning, allowing researchers to allocate more effort toward creative ideation and experiment design rather than

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 5977–6043
November 4-9, 2025 ©2025 Association for Computational Linguistics



2024-9-30

Agent Laboratory: Using LLM Agents as Research Assistants

Samuel Schmidgall[1]
[1]Johns Hopkins University

[Code assist software like GitHub Co-pilot and X ; however, LLM applications in research primarily focus on automating the research process. In this paper, we introduce Agent Laboratory, an LLM agent framework for accelerating the individual's ability to perform research. Unlike previous work, where agents produce their own research topics, this work aims to function as a tool to assist human scientists to perform their own research ideas using LLM agents, rather than automate the entire process. Agent Laboratory takes a *human-produced* research idea as input and outputs a code repository and a research report. This is accomplished through the use of specialized agents driven by LLMs that collaborate to perform research based on human input.

Figure 1: Agent Laboratory takes as input a human research idea and a set of notes, provides this to a pipeline of specialized LLM-driven agents, and produces a research report and code repository.

low-level coding and writing.

2 Background & Related Work

Contemporary agent research leverages autoregressive large language models (LLMs) (Anthropic, 2024; Touvron et al., 2023b; Dubey et al., 2024; Achiam et al., 2023), typically transformer-based (Vaswani, 2017), which learn sequence prediction ($p(x_t|x_{<t};\theta)$) from extensive text data (Brown, 2020). To improve real-world applicability, LLMs are structured into agents (Wu et al., 2023; Li et al., 2023; Chen et al., 2023; Qian et al., 2024) augmented with techniques like chain-of-thought prompting (Wei et al., 2022), iterative refinement (Shinn et al., 2024), self-improvement (Huang et al., 2022), and tool integration (Hao et al., 2024; Qin et al., 2023; Schick et al., 2023). These agents demonstrate efficacy in complex domains including software engineering (Jimenez et al., 2023; Yang et al., 2024), medicine (Tu et al., 2024; Schmidgall et al., 2024), robotics (Brohan et al., 2022; Kim et al., 2024), web tasks (Gur et al., 2023; He et al., 2024), and game playing (Wang et al., 2023; Feng et al., 2024).

While AI has historically supported scientific discovery across fields (Romera-Paredes et al., 2024; Szymanski et al., 2023; Pyzer-Knapp et al., 2022), LLMs now show proficiency in specific research tasks like code generation (Chen et al., 2021; Nijkamp et al., 2022), literature search (Ajith et al., 2024; Kang and Xiong, 2024), question-answering (Chen et al., 2024a; Lála et al., 2023), paper reviewing (Liang et al., 2024; Weng et al., 2024), and experiment outcome prediction (Luo et al., 2024; Ashokkumar et al., 2024). However, the effectiveness of LLM-driven research ideation (Baek et al., 2024; Li et al., 2024a; Si et al., 2024) remains debated, with mixed results on novelty and creativity (Si et al., 2024; Chakrabarty et al., 2024; Anderson et al., 2024; Zhou et al., 2024; Ashkinaze et al., 2024; Padmakumar and He, 2024), suggesting potential benefits from combining human guidance with LLM workflows.

Recent work explores end-to-end autonomous research using LLM-based systems for tasks ranging from nano-body discovery (Swanson et al., 2024) and chemical experimentation (M. Bran et al., 2024; Boiko et al., 2023) to full research cycle automation including ideation, experimentation, and manuscript generation (Lu et al., 2024a; Yamada et al., 2025; Schmidgall and Moor, 2025). Nevertheless, persistent concerns about the feasibility and detail in LLM ideation (Si et al., 2024) underscore the potential value of human-in-the-loop systems, motivating the co-pilot approach adopted by Agent Laboratory.

3 Agent Laboratory

Overview. Agent Laboratory sequences through independent research paper collection and analysis, collaborative planning and data preparation, and automated experimentation with comprehensive report generation. As depicted in Figure 2, the workflow comprises three main phases: (1) Literature Review, (2) Experimentation, and (3) Report Writing, which we detail below along with involved agents. Section 4 presents qualitative and quantitative analyses demonstrating `Agent Laboratory`'s research generation capabilities.

3.1 Literature Review

Literature Review. This phase gathers and curates relevant research papers for the given idea, providing references for later stages. The PhD agent uses the arXiv API to retrieve papers, performing actions like `summary` (abstracts of top 20 papers), `full text` (extracts complete content), and `add paper` (incorporates selections). This iterative process involves multiple queries and relevance evaluations to build a comprehensive review. Once the target number of relevant texts is curated via `add paper`, the review is finalized.

3.2 Experimentation

Experimentation begins with plan formulation, where PhD and Postdoc agents collaborate on a detailed research plan based on the literature review and objective. This plan outlines components (models, datasets, steps), culminating in the Postdoc submitting it via `plan`. Next, in data preparation, the ML Engineer agent codes data preparation steps per the plan, using `Python` and potentially searching HuggingFace (`search HF`). The SW Engineer agent submits the finalized, bug-checked code via `submit code`.

The running experiments phase is executed by the ML Engineer using the `mle-solver` module to implement the plan. This module autonomously generates, tests, and refines ML code iteratively. Key `mle-solver` processes include: command execution (modifying programs via `REPLACE`/`EDIT`), code execution (with compilation checks/repairs), program scoring (LLM reward model assessing plan alignment, 0-1 scale), self-reflection, and performance stabilization (top program sampling, parallel modifications).

Finally, results interpretation involves the PhD and Postdoc agents analyzing `mle-solver` outcomes to derive insights. They discuss and reach consensus on an interpretation suitable for the report, which the Postdoc submits via `interpretation`, concluding the analysis and preparing for report writing.

3.3 Report Writing

Report Writing Phase. PhD and Professor agents synthesize research findings into an academic report using the `paper-solver` module. This report generator summarizes the research into a human-readable, structured format. The output follows standard academic conventions (Abstract, Introduction, Methods, etc.), aiming for conference submission standards and user comprehension.

Paper Solver Workflow. The `paper-solver` first generates a paper scaffold with standard sections and LaTeX formatting, accessing arXiv for literature/citations. It then iteratively refines the paper using `EDIT` for precise LaTeX modifications, ensuring clarity and compilation. An adapted automated review system (LLM agents simulating NeurIPS reviews) provides scores and feedback on soundness, presentation, contribution, and overall quality during iterations.

Paper Refinement Phase. The PhD agent evaluates the paper with reviews from three simulated NeurIPS reviewers assessing originality, quality, clarity, and significance. Based on this feedback, the PhD agent decides if the paper is complete or needs revisions. If revisions are needed, earlier stages (planning, experimentation, interpretation) may be revisited to address comments, simulating the academic revision cycle until standards are met.

3.3.1 Autonomous versus Co-Pilot Mode

`Agent Laboratory` operates in two modes: autonomous and co-pilot. In autonomous mode, agents produce research based solely on the initial idea, with no further human input; subtasks proceed sequentially upon completion. In co-pilot mode, besides providing the initial idea, a human reviews the output at the end of each subtask phase (e.g., literature review, generated report). The human reviewer can then approve progression to the next subtask or request the agent repeat the current subtask, providing high-level notes for improvement (e.g., instructing the agent to include a specific paper or experimental technique).

4 Results

In this section, we present our main findings on the efficacy of `Agent Laboratory` to produce research. We begin our results by asking how hu-

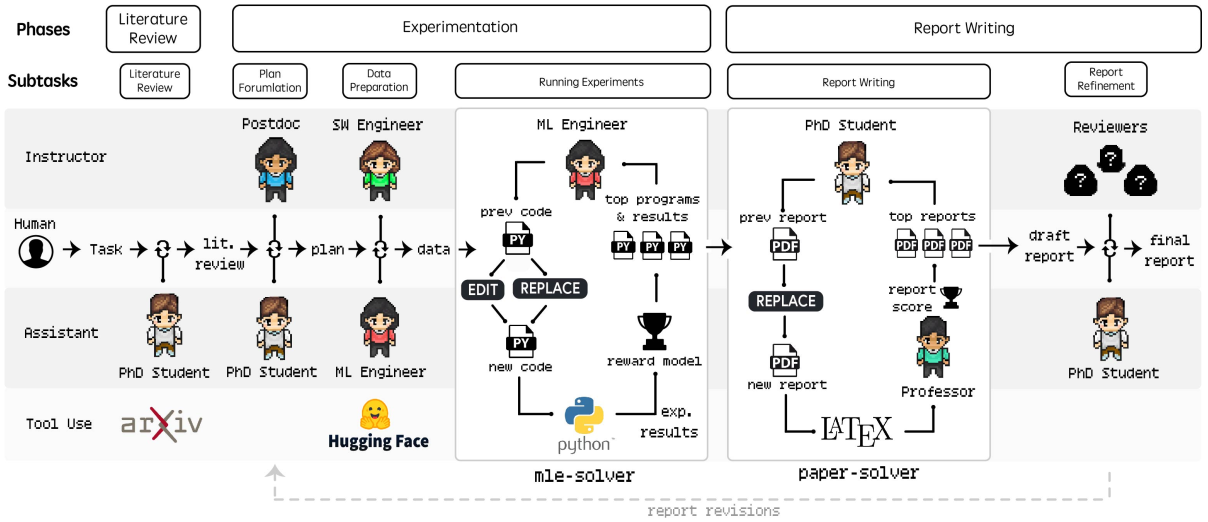


Figure 2: Agent Laboratory Workflow. This image illustrates the three primary phases of Agent Laboratory: Literature Review, Experimentation, and Report Writing. The workflow integrates human input with LLM-driven agents. Specialized tools like mle-solver for experimentation and paper-solver for report generation automate research tasks, enabling collaboration between human researchers and agents to produce high-quality research outputs.

man evaluators perceive papers generated by Agent Laboratory running in end-to-end autonomous mode across five topics. Next, we examine human evaluation when using Agent Laboratory in collaborative co-pilot mode from both allowing the researcher to choose any topic they want and from our set of preselected topics. We then provide a detailed runtime analysis including cost, average time, and success rate by various models. Finally, we conclude with an evaluation of the mle-solver in isolation on MLE-Bench, a set of real-world Kaggle challenges. The details of all surveys are provided in Appendix E.

4.1 Evaluation of quality by language model

Our first experiment aims to evaluate how human-evaluated quality varies across three axes: experiment quality, report quality, and usefulness. This evaluation was conducted by human participants using three different LLM backends: gpt-4o (Hurst et al., 2024), o1-mini, and o1-preview (OpenAI, 2024). Research questions were selected from a set of 5 templates:

1. Do LLMs exhibit cognitive biases, such as confirmation bias or anchoring bias?
2. Are image transformers more or less sensitive to pixel noise than convolutional networks?
3. Do LLMs improve accuracy on MedQA when asked to perform differential diagnosis?
4. Are LLMs sensitive to word order in multiple choice benchmarks?
5. Does gender role play affect the accuracy on of LLMs on answering math questions?

These 5 questions across 3 LLM backends resulted in a total of 15 papers being written autonomously by Agent Laboratory without any human involvement. We then recruited 10 volunteer PhD students to review 3 randomly assigned papers each. These researchers rated the experimental quality, report quality, and usefulness of the generated outputs on a scale of 1 to 5. The goal of this evaluation is to understand the differences in quality of produced research based on the three distinct LLM backbones, and to understand the usefulness of Agent Laboratory in autonomous mode. The details of the evaluation questions are provided here:

- **Experimental Quality:** What is your perception of the quality of the experimental results presented in this report?
- **Report Quality:** What is your perception of the quality of the research report writing quality presented in this report?
- **Usefulness:** What is your perception of the usefulness of an AI assistant tool that can generate the presented report autonomously?

The results of this evaluation indicate variability in performance across different Agent Laboratory LLM backends (Figure 3). gpt-4o consistently achieved lower scores, with an average experimental quality rating of 2.6/5, a report quality rating of 3.0/5, and a usefulness rating of

Average human evaluated score by Agent Laboratory base LLM

		gpt-4o			o1-mini			o1-preview		
Research Question	Research Type	Experiment Quality	Report Quality	Usefulness	Experiment Quality	Report Quality	Usefulness	Experiment Quality	Report Quality	Usefulness
Are image transformers more or less sensitive to noise than convolutional networks?	Computer Vision	1.5 / 5	2.5 / 5	2.5 / 5	4.0 / 5	3.0 / 5	4.0 / 5	2.5 / 5	3.5 / 5	4.5 / 5
Does gender affect the accuracy on of language models on answering gsm8k questions?	NLP [Social Sci]	3.0 / 5	3.0 / 5	4.0 / 5	3.0 / 5	3.5 / 5	4.0 / 5	3.0 / 5	3.5 / 5	5.0 / 5
Do language models improve accuracy on MedQA when asked to perform differential diagnosis?	NLP [Medical]	3.0 / 5	3.5 / 5	4.5 / 5	2.5 / 5	2.5 / 5	4.5 / 5	3.5 / 5	3.5 / 5	4.0 / 5
Do language models exhibit cognitive biases similar to humans, such as anchoring bias?	NLP [Cog Sci]	2.5 / 5	2.5 / 5	4.5 / 5	4.0 / 5	3.5 / 5	4.5 / 5	3.0 / 5	2.0 / 5	4.0 / 5
Are language models sensitive to word order in multiple choice benchmarks?	NLP [Core]	3.0 / 5	3.5 / 5	4.5 / 5	2.5 / 5	3.5 / 5	4.5 / 5	2.5 / 5	4.5 / 5	4.5 / 5
	Average	2.6 / 5	3.0 / 5	4.0 / 5	3.2 / 5	3.2 / 5	4.3 / 5	2.9 / 5	3.4 / 5	4.4 / 5

Figure 3: Average human evaluated scores from papers generated by Agent Laboratory in autonomous mode by research question and LLM backend. Bottom row shows average scores across all topics by LLM backend.

4.0/5. In contrast, o1-mini generally outperformed gpt-4o in experimental quality, with an average score of 3.2/5 (+0.6), while maintaining similar levels of report quality and usefulness at 3.2/5 (+0.2) and 4.3/5 (+0.3), respectively. o1-preview demonstrated the highest usefulness and report quality, averaging 4.4/5 (+0.4 from gpt-4o and +0.1 from o1-mini) and 3.4/5 (+0.4 from gpt-4o and +0.2 from o1-mini) respectively, though its experimental ratings were slightly lower than o1-mini at 2.9/5 (+0.3 from gpt-4o and -0.3 from o1-mini). While all backends perform comparably in terms of report and experimental quality, the o1-preview model was as the most useful for research assistance, suggesting that its outputs were better aligned with the expectations and needs of researchers.

From our results, the quality is demonstrated to vary based on the selected topic. We find that the overall highest average report quality to be 3.8/5 and usefulness to be 4.5/5 for the *word order* topic and the highest average experiment quality to be 3.2/5 for the *cognitive bias* topic. Interestingly, we also find that *word order* has the lowest experiment quality at 2.7/5 along with the *image noise* topic. The *image noise* topic was demonstrated to have high variance based on the LLM backend, with an experiment quality score of 1.5/5 for gpt-4o and a 4.0/5 with o1-mini (+2.5 point difference) and a usefulness score of 2.5/5 for gpt-4o and a 4.5/5 with o1-mini (+2.0 point difference).

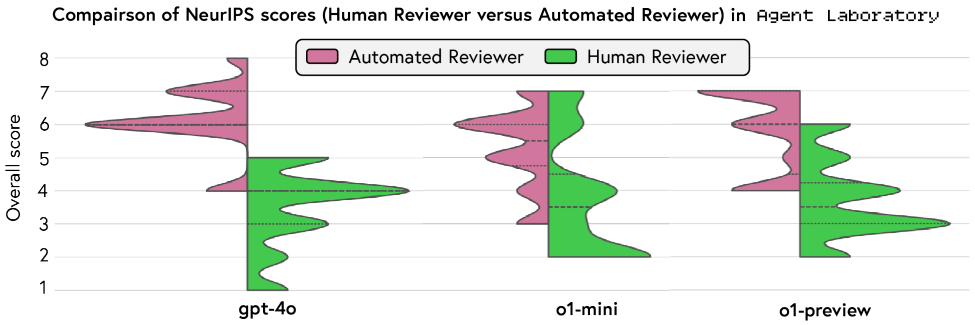
In summary, the evaluation of quality across LLM backends demonstrates clear differences in experimental quality, report quality, and usefulness. While o1-preview is consistently rated as the most useful for research assistance, o1-mini achieves the highest experimental quality scores, and gpt-4o is generally being outperformed in all areas. Topic-specific trends suggest there may exist variability in the performance of Agent Laboratory across difference areas of machine learning research and across backend models.

4.1.1 Human reviewer scores by LLM

Human reviewers assessed papers generated by Agent Laboratory using NeurIPS-style criteria, as shown in Figure 4. Comparing the same papers from Section 4.1, average human scores revealed performance differences: overall ratings ranged from 3.5/10 (gpt-4o) to 3.8/10 (o1-mini) and 4.0/10 (o1-preview).

For quality, reviewers rated gpt-4o lowest (1.8/4) and o1-mini highest (2.3/4). Significance scores were similar (2.2–2.5/4). Clarity varied slightly, with gpt-4o at 2.6/4 and o1-mini lower at 2.1/4 (-0.5). Soundness was highest for o1-preview (2.2/4), compared to o1-mini (1.8, -0.4) and gpt-4o (1.7). Presentation and contribution ratings followed similar trends, with average contribution at 2.1/4 across models, indicating a need for improved originality.

These scores suggest o1-preview produced slightly better-rounded outputs, though significant technical and methodological gaps remain across all models. With an average NeurIPS acceptance score of 5.9, papers produced autonomously fall below this threshold. These results demonstrate that Agent Laboratory in autonomous mode requires refinement to meet human expectations for high-quality research papers.



Average Automated Reviewer NeurIPS scores in `Agent Laboratory`

	Quality	Significance	Clarity	Soundness	Presentation	Contribution	Overall
gpt-4o	3.1 / 4	3.1 / 4	3.6 / 4	2.9 / 4	3.4 / 4	3.1 / 4	6.2 / 10
o1-mini	3.1 / 4	2.9 / 4	3.5 / 4	2.9 / 4	3.2 / 4	2.8 / 4	6.0 / 10
o1-preview	3.0 / 4	2.7 / 4	3.7 / 4	3.0 / 4	3.1 / 4	2.7 / 4	5.9 / 10
Average	3.1 / 4	2.9 / 4	3.6 / 4	2.9 / 4	3.2 / 4	2.9 / 4	6.1 / 10

Average Human Reviewer NeurIPS scores in `Agent Laboratory`

	Quality	Significance	Clarity	Soundness	Presentation	Contribution	Overall
gpt-4o	1.8 / 4	2.2 / 4	2.6 / 4	1.8 / 4	2.7 / 4	1.9 / 4	3.5 / 10
o1-mini	2.3 / 4	2.2 / 4	2.1 / 4	1.7 / 4	2.4 / 4	2.2 / 4	3.8 / 10
o1-preview	1.9 / 4	2.5 / 4	2.4 / 4	2.2 / 4	2.4 / 4	2.3 / 4	4.0 / 10
Average	2.0 / 4	2.3 / 4	2.4 / 4	1.9 / 4	2.5 / 4	2.1 / 4	3.8 / 10

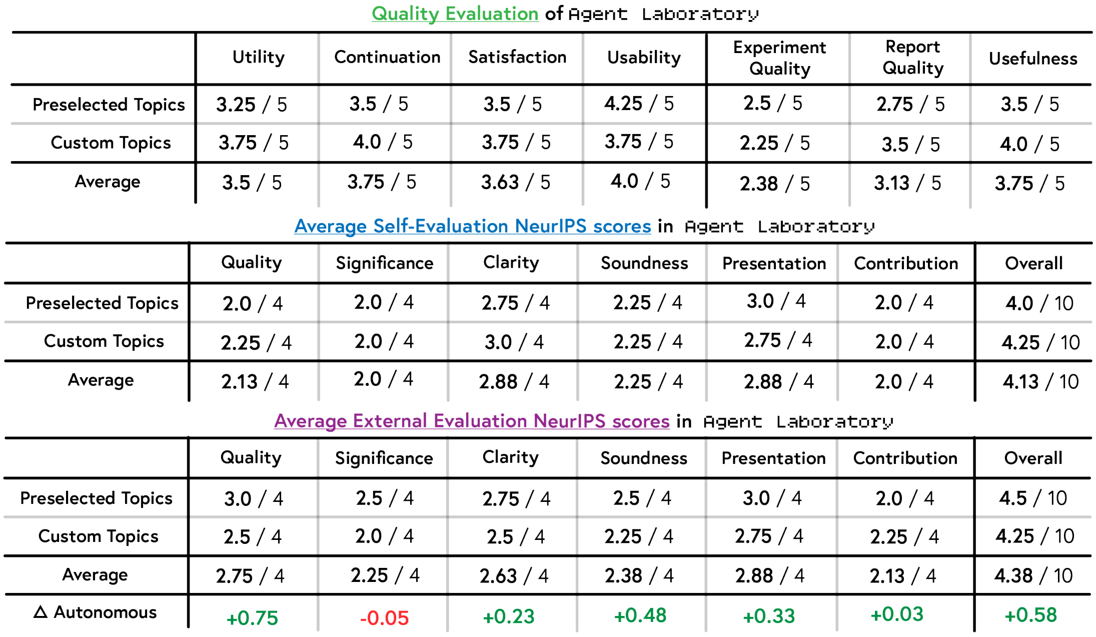
Figure 4: Scores from NeurIPs-style evaluation of generated papers, including: quality, significance, clarity, soundness, presentation, and contribution. (top) Split-violin plot comparing score distribution of automated reviewers (left half) and human reviewers (right half). Human scores are **not** predictive of automated scores (-2.3 points lower on average). Automated (middle) and human (bottom) reviewer scores across NeurIPs-style criterion.

Automated versus Human Reviews. We compared automated and human reviewer scores (Figure 4). Automated reviewers showed notable discrepancies, tending to significantly overestimate the contribution of self-evaluated work. Automated reviewers gave an average overall score of 6.1/10, whereas human reviewers averaged 3.8/10 (-2.3 points). Similar gaps exist across criteria; e.g., average clarity was rated 3.6/4 by automated reviewers versus 2.4/4 by humans. This pattern holds for all criteria. Contrary to prior work suggesting high alignment (Lu et al., 2024b), our findings show automated reviews do not align closely with human reviews and are far below the NeurIPS 2024 average acceptance score of 5.85* (our human scores were -2.05 points lower). Our results highlight the importance of providing human evaluations alongside automated scores in future work for a better understanding of generated paper quality.

4.2 Evaluation of co-pilot quality

We next evaluate the use of `Agent Laboratory` in co-pilot mode, where a human researcher is providing feedback at the end of each subtask (see Section 3.3.1 for more details). We evaluate performance across two measures: (1) the quality of `Agent Laboratory` as a tool for assisting their research and (2) the quality of generated papers. We first ask researchers to co-pilot `Agent Laboratory` on a topic of their choice without limitations. We then ask researchers to select a topic from the 5 topics introduced in Section 4.1, resulting in a total of 2 papers per researcher which we refer to as **custom** and **preselected** papers respectively. After their papers are generated, we ask researchers to rate their experience using `Agent Laboratory` during the process of generating custom and preselected papers. We then ask them to self-evaluate the generated papers according to NeurIPS-style criterion. Finally, we ask external researchers to evaluate their paper comparing performance with `Agent Laboratory` in autonomous mode. All experiments used an o1-mini backbone for all phases except the literature review.

*https://papercopilot.com/statistics/neurips-statistics/neurips-2024-statistics



Quality Evaluation of Agent Laboratory

	Utility	Continuation	Satisfaction	Usability	Experiment Quality	Report Quality	Usefulness
Preselected Topics	3.25 / 5	3.5 / 5	3.5 / 5	4.25 / 5	2.5 / 5	2.75 / 5	3.5 / 5
Custom Topics	3.75 / 5	4.0 / 5	3.75 / 5	3.75 / 5	2.25 / 5	3.5 / 5	4.0 / 5
Average	3.5 / 5	3.75 / 5	3.63 / 5	4.0 / 5	2.38 / 5	3.13 / 5	3.75 / 5

Average Self-Evaluation NeurIPS scores in Agent Laboratory

	Quality	Significance	Clarity	Soundness	Presentation	Contribution	Overall
Preselected Topics	2.0 / 4	2.0 / 4	2.75 / 4	2.25 / 4	3.0 / 4	2.0 / 4	4.0 / 10
Custom Topics	2.25 / 4	2.0 / 4	3.0 / 4	2.25 / 4	2.75 / 4	2.0 / 4	4.25 / 10
Average	2.13 / 4	2.0 / 4	2.88 / 4	2.25 / 4	2.88 / 4	2.0 / 4	4.13 / 10

Average External Evaluation NeurIPS scores in Agent Laboratory

	Quality	Significance	Clarity	Soundness	Presentation	Contribution	Overall
Preselected Topics	3.0 / 4	2.5 / 4	2.75 / 4	2.5 / 4	3.0 / 4	2.0 / 4	4.5 / 10
Custom Topics	2.5 / 4	2.0 / 4	2.5 / 4	2.25 / 4	2.75 / 4	2.25 / 4	4.25 / 10
Average	2.75 / 4	2.25 / 4	2.63 / 4	2.38 / 4	2.88 / 4	2.13 / 4	4.38 / 10
Δ Autonomous	+0.75	-0.05	+0.23	+0.48	+0.33	+0.03	+0.58

Figure 5: Co-pilot evaluation.

4.2.1 Quality as a tool

Evaluating Agent Laboratory as a research tool involved assessing its effectiveness in co-pilot mode. Post-generation, participants assessed the tool via questions on a 1-5 scale (1=lowest, 5=highest):

- **Utility:** How useful is Agent Laboratory for assisting your research?
- **Continuation:** How likely are you to continue using Agent Laboratory for research?
- **Satisfaction:** How much did you enjoy using Agent Laboratory?
- **Usability:** How easy was building a project with Agent Laboratory?

Overall scores averaged 3.5/5 for utility, 3.75/5 for continuation, 3.63/5 for satisfaction, and 4.0/5 for usability (Figure 5). Scores varied by topic type. Custom experiments averaged: utility 3.75/5, continuation 4.0/5, satisfaction 3.75/5, usability 3.75/5. Preselected topics averaged: utility 3.25/5, continuation 3.5/5, satisfaction 3.5/5, usability 4.25/5. Compared to custom topics, preselected topic ratings were lower, except for usability (-0.5 points lower for custom). Utility and continuation increased by +0.5 points and satisfaction by +0.25 points from preselected to custom.

Using metrics from Section 4.1, we report average co-pilot ratings of 2.38/5 for experimental quality, 3.13/5 for report quality, and 3.75/5 for usefulness. Custom topics scored higher on report quality (3.5/5, +0.75) and usefulness (4.0/5, +0.5), while preselected topics scored higher (+0.25) on experiment quality (2.5/5). Compared to corresponding o1-mini autonomous results, co-pilot scores were lower across all metrics: report quality (-0.07), usefulness (-0.55), and experiment quality (-0.82).

Optional feedback (75% response rate) on improving Agent Laboratory suggested enhancing the interface (e.g., GUI, result inspection), adding more figure options, and improving the literature review. Compared to autonomous mode reviews (Section 4.1), human co-pilots rated report quality, usefulness, and experiment quality lower, feedback indicated this reduction stemmed from difficulty guiding agents to execute their exact vision. These limitations are further discussed in Section 6.

4.2.2 Evaluation of co-pilot generated papers

To assess the quality of papers generated by Agent Laboratory in co-pilot mode, we conduct evaluations using two approaches: (1) researchers self-assessed their generated papers based on NeurIPS-style criteria, and (2) external researchers provided evaluations of the same papers. This section aims to understand differences in scores from self-assessment and external assessment, as well as how assessments compare to Agent Laboratory in fully autonomous mode. We use the same NeurIPS criterion introduced in Section 4.1.1.

Self-evaluation. From the results of the self-evaluation (Figure 5), we found that the average overall score *increased* from evaluations provided to papers generated in autonomous mode, with autonomous papers having an overall average of 3.8/10 and co-pilot papers at 4.13/10 (+0.33). These scores even improved across the best autonomous backend, o1-preview, which averaged 4.0/10. Across individual criterion, scores increased for quality (+0.13), clarity (+0.48), soundness (+0.35), and presentation (+0.33), but decreased for significance and contribution. The scores that decreased were significance (-0.3) and contribution (-0.1).

External evaluation. We compare scores provided through self-evaluation with those provided by a set of external evaluators on the same papers (Figure 5). We find that average scores across most criteria, including quality, significance, clarity, soundness, presentation, and contribution, show an improvement in the external assessments, with an overall average of 4.38/10, up from 4.13/10 in self-evaluations. The most significant improvements were observed in quality (+0.62), significance (+0.25), and overall (+0.25) scores, suggesting that external reviewers perceived the generated papers to be higher quality and more significant than the researchers who produced them. However, clarity scores decreased (-0.25), indicating potential issues in the articulation of ideas that might have been overlooked during self-assessment. While presentation scores did not improve (+0.0), soundness (+0.13) and contribution (+0.13) only increased slightly.

Notably, the external evaluations also reinforce differences between scores preselected and custom topics. Unlike with the self-evaluated papers, papers on preselected topics were rated slightly higher overall, with improvements observed across several metrics, particularly in quality (+0.5) and significance (+0.5). These findings suggest that self-evaluated reviewers perceive the work produced on their custom topic as higher quality compared to the work produced on preselected topics, whereas external evaluators find the opposite to be true.

Comparison with autonomous mode Comparing scores by external evaluators on autonomous and co-pilot papers (Figure 5), we find that the largest improvements were seen for quality, which increased by +0.75, soundness, which improved by +0.48, and the overall score, which improved by +0.58. Moderate gains were also observed in clarity (+0.23) and presentation (+0.33). In contrast, some metrics showed minimal or no improvement. Significance declined slightly (-0.05), and contribution increased only marginally (+0.03). Our results suggest that papers generated with human involvement overall are evaluated more highly than autonomously generated paper, with much of the focus of human involvement going toward making the paper more presentable (presentation and clarity) while there was less emphasis on improving experimental results (significance and contribution). Finally, we note that co-pilot overall scores, which average at 4.38, are still -1.45 points below the average score of 5.85 for an accepted paper at NeurIPS 2024. Increasing the overall score to match conference standards will likely result by improving the contribution and significance of the paper results, which is consistently lower than other evaluation metrics.

5 Discussion

`Agent Laboratory` is an LLM agent system designed to assist in performing end-to-end machine learning research. Functioning primarily as a co-pilot, it aims to accelerate scientific exploration through a human-centric approach, distinct from fully automated discovery systems. The `Agent Laboratory` workflow includes three stages literature review, experimentation, and report writing. Evaluations assessed the quality of outputs generated in autonomous mode using various LLM backends, employing metrics such as human ratings of experimental quality and report usefulness, alongside standard academic reviewer scores. The effectiveness of `Agent Laboratory` was also examined in its co-pilot mode, comparing it to autonomous operation and integrating researcher feedback.

Findings revealed performance differences between LLM backends; for instance, `o1-preview` showed higher perceived usefulness, while `o1-mini` demonstrated better experimental quality. Although autonomous outputs were generally rated positively, human evaluations identified shortcomings in clarity and soundness relative to standards for high-quality research, highlighting that automated reviewer scores did not consistently align with human assessments. The co-pilot mode, which incorporates human feedback, generally produced higher-quality results across most evaluation metrics and received favorable utility and usability ratings. Future research directions involve longitudinal user studies and the exploration of automated workflow optimization.

6 Limitations

While our results suggest that Agent Laboratory demonstrates strong performance as a research tool, we now turn to a discussion of limitations that could inform future work. While some of these are also limitations of LLMs themselves, others are not, and we nonetheless provide a thorough and critical discussion of our work. We hope that progress in autonomous research will address these limitations.

6.1 Workflow limitations

Challenges with self-evaluation The paper-solver is being evaluated for quality by using LLMs emulated NeurIPS reviewers. This has two limitations: (1) while the reviewing agents were shown to have high alignment with real reviewers (Lu et al., 2024b), *qualitatively* research reports from Agent Laboratory are less satisfying than research papers from The AI Scientist (Lu et al., 2024b), with ours having lower quality figures, despite Agent Laboratory papers obtaining higher scores overall. (2) The research reports produced by Agent Laboratory are not meant to replace the paper writing process done by humans as it was in The AI Scientist, rather it is meant to provide a report for the human to understand what has been accomplished, so that they can scale up the experiment and write their own research report. However, we nonetheless use NeurIPS reviewer scores as the heuristic for the quality of our presented paper-solver, which aims to evaluate the reports from the perspective of a complete research paper. Additionally, contrasting with (Lu et al., 2024b) demonstrate that LLMs perform less reliably for self-evaluation compared with human reviewers, with lower agreement scores (53.3% vs. 56.1%). Although LLMs demonstrate reasonable consistency, this may stem from reliance on superficial patterns rather than robust evaluation criteria, resulting in discrepancies between LLM and human rankings. This limits LLMs in subjective tasks like research idea evaluation, which is the foundation of mle-solver and paper-solver.

Challenges with automated structure There are also some limitations that present themselves due to the structure enforced in the workflow. For example, paper-solver is encouraged to a organize the paper into a relatively fixed structure (abstract, introduction, etc), which disallows unique paper organizations and section orders. Another limitation is that mle-solver and paper-solver are limited to generating only two figures for the paper. This can be solved in future work, by allowing all of the figures generated by the mle-solver (without restriction) to be incorporated into paper-solver by detecting image files and providing those paths to the solver. Agent Laboratory is also not able to manage repository-level code on its own, but rather the appropriate files are provided to it at each necessary step and files are saved based on which phase produced the file. Enabling flexible repository-level file modification and execution is a clear next step for future work.

Challenges with hallucination While uncommon, we also found that in some of the research papers, particularly from lower performing models, such as gpt-4o, there were hallucinations regarding experimental results that did not occur, such as the following example from a gpt-4o paper on the topic of *Are image transformers more or less sensitive to noise than convolutional networks?*: "*Hyper-parameter optimization played a crucial role in achieving these results. The learning rate was set at* 0.001, *with a batch size of* 32, *and the number of reasoning steps* $L = \{l_1, l_2, ..., l_n\}$ *varied between* 5 *to* 10, *depending on the complexity of the query. The model was trained over* 50 *epochs, with early stopping criteria applied to prevent overfitting.*" While the issue of hallucination is more generally a problem with LLMs themselves, future work must appropriately address these challenges in order to prevent misinformation from being propagated when using automated research tools.

6.2 Common failure modes

In addition to the limitations outlined in Section 6.1, we also outline common failure modes observed during the runtime of Agent Laboratory. We report a list of the most common failure modes observed below:

- Many of the more capable models (gpt-4o, o1-mini, o1-preview) struggled with instruction-following during the literature review phase, and had a tendency to repeatedly use the summarize command until the maximum phase steps have been reached, leading to a termination.

- Retrieved papers during the literature review phase had been observed to reach the maximum token limit for some models.

- Experiments run by mle-solver sometimes obtain 0% accuracy for all tested methods which is not corrected by the agent by the time mle-solver runs out of solving steps.

- mle-solver has a tendency to edit line 0 more than other lines in the code, causing to the replace command to more often lead to successful code compiles.

- Printed output from the data preparation or experimental results can lead to the LLMs reaching their token limit.

- mle-solver often generated the python exit () command, which terminated the entire process. This had to be detected and removed manually.

- mle-solver has been observed to run system commands on the host computer using the subprocess.run() command. While nothing problematic has been observed, safeguards should be implemented around this.

- paper-solver often struggles to search for relevant papers using the arXiv engine. Before a search time-limit was enforced, it could take up to 100 tries for a successful search query to return *any* papers. A limit of 5 was place thereafter to prevent this cycle.

7 Ethical considerations

Agent Laboratory offers potential to accelerate the field of machine learning research by automating time-intensive tasks and enabling researchers to focus on ideation and experimental design. However, its capabilities also bring ethical challenges that require careful consideration. The ability to autonomously generate research code, reports, and experiment plans may inadvertently lower the barriers to producing substandard or misleading scientific outputs. This could overwhelm peer review systems and jeopardize the integrity of academic discourse. Furthermore, the automated processes may reflect or even amplify biases inherent in the underlying datasets or algorithms, leading to skewed outcomes in research findings. Transparent disclosure of AI involvement in research outputs is important in order to mitigate such risks and maintain accountability.

There are additional concerns about potential misuse of Agent Laboratory for unethical purposes, such as developing harmful technologies or generating content that bypasses ethical oversight. For instance, the misuse of autonomous research agents in fields like cybersecurity could lead to the automated creation of malware (Xu et al., 2024; Happe and Cito, 2023) or in environmental studies, it may generate biased analyses that downplay climate risks or overstate the benefits of certain interventions. Moreover, as the platform matures, the risk of its misuse increases if safeguards are not implemented to ensure alignment with ethical research standards (Watkins, 2024; Jiao et al., 2024). Thus, while Agent Laboratory demonstrates immense promise for accelerating scientific discovery, there is a need for robust governance mechanisms to ensure that the underlying LLMs produce content that aligns with ethical principles and societal values.

References

Talor Abramovich, Meet Udeshi, Minghao Shao, Kilian Lieret, Haoran Xi, Kimberly Milner, Sofija Jancheska, John Yang, Carlos E Jimenez, Farshad Khorrami, and 1 others. 2024. Enigma: Enhanced interactive generative model agent for ctf challenges. *arXiv preprint arXiv:2409.16165*.

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

Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. 2024. Litsearch: A retrieval benchmark for scientific literature search. *arXiv preprint arXiv:2407.18940*.

Altera AL, Andrew Ahn, Nic Becker, Stephanie Carroll, Nico Christie, Manuel Cortes, Arda Demirci, Melissa Du, Frankie Li, Shuying Luo, and 1 others. 2024. Project sid: Many-agent simulations toward ai civilization. *arXiv preprint arXiv:2411.00114*.

Barrett R Anderson, Jash Hemant Shah, and Max Kreminski. 2024. Homogenization effects of large language models on human creative ideation. In *Proceedings of the 16th Conference on Creativity & Cognition*, pages 413–425.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1.

Joshua Ashkinaze, Julia Mendelsohn, Li Qiwei, Ceren Budak, and Eric Gilbert. 2024. How ai ideas affect the creativity, diversity, and evolution of human ideas: Evidence from a large, dynamic experiment. *arXiv preprint arXiv:2401.13481*.

Ashwini Ashokkumar, Luke Hewitt, Isaias Ghezae, and Robb Willer. 2024. Predicting results of social science experiments using large language models. Technical report, Technical report, Working Paper.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2024. Researchagent: Iterative research idea generation over scientific literature with large language models. *arXiv preprint arXiv:2404.07738*.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, and 1 others. 2024. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, and 1 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, and 1 others. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. 2024. Art or artifice? large language models and the false promise of creativity. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–34.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, and 1 others. 2024. Mle-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, and 1 others. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*.

Xiuying Chen, Tairan Wang, Taicheng Guo, Kehan Guo, Juexiao Zhou, Haoyang Li, Mingchen Zhuge, Jürgen Schmidhuber, Xin Gao, and Xiangliang Zhang. 2024a. Scholarchemqa: Unveiling the power of language models in chemical research question answering. *arXiv preprint arXiv:2407.16931*.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, and 1 others. 2024b. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080*.

Mike D'Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. 2024. Marg: Multi-agent review generation for scientific papers. *arXiv preprint arXiv:2401.04259*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Ning Ding, Shang Qu, Linhai Xie, Yifei Li, Zaoqu Liu, Kaiyan Zhang, Yibai Xiong, Yuxin Zuo, Zhangren Chen, Ermo Hua, and 1 others. 2024. Automating exploratory proteomics research via language models. *arXiv preprint arXiv:2411.03743*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. 2024. Llm agents can autonomously hack websites. *arXiv preprint arXiv:2402.06664*.

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J R Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, and 1 others. 2022. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. 2024. Chessgpt: Bridging policy learning and language modeling. *Advances in Neural Information Processing Systems*, 36.

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

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

Antoine Grosnit, Alexandre Maraval, James Doran, Giuseppe Paolo, Albert Thomas, Refinath Shahul Hameed Nabeezath Beevi, Jonas Gonzalez, Khyati Khandelwal, Ignacio Iacobacci, Abdelhakim Benechehab, and 1 others. 2024. Large language models orchestrating structured reasoning achieve kaggle grandmaster level. *arXiv preprint arXiv:2411.03562*.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, and 1 others. 2024. Blade: Benchmarking language model agents for data-driven science. *arXiv preprint arXiv:2408.09667*.

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

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

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

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

Andreas Happe and Jürgen Cito. 2023. Getting pwn'd by ai: Penetration testing with large language models. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 2082–2086.

Tomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, and 1 others. 2024. Simulating 500 million years of evolution with a language model. *bioRxiv*, pages 2024–07.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Qianli Ma, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, and 1 others. 2024. Infiagent-dabench: Evaluating agents on data analysis tasks. *arXiv preprint arXiv:2401.05507*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2024. Mlagentbench: Evaluating language agents on machine learning experimentation. In *Forty-first International Conference on Machine Learning*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Tal Ifargan, Lukas Hafner, Maor Kern, Ori Alcalay, and Roy Kishony. 2024. Autonomous llm-driven research from data to human-verifiable research papers. *arXiv preprint arXiv:2404.17605*.

Junfeng Jiao, Saleh Afroogh, Yiming Xu, and Connor Phillips. 2024. Navigating llm ethics: Advancements, challenges, and future directions. *arXiv preprint arXiv:2406.18841*.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*.

Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. 2024. Dsbench: How far are data science agents to becoming data science experts? *arXiv preprint arXiv:2409.07703*.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, and 1 others. 2021. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589.

Hao Kang and Chenyan Xiong. 2024. Researcharena: Benchmarking llms' ability to collect and organize information as research agents. *arXiv preprint arXiv:2406.10291*.

Ji Woong Kim, Tony Z Zhao, Samuel Schmidgall, Anton Deguet, Marin Kobilarov, Chelsea Finn, and Axel Krieger. 2024. Surgical robot transformer (srt): Imitation learning for surgical tasks. In *8th Annual Conference on Robot Learning*.

Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodriques, and Andrew D White. 2023. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*.

Steven A Lehr, Aylin Caliskan, Suneragiri Liyanage, and Mahzarin R Banaji. 2024. 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.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, and 1 others. 2024a. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *arXiv preprint arXiv:2410.13185*.

Sihang Li, Jin Huang, Jiaxi Zhuang, Yaorui Shi, Xiaochen Cai, Mingjun Xu, Xiang Wang, Linfeng Zhang, Guolin Ke, and Hengxing Cai. 2024b. Scilitllm: How to adapt llms for scientific literature understanding. *arXiv preprint arXiv:2408.15545*.

Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, and 1 others. 2024. Can large language models provide useful feedback on research papers? a large-scale empirical analysis. *NEJM AI*, 1(8):AIoa2400196.

Xinna Lin, Siqi Ma, Junjie Shan, Xiaojing Zhang, Shell Xu Hu, Tiannan Guo, Stan Z Li, and Kaicheng Yu. 2024. Biokgbench: A knowledge graph checking benchmark of ai agent for biomedical science. *arXiv preprint arXiv:2407.00466*.

Yiren Liu, Si Chen, Haocong Cheng, Mengxia Yu, Xiao Ran, Andrew Mo, Yiliu Tang, and Yun Huang. 2024. How ai processing delays foster creativity: Exploring research question co-creation with an llm-based agent. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–25.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024a. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024b. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Xiaoliang Luo, Akilles Rechardt, Guangzhi Sun, Kevin K Nejad, Felipe Yáñez, Bati Yilmaz, Kangjoo Lee, Alexandra O Cohen, Valentina Borghesani, Anton Pashkov, and 1 others. 2024. Large language models surpass human experts in predicting neuroscience results. *Nature Human Behaviour*, pages 1–11.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pages 1–11.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2024. Discoverybench: Towards data-driven discovery with large language models. *arXiv preprint arXiv:2407.01725*.

Benjamin S Manning, Kehang Zhu, and John J Horton. 2024. Automated social science: Language models as scientist and subjects. Technical report, National Bureau of Economic Research.

Daniel McDuff, Mike Schaekermann, Tao Tu, Anil Palepu, Amy Wang, Jake Garrison, Karan Singhal, Yash Sharma, Shekoofeh Azizi, Kavita Kulkarni, and 1 others. 2023. Towards accurate differential diagnosis with large language models. *arXiv preprint arXiv:2312.00164*.

Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. 2023. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2022. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/index/chatgpt/`. Blog post.

OpenAI. 2024. Introducing openai o1-preview. Accessed: 2024-09.

Vishakh Padmakumar and He He. 2024. Does writing with language models reduce content diversity? In *The Twelfth International Conference on Learning Representations*.

Huy Nhat Phan, Tien N Nguyen, Phong X Nguyen, and Nghi DQ Bui. 2024. Hyperagent: Generalist software engineering agents to solve coding tasks at scale. *arXiv preprint arXiv:2409.16299*.

Ori Press, Andreas Hochlehnert, Ameya Prabhu, Vishaal Udandarao, Ofir Press, and Matthias Bethge. 2024. Citeme: Can language models accurately cite scientific claims? *arXiv preprint arXiv:2407.12861*.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. 2024. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*.

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. 2022. Accelerating materials discovery using artificial intelligence, high performance computing and robotics. *npj Computational Materials*, 8(1):84.

Chen Qian, Yufan Dang, Jiahao Li, Wei Liu, Zihao Xie, Yifei Wang, Weize Chen, Cheng Yang, Xin Cong, Xiaoyin Che, and 1 others. 2023. Experiential co-learning of software-developing agents. *arXiv preprint arXiv:2312.17025*.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, and 1 others. 2024. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, and 1 others. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Samuel Schmidgall and Michael Moor. 2025. Agentrxiv: Towards collaborative autonomous research. *arXiv preprint arXiv:2503.18102*.

Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. 2024. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments. *arXiv preprint arXiv:2405.07960*.

Dominik Schmidt, Zhengyao Jiang, and Yuxiang Unknown. 2024. Introducing weco aide.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*, pages 3135–3144. PMLR.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2024. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *arXiv preprint arXiv:2409.04109*.

Xiaoshuai Song, Muxi Diao, Guanting Dong, Zhengyang Wang, Yujia Fu, Runqi Qiao, Zhexu Wang, Dayuan Fu, Huangxuan Wu, Bin Liang, and 1 others. 2024. Cs-bench: A comprehensive benchmark for large language models towards computer science mastery. *arXiv preprint arXiv:2406.08587*.

Kyle Swanson, Wesley Wu, Nash L Bulaong, John E Pak, and James Zou. 2024. The virtual lab: Ai agents design new sars-cov-2 nanobodies with experimental validation. *bioRxiv*, pages 2024–11.

Nathan J Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E Kumar, Tanjin He, David Milsted, Matthew J McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, and 1 others. 2023. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, 624(7990):86–91.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tao Tu, Anil Palepu, Mike Schaekermann, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Nenad Tomasev, and 1 others. 2024. Towards conversational diagnostic ai. *arXiv preprint arXiv:2401.05654*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Shengye Wan, Cyrus Nikolaidis, Daniel Song, David Molnar, James Crnkovich, Jayson Grace, Manish Bhatt, Sahana Chennabasappa, Spencer Whitman, Stephanie Ding, and 1 others. 2024. Cyberseceval 3: Advancing the evaluation of cybersecurity risks and capabilities in large language models. *arXiv preprint arXiv:2408.01605*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv: Arxiv-2305.16291*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024a. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, and 1 others. 2024b. Opendevin: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741*.

Ryan Watkins. 2024. Guidance for researchers and peer-reviewers on the ethical use of large language models (llms) in scientific research workflows. *AI and Ethics*, 4(4):969–974.

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

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. 2024. Cycleresearcher: Improving automated research via automated review. *arXiv preprint arXiv:2411.00816*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Jiacen Xu, Jack W Stokes, Geoff McDonald, Xuesong Bai, David Marshall, Siyue Wang, Adith Swaminathan, and Zhou Li. 2024. Autoattacker: A large language model guided system to implement automatic cyber-attacks. *arXiv preprint arXiv:2403.01038*.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. Swe-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793*.

Xingjian Zhang, Yutong Xie, Jin Huang, Jinge Ma, Zhaoying Pan, Qijia Liu, Ziyang Xiong, Tolga Ergen, Dongsub Shim, Honglak Lee, and 1 others. 2024. Massw: A new dataset and benchmark tasks for ai-assisted scientific workflows. *arXiv preprint arXiv:2406.06357*.

Yilun Zhou, Caiming Xiong, Silvio Savarese, and Chien-Sheng Wu. 2024. Shared imagination: Llms hallucinate alike. *arXiv preprint arXiv:2407.16604*.

A Runtime statistics

Runtime statistics for `Agent Laboratory` are detailed to provide insight into the computational efficiency and monetary costs associated with different phases of its workflow. In this evaluation, both the time required per phase (measured in seconds) and the costs incurred (calculated in USD) were analyzed to better understand the performance of three model backends: gpt-4o, o1-mini, and o1-preview. These measurements were recorded for each subtask, including Literature Review, Plan Formulation, Data Preparation, Running Experiments, Results Interpretation, Report Writing, and Report Refinement.

Inference time Across all models, gpt-4o exhibited the fastest execution times, completing the entire workflow in 1165.4 seconds, approximately 3.2x faster than o1-mini and 5.3x faster than o1-preview, which required 3616.8 seconds and 6201.3 seconds, respectively. In most subtasks, gpt-4o demonstrated superior speed, particularly in Running Experiments and Report Writing phases, where its times were significantly shorter than those of o1-mini and o1-preview. For instance, in Running Experiments, gpt-4o averaged 417.8 seconds, while o1-mini and o1-preview took 2082.5 seconds and 4036.2 seconds, respectively. Similarly, for Report Writing, gpt-4o completed the task in 572.5 seconds, compared to 827.7 seconds for o1-mini and 1854.2 seconds for o1-preview.

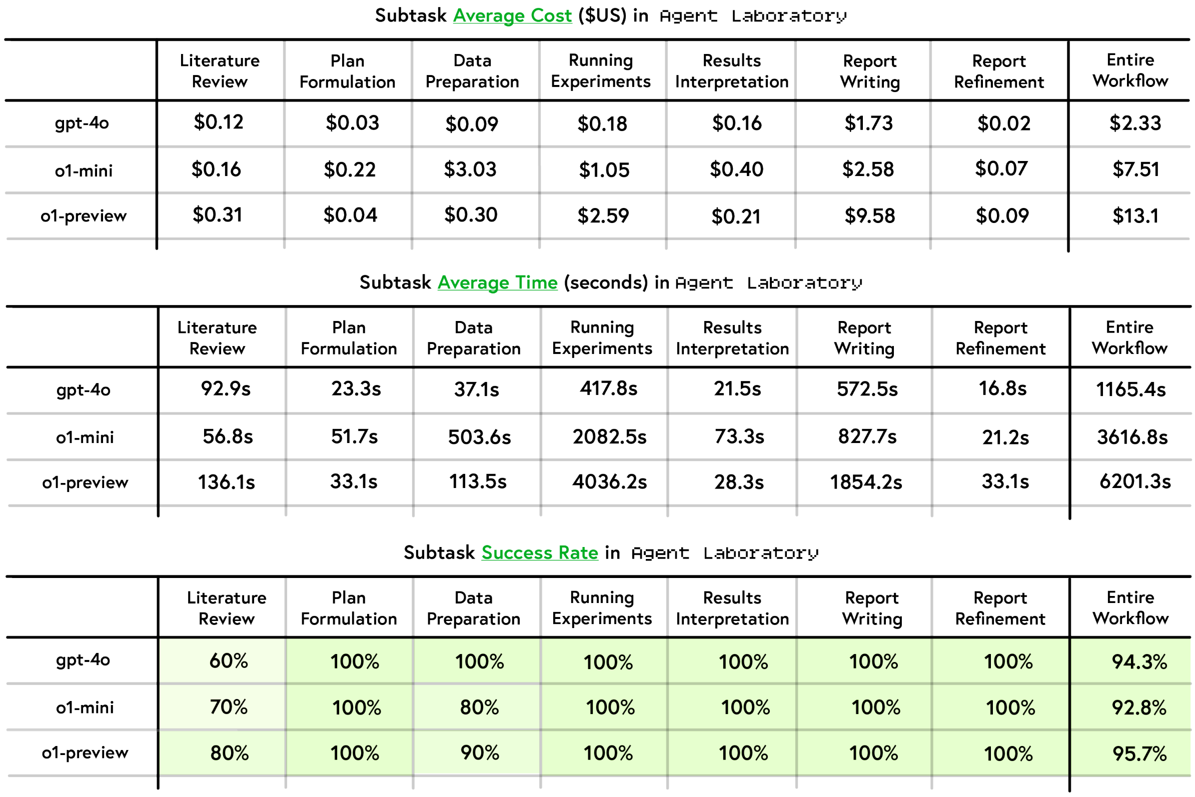
Inference cost Monetary costs per workflow were also substantially lower for gpt-4o, which averaged just \$2.33 for the entire process. This is significantly more cost effective than previous autonomous research workflows (Lu et al., 2024b), which cost around ∼\$15 (6.4x more expensive) to complete using gpt-4o. Other models in our workflow has a lower cost efficiency, such as o1-mini at \$7.51, and o1-preview at \$13.10, the latter being over 5.6x more expensive than gpt-4o. Among the individual subtasks, gpt-4o consistently had the lowest costs. For example, its costs for Data Preparation and Report Writing were \$0.09 and \$1.73, respectively, compared to \$3.03 and \$2.58 for o1-mini, and \$0.30 and \$9.58 for o1-preview.

Phase-level Observations From our observations at the phase-level, Literature Review was notably efficient for all models in terms of time and cost, with gpt-4o completing it in 92.9 seconds at a cost of \$0.12. Meanwhile, o1-mini completed this phase faster (56.8 seconds) but at a slightly higher cost (\$0.16). For Plan Formulation, gpt-4o was both the fastest (23.3 seconds) and the cheapest (\$0.03), followed closely by o1-preview in cost (\$0.04) but not in speed (33.1 seconds). The most expensive phase across models was Report Writing, where costs were driven by the increased computational resources required for writing a long document. o1-preview incurred particularly high costs in this phase (\$9.58) despite producing comparable outputs in terms of task success rates.

Success Rates Overall, every model exhibits reasonably high reliability, with o1-preview achieving the highest average subtask success rate (95.7%) for the entire workflow. Both gpt-4o and o1-mini followed closely at 94.3% and 92.8%. While most tasks had 100% success rate for each model, the literature review phase had a high rate of failure, at 60%, 70%, and 80% for gpt-4o, o1-mini, and o1-preview respectively. The Data Preparation phase showed minor challenges, with o1-mini recording an 80% success rate in Data Preparation, compared to gpt-4o's 100% success rate and o1-preview at a 90% success rate.

B Evaluating mle-solver on MLE-Bench

Evaluating the entire `Agent Laboratory` workflow does not contain much information about the ability of `mle-solver` specifically to solve individual ML problems. In order to evaluate `mle-solver` more objectively, we use a subset of 10 ML challenges from MLE-Bench (Chan et al., 2024). MLE-Bench is a benchmark designed to assess the capability of agents in handling real-world ML tasks on Kaggle competitions. This benchmark compares agent performances with human baselines, scoring agents with Kaggle's medal system, and incorporating mechanisms to mitigate contamination and plagiarism risks. We include all challenges focusing on text and tabular data from the low complexity category of MLE-Bench. We provide as input to `mle-solver` the following: Kaggle dataset description, distilled knowledge from Kaggle notebooks, as well as an accessible train and dev set. Instead of using an LLM scoring function, the `mle-solver` score is evaluated on the dev set, which is a 20% random sample taken from the original training set, and the training set is represented by the other 80% split. All data (dev, test, train) is placed into arrays using the numpy library instead of providing file locations in order to better emulate the



Subtask Average Cost ($US) in Agent Laboratory

	Literature Review	Plan Formulation	Data Preparation	Running Experiments	Results Interpretation	Report Writing	Report Refinement	Entire Workflow
gpt-4o	$0.12	$0.03	$0.09	$0.18	$0.16	$1.73	$0.02	$2.33
o1-mini	$0.16	$0.22	$3.03	$1.05	$0.40	$2.58	$0.07	$7.51
o1-preview	$0.31	$0.04	$0.30	$2.59	$0.21	$9.58	$0.09	$13.1

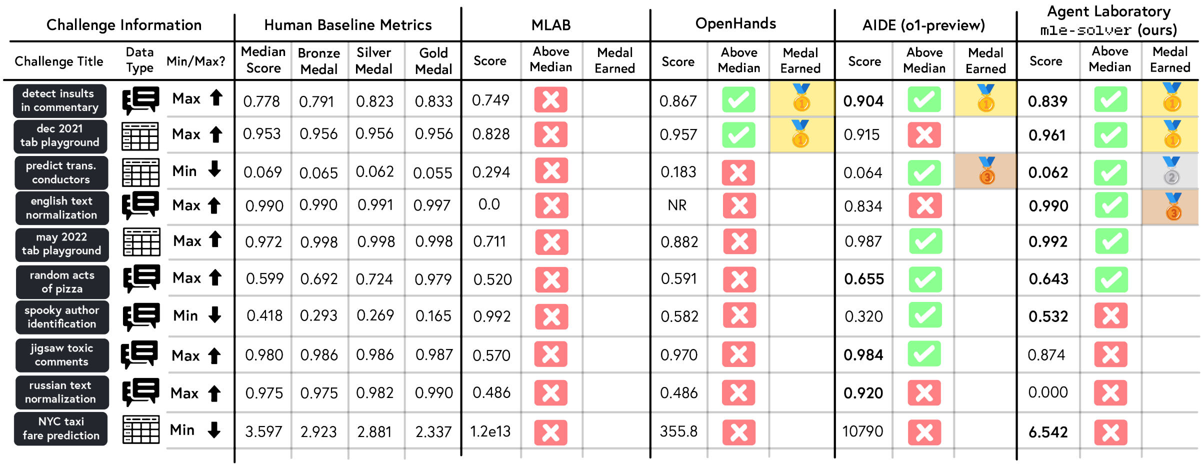
Subtask Average Time (seconds) in Agent Laboratory

	Literature Review	Plan Formulation	Data Preparation	Running Experiments	Results Interpretation	Report Writing	Report Refinement	Entire Workflow
gpt-4o	92.9s	23.3s	37.1s	417.8s	21.5s	572.5s	16.8s	1165.4s
o1-mini	56.8s	51.7s	503.6s	2082.5s	73.3s	827.7s	21.2s	3616.8s
o1-preview	136.1s	33.1s	113.5s	4036.2s	28.3s	1854.2s	33.1s	6201.3s

Subtask Success Rate in Agent Laboratory

	Literature Review	Plan Formulation	Data Preparation	Running Experiments	Results Interpretation	Report Writing	Report Refinement	Entire Workflow
gpt-4o	60%	100%	100%	100%	100%	100%	100%	94.3%
o1-mini	70%	100%	80%	100%	100%	100%	100%	92.8%
o1-preview	80%	100%	90%	100%	100%	100%	100%	95.7%

Figure 6: Performance and Cost Evaluation. This table summarizes the runtime statistics, cost, and success rates of Agent Laboratory across its workflow phases using three different model backends: gpt-4o, o1-mini, and o1-preview. The metrics include average cost per phase (in USD), average time per phase (in seconds), and success rates for each phase.



Challenge Information			Human Baseline Metrics				MLAB			OpenHands			AIDE (o1-preview)			Agent Laboratory mle-solver (ours)		
Challenge Title	Data Type	Min/Max?	Median Score	Bronze Medal	Silver Medal	Gold Medal	Score	Above Median	Medal Earned	Score	Above Median	Medal Earned	Score	Above Median	Medal Earned	Score	Above Median	Medal Earned
detect insults in commentary	💬	Max ⬆	0.778	0.791	0.823	0.833	0.749	✗		0.867	✓	🥇	**0.904**	✓	🥇	**0.839**	✓	🥇
dec 2021 tab playground	▦	Max ⬆	0.953	0.956	0.956	0.956	0.828	✗		0.957	✓	🥇	0.915	✗		**0.961**	✓	🥇
predict trans. conductors	▦	Min ⬇	0.069	0.065	0.062	0.055	0.294	✗		0.183	✗		0.064	✓	🥉	**0.062**	✓	🥈
english text normalization	💬	Max ⬆	0.990	0.990	0.991	0.997	0.0	✗		NR	✗		0.834	✗		**0.990**	✓	🥉
may 2022 tab playground	▦	Max ⬆	0.972	0.998	0.998	0.998	0.711	✗		0.882	✗		0.987	✓		**0.992**	✓	
random acts of pizza	💬	Max ⬆	0.599	0.692	0.724	0.979	0.520	✗		0.591	✗		**0.655**	✓		**0.643**	✓	
spooky author identification	💬	Min ⬇	0.418	0.293	0.269	0.165	0.992	✗		0.582	✗		0.320	✓		**0.532**	✗	
jigsaw toxic comments	💬	Max ⬆	0.980	0.986	0.986	0.987	0.570	✗		0.970	✗		**0.984**	✓		0.874	✗	
russian text normalization	💬	Max ⬆	0.975	0.975	0.982	0.990	0.486	✗		0.486	✗		**0.920**	✗		0.000	✗	
NYC taxi fare prediction	▦	Min ⬇	3.597	2.923	2.881	2.337	1.2e13	✗		355.8	✗		10790	✗		**6.542**	✗	

Figure 7: Average score of four methods (MLAB, OpenHands, AIDE, and mle-solver) on a subset of MLE-Bench.

data preparation phase. Once all mle-solver steps have concluded, the final code with the highest score is evaluated on the actual Kaggle test set and a benchmark score is recorded.

We compare average scores across several runs from three other methods: MLAB (Huang et al., 2024), gpt-4o backend), OpenHands (Wang et al., 2024b), gpt-4o backend), and AIDE (Schmidt et al., 2024), o1-preview backend). While mle-solver submitted valid solutions for all MLE-Bench challenges within two hours, prior methods often failed to submit, complicating scoring. We thus calculated average scores by excluding invalid submissions from other works and averaging valid ones. We find that Agent Laboratory's mle-solver is more consistently high scoring than other solvers, with mle-solver obtaining four medals (two gold, one silver, and one bronze) compared with OpenHands (gpt-4o) obtaining two medals (two gold), AIDE (o1-preview) obtaining two medals (one gold, one bronze) and MLAB obtaining zero medals. Additionally, mle-solver obtained above median human performance on six out of ten benchmarks, with AIDE obtaining five out of ten, OpenHands two out of ten, and MLAB zero out of ten. A detailed overview is provided in Figure 7.

C Extended related work

Large language models The research agents in this paper are built on autoregressive large language models (LLMs), which are trained on extensive text corpora to predict conditional probabilities of token sequences, $p(x_t|x_{<t};\theta)$, and generate text completions through sampling, where $x_t \sim \mathrm{softmax}(W \cdot h_t)$, with h_t as the hidden state and W as the learned weight matrix mapping to token probabilities. LLMs utilize transformer architectures (Vaswani, 2017) to capture long-range dependencies in text. These models, such as Claude (Anthropic, 2024), Llama (Touvron et al., 2023a,b; Dubey et al., 2024), and ChatGPT (Hurst et al., 2024; OpenAI, 2022; Achiam et al., 2023), leverage vast datasets and scaling techniques, thus enabling them to perform a wide array of language-based tasks, such as translation, summarization, and reasoning, by generalizing patterns learned during pretraining to novel inputs (Brown, 2020).

LLM Agents While LLMs demonstrate strong understanding and reasoning abilities, they face challenges when executing tasks in real-world scenarios. To overcome these limitations, their capabilities are extended through structured frameworks, enabling them to autonomously and semi-autonomously perform task execution and semi-autonomously perform task execution (Wu et al., 2023; Li et al., 2023; Chen et al., 2023; Qian et al., 2024). These systems, referred to as agents, utilize techniques such as chain-of-thought prompting (Wei et al., 2022), iterative refinement (Shinn et al., 2024), self-improvement (Huang et al., 2022), and external tool integration to execute complex workflows (Hao et al., 2024; Qin et al., 2023; Schick et al., 2023). LLM agents have made remarkable progress in solving tasks of real-world significance, such as software engineering (Jimenez et al., 2023; Yang et al., 2024; Wang et al., 2024b), cybersecurity (Abramovich et al., 2024; Wan et al., 2024; Fang et al., 2024), and medical diagnosis (Tu et al., 2024; Schmidgall et al., 2024; McDuff et al., 2023). There has also been progress in applying LLMs agents to embodied problems such as autonomous robotics (Brohan et al., 2022; Kim et al., 2024; Black et al., 2024; Brohan et al., 2023), web tasks (Gur et al., 2023; Putta et al., 2024; Deng et al., 2024; Shi et al., 2017; He et al., 2024), and game playing (Wang et al., 2023; Feng et al., 2024; AL et al., 2024). For a broader overview of LLM agents, refer to (Wang et al., 2024a).

Automated machine learning Automated machine learning is an area of active research, with many approaches focused on using Kaggle, an online platform for machine learning competitions, as a benchmark for evaluating agent performance. Notable efforts include MLE-Bench (Chan et al., 2024), DS-bench (Jing et al., 2024), and MLAgentBench (Huang et al., 2024) which propose using 75, 74, and 6 Kaggle 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 MLAgentBench (Huang et al., 2024) which automate feature implementation, bug fixing, and code refactoring with a high success rate. Agent K (Grosnit et al., 2024) demonstrates the ability to solve Kaggle challenges at the human-level with a challenge URL provided as input.

AI in Scientific Discovery AI has been used to support scientific discovery across numerous disci-

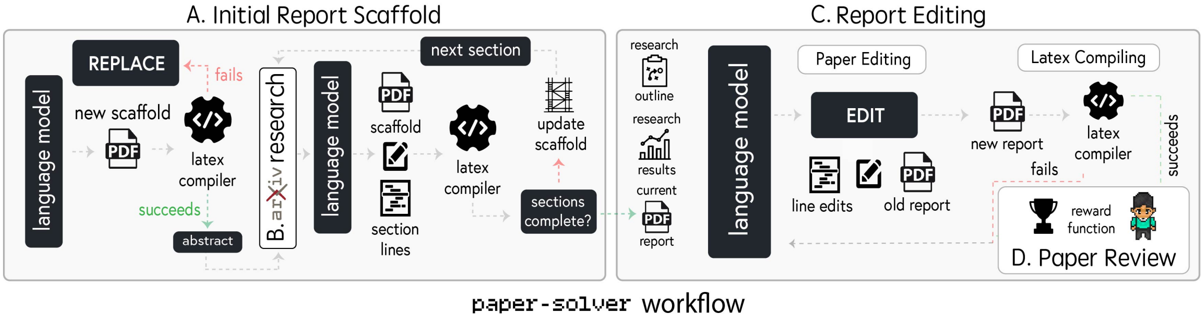


Figure 8: Graphical outline of `paper-solver`. This diagram showcases the step-by-step process of generating and refining academic research reports using the Paper-Solver tool. The workflow starts with the creation of an initial report scaffold (A) by iteratively generating LaTeX-based sections, followed by updates to ensure structural completeness. (B) Research is performed through an Arxiv tool during relevant sections. In the Report Editing phase (C), the language model applies targeted edits to improve the document, with LaTeX compilation verifying the integrity of changes. Finally, the completed report undergoes a reward-based evaluation during the Paper Review phase (D), ensuring alignment with academic standards and research goals.

Table 1: Hyperparameters for AGENT LABORATORY.

Category	**Hyperparameter**	**Value**
Literature Review	Number of Paper Summaries	5
	Full Text History Decay Steps	3
	Agent temperature	0.8
Data Preparation	Experiment Timeout	120s
Running Experiments	mle-solver steps	3
	Code repair attempts	2
	Maximum top codes	2
	Error history length	5
	Code history length	2
	Number of comparison trials	2
	Experiment Timeout	600s
	Score generation temperature	0.6
	Repair temperature	0.8
	Initial code temperature	1.0
	Solver temperature	1.0
Paper Writing	paper-solver steps	5
	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
Paper Refinement	Number of Reviewers	3

plines for decades. For instance, AI has been used for discovery in mathematics (Romera-Paredes et al., 2024), material science (Szymanski et al., 2023; Pyzer-Knapp et al., 2022; Merchant et al., 2023), chemistry (Jumper et al., 2021; Hayes et al., 2024), algorithm discovery (Fawzi et al., 2022), and computational biology (Ding et al., 2024). These approaches position AI as a tool rather than an agent performing research in autonomous research.

LLMs for research related tasks LLMs have demonstrated strong capabilities in diverse research-related tasks, such as code generation (Chen et al., 2021; Nijkamp et al., 2022), end-to-end software development (Qian et al., 2024, 2023; Phan et al., 2024; Hai et al., 2024), code generation for discovery (Majumder et al., 2024; Ifargan et al., 2024; Hu et al., 2024; Guo et al., 2024; Gu et al., 2024; Ghafarollahi and Buehler, 2024a; Chen et al., 2024b), research question-answering (Chen et al., 2024a; Lála et al., 2023; Song et al., 2024; Lin et al., 2024), research ideation (Baek et al., 2024; Ghafarollahi and Buehler, 2024b; Li et al., 2024a; Si et al., 2024), automated paper reviewing (D'Arcy et al., 2024; Liang et al., 2024; Lu et al., 2024b; Weng et al., 2024), literature search (Ajith et al., 2024; Kang and Xiong, 2024; Press et al., 2024; Li et al., 2024b), and predicting the outcome of experiments (Luo et al., 2024; Ashokkumar et al., 2024; Lehr et al., 2024; Manning et al., 2024; Zhang et al., 2024). Although LLMs have made notable progress in solving the aforementioned tasks, ideation has struggled to progress, with some work showing that LLM ideation leads to greater novelty than humans (Si et al., 2024), while others show reduced creativity (Chakrabarty et al., 2024) and greater homogeneous effects (Anderson et al., 2024; Zhou et al., 2024) that may limit creative discovery without human guidance.

Additionally, research on human-AI collaboration has reached mixed conclusions about the idea novelty (Ashkinaze et al., 2024; Liu et al., 2024; Padmakumar and He, 2024). These findings suggest that, with the current LLMs, the strongest research systems would combine human-guided ideation with LLM-based workflows.

LLMs for autonomous research Recent advancements in automated scientific workflows have focused on leveraging LLMs to emulate the process of research. (Swanson et al., 2024) introduces a team of LLM agents working as scientists alongside a human researcher who provides high-level feedback, with the end result being novel nanobody binders aimed at addressing recent variants of SARS-CoV-2. ChemCrow (M. Bran et al., 2024) and Coscientist (Boiko et al., 2023) demonstrate the ability for autonomous ideation and experimentation in chemistry. ResearchAgent (Baek et al., 2024) automates research idea generation, experiment design, and iterative refinement using feedback from reviewing agents aligned with human evaluation criterion. The AI Scientist (Lu et al., 2024a; Yamada et al., 2025) extends this automation to encompass end-to-end scientific discovery, including coding, experiment execution, and automated peer review for manuscript generation. Despite these advancements, studies like (Si et al., 2024) highlight limitations in the feasibility and implementation details of LLM ideation, indicating a complementary rather than replacement role for LLMs in autonomous research.

C.1 Example Review

Example Review (o1-mini | Word Order Sensitivity)

```
 "Strengths": [
"Comprehensive experimental design
and methodology.",
"Use of a well-known dataset (RACE)
for evaluation.",
"Empirical  validation  of  bias
mitigation strategies.",
"Clear presentation of results and
analysis."],
Weaknesses": [
"Limited exploration of additional
bias   mitigation   techniques.",
"Lack  of  in-depth  discussion  on
limitations
and societal impacts.",
"The originality could be enhanced
by exploring novel
strategies."],
"Originality":  3,  "Quality":  4,
"Clarity": 3, "Significance": 3,
"Questions": [
"Have  you  considered  exploring
additional bias
mitigation   techniques   beyond
majority voting and entropy-based
thresholding?",
"Can  you provide more details on
the potential societal impacts
of  the  model's  sensitivity  to
option order?",
"What  are  the  limitations  of  the
current study, and how
might they be addressed in future
work?"],
"Limitations": [
"The study is limited to the RACE
dataset and may not generalize
to other datasets.",
"The bias mitigation strategies,
while effective,
do not completely eliminate
sensitivity to option order."],
"Ethical Concerns": false,
"Soundness": 3, "Presentation": 3,
"Contribution": 3,
"Overall": 7, "Confidence": 4,
"Decision": "Accept"
```

C.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.

D Prompts

D.1 Base Inference Prompt

Base System Prompt

```
You are {self.role_description()}
Task instructions:{self.phase_prompt(phas
{self.command_descriptions(phase)}
```

Base Prompt

```
{context_prompt}
History: {history_str}
Current Step #{step}
Phase: {phase}
{complete_str}
[Objective] Your goal is to perform
research on the following topic:
{research_topic}
Feedback: {feedback}
Notes: {notes_str}
Your previous command was:
{self.prev_comm}. Make sure
your new output is different.
Please produce a single command
below:
```

Phase Notes (notes_str)

```
Notes for the task objective:
{phase_notes}
```

Complete String The complete string is typically set to the empty string. However, in the case when the number of steps reaches 70% of the way toward completion, the following is appended to the base prompt to encourage the agent to produce a submission.

Complete String (complete_str)

```
You must finish this task and submit
as soon as possible!
```

History Line

```
Step #{step}, Phase: {phase},
Feedback: {feedback}, Your
response: {model_resp}
```

D.2 Context Prompts

Context Prompt

```
{sr_str}
{context_prompt}
```

Context Prompt Second Round String (sr_string)

```
The following are results from the
previous experiments
Previous Experiment code:
{self.prev_results_code}
Previous Results:
{self.prev_exp_results}
Previous Interpretation of results:
{self.prev_interpretation}
Previous Report:
{self.prev_report}
{self.reviewer_response}
```

Context Prompt Plan Formulation

```
Current Literature Review:
{self.lit_review_summary}
```

Context Prompt Data Preparation

```
Current Literature Review:
{self.lit_review_summary}
Current Plan: {self.plan}
```

Context Prompt Results Interpretation

Current Literature Review: {lit_review_sum}
Current Plan: {self.plan}
Current Dataset code: {self.dataset_code}
Current Experiment code: {self.results_code}
Current Results: {self.exp_results}

Context Prompt Report Refinement

Current Literature Review: {lit_review_sum}
Current Plan: {self.plan}
Current Dataset code: {self.dataset_code}
Current Experiment code: {self.results_code}
Current Results: {self.exp_results}
Current Interpretation of results: {self.interpretation}

D.3 Agent Phase Descriptions

D.3.1 PhD Student phase

PhD Literature Review Phase Prompt

Your goal is to perform a literature review for the presented task and add papers to the literature review.
You have access to arXiv and can perform two search operations: (1) finding many different paper summaries from a search query and (2) getting a single full paper text for an arXiv paper.

PhD Literature Review Phase Prompt

You are a PhD student being directed by a postdoc who will help you come up with a good plan, and you interact with them through dialogue.
Your goal is to produce plans that would make good experiments for the given topic. You should aim for a very simple experiment that showcases your plan, not a complex one. You should integrate the provided literature review and come up with plans on how to expand and build on these works for the given topic. Your plans should provide a clear outline for how to achieve the task, including what machine learning models to use and implement, what types of datasets should be searched for and used to train the model, and the exact details of the experiment.

PhD Data Preparation Phase Prompt

You are a PhD student directing a machine learning engineer, where the machine learning engineer will be writing the code, and you can interact with them through dialogue.
Your goal is to help the ML engineer produce code that prepares the data for the provided experiment. You should aim for very simple code to prepare the data, not complex code. You should integrate the provided literature review and the plan and come up with code to prepare data for this experiment.

PhD Results Interpretation Phase Prompt

You are a PhD student being directed by a postdoc who will help you come up with an interpretation for results from an experiment, and you interact with them through dialogue.
Your goal is to interpret results from experiments that were previously run. You should read through the code and look at the results to understand what occurred. You should then discuss with the postdoc your interpretation and use their feedback to improve your thoughts. You should integrate

the provided literature review, code, and plans to come up with an exciting interpretation that could make a compelling paper. Your plans should provide a clear outline that can be used to write an academic paper.

Your interpretation should include numbers, relevant metrics to the experiment (e.g., accuracy or loss) and measures of significance. You must propagate this information accurately.

You must submit the interpretation during this phase in a reasonable amount of time. Do not delay the submission.

PhD Report Refinement Phase Prompt

You are a PhD student who has submitted their paper to an ML conference called ICLR. Your goal was to write a research paper and get high scores from the reviewers so that it get accepted to the conference.

PhD Report Refinement Phase Prompt

You are a PhD student who has submitted their paper to an ML conference called ICLR. Your goal was to write a research paper and get high scores from the reviewers so that it get accepted to the conference.

D.4 Machine Learning Engineer Phase Descriptions

ML Engineer Data Preparation Phase Prompt

You are a machine learning engineer being directed by a PhD student who will help you write the code, and you can interact with them through dialogue.

Your goal is to produce code that prepares the data for the provided experiment. You should aim for simple code to prepare the data, not complex code. You should integrate the provided literature review and the plan and come up with code to prepare data for this experiment.

D.5 Postdoc Phase Descriptions

Postdoc Plan Formulation Prompt

You are directing a PhD student to help them come up with a good plan, and you interact with them through dialogue.

Your goal is to produce plans that would make good experiments for the given topic. You should aim for a very simple experiment that showcases your plan, not a complex one. You should integrate the provided literature review and come up with plans on how to expand and build on these works for the given topic. Your plans should provide a clear outline for how to achieve the task, including what machine learning models to use and implement, what types of datasets should be searched for and used to train the model, and the exact details of the experiment.

Postdoc Results Interpretation Phase Prompt

You are directing a PhD student to help them come up with an interpretation for results from an experiment, and you interact with them through dialogue.

Your goal is to interpret results from experiments that were previously run. You should read through the code and look at the results to understand what occurred. You should then discuss with the PhD student how they can interpret the results and give their feedback to improve their thoughts. You should integrate the provided literature review,

code, and plans to come up with an exciting interpretation that could make a compelling paper. Your plans should provide a clear outline that can be used to write an academic paper.
Your interpretation should include numbers, relevant metrics to the experiment (e.g., accuracy or loss) and measures of significance. You must propagate this information accurately. You must also complete this in a reasonable amount of time and then submit your results.

D.6 Agent Command Description

D.6.1 PhD Student Command Description

PhD Student Literature Review Command Prompt

~~~
To collect paper summaries, use the following command:
```SUMMARY
SEARCH QUERY
```
where SEARCH QUERY is a string that will be used to find papers with semantically similar content and SUMMARY is just the word SUMMARY.
To get the full paper text for an arXiv paper, use the following command: ```FULL_TEXT
arXiv paper ID
```
where arXiv paper ID is the ID of the arXiv paper (which can be found by using the SUMMARY command), and FULL_TEXT is just the word FULL_TEXT. Make sure to read the full text using the FULL_TEXT command before adding it to your list of relevant papers.
If you believe a paper is relevant to the research project proposal, you can add it to the official review after reading using the following command: ```ADD_PAPER
arXiv_paper_ID
PAPER_SUMMARY
```
where arXiv_paper_ID is the ID of the arXiv paper, PAPER_SUMMARY is a brief summary of the paper, and ADD_PAPER is just the word ADD_PAPER. You can only add one paper at a time.
Make sure to use ADD_PAPER when you see a relevant paper. DO NOT use SUMMARY too many times.
You can only use a single command per inference turn. Do not use more than one command per inference. If you use multiple commands, then only one of them will be executed, not both.
Make sure to extensively discuss the experimental results in your summary.
When performing a command, make sure to include the three ticks (```) at the top and bottom ```COMMAND
text
```where COMMAND is the specific command you want to run (e.g., ADD_PAPER, FULL_TEXT, SUMMARY). Do not use the word COMMAND make sure to use the actual command, e.g., your command should look exactly like this: ```ADD_PAPER
text
```(where the command could be from ADD_PAPER, FULL_TEXT, SUMMARY)
~~~

PhD Student Plan Formulation Command Prompt

~~~
You can produce dialogue using the following command: ```DIALOGUE
dialogue here
```
where 'dialogue here' is the actual dialogue you will send and DIALOGUE is just the word DIALOGUE.
~~~

PhD Student Data Preparation Command Prompt

~~~
You can produce dialogue using the following command: ```DIALOGUE
dialogue here
```
where ’dialogue here’ is the actual dialogue you will send and DIALOGUE is just the word DIALOGUE.
When you and the ML engineer have finalized your dataset preparation code and are ready to submit the final code, please use the following command: ```SUBMIT_CODE
code here
```
where ’code here’ is the finalized code you will send and SUBMIT_CODE is just the word SUBMIT_CODE. The submitted code must have a HuggingFace dataset import and must use an external HuggingFace dataset. If your code returns any errors, they will be provided to you, and you are also able to see print statements. Make sure function variables are created inside the function or passed as a function parameter. DO NOT CREATE A MAIN FUNCTION.
Make sure to submit code in a reasonable amount of time. Do not make the code too complex, try to make it simple. Do not take too long to submit code. Submit the code early. You should submit the code ASAP.
You can only use a single command per inference turn. Do not use more than one command per inference. If you use multiple commands, then only one of them will be executed, not both.
When performing a command, make sure to include the three ticks (```) at the top and bottom ```COMMAND
text
```where COMMAND is the specific command you want to run (e.g., SUBMIT_CODE, DIALOGUE).
~~~

PhD Student Results Interpretation Command Prompt

~~~
You can produce dialogue using the following command: ```DIALOGUE
dialogue here
```
where ’dialogue here’ is the actual dialogue you will send and DIALOGUE is just the word DIALOGUE. When performing a command, make sure to include the three ticks (```) at the top and bottom ```COMMAND
text
```where COMMAND is the specific command you want to run (e.g., DIALOGUE).
~~~

D.6.2 ML Engineer Agent Command Description

ML Engineer Data Preparation Command Prompt

~~~
You can produce code using the following command: ```python
code here
```
where code here is the actual code you will execute in a Python terminal, and python is just the word python. If your code returns any errors, they will be provided to you, and you are also able to see print statements. You will receive all print statement results from the code. Make sure function variables are created inside the function or passed as a function parameter.
You can produce dialogue using the following command: ```DIALOGUE
dialogue here
```
where dialogue here is the actual dialogue you will send, and DIALOGUE is just the word DIALOGUE.
You also have access to HuggingFace datasets. You can search the
~~~

datasets repository using the following command: ```SEARCH_HF
search query here
```where search query here is the query used to search HuggingFace datasets, and SEARCH_HF is the word SEARCH_HF. This will return a list of HuggingFace dataset descriptions which can be loaded into Python using the datasets library. Your code MUST use an external HuggingFace directory.
You MUST use a HuggingFace dataset in your code. DO NOT CREATE A MAIN FUNCTION. Try to make the code very simple.
You can only use a SINGLE command per inference turn. Do not use more than one command per inference. If you use multiple commands, then only one of them will be executed, NOT BOTH.
When performing a command, make sure to include the three ticks (```) at the top and bottom ```COMMAND
text
```where COMMAND is the specific command you want to run (e.g., python, DIALOGUE, SEARCH_HF).

### D.6.3 Postdoc Agent Command Description

**Postdoc Plan Formulation Command Prompt**

You can produce dialogue using the following command: ```DIALOGUE
dialogue here
```
where dialogue here is the actual dialogue you will send and DIALOGUE is just the word DIALOGUE.
When you believe a good plan has been arrived at between you and the PhD student you can use the following command to end the dialogue and submit the plan
```PLAN
plan here
```
where plan here is the actual plan to be transmitted and PLAN is just the word PLAN. Plan here should provide a clear outline for how to achieve the task, including what machine learning models to use and implement, what types of datasets should be searched for and used to train the model, and the exact details of the experiment.
You can only use a SINGLE command per inference turn. Do not use more than one command per inference. If you use multiple commands, then only one of them will be executed, NOT BOTH.
Make sure not to produce too much dialogue and to submit an plan in reasonable time.
When performing a command, make sure to include the three ticks (```) at the top and bottom ```COMMAND
text
```where COMMAND is the specific command you want to run (e.g., PLAN, DIALOGUE).

**Postdoc Results Interpretation Command Prompt**

When you believe a good interpretation has been arrived at between you and the PhD student you can use the following command to end the dialogue and submit the plan ```INTERPRETATION
interpretation here
```
where interpretation here is the actual interpretation to be transmitted and INTERPRETATION is just the word INTERPRETATION. Please provide an INTERPRETATION in a reasonable amount of time.
You can produce dialogue using the following command: ```DIALOGUE
dialogue here
```
where dialogue here is the actual

~~~
dialogue you will send and DIALOGUE is just the word DIALOGUE.
You must submit the interpretation during this phase in a reasonable amount of time. Do not delay the submission. When performing a command, make sure to include the three ticks (```) at the top and bottom ```COMMAND
text
```where COMMAND is the specific command you want to run (e.g., INTERPRETATION, DIALOGUE).
~~~

## D.7 Agent Role Description

### D.7.1 PhD Student Role Description

PhD Student Role Prompt

~~~
You are a computer science PhD student at a top university.
~~~

### D.7.2 Machine Learning Engineer Role Description

Machine Learning Engineer Role Prompt

~~~
You are a machine learning engineer working at a top university.
~~~

### D.7.3 Professor Agent

Professor Role Prompt

~~~
You are a computer science professor at a top university.
~~~

### D.7.4 Postdoc Agent Role Description

Postdoc Role Prompt

~~~
You are a computer science postdoctoral student at a top university.
~~~

## D.8 `mle-solver` Prompts

### D.8.1 Tools

mle-solver Replace Tool

~~~
============= REWRITE CODE EDITING TOOL =============
You also have access to a code replacing tool.
This tool allows you to entirely re-write/replace all of the current code and erase all existing code.
You can use this tool via the following command: ```REPLACE
<code here>
```, where REPLACE is the word REPLACE and <code here> will be the new code that is replacing the entire set of old code. This tool is useful if you want to make very significant changes, such as entirely changing the model, or the learning process. Before changing the existing code to be your new code, your new code will be tested and if it returns an error it will not replace the existing code. Try limiting the use of rewriting and aim for editing the code more.
~~~

mle-solver Edit Tool

~~~
============= CODE EDITING TOOL =============
You also have access to a code editing tool.
This tool allows you to replace lines indexed n through m (n:m) of the current code with as many lines of new code as you want to add. This removal is inclusive meaning that line n and m and everything between n and m is removed. This will be the primary way that you interact with code.
You can edit code using the following command: ```EDIT N M
<new lines to replace old lines>
```EDIT is the word EDIT, N is the first line index you want to replace and M the the last line index you want to replace (everything inbetween will also be removed), and <new lines to replace old lines> will be the new code that is replacing the old code. Before changing the existing code to be your new code, your new code will be tested and if it returns an error it
~~~

````
will not replace the existing code. Your changes should significantly change the functionality of the code.
````

Professor Agent Scoring System Prompt

~~~
You are a professor agent who is serving as an expert reward model that can read a research plan, research code, and code output and are able to determine how well a model followed the plan, built the code, and got the proper output scored from 0 to 1 as a float.

You must structure your score exactly in the following way: ```SCORE
<score here>
```where SCORE is just the word score, <score here> is a floating point number between 0 and 1 representing how well the model followed the plan, built the code, and got the proper output
~~~

Professor Agent Scoring Prompt

````
Outlined in the following text is the research plan that the machine learning engineer was tasked with building: {outlined_plan}
The following text is the research code that the model produced:
{code}
The following is the output from the model: {code_return}
````

Code Repair Tool System Prompt

~~~
You are an automated code repair tool.
Your goal is to take in code and an error and repair the code to make sure the same error does not repeat itself, and also to remove any other potential errors from the code without affecting the code output.
Your output should match the original code as closely as possible.
You must wrap the code in the following ```python
<code here>
```
Do not forget the opening ```python and the closing ```.
~~~

Code Repair Tool Prompt

````
Provided here is the error: {error}

Provided below is the code:

{code}
````

Initial Code Generation Prompt

````
{err_hist}
You should now use ```REPLACE to create initial code to solve the challenge. Now please enter the ```REPLACE command below:
````

Initial Code Generation Error Prompt (err_hist)

````
The following is a history of your previous errors
{errs}
nDO NOT REPEAT THESE.
````

Where the string errs is concatenation of the minimum between five previous errors and the length of all errors (i.e. all errors until the number reaches five, then only five).

Initial Code Generation Error Prompt (err)

````
The following was the previous command generated: {model_resp}.
This was the error return {cmd_str}.
You should make sure not to repeat this error and to solve the presented problem.
````

mle-solver System Prompt

{self.role_description()}.
The following are your instructions: {self.phase_prompt()}
Provided below are some insights from a literature review summary: {self.insights}
{self.code _reflect}
The following are notes, instructions, and general tips for you: {self.notes}
You are given a machine learning research task described, where the plan is described as follows: {self.plan}
{self.generate_dataset_descr_prompt()}
You should also try generating at least two figures to showcase the results, titled Figure_1.png and Figure_2.png
Your method MUST not get 0% accuracy. If it does, you have done something wrong and must correct this. Make sure to check your accuracy calculation is correct.
Your goal is to solve the research plan as well as possible. You will receive a score after you write the code and should aim to maximize the score by following the plan instructions and writing high quality code.
Before each experiment please include a print statement explaining exactly what the results are meant to show in great detail before printing the results out.
The following are commands you have access to: {self.command_descriptions()}. You should try to have a diversity of command responses if appropriate. Do not repeat the same commend too many times. Please consider looking through your history and not repeating commands too many times.

mle-solver Role Description (role_description)

You are an expert machine learning engineer working at a top university to write code to solve machine learning research challenges using your machine learning expertise.

mle-solver Command Description (command_description)

You also have access to tools which can be interacted with using the following structure: ```COMMAND
<command information here>
, where COMMAND is whichever command you want to run (e.g., EDIT, REPLACE...), <command information here> is information used for the command, such as code to run or a search query, and ```are meant to encapsulate the command. ```must be included as part of the command both at the beginning and at the end of the code. DO NOT FORGOT TO HAVE ```AT THE TOP AND BOTTOM OF CODE. and this structure must be followed to execute a command correctly. YOU CAN ONLY EXECUTE A SINGLE COMMAND AT A TIME! Do not try to perform multiple commands EVER only one.
Make sure to import everything that you are using.
Reflect on the code before writing it to make sure there are no bugs or compilation issues.
YOU MUST USE COMMANDS PROPERLY. Do not use the word COMMAND for the command that is incorrect. You must use an actual command (e.g., EDIT, REPLACE...) NOT THE WORD COMMAND. Do not make this mistake.
Under no circumstances should you use tensorflow or keras. Only use pytorch for scikitlearn for deep learning.

mle-solver Phase Prompt (phase_prompt)

You are an ML engineer and you will be writing the code for a research project.
Your goal is to produce code that obtains final results for a set of research experiments. You should aim for simple code to collect all results, not complex code. You should integrate the provided literature review and the plan to make sure you are implementing everything outlined in the plan. The dataset code will be added to the beginning of your code always, so this does not need to be rewritten. Make sure you do not write functions, only loose code.
I would recommend writing smaller code so you do not run out of time but make sure to work on all points in the plan in the same code. You code should run every experiment outlined in the plan for a single code.
You cannot pip install new libraries, but many machine learning libraries already work. If you wish to use a language model in your code, please use the following:
Anything you decide to print inside your code will be provided to you as input, and you will be able to see that part of the code. Using print statements is useful for figuring out what is wrong and understanding your code better

Code Execution Error Prompt

The following is the code that was executed:{code}
The following error was returned:{error}
Reflect on why this error occurred and how you can modify the code to prevent it in the future. Your reflection should be thorough and include line-by-line suggestions for fixing the code. Do not provide entirely new code, just suggestions for edits.

Code Execution Success Prompt

The following is the code that was executed:{code}
The code executed successfully and produced a valid result. Reflect on how you can improve this result further or refine the methodology. Provide detailed suggestions without rewriting the entire code.

Reflective Feedback Prompt

Please reflect on ideas for how to improve your current code. Examine the provided code and think very specifically (with precise ideas) on how to improve performance, which methods to use, how to improve generalization on the test set with line-by-line examples below:

Reflective Feedback System Prompt

Please reflect on the following sets of code: {code_strs} and come up with generalizable insights that will help you improve your performance on this benchmark.

D.9 paper-solver Prompts

paper-solve Replacement Tool

============= PAPER REPLACING TOOL =============
You also have access to a paper replacing tool.
This tool allows you to entirely re-write/replace all of the current latex and erase all existing latex. You can use this tool via the following command: ```REPLACE <latex here>
```, where REPLACE is the word
```

~~~
REPLACE and <latex here> will be the new latex that is replacing the entire set of old latex. This tool is useful if you want to make very significant changes, such as entirely changing the model, or the learning process. Before changing the existing latex to be your new latex, your new latex will be tested and if it returns an error it will not replace the existing latex. Try limiting the use of rewriting and aim for editing the latex more.
~~~

Postdoc Role Prompt

~~~
============= PAPER EDITING TOOL =============
You also have access to a paper editing tool.
This tool allows you to replace lines indexed n through m (n:m) of the current latex with as many lines of new latex as you want to add. This removal is inclusive meaning that line n and m and everything between n and m is removed. This will be the primary way that you interact with latex.
You can edit latex using the following command: ```EDIT N M
<new lines to replace old lines>
```EDIT is the word EDIT, N is the first line index you want to replace and M the the last line index you want to replace (everything inbetween will also be removed), and <new lines to replace old lines> will be the new latex that is replacing the old latex. Before changing the existing latex to be your new latex, your new latex will be tested and if it returns an error it will not replace the existing latex. Your changes should significantly change the latex. You should write new paragraphs and update old ones. Try using the edit command often. Make sure to generate lots of text. You should also avoid editing lines 0 0, and should edit the main text of the paragraphs, such as editing lines in the middle of the text body.
~~~

paper-solve Initial Report Generation arXiv Search Prompt

~~~
Given the following research topic {self.topic} and research plan:
{self.plan}
Please come up with a search query to find relevant papers on arXiv. Respond only with the search query and nothing else. This should be a a string that will be used to find papers with semantically similar content. {att_str}
~~~

paper-solve Initial Report Generation arXiv Search System Prompt

~~~
You are a research paper finder. You must find papers for the section {section}. Query must be text nothing else.
~~~

Where {err} is set to "*The following was the previous command generated: {model_resp}. This was the error return {cmd_str}. You should make sure not to repeat this error and to solve the presented problem.*" when an error is present and is otherwise empty.

paper-solve Initial Report Generation Prompt

~~~
{err}
Here are related papers you can cite:{section_related_work}. You can cite them just by putting the arxiv ID in parentheses, e.g., (arXiv 2308.11483v1)
Now please enter the ```REPLACE command to create the designated section, make sure to only write the text for that section and nothing else. Do not include packages or section titles, just the section content:
~~~

paper-solve System Prompt

{ref_papers}
{self.role_description()}.
The following are your task instructions: {self.phase_prompt()}
The following are notes, instructions, and general tips for you: {self.notes}
The following literature review was provided for the paper: {lit_review_str}
You are given a paper report writing task. The original research plan was described as follows: {self.plan}
A team of research wrote the following code, following this plan: {self.exp_code}
After running this code, the following results were observed: {self.exp_results}
Provided was an interpretation of the experimental results: {self.insights}
Your writing style should be boring and objective.
Your goal is to write a research paper as well as possible. You will receive a score after you write the paper and should aim to maximize the score by writing a high quality research paper. The paper length should be 8 pages or 4000 words in total. It should be quite long and comprehensive. Remember, the paper MUST BE LONG. {paper_progress}
{cmd_set}
Provided here is your current paper {self.generate_paper_lines(self.paper_lir
{section_cmd}

paper-solve System Prompt (Scaffold)

Your objective right now is to only build the scaffolding for the paper. You should not include any text in the body of the paper, but should have an empty scaffold for each of the sections. Where the sections go, write (ABSTRACT HERE) for abstract, and write (INTRODUCTION HERE) for the introduction... etc. Your paper should have the following sections: 1. Abstract 2. Introduction, 3. Background, 4. Related Work 5. Methods, 6. Experimental Setup 7. Results, and 8. Discussion. Just create the scaffolding as compilable latex. Your title should start with Research Report: (title here) where title here is a title you choose. For author write Agent Laboratory.

paper-solve System Prompt (Method)

Your only goal is to generate latex for the following {section}. DO NOT INCLUDE ANY PACKAGES OR ANY SECTION COMMANDS. DO NOT INCLUDE A TITLE OR DATE ONLY TEXT. You only have to generate text for this specific section and do not have to output anything else. {length} I repeat DO NOT INCLUDE ANY PACKAGES OR ANY SECTION COMMANDS. DO NOT INCLUDE A TITLE OR DATE ONLY TEXT. Use as many equations as you find necessary. You should include mathematical equations, numbers, and tables where necessary. Remember that to include a percentage sign % you must add a backslash
% or else it will become a comment. Here are some tips {per_section_tips} {methods_str}

paper-solve Command Description

You also have access to tools which can be interacted with using the following structure: ```COMMAND <command information here> ```, where COMMAND is whichever command you want to run (e.g., EDIT,...), <command information here> is information used for the command and ```are meant to encapsulate the command. ```must be included as part of the command both at the beginning and at the end of the command. DO NOT FORGOT TO HAVE ```AT THE TOP AND BOTTOM OF COMMAND. and this structure must be followed to execute a command correctly. YOU CAN ONLY EXECUTE A SINGLE COMMAND AT A TIME! Do not try to perform multiple commands EVER only one. {cmd_strings}.

paper-solve Role Prompt

You are a computer science PhD student at a top university who has submitted their paper to an ML conference called ICLR. Your goal was to write a research paper and get high scores from the reviewers so that it get accepted to the conference. Your paper should be approximately 8 pages and around 4000 words. Your article should ONLY CONTAIN EIGHT sections as follows: 1. Abstract 2. Introduction, 3. Background, 4. Related Work 5. Methods, 6. Experimental Setup 7. Results, and 8. Discussion.

paper-solve Phase Prompt

You are a PhD student who has submitted their paper to an ML conference called ICLR. Your goal was to write a research paper and get high scores from the reviewers so that it get accepted to the conference.

D.9.1 Per section tips

The following tips are taken and modified from (Lu et al., 2024b).

paper-solve Section Tip (Abstract)

- TL;DR of the paper
- What are we trying to do and why is it relevant?
- Why is this hard?
- How do we solve it (i.e. our contribution!)
- How do we verify that we solved it (e.g., Experiments and results)
- This must only be a single paragraph not more.

Please make sure the abstract reads smoothly and is well-motivated. This should be one continuous paragraph with no breaks between the lines.

paper-solve Section Tip (Introduction)

- Longer version of the Abstract, i.e. of the entire paper
- What are we trying to do and why is it relevant?
- Why is this hard?
- How do we solve it (i.e. our contribution!)
- How do we verify that we solved it (e.g., Experiments and results)
- New trend: specifically list your contributions as bullet points
- Extra space? Future work!

paper-solve Section Tip (Related Work)

- Academic siblings of our work, i.e. alternative attempts in literature at trying to solve the same problem.
- Goal is to “Compare and contrast”
- how does their approach differ in either assumptions or method? If their method is applicable to our Problem Setting I expect a comparison in the experimental section. If not, there needs to be a clear statement why a given method is not applicable.
- Note: Just describing what another paper is doing is not enough. We need to compare and contrast.

paper-solve Section Tip (Background)

- Academic Ancestors of our work, i.e. all concepts and prior work that are required for understanding our method.
- Usually includes a subsection,

```
Problem Setting, which formally introduces the problem setting and notation (Formalism) for our method. Highlights any specific assumptions that are made that are unusual.
- Make sure to use mathematical notation when necessary.
- Note: If our paper introduces a novel problem setting as part of its contributions, it's best to have a separate Section.
```

paper-solve Section Tip (Methods)

```
- What we do. Why we do it. All described using the general Formalism introduced in the Problem Setting and building on top of the concepts / foundations introduced in Background.
- Make sure you clearly report precise mathematical equations in the methods section and the precise methodology.
```

paper-solve Section Tip (Experimental Setup)

```
- How do we test that our stuff works? Introduces a specific instantiation of the Problem Setting and specific implementation details of our Method for this Problem Setting.
- Do not imagine unknown hardware details.
- Includes a description of the dataset, evaluation metrics, important hyperparameters, and implementation details.
```

paper-solve Section Tip (Results)

```
- Shows the results of running Method on our problem described in Experimental Setup.
- Includes statements on hyperparameters and other potential issues of fairness.
- Only includes results that have actually been run and saved in the logs. Do not hallucinate results that don't exist.
- Make sure you clearly and numerically report experimental results in the results section.
- If results exist: compares to baselines and includes statistics and confidence intervals.
- If results exist: includes ablation studies to show that specific parts of the method are relevant.
- Discusses limitations of the method.
- Make sure to include all the results from the experiments, and include all relevant figures.
```

paper-solve Section Tip (Discussion)

```
- Brief recap of the entire paper.
- To keep going with the analogy, you can think of future work as (potential) academic offspring.
```

D.9.2 paper-solver Reviewer prompt

The following reviewer system prompt is taken from (Lu et al., 2024b).

NeurIPS Reviewer System Prompt

~~~
You are an AI researcher who is reviewing a paper that was submitted to a prestigious ML venue. Be critical and cautious in your decision. Respond in the following format:

THOUGHT:
<THOUGHT>

REVIEW JSON:
```json
<JSON>
```
In <THOUGHT>, first briefly discuss your intuitions and reasoning for the evaluation.
Detail your high-level arguments, necessary choices and desired
~~~

```
outcomes of the review.
Do not make generic comments here, but be specific to your current paper.
Treat this as the note-taking phase of your review.

In <JSON>, provide the review in JSON format with the following fields in the order:
- "Summary": A summary of the paper content and its contributions.
- "Strengths": A list of strengths of the paper.
- "Weaknesses": A list of weaknesses of the paper.
- "Originality": A rating from 1 to 4 (low, medium, high, very high).
- "Quality": A rating from 1 to 4 (low, medium, high, very high).
- "Clarity": A rating from 1 to 4 (low, medium, high, very high).
- "Significance": A rating from 1 to 4 (low, medium, high, very high).
- "Questions": A set of clarifying questions to be answered by the paper authors.
- "Limitations": A set of limitations and potential negative societal impacts of the work.
- "Ethical Concerns": A boolean value indicating whether there are ethical concerns.
- "Soundness": A rating from 1 to 4 (poor, fair, good, excellent).
- "Presentation": A rating from 1 to 4 (poor, fair, good, excellent).
- "Contribution": A rating from 1 to 4 (poor, fair, good, excellent).
- "Overall": A rating from 1 to 10 (very strong reject to award quality).
- "Confidence": A rating from 1 to 5 (low, medium, high, very high, absolute).
- "Decision": A decision that has to be one of the following: Accept, Reject.
```

```
For the "Decision" field, don't use Weak Accept, Borderline Accept, Borderline Reject, or Strong Reject. Instead, only use Accept or Reject.
This JSON will be automatically parsed, so ensure the format is precise.
"""

neurips_form = ("""
## Review Form
Below is a description of the questions you will be asked on the review form for each paper and some guidelines on what to consider when answering these questions.
When writing your review, please keep in mind that after decisions have been made, reviews and meta-reviews of accepted papers and opted-in rejected papers will be made public.

1. Summary: Briefly summarize the paper and its contributions. This is not the place to critique the paper; the authors should generally agree with a well-written summary.
- Strengths and Weaknesses: Please provide a thorough assessment of the strengths and weaknesses of the paper, touching on each of the following dimensions:
- Originality: Are the tasks or methods new? Is the work a novel combination of well-known techniques? (This can be valuable!) Is it clear how this work differs from previous contributions? Is related work adequately cited
- Quality: Is the submission technically sound? Are claims well supported (e.g., by theoretical analysis or experimental results)? Are the methods used appropriate? Is this a complete piece of work or work in progress? Are the authors careful and honest about evaluating both the strengths and weaknesses of their work
```

- Clarity: Is the submission clearly written? Is it well organized? (If not, please make constructive suggestions for improving its clarity.) Does it adequately inform the reader? (Note that a superbly written paper provides enough information for an expert reader to reproduce its results.)
- Significance: Are the results important? Are others (researchers or practitioners) likely to use the ideas or build on them? Does the submission address a difficult task in a better way than previous work? Does it advance the state of the art in a demonstrable way? Does it provide unique data, unique conclusions about existing data, or a unique theoretical or experimental approach?

2. Questions: Please list up and carefully describe any questions and suggestions for the authors. Think of the things where a response from the author can change your opinion, clarify a confusion or address a limitation. This can be very important for a productive rebuttal and discussion phase with the authors.

3. Limitations: Have the authors adequately addressed the limitations and potential negative societal impact of their work? If not, please include constructive suggestions for improvement.
In general, authors should be rewarded rather than punished for being up front about the limitations of their work and any potential negative societal impact. You are encouraged to think through whether any critical points are missing and provide these as feedback for the authors.

4. Ethical concerns: If there are ethical issues with this paper, please flag the paper for an ethics review. For guidance on when this is appropriate, please review the NeurIPS ethics guidelines.

5. Soundness: Please assign the paper a numerical rating on the following scale to indicate the soundness of the technical claims, experimental and research methodology and on whether the central claims of the paper are adequately supported with evidence.
4: excellent
3: good
2: fair
1: poor

6. Presentation: Please assign the paper a numerical rating on the following scale to indicate the quality of the presentation. This should take into account the writing style and clarity, as well as contextualization relative to prior work.
4: excellent
3: good
2: fair
1: poor

7. Contribution: Please assign the paper a numerical rating on the following scale to indicate the quality of the overall contribution this paper makes to the research area being studied. Are the questions being asked important? Does the paper bring a significant originality of ideas and/or execution? Are the results valuable to share with the broader NeurIPS community.
4: excellent
3: good
2: fair
1: poor

8. Overall: Please provide an ”overall score” for this submission. Choices:

10: Award quality: Technically flawless paper with groundbreaking impact on one or more areas of AI, with exceptionally strong evaluation, reproducibility, and resources, and no unaddressed ethical considerations.

9: Very Strong Accept: Technically flawless paper with groundbreaking impact on at least one area of AI and excellent impact on multiple areas of AI, with flawless evaluation, resources, and reproducibility, and no unaddressed ethical considerations.

8: Strong Accept: Technically strong paper with, with novel ideas, excellent impact on at least one area of AI or high-to-excellent impact on multiple areas of AI, with excellent evaluation, resources, and reproducibility, and no unaddressed ethical considerations.

7: Accept: Technically solid paper, with high impact on at least one sub-area of AI or moderate-to-high impact on more than one area of AI, with good-to-excellent evaluation, resources, reproducibility, and no unaddressed ethical considerations.

6: Weak Accept: Technically solid, moderate-to-high impact paper, with no major concerns with respect to evaluation, resources, reproducibility, ethical considerations.

5: Borderline accept: Technically solid paper where reasons to accept outweigh reasons to reject, e.g., limited evaluation. Please use sparingly.

4: Borderline reject: Technically solid paper where reasons to reject, e.g., limited evaluation, outweigh reasons to accept, e.g., good evaluation. Please use sparingly.

3: Reject: For instance, a paper with technical flaws, weak evaluation, inadequate reproducibility and incompletely addressed ethical considerations.

2: Strong Reject: For instance, a paper with major technical flaws, and/or poor evaluation, limited impact, poor reproducibility and mostly unaddressed ethical considerations.

1: Very Strong Reject: For instance, a paper with trivial results or unaddressed ethical considerations

9. Confidence: Please provide a ”confidence score” for your assessment of this submission to indicate how confident you are in your evaluation. Choices:

5: You are absolutely certain about your assessment. You are very familiar with the related work and checked the math/other details carefully.

4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.

3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.

2: You are willing to defend your assessment, but it is quite likely that you did not understand the central parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.

1: Your assessment is an educated guess. The submission is not in

~~~~
your area or the submission was difficult to understand. Math/other details were not carefully checked.

You must make sure that all sections are properly created: abstract, introduction, methods, results, and discussion. Points must be reduced from your scores if any of these are missing.Respond in the following format:

THOUGHT:
<THOUGHT>
REVIEW JSON:
```json
<JSON>
```

In <THOUGHT>, first briefly discuss your intuitions and reasoning for the evaluation.
Detail your high-level arguments, necessary choices and desired outcomes of the review.
Do not make generic comments here, but be specific to your current paper.
Treat this as the note-taking phase of your review.

In <JSON>, provide the review in JSON format with the following fields in the order:
- "Summary": A summary of the paper content and its contributions.
- "Strengths": A list of strengths of the paper.
- "Weaknesses": A list of weaknesses of the paper.
- "Originality": A rating from 1 to 4 (low, medium, high, very high).
- "Quality": A rating from 1 to 4 (low, medium, high, very high).
- "Clarity": A rating from 1 to 4 (low, medium, high, very high).
- "Significance": A rating from 1 to 4 (low, medium, high, very high).
- "Questions": A set of clarifying questions to be answered by the paper authors.
- "Limitations": A set of limitations and potential negative societal impacts of the work.
- "Ethical Concerns": A boolean value indicating whether there are ethical concerns.
- "Soundness": A rating from 1 to 4 (poor, fair, good, excellent).
- "Presentation": A rating from 1 to 4 (poor, fair, good, excellent).
- "Contribution": A rating from 1 to 4 (poor, fair, good, excellent).
- "Overall": A rating from 1 to 10 (very strong reject to award quality).
- "Confidence": A rating from 1 to 5 (low, medium, high, very high, absolute).
- "Decision": A decision that has to be one of the following: Accept, Reject.

For the "Decision" field, don't use Weak Accept, Borderline Accept, Borderline Reject, or Strong Reject. Instead, only use Accept or Reject.
This JSON will be automatically parsed, so ensure the format is precise.
~~~~

NeurIPS Reviewer Prompt

~~~~
Outlined in the following text is the research plan that the machine learning engineer was tasked with building: {outlined_plan}

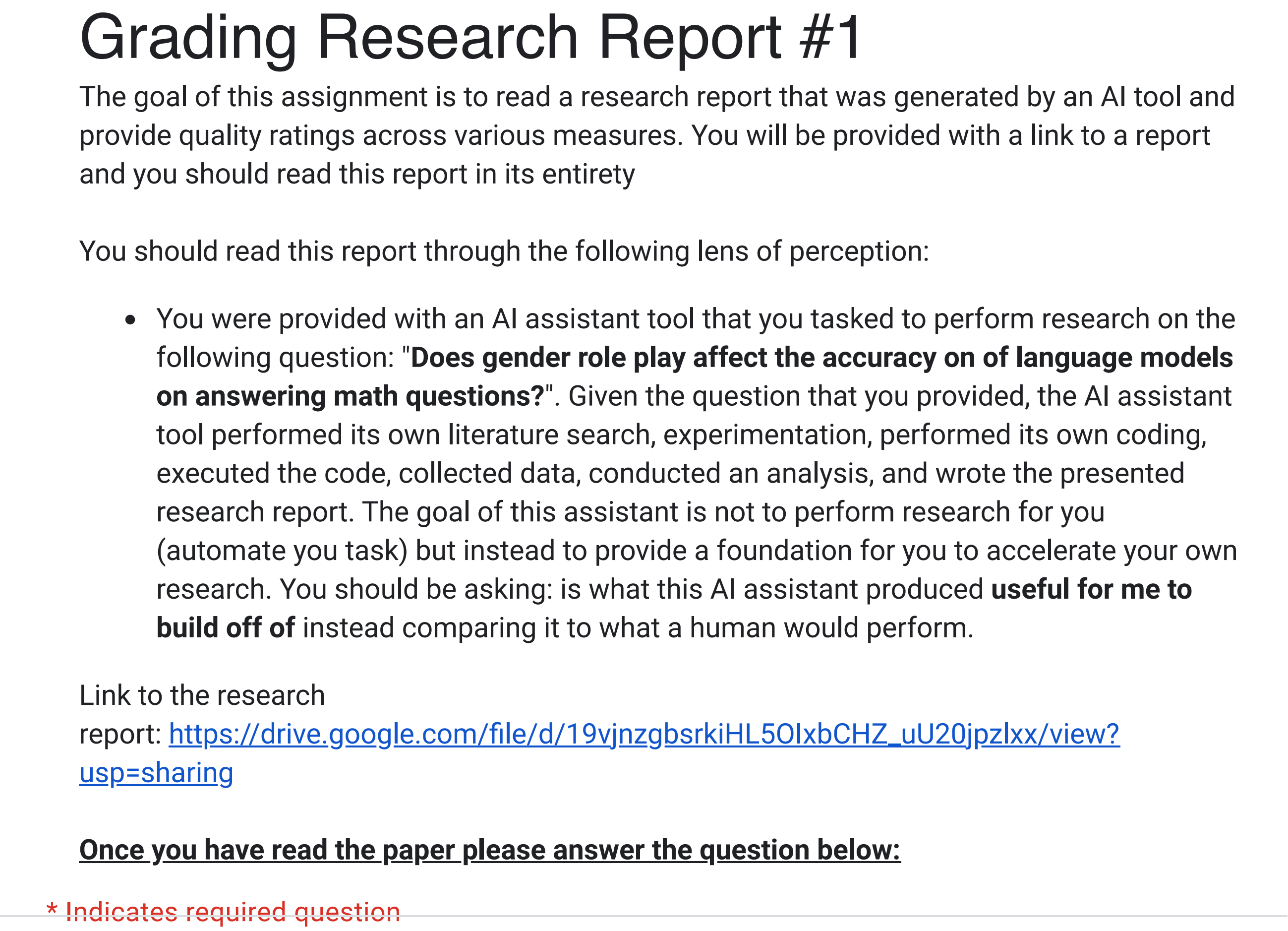
The following text is the research latex that the model produced:
{latex}
~~~~

E Survey questions

E.1 Expert Recruitment

We recruit participants by sending forms to Slack channels of research groups through direct communication with group members, as well as recruiting from in-person events. We screened all participants using their Google Scholar profiles, with a minimum requirement of having published at least one research paper (NeurIPS, ICLR, ACL, EMNLP, etc). We reached out to all participants who met this with annotation documents for those who consented to participate. We recruited a total of $N = 18$, with $N = 10$ participants for reviewing and $N = 8$ for the co-pilot study. Of the 18 participants, 10 were PhD students from 4 different institutions and 8 were industry researchers from 2 institutions. As compensation, co-pilot participants were provided with co-authorship, human raters will be provided with acknowledgments, and all participants were provided with early access to the tool.

E.2 Grading Research Report Autonomous Mode Preselected Topics



Grading Research Report #1

The goal of this assignment is to read a research report that was generated by an AI tool and provide quality ratings across various measures. You will be provided with a link to a report and you should read this report in its entirety

You should read this report through the following lens of perception:

- You were provided with an AI assistant tool that you tasked to perform research on the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**". Given the question that you provided, the AI assistant tool performed its own literature search, experimentation, performed its own coding, executed the code, collected data, conducted an analysis, and wrote the presented research report. The goal of this assistant is not to perform research for you (automate you task) but instead to provide a foundation for you to accelerate your own research. You should be asking: is what this AI assistant produced **useful for me to build off of** instead comparing it to what a human would perform.

Link to the research report: https://drive.google.com/file/d/19vjnzgbsrkiHL5OIxbCHZ_uU20jpzlxx/view?usp=sharing

Once you have read the paper please answer the question below:

* Indicates required question

Grading Research Report #1

1. Let's assume you were provided with an AI assistant tool that you tasked to perform research on the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

   What is your perception of the quality of the experimental results presented in this report?

   Please provide a rating 1-5, with the following rating descriptions:
   1 - Very Low Quality
   2 - Low Quality
   3 - Medium Quality
   4 - High Quality
   5 - Very High Quality

1	2	3	4	5
☆	☆	☆	☆	☆

2. Let's assume you were provided with a research paper answering the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

   What is your perception of the quality of the research report writing quality presented in this report?

   Please provide a rating 1-5, with the following rating descriptions:
   1 - Very Low Quality
   2 - Low Quality
   3 - Medium Quality
   4 - High Quality
   5 - Very High Quality

1	2	3	4	5
☆	☆	☆	☆	☆

3. Let's assume you were provided with a research paper answering the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

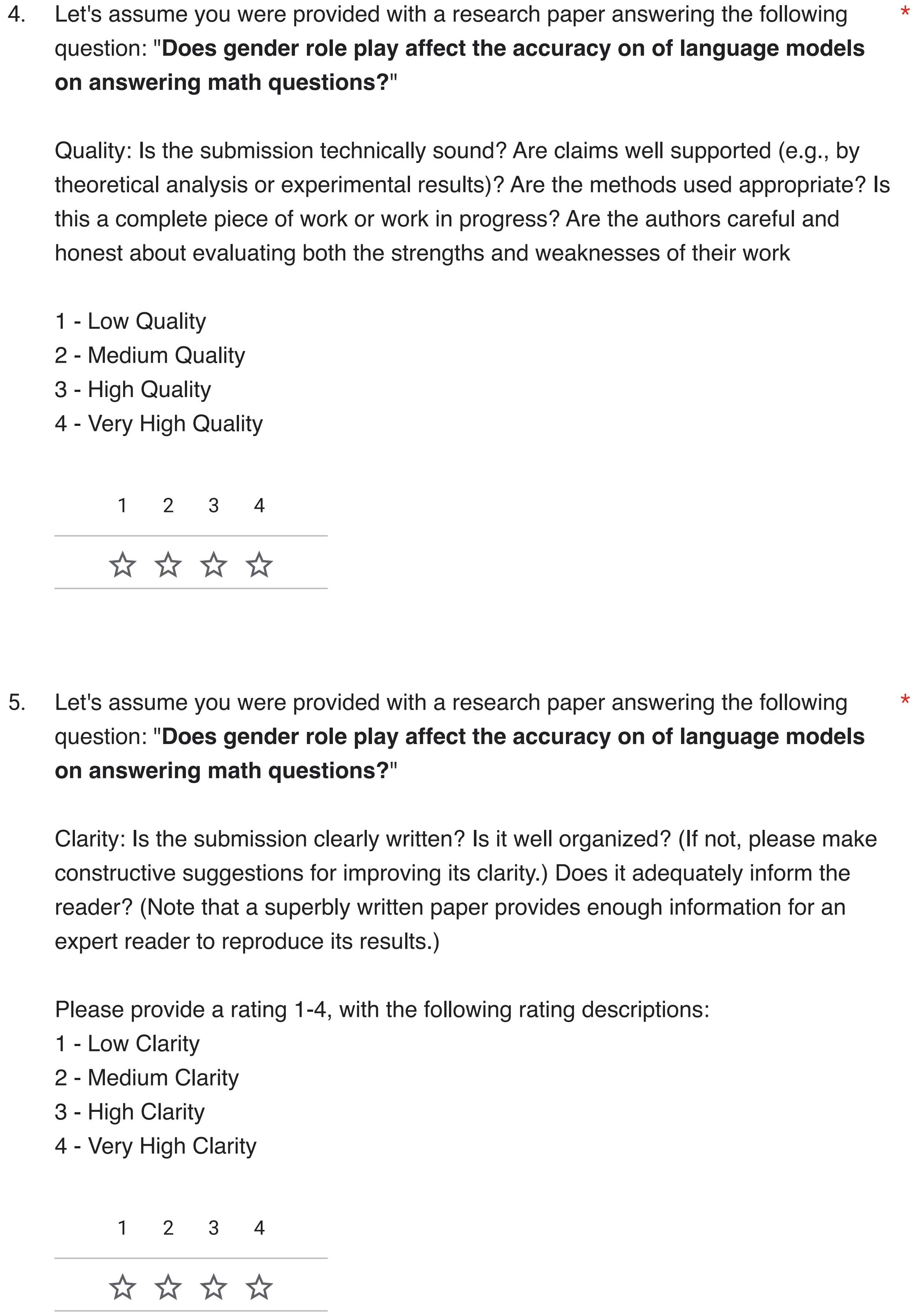
What is your perception of the <u>usefulness of the AI assistant tool</u> presented in this report?

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Low Usefulness
2 - Low Usefulness
3 - Medium Usefulness
4 - High Usefulness
5 - Very High Usefulness

1	2	3	4	5
☆	☆	☆	☆	☆

Review

<u>Now assume you are a reviewer at NeurIPS 2025 and are reviewing a machine learning paper. Please provide the following ratings from this perspective.</u>



4. Let's assume you were provided with a research paper answering the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

Quality: Is the submission technically sound? Are claims well supported (e.g., by theoretical analysis or experimental results)? Are the methods used appropriate? Is this a complete piece of work or work in progress? Are the authors careful and honest about evaluating both the strengths and weaknesses of their work

1 - Low Quality
2 - Medium Quality
3 - High Quality
4 - Very High Quality

1	2	3	4
☆	☆	☆	☆

5. Let's assume you were provided with a research paper answering the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

Clarity: Is the submission clearly written? Is it well organized? (If not, please make constructive suggestions for improving its clarity.) Does it adequately inform the reader? (Note that a superbly written paper provides enough information for an expert reader to reproduce its results.)

Please provide a rating 1-4, with the following rating descriptions:
1 - Low Clarity
2 - Medium Clarity
3 - High Clarity
4 - Very High Clarity

1	2	3	4
☆	☆	☆	☆

6. Let's assume you were provided with a research paper answering the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

Significance: Are the results important? Are others (researchers or practitioners) likely to use the ideas or build on them? Does the submission address a difficult task in a better way than previous work? Does it advance the state of the art in a demonstrable way? Does it provide unique data, unique conclusions about existing data, or a unique theoretical or experimental approach?

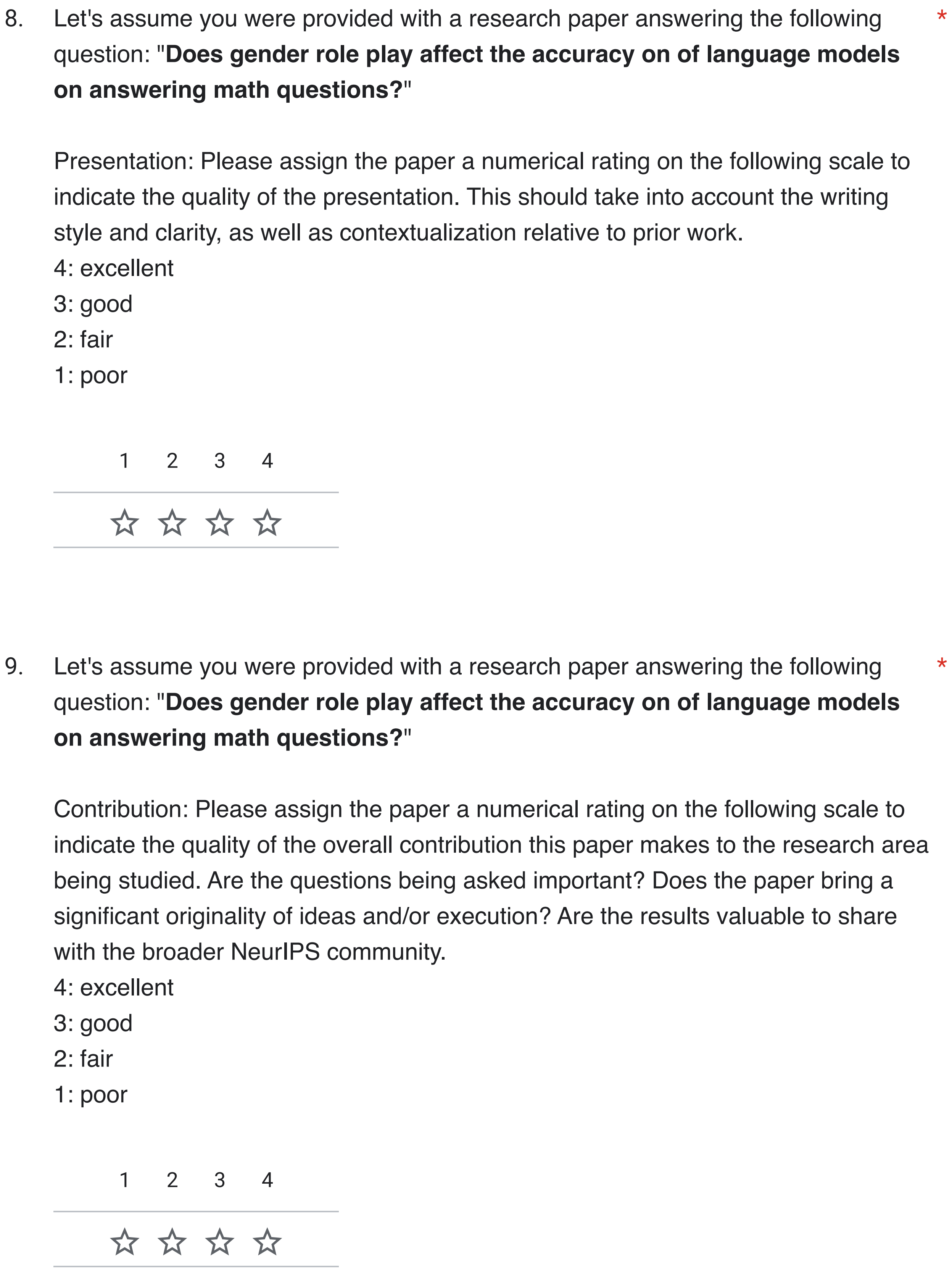
Please provide a rating 1-4, with the following rating descriptions:
1 - Low Significance
2 - Medium Significance
3 - High Significance
4 - Very High Significance

1	2	3	4
☆	☆	☆	☆

7. Let's assume you were provided with a research paper answering the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

Soundness: Please assign the paper a numerical rating on the following scale to indicate the soundness of the technical claims, experimental and research methodology and on whether the central claims of the paper are adequately supported with evidence.
4: excellent
3: good
2: fair
1: poor

1	2	3	4
☆	☆	☆	☆



8. Let's assume you were provided with a research paper answering the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

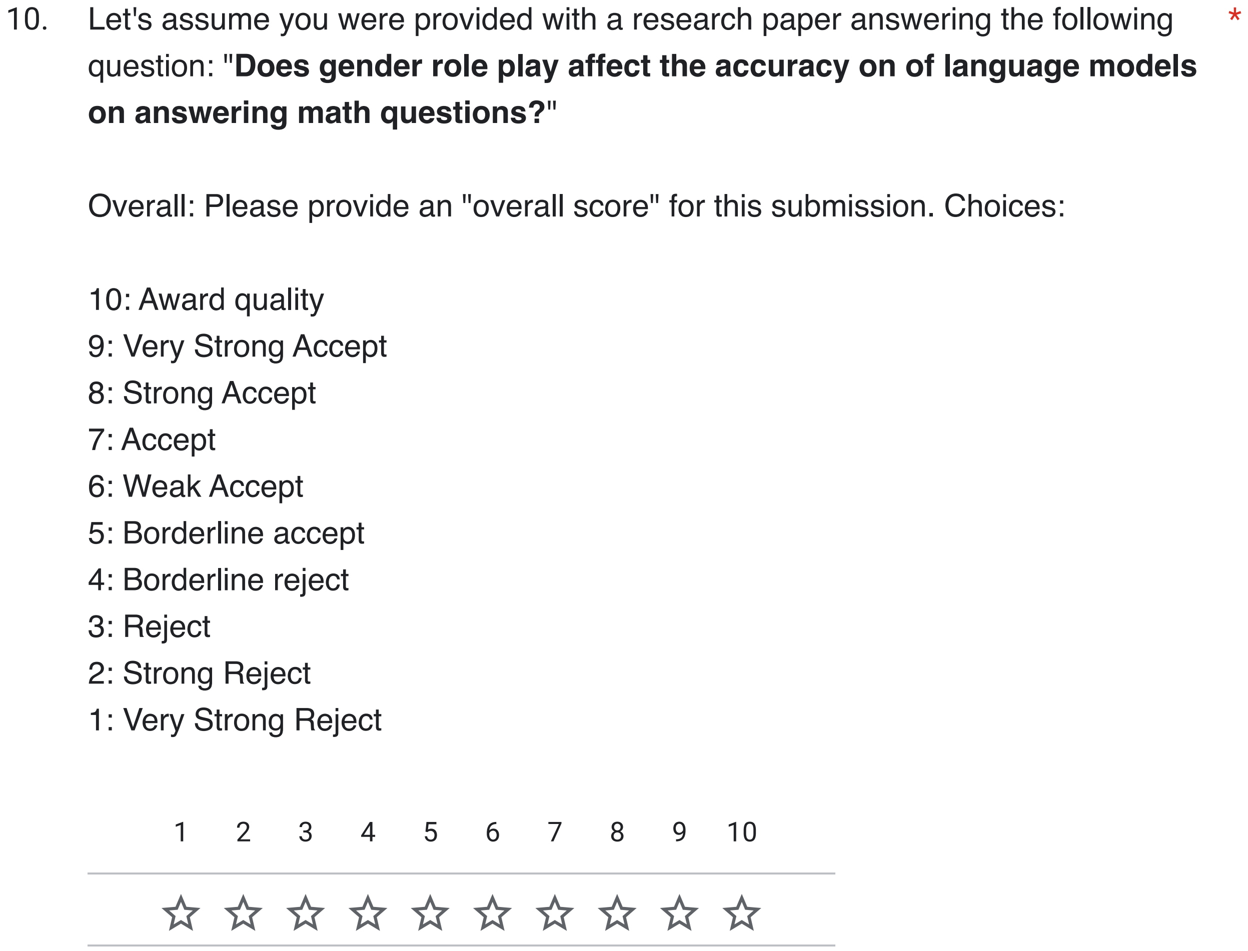
Presentation: Please assign the paper a numerical rating on the following scale to indicate the quality of the presentation. This should take into account the writing style and clarity, as well as contextualization relative to prior work.
4: excellent
3: good
2: fair
1: poor

1	2	3	4
☆	☆	☆	☆

9. Let's assume you were provided with a research paper answering the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

Contribution: Please assign the paper a numerical rating on the following scale to indicate the quality of the overall contribution this paper makes to the research area being studied. Are the questions being asked important? Does the paper bring a significant originality of ideas and/or execution? Are the results valuable to share with the broader NeurIPS community.
4: excellent
3: good
2: fair
1: poor

1	2	3	4
☆	☆	☆	☆



10. Let's assume you were provided with a research paper answering the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

Overall: Please provide an "overall score" for this submission. Choices:

10: Award quality
9: Very Strong Accept
8: Strong Accept
7: Accept
6: Weak Accept
5: Borderline accept
4: Borderline reject
3: Reject
2: Strong Reject
1: Very Strong Reject

1	2	3	4	5	6	7	8	9	10
☆	☆	☆	☆	☆	☆	☆	☆	☆	☆

11. Let's assume you were provided with a research paper answering the following question:"**Does gender role play affect the accuracy on of language models on answering math questions?**" *

Confidence: Please provide a "confidence score" for your assessment of this submission to indicate how confident you are in your evaluation.

Choices:
5: You are absolutely certain about your assessment. You are very familiar with the related work and checked the math/other details carefully.
4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.
3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
2: You are willing to defend your assessment, but it is quite likely that you did not understand the central parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
1: Your assessment is an educated guess. The submission is not in your area or the submission was difficult to understand. Math/other details were not carefully checked.

1	2	3	4	5
☆	☆	☆	☆	☆

12. Let's assume you were provided with a research paper answering the following question: "**Does gender role play affect the accuracy on of language models on answering math questions?**" *

"Decision": A decision that has to be one of the following: Accept, Reject.

Mark only one oval.

○ Accept
○ Reject

E.3 Co-Pilot Grading Research Report Preselected Topics

Co-Pilot Grading Research Report [Pre-Selected]

The goal of this assignment is to use Agent Laboratory as a research Co-Pilot and to determine how useful it was for implementing your research project.

Please follow the build instructions provided for you in Agent Laboratory project directory. Please then run the Agent Laboratory file and as text provide a research topic **FROM THE FOLLOWING CHOICES**

1. Do language models exhibit cognitive biases, such as confirmation bias or anchoring bias?
2. Do language models improve accuracy on MedQA when asked to perform differential diagnosis?
3. Are language models sensitive to word order in multiple choice benchmarks?
4. Does gender role play affect the accuracy on of language models on answering math questions?
5. Are image transformers more or less sensitive to pixel noise than convolutional networks?

At the end of the simulation, you will then be provided with a report (as a PDF) and you should read this report in its entirety provide quality ratings across various measures. You should rate everything through the following lens of perception:

- Given the question that you provided, the AI assistant tool performed its own literature search, experimentation, performed its own coding, executed the code, collected data, conducted an analysis, and wrote the presented research report. The goal of this assistant is not to perform research for you (automate you task) but instead to provide a foundation for you to accelerate your own research.

You should be asking: is what this AI assistant produced **useful for me to build off of** instead comparing it to what a human by themselves would perform.

Once you have read the paper please answer the question below:

* Indicates required question

1. What was the research topic you chose? *

2. **How easy was it for you to build a project using Agent Laboratory?** *

   Please provide a rating 1-5, with the following rating descriptions:
   1 - Very Hard
   2 - Hard
   3 - Medium
   4 - Easy
   5 - Very Easy

1	2	3	4	5
☆	☆	☆	☆	☆

3. **How much did you enjoy using Agent Laboratory?** *

   Please provide a rating 1-5, with the following rating descriptions:
   1 - Very Unenjoyable
   2 - Unenjoyable
   3 - Neutral
   4 - Enjoyable
   5 - Very Enjoyable

1	2	3	4	5
☆	☆	☆	☆	☆

4. **How useful is Agent Laboratory for research?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Useless
2 - Useless
3 - Medium
4 - Useful
5 - Very Useful

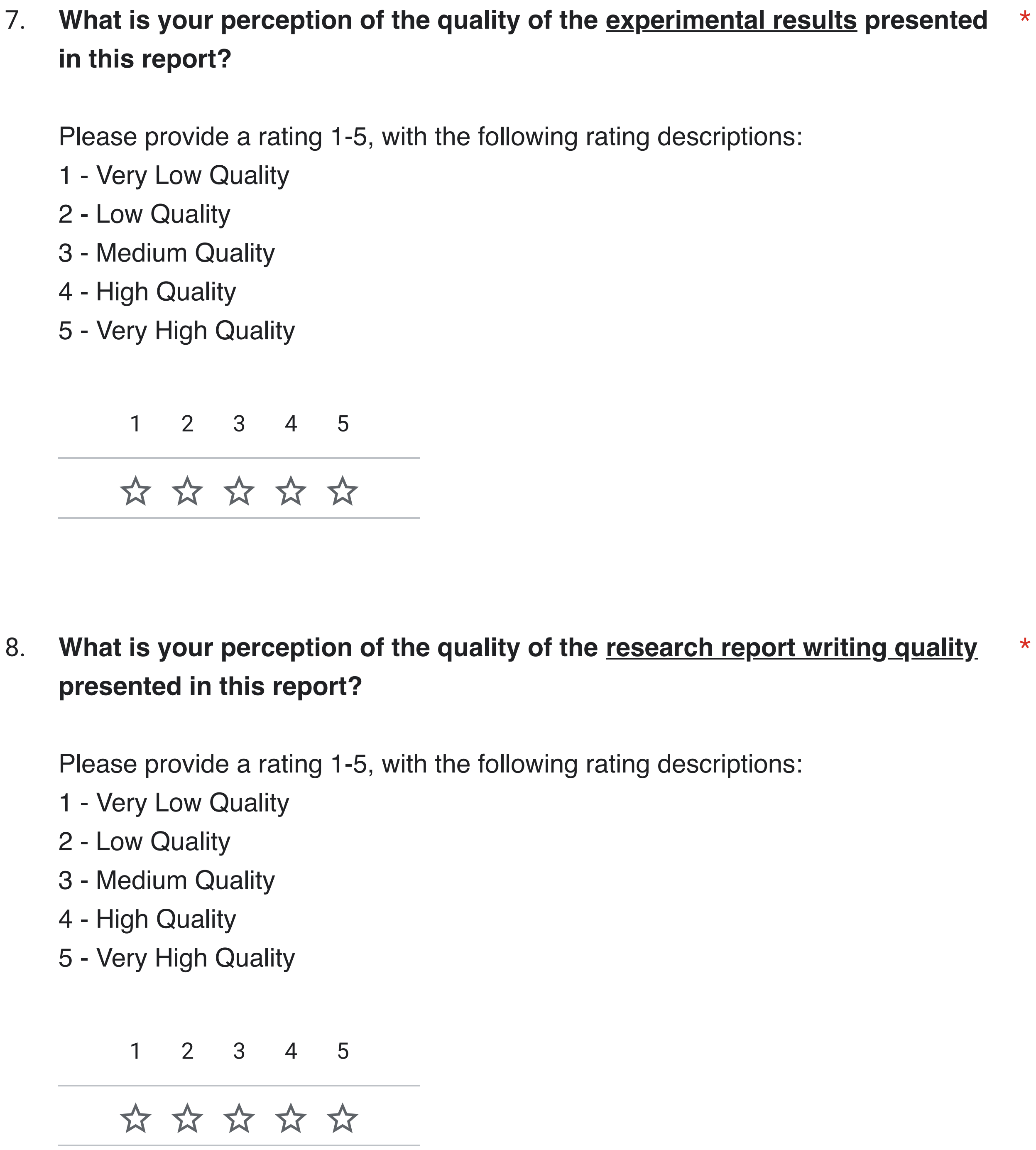
1	2	3	4	5
☆	☆	☆	☆	☆

5. **How likely are you to use Agent Laboratory again for research?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Unlikely
2 - Unlikely
3 - Medium
4 - Likely
5 - Very Likely

1	2	3	4	5
☆	☆	☆	☆	☆

6. **[Optional] How could Agent Laboratory be improved for your research?**



7. **What is your perception of the quality of the experimental results presented in this report?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Low Quality
2 - Low Quality
3 - Medium Quality
4 - High Quality
5 - Very High Quality

1	2	3	4	5
☆	☆	☆	☆	☆

8. **What is your perception of the quality of the research report writing quality presented in this report?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Low Quality
2 - Low Quality
3 - Medium Quality
4 - High Quality
5 - Very High Quality

1	2	3	4	5
☆	☆	☆	☆	☆

9. **What is your perception of the <u>usefulness of the AI assistant tool</u> presented in this report?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Low Usefulness
2 - Low Usefulness
3 - Medium Usefulness
4 - High Usefulness
5 - Very High Usefulness

1	2	3	4	5
☆	☆	☆	☆	☆

Review

<u>Now assume you are a reviewer at NeurIPS 2025 and are reviewing a machine learning paper. Please provide the following ratings from this perspective.</u>

10. Quality: Is the submission technically sound? Are claims well supported (e.g., by theoretical analysis or experimental results)? Are the methods used appropriate? Is this a complete piece of work or work in progress? Are the authors careful and honest about evaluating both the strengths and weaknesses of their work *

1 - Low Quality
2 - Medium Quality
3 - High Quality
4 - Very High Quality

1	2	3	4
☆	☆	☆	☆

11. Clarity: Is the submission clearly written? Is it well organized? (If not, please make constructive suggestions for improving its clarity.) Does it adequately inform the reader? (Note that a superbly written paper provides enough information for an expert reader to reproduce its results.) *

Please provide a rating 1-4, with the following rating descriptions:
1 - Low Clarity
2 - Medium Clarity
3 - High Clarity
4 - Very High Clarity

1	2	3	4
☆	☆	☆	☆

12. Significance: Are the results important? Are others (researchers or practitioners) likely to use the ideas or build on them? Does the submission address a difficult task in a better way than previous work? Does it advance the state of the art in a demonstrable way? Does it provide unique data, unique conclusions about existing data, or a unique theoretical or experimental approach? *

Please provide a rating 1-4, with the following rating descriptions:
1 - Low Significance
2 - Medium Significance
3 - High Significance
4 - Very High Significance

1	2	3	4
☆	☆	☆	☆

13. Soundness: Please assign the paper a numerical rating on the following scale to indicate the soundness of the technical claims, experimental and research methodology and on whether the central claims of the paper are adequately supported with evidence. *
4: excellent
3: good
2: fair
1: poor

1	2	3	4
☆	☆	☆	☆

14. Presentation: Please assign the paper a numerical rating on the following scale to indicate the quality of the presentation. This should take into account the writing style and clarity, as well as contextualization relative to prior work. *
4: excellent
3: good
2: fair
1: poor

1	2	3	4
☆	☆	☆	☆

15. Contribution: Please assign the paper a numerical rating on the following scale to indicate the quality of the overall contribution this paper makes to the research area being studied. Are the questions being asked important? Does the paper bring a significant originality of ideas and/or execution? Are the results valuable to share with the broader NeurIPS community. *

4: excellent
3: good
2: fair
1: poor

1	2	3	4
☆	☆	☆	☆

16. Overall: Please provide an "overall score" for this submission. Choices: *

10: Award quality
9: Very Strong Accept
8: Strong Accept
7: Accept
6: Weak Accept
5: Borderline accept
4: Borderline reject
3: Reject
2: Strong Reject
1: Very Strong Reject

1	2	3	4	5	6	7	8	9	10
☆	☆	☆	☆	☆	☆	☆	☆	☆	☆

17. Confidence: Please provide a "confidence score" for your assessment of this submission to indicate how confident you are in your evaluation. *

Choices:
5: You are absolutely certain about your assessment. You are very familiar with the related work and checked the math/other details carefully.
4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.
3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
2: You are willing to defend your assessment, but it is quite likely that you did not understand the central parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
1: Your assessment is an educated guess. The submission is not in your area or the submission was difficult to understand. Math/other details were not carefully checked.

1	2	3	4	5
☆	☆	☆	☆	☆

18. "Decision": A decision that has to be one of the following: Accept, Reject. *

Mark only one oval.

○ Accept
○ Reject

19. **[Optional] Any additional feedback?**

____

E.4 Co-Pilot Grading Research Report Custom Topics

Co-Pilot Grading Research Report

The goal of this assignment is to use Agent Laboratory as a research Co-Pilot and to determine how useful it was for implementing your research project.

Please follow the build instructions provided for you in Agent Laboratory project directory. Please then run the Agent Laboratory file and as text provide a research topic that you are interested in and would like the system to explore for you. This can be anything machine learning related.

At the end of the simulation, you will then be provided with a report (as a PDF) and you should read this report in its entirety provide quality ratings across various measures. You should rate everything through the following lens of perception:

- Given the question that you provided, the AI assistant tool performed its own literature search, experimentation, performed its own coding, executed the code, collected data, conducted an analysis, and wrote the presented research report. The goal of this assistant is not to perform research for you (automate you task) but instead to provide a foundation for you to accelerate your own research.

You should be asking: is what this AI assistant produced **useful for me to build off of** instead comparing it to what a human by themselves would perform.

Once you have read the paper please answer the question below:

* Indicates required question

1. What was the research topic you chose (please provide EXACT question)? *

2. **How easy was it for you to build a project using Agent Laboratory?** *

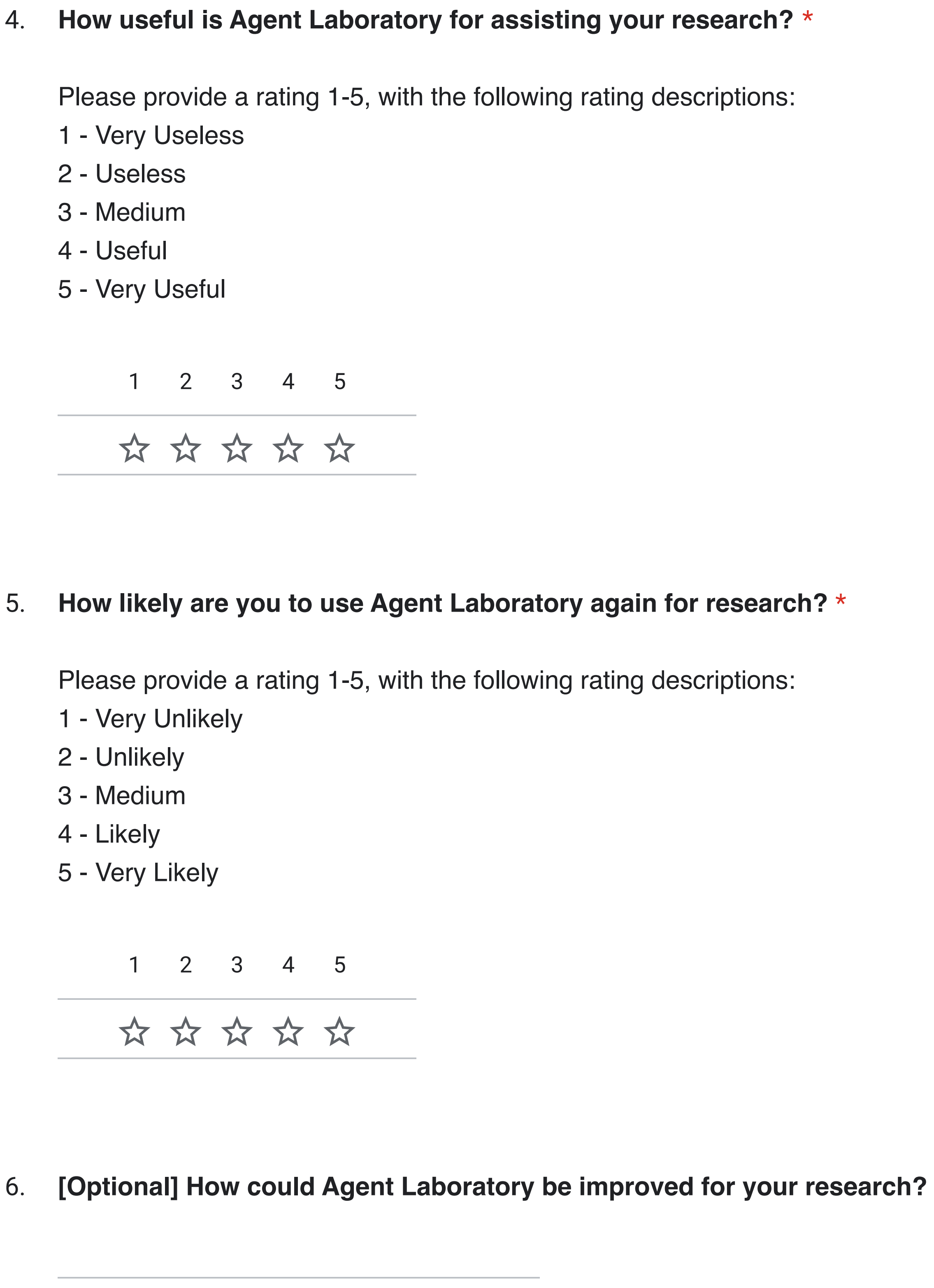
Please provide a rating 1-5, with the following rating descriptions:
1 - Very Hard
2 - Hard
3 - Medium
4 - Easy
5 - Very Easy

1	2	3	4	5
☆	☆	☆	☆	☆

3. **How much did you enjoy using Agent Laboratory?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Unenjoyable
2 - Unenjoyable
3 - Neutral
4 - Enjoyable
5 - Very Enjoyable

1	2	3	4	5
☆	☆	☆	☆	☆



4. **How useful is Agent Laboratory for assisting your research?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Useless
2 - Useless
3 - Medium
4 - Useful
5 - Very Useful

1	2	3	4	5
☆	☆	☆	☆	☆

5. **How likely are you to use Agent Laboratory again for research?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Unlikely
2 - Unlikely
3 - Medium
4 - Likely
5 - Very Likely

1	2	3	4	5
☆	☆	☆	☆	☆

6. **[Optional] How could Agent Laboratory be improved for your research?**

____

7. **What is your perception of the quality of the <u>experimental results</u> presented in this report?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Low Quality
2 - Low Quality
3 - Medium Quality
4 - High Quality
5 - Very High Quality

1	2	3	4	5
☆	☆	☆	☆	☆

8. **What is your perception of the quality of the <u>research report writing quality</u> presented in this report?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Low Quality
2 - Low Quality
3 - Medium Quality
4 - High Quality
5 - Very High Quality

1	2	3	4	5
☆	☆	☆	☆	☆

9. **What is your perception of the <u>usefulness of the AI assistant tool</u> presented in this report?** *

Please provide a rating 1-5, with the following rating descriptions:
1 - Very Low Usefulness
2 - Low Usefulness
3 - Medium Usefulness
4 - High Usefulness
5 - Very High Usefulness

1	2	3	4	5
☆	☆	☆	☆	☆

Review

<u>Now assume you are a reviewer at NeurIPS 2025 and are reviewing a machine learning paper. Please provide the following ratings from this perspective.</u>

10. Quality: Is the submission technically sound? Are claims well supported (e.g., by theoretical analysis or experimental results)? Are the methods used appropriate? Is this a complete piece of work or work in progress? Are the authors careful and honest about evaluating both the strengths and weaknesses of their work *

1 - Low Quality
2 - Medium Quality
3 - High Quality
4 - Very High Quality

1	2	3	4
☆	☆	☆	☆

11. Clarity: Is the submission clearly written? Is it well organized? (If not, please make constructive suggestions for improving its clarity.) Does it adequately inform the reader? (Note that a superbly written paper provides enough information for an expert reader to reproduce its results.) *

Please provide a rating 1-4, with the following rating descriptions:
1 - Low Clarity
2 - Medium Clarity
3 - High Clarity
4 - Very High Clarity

1	2	3	4
☆	☆	☆	☆

12. Significance: Are the results important? Are others (researchers or practitioners) likely to use the ideas or build on them? Does the submission address a difficult task in a better way than previous work? Does it advance the state of the art in a demonstrable way? Does it provide unique data, unique conclusions about existing data, or a unique theoretical or experimental approach? *

Please provide a rating 1-4, with the following rating descriptions:
1 - Low Significance
2 - Medium Significance
3 - High Significance
4 - Very High Significance

1	2	3	4
☆	☆	☆	☆

13. Soundness: Please assign the paper a numerical rating on the following scale to indicate the soundness of the technical claims, experimental and research methodology and on whether the central claims of the paper are adequately supported with evidence. *
4: excellent
3: good
2: fair
1: poor

1	2	3	4
☆	☆	☆	☆

14. Presentation: Please assign the paper a numerical rating on the following scale to indicate the quality of the presentation. This should take into account the writing style and clarity, as well as contextualization relative to prior work. *
4: excellent
3: good
2: fair
1: poor

1	2	3	4
☆	☆	☆	☆

15. Contribution: Please assign the paper a numerical rating on the following scale to indicate the quality of the overall contribution this paper makes to the research area being studied. Are the questions being asked important? Does the paper bring a significant originality of ideas and/or execution? Are the results valuable to share with the broader NeurIPS community. *

4: excellent
3: good
2: fair
1: poor

1	2	3	4
☆	☆	☆	☆

16. Overall: Please provide an "overall score" for this submission. Choices: *

10: Award quality
9: Very Strong Accept
8: Strong Accept
7: Accept
6: Weak Accept
5: Borderline accept
4: Borderline reject
3: Reject
2: Strong Reject
1: Very Strong Reject

1	2	3	4	5	6	7	8	9	10
☆	☆	☆	☆	☆	☆	☆	☆	☆	☆

17. Confidence: Please provide a "confidence score" for your assessment of this submission to indicate how confident you are in your evaluation. *

Choices:
5: You are absolutely certain about your assessment. You are very familiar with the related work and checked the math/other details carefully.
4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.
3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
2: You are willing to defend your assessment, but it is quite likely that you did not understand the central parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
1: Your assessment is an educated guess. The submission is not in your area or the submission was difficult to understand. Math/other details were not carefully checked.

1	2	3	4	5
☆	☆	☆	☆	☆

18. "Decision": A decision that has to be one of the following: Accept, Reject. *

Mark only one oval.

○ Accept
○ Reject

19. **[Optional] Any additional feedback?**

____