MLR-COPILOT: Autonomous Machine Learning Research based on Large Language Model Agents

Ruochen Li[1], **Teerth Patel**[1], **Qingyun Wang**[2], **Xinya Du**[1]
[1]University of Texas at Dallas [2]UIUC
{ruochen.li, teerth.patel, xinya.du}@utdallas.edu
qingyun4@illinois.edu

Abstract

Autonomous machine learning research has gained significant attention recently. We present MLR-COPILOT, an autonomous Machine Learning Research framework powered by large language model agents. The system is designed to enhance ML research productivity through automatic generation and implementation of research ideas within constraints. Our work was released in August 2024 (concurrent to AI-SCIENTIST) and has gained notable recognition from leading projects. We further enhance our ideation with training afterwards. The framework consists of three stages: idea generation, experiment implementation, and code execution. First, existing research papers are used to generate feasible ideas and experiment plans with IdeaAgent powered by an RL-tuned LLM. Next, ExperimentAgent leverages retrieved prototype code to convert plans into executable code with optionally retrieved candidate models and data from HuggingFace. In the final stage, ExperimentAgent runs experiments, and allows subsequent iterations of debugging and human feedback for a better chance of success with executable outcomes. We evaluate our framework on five machine learning research tasks. Experiment results demonstrate the potential of our framework to facilitate ML research progress and innovation.[1] [2] [3]

1 Introduction

The increasing complexity of modern scientific research and the rapid expansion of scientific knowledge pose significant challenges for researchers (Choudhury, 2021). Traditional research workflow typically follows a structured process:

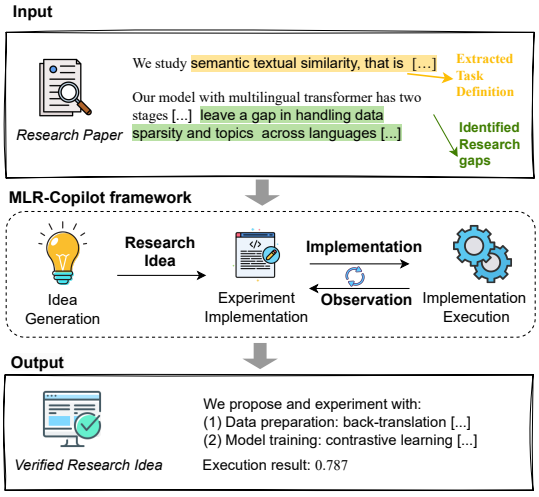

Figure 1: The autonomous machine learning research task. We take the research paper as input and output the research idea (i.e. research methodology and experiment plan) with execution results.

researchers begin with a literature review to identify existing knowledge and gaps, then proceed to method formulation and experimental design. Afterward, they move to the implementation and execution phases to obtain results. While effective, these steps are highly labor-intensive and time-consuming (Powell, 2015), especially given the accelerated pace of advancements that shorten research cycles, which leads to tighter time constraints and a higher risk of errors in decision-making, potentially hindering progress (Bornmann et al., 2010).

Recent advancements in Large Language Models (LLMs) and agents offer a promising opportunity to augment and accelerate this traditional workflow. With their impressive ability to generate text, code, and hypotheses (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023; Park et al., 2023; Zhou et al., 2023), LLMs can act as a "copilot" for researchers (Figure 1), supporting every phase of the research process to enable higher efficiency and productivity (Dakhel et al., 2023; GitHub, Inc.).

[1]Code package, data, models, and demonstration can be found at: https://github.com/du-nlp-lab/MLR-Copilot.

[2]Our software demonstration video is at: https://youtu.be/y_yBKUtvln8.

[3]Our examples with Graphical User Interface can be found at: https://huggingface.co/spaces/du-lab/MLR-Copilot.

There have been some recent efforts to explore LLMs for scientific discovery, but these are often limited in scope. For example, Yang et al. (2023); Wang et al. (2024); Qi et al. (2023a); Baek et al. (2024) focus only on idea generation from general scientific literature (Stage 1 in our process), but they are not tailored to Machine Learning Research (MLR) and fail to address the limitations of prior work for specific problems, which results in solutions that are too broad for targeted applications. In addition, their reliance on pertaining through prompting limits their creativity. Other studies, such as Huang et al. (2023); Zhang et al. (2023) target auto-experimenting for ML tasks (Stages 2 and 3). However, their settings are much more constricted as they start with a predefined task and mature code template instead of research literature as in real-world scenarios. Also, they typically apply small code edits like hyperparameter tuning with limited exploration of novel models, structures, or data. Moreover, they lack robust feedback mechanisms, leaving no guarantee that the experiment will converge with no address of the root cause of errors in code.

Different from all the above (and concurrent to AI Scientist (Lu et al., 2024)), we aim to build a fully autonomous framework to tackle the entire process of machine learning research. We present MLR-COPILOT (Figure 2), a systematic framework designed to enhance productivity through automatic generation and implementation/verification with LLM agents. It takes the paper as input and operates in three integrated phases: research idea generation, experiment implementation, and implementation execution. In this first stage, we construct input prompts that incorporate relevant research papers and extracted research problems (including task definition); these prompts are then processed by IdeaAgent, a fine-tuned LLM agent, to generate research methodologies and experimental plans. The structured generation ensures that the proposed ideas are well-grounded in the existing literature and address current gaps (Zhang and Teng, 2023; Cohan and Goharian, 2018; Baek et al., 2024), while the fine-tuning tailors them specifically to the needs of MLR. In the second stage, the framework translates these experimental plans into executable experiments. It is facilitated by ExperimentAgent, which incorporates the utility of model and data retrieval, and leverages retrieved prototype code (from relevant papers) to generate the necessary implementations (Smith et al., 2023; Hocky and White, 2022; Viswanathan et al., 2023). Later, ExperimentAgent leverages feedback from the execution results from Stage 3. Finally, the implementation execution phase, also managed by ExperimentAgent, involves running the experiments and generating execution/debugging feedback, as well as optional human feedback. The feedback allows for the refinement of the experiment implementations (Stage 2). The implementation and execution process is iterative, and the human-in-the-loop feature ensures that the final research outcomes are robust, reproducible, and scientifically sound (Viswanathan et al., 2023).

We conduct manual and automatic evaluations on generated hypotheses and experimental executions/results. We also present case studies demonstrating the practical applications of our system on five ML research papers/problems. Through evaluations and examples, we illustrate that our framework can generate novel and feasible hypotheses for research, enabling researchers to focus on high-level scientific inquiry and innovation. We also show that MLR-COPILOT is able to help finish the full research process and obtain significant results/improvements and conclusions.

2 MLR-COPILOT Framework

MLR-COPILOT automates the generation and implementation of research ideas using LLM agents, organized into three integrated phases: research idea generation, experiment implementation, and implementation execution.

2.1 Research Idea Generation

In the first stage, IdeaAgent an LLM-powered agent, generates research methodologies and experimental plans.

The agent is fine-tuned with fine-grained Reinforcement Learning (RL) technique following our training-based method (Li et al., 2024a)[4] on collected feedback data of the top ML topic conference papers[5] collected from OpenReview[6]. The overall rate, novelty, feasibility, and effectiveness scores are collected and used to guide the optimization. In this process, the model first under-

[4]Without steering at inference time.

[5]4,271 papers are collected as of year 2023 and 2024. ICLR: https://iclr.cc/ NeurIP: https://neurips.cc/.

[6]Open Review API :https://docs.openreview.net/reference/api-v2.

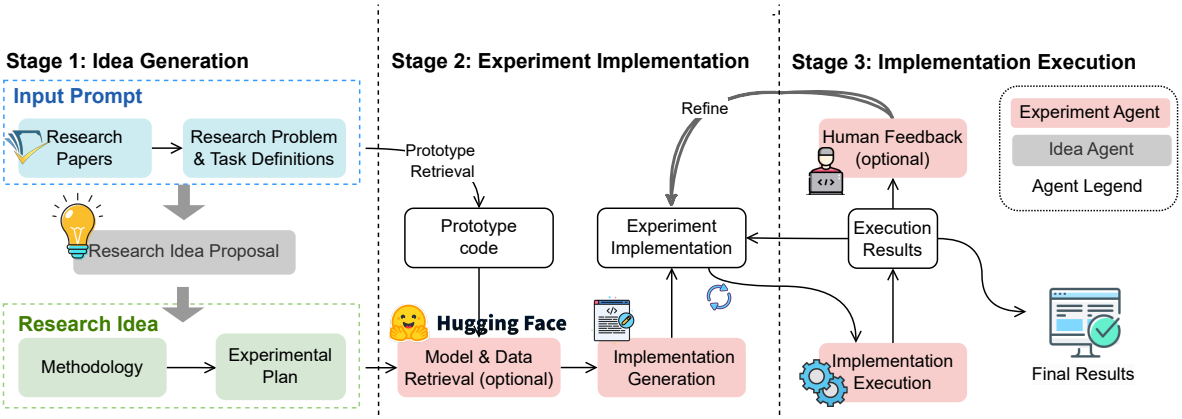

Figure 2: Our MLR-COPILOT Framework. LLM IdeaAgent (leftmost grey component) performs research idea generation including hypothesis and experimental design (Stage 1). ExperimentAgent implements and executes the experiments (Stage 2 and 3).

goes supervised fine-tuning with a derived dataset of 1,000 papers and extracted research ideas and plans, ensuring an initial understanding of the task. Next, the model is refined on the dataset using RL with multi-dimensional feedback from respective reward models for novelty, feasibility, and effectiveness. These reward models evaluate and score ideas, guiding the reinforcement learning process to improve the quality of generated ideas. This alignment helps IdeaAgent to produce ideas that are not only grounded in the literature but also tailored to the unique requirements of machine learning research.

During idea generation, for each task, the process begins with an individual research paper $c = \{c_1, c_2, \ldots, c_n\}$, where c_i represents the selected contents of the paper with *Semantic Scholar API*[7], including the title, abstract, introduction, and related work. The input processing involves analyzing the literature to extract essential information. Specifically, the initial input prompt is used to extract research tasks t, research gaps g, and keywords $k = \{k_1, k_2, \ldots, k_m\}$ with LLM. Then $\mathcal{P} = \{c, t, g, k\}$ are provided to retrieve a set of recent works in the literature, denoted as $\mathcal{R} = \{r_1, r_2, \ldots, r_l\}$. IdeaAgent extracts and synthesizes relevant information from the literature. Using updated information, the LLM generates a new methodology with a prompt detailed as $\mathcal{P}_1 = \{\mathcal{P}, \mathcal{R}\} \rightarrow h$ based on identified trends and gaps in the existing research, ensuring both relevance and grounding in current studies. This initial methodology set $\mathcal{P}_1$ is then appended to create a detailed experimental plan $\mathcal{P}_2 = \{\mathcal{P}_1, h\} \rightarrow e$. The experiment plan outlines the methodology, expected outcomes, and potential challenges associated with testing the methodology.

Finally, we represent a research idea as:

$$RI = \{\mathcal{P}, \mathcal{R}, h, e\}$$

where: $\mathcal{P}$ denotes the information from original paper, $\mathcal{R}$ denotes the recent research findings, h represents the generated methodology, e outlines the experiment plan.

2.2 Experiment Implementation

The second phase involves translating experimental plans into executable experiments. This phase is facilitated by ExperimentAgent an LLM-based agent. Given research idea RI that contains experiment plan e, ExperimentAgent performs several critical actions:

First, it retrieves prototype implementation I from the original paper. Leveraging existing I, ExperimentAgent adapts and integrates this code, and optionally retrieves suitable models $\mathcal{M}_\nabla$ from a model repository $\mathcal{M} = \{M_1, M_2, \ldots, M_p\}$ to fit the specific needs of the experimental plan. The selection process is guided by the requirements of the experimental plan e_j, ensuring that the chosen models are appropriate for the specified tasks. If needed, relevant datasets $\mathcal{D} \in \{D_1, D_2, \ldots, D_q\}$ are identified and retrieved. We ensure that these datasets align with the experimental requirements by post-checkup, facilitating accurate and comprehensive testing of the methodologies (Hocky and White, 2022).

The ExperimentAgent modifies the code to ensure compatibility with the selected models and

[7] https://www.semanticscholar.org/product/api.

datasets (Viswanathan et al., 2023). Finally, the retrieved models, datasets, and prototype code are integrated into a cohesive experimental setup with experimental implementation $(\mathcal{I}, \mathcal{M}_{\nabla}, \mathcal{D}) \rightarrow \mathcal{S}$, ExperimentAgent ensures seamless interaction between these components, preparing the experimental setup for execution.

2.3 Implementation Execution

In the final phase, ExperimentAgent manages the execution of the experiments. The execution phase encompasses running the experiments, incorporating mechanisms for human feedback, and supporting iterative debugging.

The experimental setups $(\mathcal{I}, \mathcal{M}_{\nabla}, \mathcal{D}) \rightarrow \mathcal{S}$ are executed under the management of ExperimentAgent. The agent oversees the allocation of computational resources, monitoring the progress and performance of the experiments. Additionally, ExperimentAgent integrates mechanisms for human feedback, allowing researchers to provide input and adjustments during the execution phase. This feedback loop ensures that the experimental design and implementation can be refined in real time.

From the global point of view, ExperimentAgent provides feedback and enables researchers (or stage 1) to refine their methodologies and experimental designs based on intermediate and final execution results (e.g. feasibility). This iterative approach ensures that the final research outcomes are robust, reproducible, and scientifically sound.

3 Experiments

3.1 Experimental Setup and Datasets

To evaluate the effectiveness of MLR-COPILOT , we conduct experiments across five machine learning research task papers following (Smith et al., 2023). These tasks of the papers were chosen to cover a range of domains and complexities, demonstrating the versatility and robustness of our framework. ***SemRel*** (Ousidhoum et al., 2024) from SemEval 2024 Task 1 focuses on semantic textual relatedness across 13 languages and is popular for its diversity and real-world relevance. We use the supervised track for our experiments and adopt Pearson correlation as the metrics. ***MLAgentBenchmark*** (Huang et al., 2023) includes several datasets for evaluating LLMs in automated research idea generation and implementation. We use the following datasets: ***feedback (ELLIPSE)*** (Franklin et al., 2022; Doe and Smith, 2023) used for machine learning-based feedback prediction, suitable for regression tasks like MCRMSE. ***IMDB*** (Maas et al., 2011) consists of movie reviews labeled by sentiment, commonly used for sentiment analysis and NLP tasks. ***Spaceship-Titanic*** dataset predicts passenger survival based on features like passenger class, age, and ticket fare. ***Identify-Contrails*** involves identifying contrails in satellite images, suitable for image classification tasks. Classification accuracy is used as the metric for these tasks.

3.2 Evaluation and Results

We evaluate different stages of our framework, i.e. the hypothesis generation stage (Section 3.2.1), the experiment implementation and implementation execution stages (Section 3.2.2) separately.

3.2.1 Research Idea Generation

We conduct both manual evaluations and automated evaluations. For baselines, we adopt *BaseLLM* which prompts with only a core paper to generate research ideas, and *ResearchAgent* (Baek et al., 2024) with our implementation.

For manual evaluation, we invite five domain expert reviewers to assess 45 generated hypotheses based on the following criteria: clarity, validity, rigor, innovativeness, and generalizability. Additionally, the experimental designs are evaluated for clarity, validity, robustness, feasibility, and reproducibility. Each criterion is scored on a 5-point Likert scale (detailed definitions from Baek et al. (2024)), with human researchers who have published at least three papers providing the annotations. For automated evaluation, we employ GPT-4 as our reviewing agent to assess the clarity and validity of the hypotheses and the robustness and feasibility of the experimental designs, scoring each criterion on a 5-point Likert scale. Similarity analysis is performed to compare the new hypotheses with the original hypotheses from existing papers on a scale from 0 to 1.

Table 1 and 2 present evaluation results comparing IdeaAgent trained with Llama3-7B to

Method	Criteria	BaseLLM	ResearchAgent	IdeaAgent
Manual	Clarity	3.7	4.2	4.4
	Validity	3.8	3.8	3.9
	Rigor	3.5	4.0	4.3
	Innovativeness	3.1	3.8	3.9
	Generalizability	3.6	3.8	4.1
Automated	Clarity	2.9	4.4	4.6
	Validity	3.2	4.2	4.7
	Similarity	0.32	0.15	0.13

Table 1: Evaluation results for generated hypotheses.

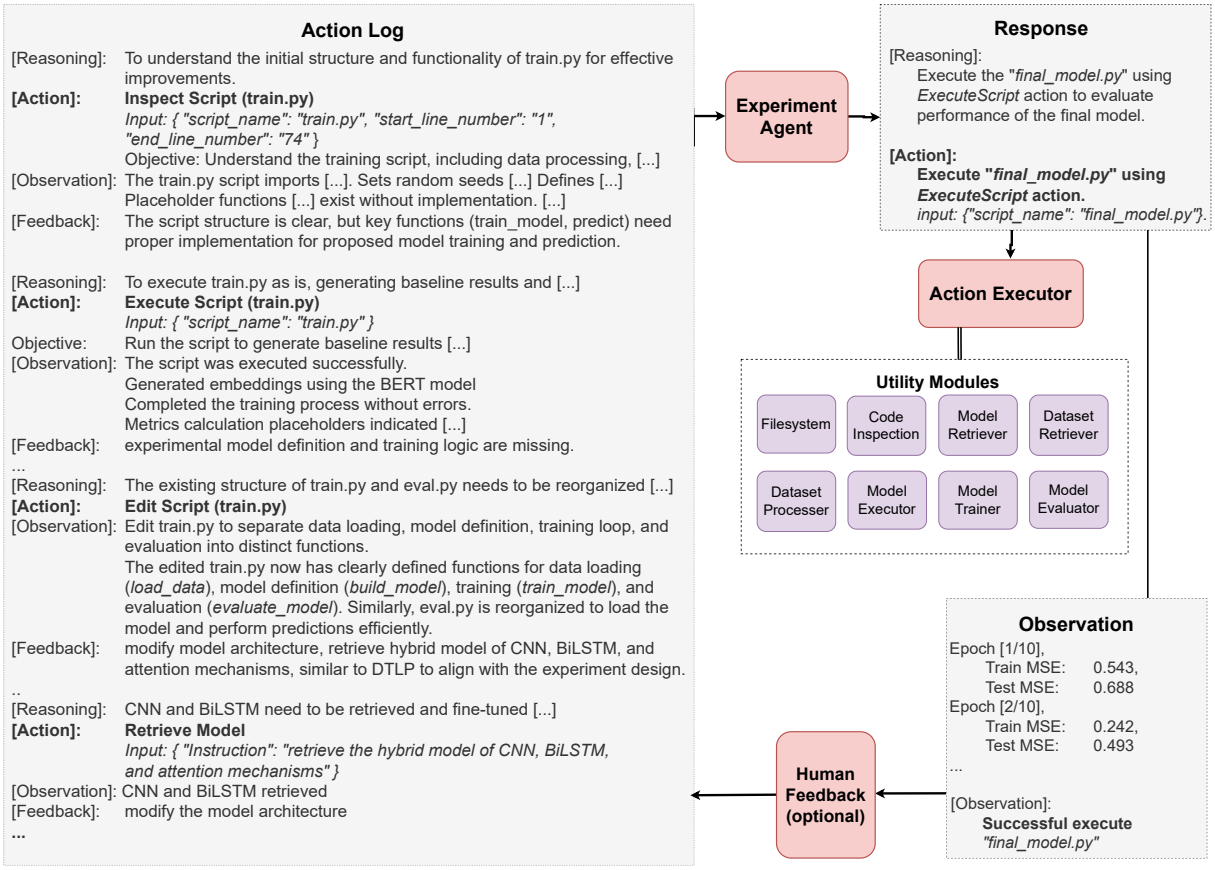

Figure 3: An illustrative case study demonstrating the practical application of MLR-COPILOT for sentiment analysis on the ELLIPSE dataset. The diagram shows the interaction between the ExperimentAgent, Action Executor, and various Utility Modules. The action log details steps taken to inspect, execute, and retrieve models, with observations and feedback guiding iterative improvements in the experimental implementation and model performance.

BaseLLM and ResearchAgent across various criteria for generated hypotheses and experimental design. IdeaAgent consistently outperforms the BaseLLM and better than ResearchAgent in both manual and automated assessments. Furthermore, the similarity scores indicate that IdeaAgent generates hypotheses with lower similarity to existing ones, suggesting more novel contributions.

Method	Criteria	BaseLLM	ResearchAgent	IdeaAgent
Manual	Clarity	3.4	4.1	4.3
	Validity	3.7	3.9	4.2
	Robustness	3.5	3.8	4.1
	Feasibility	3.8	4.0	4.3
	Reproducibility	3.6	3.9	3.9
Automated	Robustness	3.1	3.9	4.4
	Feasibility	3.3	4.0	4.6

Table 2: Evaluation results for experimental design.

3.2.2 Experiment Implementation and Implementation Execution

We evaluate the effectiveness of experiment implementation and execution by measuring the average task performance improvement and success rate across 8 trials aided with human instructions. For each assessment, we utilize the retrieved state-of-the-art prototype code as the starting point for improvement. Our MLR-COPILOT is compared against a *One-Pass Prompting (1-Prompt)* baseline, a single-step method that directly modifies code without iterative feedback.

Task	1Prompt	Ours (Claude-2.1)	Ours (Claude-3.7)	Ours (GPT-4)
SemRel	N/A	14.5	21.5	15.2
imdb	N/A	67.3	76.2	78.5
spaceship-titanic	N/A	48.4	48.4	45.8
feedback (ELLIPSE)	N/A	55.3	60.2	49.2
identify-contrails	N/A	4.6	14.5	10.0
Average	N/A	38.0	**44.16**	39.74

Table 3: Average percentage improvement over the SOTA prototype. N/A indicates execution failure.

Task	1Prompt	Ours (Claude-2.1)	Ours (Claude-3.7)	Ours (GPT-4)
SemRel	0.0	37.5	62.5	50.0
imdb	0.0	12.5	50.0	50.0
spaceship-titanic	0.0	75.0	75.0	62.5
feedback (ELLIPSE)	0.0	12.5	50.0	25.0
identify-contrails	0.0	0.0	12.5	12.5
Average	0.0	27.5	**50.0**	40.0

Table 4: Success rate over 8 trials, where success is defined as achieving at least 10% improvement over the SOTA prototype.

Table 3 and 4 demonstrate that both GPT-4 and Claude outperform the 1-Prompt baseline. 1-Prompt consistently fails across all trails in improvement towards the generated ideas due to its inability to detect and correct environmental and execution errors. This becomes particularly prominent when handling novel or complex research ideas, leading to persistent runtime failures and a complete lack of measurable success. Notably, Our method with Claude-3.7 achieves the highest average improvement, with a success rate of 50.0% compared to GPT-4 with 40% and Claude-2.1 with 27.5%, highlighting its superior effectiveness.

4 Case Study for Sentiment Analysis

To demonstrate the practical application of our framework, we conducted a case study where researchers used the system to generate hypotheses and conduct sentiment analysis experiments on ELLIPSE dataset. As shown in Figure 3, the process involves interaction between the ExperimentAgent, Action Executor, and various Utility Modules. The action sequences illustrate how the MLR-COPILOT system helps researchers systematically generate hypotheses and conduct experiments. The system inspects scripts, executes models, retrieves models, and analyzes results. Details are provided in the appendix. This comprehensive action log highlights the systematic approach of MLR-COPILOT , allowing researchers to understand, modify, and execute scripts for sentiment analysis. Each action, driven by reasoning, objectives, observations, and feedback, refines the model and experimental design, leading to the success of evaluation.

5 Related Work

LLM as Scientific Agents. The automation of idea generation in scientific research received great interest with the advent of LLMs. Prior works have explored their potential for research question and idea generation based on literature-based discovery (Zhong et al., 2023; Qi et al., 2023b; Yang et al., 2023; Wang et al., 2024; Li et al., 2024b). Others apply LLM agents for AutoML tasks (ScienceDirect, 2023; Zhang et al., 2023). MLAgentBench (Huang et al., 2023) is proposed to benchmark LLM performance on diverse ML tasks and datasets. Their scope is limited to predefined tasks and existing codebases without interaction. Our work supports automatic ML hypothesis generation with broader utilities and a more expressive action space.

Concurrent to our work, AI SCIENTIST (Lu et al., 2024) introduced comprehensive autonomous research of very similar stage design with further scope of paper writing and reviewing. AI-SCIENTIST V2(Yamada et al., 2025) replaces manual templates with tree-search. Subsequent works like AGENTLAB (Schmidgall et al., 2025) and AI-RESEARCHER (Tang et al., 2025) adopt similar staged designs with enhanced role specialization and coordination. Other systems, including AI CO-SCIENTIST (Gottweis et al., 2025) and RESEARCHTOWN (Yu et al., 2024), explore multi-agent collaboration to facilitate novel idea discovery. Several follow-up efforts incorporate human-in-the-loop paradigms. CODESCIENTIST (Jansen et al., 2025) supports collaborative code-based experimentation between LLMs and humans. Ifargan et al. (2025) and DEEPREVIEW (Zhu et al., 2025) integrate expert feedback to refine and align LLM-generated drafts with domain expertise.

Model and Data Retrieval Systems. Efficient models and data retrieval are critical components of modern AI systems. Hugging Face's Datasets and Model Hub provide researchers with vast repositories of datasets and pre-trained models (Lhoest et al., 2021; Wolf et al., 2020). These systems enable users to find relevant data and models quickly through natural language prompts, facilitating seamless integration into the research workflow. Our framework incorporates the model and data retrieval utilities, which play a crucial role in the experiment implementation process based on natural language prompts (Viswanathan et al., 2023). This allows for translating research questions and problem statements into specific model requirements, facilitating the automated retrieval of the most relevant models for hypothesis testing and validation.

6 Conclusion

We introduce MLR-COPILOT, a framework for automating machine learning research using LLM agents. It helps generate novel research ideas, implements and executes the experiments, and refines the implementations based on both automatic and human feedback. Evaluations from domain experts highlight it as a powerful tool for research idea generation and the experimentation process.

7 Limitation

While MLR-COPILOT demonstrates promising results in automating machine learning research, several limitations remain. First, the current pipeline treats the stages, especially idea generation and experiment implementation and execution largely as sequential. In practice, however, failed or suboptimal experiments often indicate the need to revisit and revise the original hypotheses. A tighter integration between stages, particularly enabling backward transitions from implementation or execution back to ideation, would better reflect the iterative nature of real-world scientific discovery. Second, although our framework introduces a novel paradigm and a usable end-to-end system, there remains room for improving its usability and accessibility, especially for researchers without extensive experience in LLM prompting or code debugging. Enhancing the user interface and providing more intuitive interaction mechanisms would help broaden adoption among a wider range of ML practitioners.

References

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*.

Lutz Bornmann, Ruediger Mutz, and Hans-Dieter Daniel. 2010. The growth of scientific knowledge: a bibliometric perspective on the expansion and acceleration of scientific output. *Journal of the American Society for Information Science and Technology*, 61(12):2155–2160.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Shyam Choudhury. 2021. Emerging technologies in scientific research: Opportunities and challenges. *Journal of Scientific Research and Development*, 12(3):45–56.

Arman Cohan and Nazli Goharian. 2018. Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, 19(2–3):287–303.

Arghavan Moradi Dakhel, Vahid Majdinasab, Amin Nikanjam, Foutse Khomh, Michel C Desmarais, and Zhen Ming Jack Jiang. 2023. Github copilot ai pair programmer: Asset or liability? *Journal of Systems and Software*, 203:111734.

John Doe and Jane Smith. 2023. Ellipse: A dataset for feedback prediction in machine learning. *Journal of Machine Learning Research*, 24:1–10.

A. Franklin, M. Benner, N. Rambis, P. Baffour, R. Holbrook, S. Crossley, and ulrichboser. 2022. Feedback prize - english language learning.

GitHub, Inc. Github copilot. https://github.com/features/copilot. Accessed: 2024-08-05.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. 2025. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*.

Glen M. Hocky and Andrew D. White. 2022. Natural language processing models that automate programming will transform chemistry research and teaching. *Digital Discovery*, 1:79–83.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2023. Mlagentbench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*.

Tal Ifargan, Lukas Hafner, Maor Kern, Ori Alcalay, and Roy Kishony. 2025. Autonomous llm-driven research—from data to human-verifiable research papers. *NEJM AI*, 2(1):AIoa2400555.

Peter Jansen, Oyvind Tafjord, Marissa Radensky, Pao Siangliulue, Tom Hope, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Daniel S Weld, and Peter Clark. 2025. Codescientist: End-to-end semi-automated scientific discovery with code-based experimentation. *arXiv preprint arXiv:2503.22708*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ruochen Li, Liqiang Jing, Chi Han, Jiawei Zhou, and Xinya Du. 2024a. Learning to generate research idea with dynamic control.

Ruochen Li, Liqiang Jing, Chi Han, Jiawei Zhou, and Xinya Du. 2024b. Learning to generate research idea with dynamic control. *arXiv preprint arXiv:2412.14626*.

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

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024. Semrel2024: A collection of semantic textual relatedness datasets for 13 languages. *arXiv preprint arXiv:2402.08638*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Kenny Powell. 2015. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications.

Biqing Qi, Kaiyan Zhang, Haoxiang Li, Kai Tian, Sihang Zeng, Zhang-Ren Chen, and Bowen Zhou. 2023a. Large language models are zero shot hypothesis proposers.

Biqing Qi, Kaiyan Zhang, Haoxiang Li, Kai Tian, Sihang Zeng, Zhang-Ren Chen, and Bowen Zhou. 2023b. Large language models are zero shot hypothesis proposers.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*.

ScienceDirect. 2023. Automl: A systematic review on automated machine learning. *ScienceDirect*.

John Smith, Jane Doe, and Wei Zhang. 2023. Mlagentbench2023: A framework for automating research idea generation and implementation using llm agents. *Journal of Computational Research*, 45(3):123–145.

Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. 2025. Ai-researcher: Autonomous scientific innovation. *arXiv preprint arXiv:2505.18705*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vijay Viswanathan, Chenyang Zhao, Amanda Bertsch, Tongshuang Wu, and Graham Neubig. 2023. Prompt2model: Generating deployable models from natural language instructions. *CoRR*, abs/2308.12261.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2024. Scimon: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL2024)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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*.

Zonglin Yang, Xinya Du, Junxian Li, Jie Zheng, Soujanya Poria, and Erik Cambria. 2023. Large language models for automated open-domain scientific hypotheses discovery. *arXiv preprint arXiv:2309.02726*.

Haofei Yu, Zhaochen Hong, Zirui Cheng, Kunlun Zhu, Keyang Xuan, Jinwei Yao, Tao Feng, and Jiaxuan You. 2024. Researchtown: Simulator of human research community. *arXiv preprint arXiv:2412.17767*.

Shujian Zhang, Chengyue Gong, Lemeng Wu, Xingchao Liu, and Mi Zhou. 2023. Automl-gpt: Automatic machine learning with gpt. *ArXiv*, abs/2305.02499.

Yue Zhang and Zhiyang Teng. 2023. *Natural Language Processing: A Machine Learning Perspective*. MIT Press.

Ruiqi Zhong, Peter Zhang, Steve Li, Jinwoo Ahn, Dan Klein, and Jacob Steinhardt. 2023. Goal driven discovery of distributional differences via language descriptions. *Advances in Neural Information Processing Systems*, 36:40204–40237.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. 2025. Deepreview: Improving llm-based paper review with human-like deep thinking process. *arXiv preprint arXiv:2503.08569*.

A Details of Case Study: Sentiment Analysis Research

For the purpose of brevity, the running logs are summarized. Full logs are available in git repository `https://github.com/du-nlp-lab/MLR-Copilot/blob/main/full.log`.

Hypothesis and Experiment Generation Prompts
Detailed prompts guiding the formulation of innovative scientific methods and robust experiments, emphasizing clarity, innovation, validity, and reproducibility based on provided research problems, existing studies, and relevant entities.

Generated Research Idea
Method: Advanced Aspect-Level Sentiment Analysis with Hybrid Deep Learning

- *Dataset Enhancement:* Additional data collection, preprocessing, and annotation refinement. - *Hybrid Model:* Integration of CNN, BiLSTM, and Transformers (BERT) with enhanced linguistic features. - *Training and Validation:* Cross-validation and comparison to baseline methods. - *Iterative Refinement:* Expert feedback and active learning for continuous improvement. - *Deployment:* Real-time integration with educational systems and user-friendly interfaces.

Model Training and Evaluation Scripts
Training Script (`train.py`): - Loads and preprocesses data, splits into training and validation sets. - Implements custom dataset and regression model based on BERT. - Conducts training loop with validation and outputs metrics.

Evaluation Script (`eval.py`): - Loads trained model parameters. - Evaluates predictions on test data, calculates metrics, and generates submissions.

Both scripts are structured to streamline training, evaluation, and deployment processes effectively.

B IdeaAgent Training

B.1 Dataset Analysis

Figures 4 and 5 provide an overview of the IdeaAgent training data distribution and top 10 keywords.

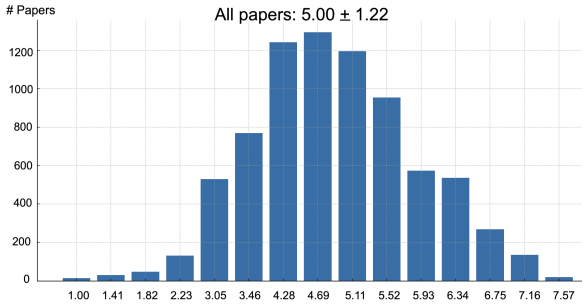

Figure 4: Rating distribution statistics of our dataset.

B.2 Definition of Novelty, Feasibility, and Effectiveness

This appendix provides detailed definitions and scoring guidelines for **Novelty**, **Feasibility**, and **Effectiveness**—the three primary dimensions used to evaluate research ideas and used in RL to train the IdeaAgent.

B.2.1 Novelty

Novelty evaluates how different a proposed research idea is compared to existing works. Following previous work, the guidelines for scoring are as follows:

- **1**: *Not novel at all* — The idea is identical to many existing works.

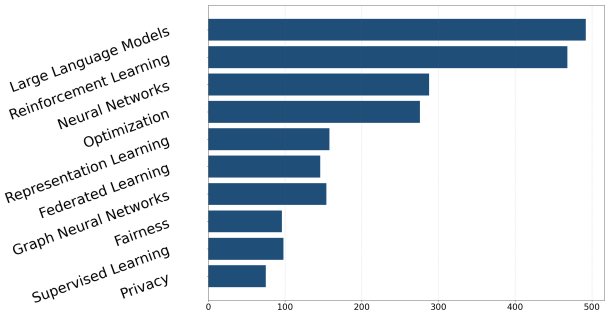

Figure 5: Top 10 topic distribution of our dataset.

- **3**: *Mostly not novel* — Very similar ideas already exist.
- **5**: *Somewhat novel* — There are differences, but not enough for a standalone paper.
- **6**: *Reasonably novel* — Notable differences, potentially sufficient for a new paper.
- **8**: *Clearly novel* — Major differences from all existing ideas.
- **10**: *Highly novel* — Highly different and creative in a clever, impactful way.

B.2.2 Feasibility

Feasibility measures how practical it is to execute the proposed idea within 1–2 months under the following assumptions:

- Ample access to OpenAI/Anthropic APIs.
- Limited GPU computing resources.

Scoring guidelines:

- **1**: *Impossible* — The idea or experiments are fundamentally flawed.
- **3**: *Very challenging* — Major flaws or significant resource limitations.
- **5**: *Moderately feasible* — Possible with careful planning and modifications.
- **6**: *Feasible* — Achievable with reasonable planning.
- **8**: *Highly feasible* — Straightforward to implement and run.
- **10**: *Easy* — Quick to implement without requiring advanced skills.

B.2.3 Effectiveness

Effectiveness assesses the likelihood of the research idea achieving meaningful experimental performance improvement. The scoring is defined as:

- **1**: *Extremely unlikely* — Significant flaws, almost certain to fail.
- **3**: *Low effectiveness* — Limited potential, might work in very specific scenarios.
- **5**: *Somewhat ineffective* — A slight chance of marginal or inconsistent improvement.
- **6**: *Somewhat effective* — A decent chance of moderate improvement on certain benchmarks.
- **8**: *Probably effective* — Likely to deliver significant improvement on benchmarks.
- **10**: *Definitely effective* — Highly likely to outperform existing benchmarks by a substantial margin.