Can AI Research Assistants Solve Research Problems? Explore-and-Evolve for Long-Horizon Planning in Scientific Research

ANONYMOUS AUTHOR(S)

Scientific research is a complex process that drives innovation and knowledge advancement. Automating research tasks can reduce researchers' workload and accelerate discovery, but long-horizon planning and execution introduce challenges such as error accumulation and misalignment with human intent. We present Explore-and-Evolve, a framework that organizes AI research assistants to collaborate with human researchers throughout the research process. In this framework, imitation agents retrieve and emulate reference plans from external sources based on the human researcher's initial idea. Evolution agents use various tools to gather new observations and iteratively refine the plan during execution, guided by human feedback and approval. By integrating imitation and evolution, the AI assistant team automates research tasks while staying aligned with human intent, leading to faithful and desirable outcomes. Experiments show that our framework improves the quality of generated research papers, as measured by review ratings.

CCS Concepts: • **Computing methodologies → Multi-agent planning**; **Natural language generation**.

Additional Key Words and Phrases: Imitation-and-Evolve, Long-Horizon Planning, Research Paper Generation

ACM Reference Format:
Anonymous Author(s). 2018. Can AI Research Assistants Solve Research Problems? Explore-and-Evolve for Long-Horizon Planning in Scientific Research. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Scientific research plays an important role in driving innovation, advancing knowledge, solving problems, expanding our understanding of the world, and ultimately improving people's lives in tangible ways [26]. However, the research process is labor-intensive, requiring researchers to read and synthesize overwhelming amounts of knowledge from a vast and rapidly growing scientific literature in order to formulate research ideas and design corresponding experiments [12]. As a result, many promising ideas are often abandoned or overlooked due to the substantial effort needed to identify and validate them.

Automating scientific discovery has been a long-standing ambition in the research community, with early explorations tracing back to the 1970s and 1980s [16]. The recent emergence of large language models (LLMs) [1] has driven remarkable progress in processing, organizing, and generating scientific text [26]. Increasing efforts now leverage commercial LLMs to develop research agents that can propose novel research ideas [2, 28, 33], assist in designing and conducting experiments [5], or function as AI scientists capable of generating complete, open-ended scientific publications automatically [17, 31]. State-of-the-art systems further permit closing the loop by automating the entire research and review cycle, including literature review, manuscript drafting, peer review, and iterative paper refinement [30].

Recent progress has pushed the boundary to the point where a workshop paper written entirely by AI has been accepted through peer review [31]. However, as highlighted by human evaluations of generated manuscripts [31], current systems still struggle to produce drafts that are technically sound, well-organized, and clearly written, especially when addressing novel research problems or methods not well represented in existing literature. One of the most challenging bottlenecks leading to these problems is error accumulation in long-horizon planning. Drafting a research paper is inherently a long-horizon task, involving multiple stages of reasoning from literature review and manuscript preparation to peer review and paper refinement. Prior studies on multi-step models have shown that such long-horizon planning inevitably introduces the risk of compounding errors: small mistakes made in early steps can accumulate and propagate through subsequent planning stages, as errors at one stage directly impact downstream decisions [10, 11, 24, 25].

As a demonstration, we employ state-of-the-art research assistants, OpenAI's Deep Research [18] and Gemini Deep Research, to draft a paper on Diffusion models for data augmentation in EEG (electroencephalogram)-based emotion recognition. We observe that similar issues exist in the outputs of both Deep Research systems. For example, the planned survey section lacks relevant literature in the retrieved results, resulting in missing citations and hallucinated related work. The proposed method is constructed on top of these hallucinated references, lacking a solid technical foundation. Furthermore, the experimental setting employs inappropriate datasets, which further undermines the validity of the research plan.

Based on these observations, a question arises: How can we mitigate error accumulation in long-horizon planning for scientific research generation? We find that if a plan cannot effectively incorporate new observations, there is a persistent risk of misalignment between the plan and the actual observations. It is hard to anticipate potential observations in advance if possible. Even expert researchers rarely plan and execute an entire research project without making revisions. Instead, they continuously track new results and iteratively realign their ideas with emerging evidence throughout the research process.

In this paper, we propose an Imitation-and-Evolve Multi-agent framework to address the challenge of error accumulation in long-horizon planning for scientific research generation. The contributions of our framework are threefold. (1) We are the first to introduce a multi-agent framework to mitigate error accumulation in long-horizon planning for scientific research generation. The proposed framework employs imitation agents to retrieve reference plans from external sources and evolve agents to utilize tools and derive observations. Based on these observations, the plan is continuously updated during execution. (2) We integrate the imitation-and-evolve paradigm into research lifecycle and propose the first self-improving scientific research generation system. In the research stage, the imitation agent emulates related work to plan the manuscript outline, while the evolve agent iteratively drafts and refines this outline. In the revision, the imitation agent simulates human review to generate a revision plan, and the evolve agent iteratively revises the manuscript and decides whether to adopt each suggestion. (3) We propose a new human–computer interaction paradigm that gives users control over the planning process in scientific research generation, enabling them to steer research outputs according to their preferences. Experiments demonstrate that our approach achieves higher reviewer scores on a public benchmark dataset [30] and a self-constructed dataset.

2 Related Work

2.1 Large Language Models for Research

In recent years, several studies have explored the use of language models for creative tasks in research, such as multi-agent collaborative writing [2] and multi-module retrieval [33] to improve research idea generation. These works aim to enhance the novelty and diversity of artificial intelligence in creative research tasks. Si et al. [21] conducted a comprehensive human evaluation of idea generation by language models. Wang et al. [29] proposed a method that uses large language models to automatically write survey papers. Additionally, large language models have been applied to automate the research process. For example, Huang et al. [13] introduced a benchmark for evaluating large language models in coding solutions for machine learning problems. Wang et al. [28] proposed an approach that leverages large language models for scientific literature retrieval. The AI Scientist project [17] introduced a fully automated, prompt-driven research pipeline. Subsequently, the AI Scientist-v2 project [31] introduced a workshop-level automated scientific discovery method based on agentic tree search. More recently, Weng et al. [30] developed an iterative self-rewarding framework that enables large language models to continuously refine their ideas, thereby enhancing both the diversity and practicality of research proposal generation. Guo et al. [9] proposed a benchmark for evaluating large language models in research idea generation. Pu et al. [20] proposed a method for iterative research idea development by evolving and composing idea facets with literature-grounded feedback. Garikaparthi et al. [8] proposed an interactive research ideation system to accelerate scientific discovery.

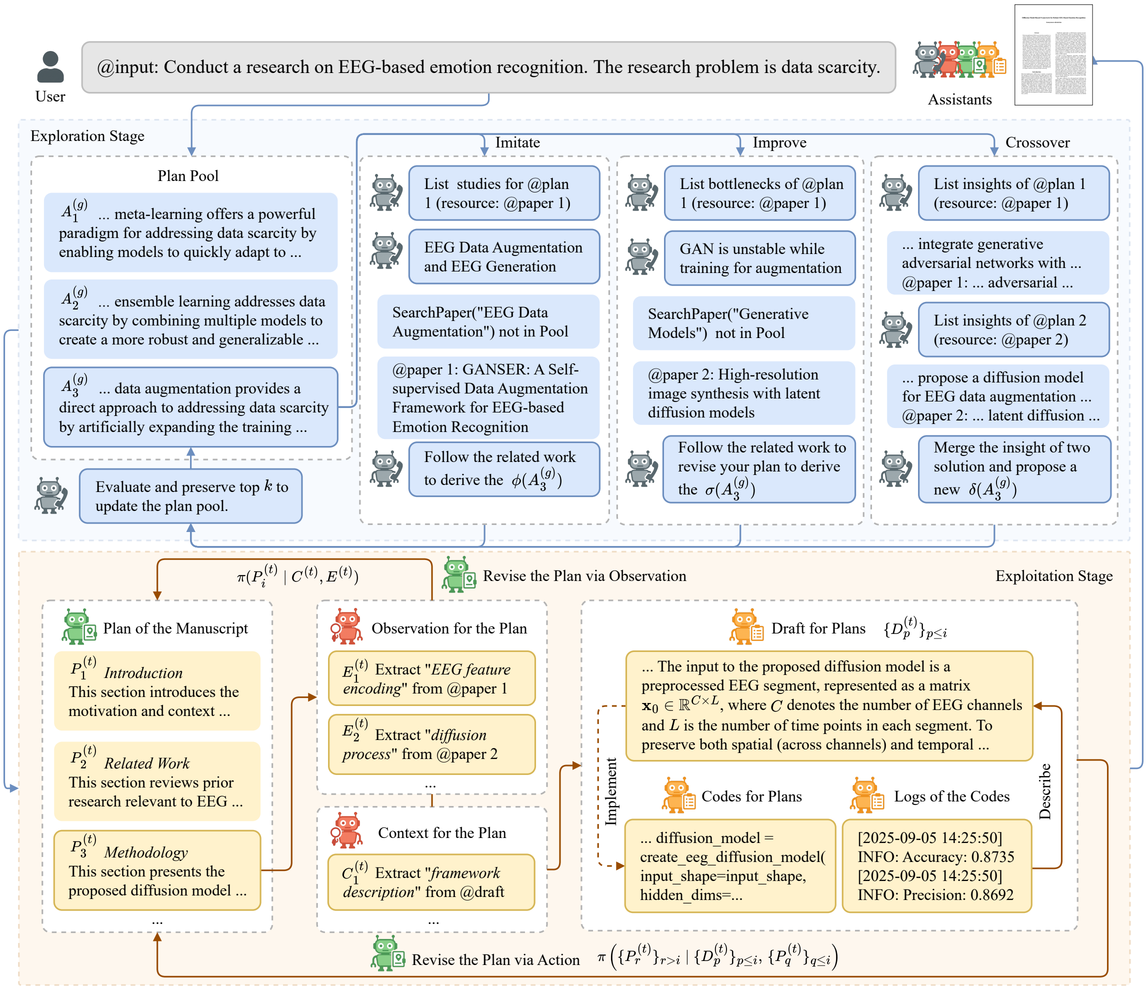
Existing work has made progress in research paper generation. However, most approaches rely on predefined or one-shot planning, which risks error accumulation and misalignment between the plan and the actual evidence.

2.2 Planning of Large Language Model Agents

Large language models have achieved remarkable success across various domains, demonstrating significant abilities in reasoning, tool usage, planning, and instruction following. The intelligence of large language models highlights their potential as the cognitive core of agents, which offers opportunities to improve planning ability [14]. For example, Plan-and-Solve [27] proposes a two-step prompt instruction, where the model first devises a plan and then carries out the plan. This zero-shot approach has achieved improvements in mathematical reasoning, common-sense reasoning, and symbolic reasoning. ProgPrompt [22] translates natural language descriptions of tasks into coding problems by symbolizing the agent's action space and objects in the environment through code, with each action formalized as a function and each object represented as a variable. Plan-Act [6] targets HTML manipulation, and LLMCompiler [15] breaks down question-answering tasks into parallel execution graphs.

Despite these promising contributions, large language models still fall short in more complex scenarios such as long-horizon planning [4]. Several hierarchical planning frameworks have been proposed. AgentOccam [32] incorporates planning into the action space using tree-like planning. WebPilot [34] uses six different agents. AdaPlanner [23] employs InPlan and Out-of-Plan Refiners for replanning, and ADaPT [19] uses recursive decomposition. PLAN-AND-ACT [7] provides a two-agent framework with a systematic approach to generating high-quality training data for open-source large language models.

However, although research on planning with large language models has made significant progress, there is still a lack of studies on long-horizon planning, especially for research paper generation.


User
Assistants
Exploration Stage
Plan Pool
Imitate
Improve
Crossover
Exploitation Stage
$\pi(P_i^{(t)} \mid C^{(t)}, E^{(t)})$
Revise the Plan via Observation
Plan of the Manuscript
Observation for the Plan
Context for the Plan
Draft for Plans $\{D_p^{(t)}\}_{p\le i}$
Codes for Plans
Logs of the Codes
Implement
Describe
Revise the Plan via Action $\pi\left(\{P_r^{(t)}\}_{r>i} \mid \{D_p^{(t)}\}_{p\le i}, \{P_q^{(t)}\}_{q\le i}\right)$

@input: Conduct a research on EEG-based emotion recognition. The research problem is data scarcity.

$A_1^{(g)}$ ... meta-learning offers a powerful paradigm for addressing data scarcity by enabling models to quickly adapt to ...

$A_2^{(g)}$ ... ensemble learning addresses data scarcity by combining multiple models to create a more robust and generalizable ...

$A_3^{(g)}$ ... data augmentation provides a direct approach to addressing data scarcity by artificially expanding the training ...

Evaluate and preserve top k to update the plan pool.

List studies for @plan 1 (resource: @paper 1)

EEG Data Augmentation and EEG Generation

SearchPaper("EEG Data Augmentation") not in Pool

@paper 1: GANSER: A Self-supervised Data Augmentation Framework for EEG-based Emotion Recognition

Follow the related work to derive the $\phi(A_3^{(g)})$

List bottlenecks of @plan 1 (resource: @paper 1)

GAN is unstable while training for augmentation

SearchPaper("Generative Models") not in Pool

@paper 2: High-resolution image synthesis with latent diffusion models

Follow the related work to revise your plan to derive the $\sigma(A_3^{(g)})$

List insights of @plan 1 (resource: @paper 1)

... integrate generative adversarial networks with ... @paper 1: ... adversarial ...

List insights of @plan 2 (resource: @paper 2)

... propose a diffusion model for EEG data augmentation ... @paper 2: ... latent diffusion ...

Merge the insight of two solution and propose a new $\delta(A_3^{(g)})$

$P_1^{(t)}$ *Introduction* This section introduces the motivation and context ...

$P_2^{(t)}$ *Related Work* This section reviews prior research relevant to EEG ...

$P_3^{(t)}$ *Methodology* This section presents the proposed diffusion model ...

...

$E_1^{(t)}$ Extract "*EEG feature encoding*" from @paper 1

$E_2^{(t)}$ Extract "*diffusion process*" from @paper 2

...

$C_1^{(t)}$ Extract "*framework description*" from @draft

...

... The input to the proposed diffusion model is a preprocessed EEG segment, represented as a matrix $\mathbf{x}_0 \in \mathbb{R}^{C\times L}$, where C denotes the number of EEG channels and L is the number of time points in each segment. To preserve both spatial (across channels) and temporal ...

```
... diffusion_model =
create_eeg_diffusion_model(
input_shape=input_shape,
hidden_dims=...
```

```
[2025-09-05 14:25:50]
INFO: Accuracy: 0.8735
[2025-09-05 14:25:50]
INFO: Precision: 0.8692
```

Fig. 1. Overview of the Imitation-and-Evolve framework.

3 Methodology

In this section, we present our Imitation-and-Evolve approach in detail. The overall framework, illustrated in Fig. 1, consists of two stages: drafting and revision. In the drafting stage, the imitation agent emulates relevant work to plan the manuscript outline, while the evolve agent iteratively drafts and refines the outline. In the revision stage, the imitation agent simulates human review to generate revision plan, and the evolve agent iteratively revises the manuscript and decides whether to adopt each point.

3.1 Imitation-and-Evolve for Drafting

Given a user request U describing a research topic, we first propose an imitation agent to imitate human expert planning. The imitation agent first prompts an LLM to analyze U and decompose it into an algorithm of interest U_A and an application of interest U_T. Formally, we represent this as $\mu(U) = (U_A, U_T; r)$, where $\mu(.)$ is the decomposition function and r denotes any additional requirements or constraints.

Then, the imitation agent retrieves related work in terms of the algorithm of interest and the application of interest. The retrieval results are represented as $\phi(U_A) = \{L_A^{(1)}, \ldots, L_A^{(K)}\}$ and $\phi(U_T) = \{L_T^{(1)}, \ldots, L_T^{(K)}\}$, where $L_A^{(m)}$ and $L_T^{(n)}$ denote the m-th and n-th retrieved papers, respectively. The imitation agent is equipped with two tools for retrieval, i.e., a search tool and a rerank tool. The search tool receives a query and invokes HuggingFace's paper API[1] to search for candidate papers. The rerank tool uses the bge-reranker-v2-m3 [3] model to encode both the query and the titles and abstracts of the searched papers, and returns the top K most similar results. To find an optimal combination of retrieved algorithm and application, we prompt the LLM to evaluate all pairs of retrieved algorithms and applications, and identify the best-matched pair. This process derives $M_A \in L_A^{(m)}$ and $M_T \in L_T^{(n)}$, representing the matched algorithm and application papers, respectively.

The imitation agent refers to the section and subsection outlines of these selected papers as human expert plans for organizing the manuscript. We prompt the LLM to extract the sections and subsections along with their summaries from M_A and M_T, resulting in $P_A = \sigma(M_A)$ and $P_T = \sigma(M_T)$. Next, the imitation agent imitates these outlines to generate a new outline for the target manuscript, with the algorithm of interest U_A, application of interest U_T, and any constraints r as additional inputs. The derived outline P serves as the initial plan for drafting the manuscript.

Then, P is handed off to the evolve agent, which is responsible for executing P and updating it whenever discrepancies between the plan and observed results arise. The evolve agent follows an iterative paradigm, starting from step $n = 1$ and adjustment index $t = 1$, which corresponds to the initial execution before any adjustments. The plan at the t-th adjustment is denoted as $P^{(t)} = \{P_1^{(t)}, \ldots, P_N^{(t)}\}$, where N is the number of planned steps. For the plan $P_n^{(t)}$ at the n-th step after $t - 1$ adjustments, the evolve agent first fetches observations relevant to $P_n^{(t)}$. The evolve agent is equipped with two tools, i.e., an external tool for fetching segments from external resources, and a contextual tool for fetching content within the existing draft. When using these tools, the evolve agent is provided with external resources, such as M_A and M_T, or the current draft as context. The LLM is prompted to identify the appropriate line ranges, and the content within these ranges is extracted and concatenated to facilitate efficient evidence gathering. The content retrieved from context for step n is denoted as C_n, while the content retrieved from external resources is denoted as E_n. Using these observations, the evolve agent updates the plan according to

$$P_i^{(t+1)} = \pi(P_i^{(t)} \mid C^{(t)}, E^{(t)}, P_i^{(t)}) \tag{1}$$

where $\pi(.)$ is a policy function that revises each step i of the current plan based on the available context $C^{(t)}$, external evidence $E^{(t)}$, and the i-th step of the current plan $P_i^{(t)}$. The policy function is implemented by prompting the LLM to update the plan according to the observations. Then, the evolve agent

The revised step $P_i^{(t+1)}$ is then executed to produce the draft D_i for the i-th section, using the contextual segments C_i and external segments E_i as context, and $P_i^{(t+1)}$ as the instruction. After deriving the draft D_i, the imitation agent further updates the remaining steps in the plan. The subsequent plan steps are modified as needed to be appropriate given the new results:

[1] https://huggingface.co/docs/hub/api

$$P_j^{(t+2)} = \begin{cases} P_j^{(t)}, & \text{if } j < i \\ P_j^{(t+1)}, & \text{if } j = i \\ \pi\left(P_j^{(t)} \mid \{D_p^{(t)}\}_{p \le i},\ \{P_q^{(t)}\}_{q<j}\right), & \text{if } j > i \end{cases} \tag{2}$$

where $P_j^{(t+2)}$ denotes the j-th step in the updated plan after considering the collection of drafts $\{D_p^{(t)}\}_{p\le i}$ already produced and the collection of plan steps $\{P_q^{(t)}\}_{q<j}$ as context. In this paper, the LLM is prompted to revise the subsequent plan steps and return a revised version if the current steps are not appropriate; otherwise, the original steps are retained.

Next, the evolve agent moves to the next step in the plan and starts a new round of iteration. This iterative process continues until all N steps are completed. Finally, the evolve agent collects the section titles according to the last version of the plan and combines them with all the drafts $\{D_1, D_2, \ldots, D_N\}$ to form the main body of the manuscript. The evolve agent then prompts the LLM to generate the title and abstract based on the completed body as context. Supplemented with references, all the content is rendered into a LaTeX project according to the conference template, and the final PDF of the paper is generated.

3.2 Imitation-and-Evolve for Revision

After deriving draft sections $\{D_1, D_2, \ldots, D_{N_D}\}$, we design the revision stage, which aims to further improve the manuscript. To plan systematic revisions, we introduce an imitation agent that mimics the peer-review and revision process commonly practiced by human experts in scientific research.

We first design the imitation agent to generate review feedback and decompose it into actionable points, which then serve as the revision plan. Following the standard peer-review protocols adopted by leading conferences that make their review comments public on OpenReview[2] (e.g., ICLR), our imitation agent prompts the LLM to generate comprehensive review feedback across multiple dimensions. Formally, given the complete manuscript $\{D_1, D_2, \ldots, D_{N_D}\}$, the imitation agent applies a feedback generation function δ such that $F = \delta(\{D_1, D_2, \ldots, D_{N_D}\})$. The generated feedback F covers aspects such as summary, soundness, presentation, contribution, strengths, weaknesses, questions, and overall rating. Next, we decompose F into a set of review points $\{G_1, G_2, \ldots, G_L\} = \beta(F)$, where β is a decomposition function that prompts the LLM to enumerate feedback items from F. Each G_l represents a comment that can be addressed independently, forming the initial revision plan.

These review points $\{G_1, G_2, \ldots, G_L\}$ are then handed off to the evolve agent, which is responsible for executing and updating the revision plan in response to observed outcomes. The evolve agent operates in an iterative paradigm, starting from review point $l = 1$ and adjustment index $t = 1$, corresponding to the initial execution before any adjustments. The revision plan at the t-th adjustment is denoted as $\{G_1^{(t)}, \ldots, G_L^{(t)}\}$, where L is the total number of review points. For a given review point $G_l^{(t)}$ at step l after $t-1$ adjustments, the evolve agent first gathers relevant observations. Similar to imitation-and-evolve for drafting, it is equipped with two tools: an external tool for retrieving evidence from external resources, and a contextual tool for extracting content from the current manuscript draft. The content extracted from the current draft for step l is denoted as C_l, while content from external resources is denoted as E_l.

Not all review feedback should be followed unconditionally, as some points may arise from misunderstandings. Therefore, for each plan G_l, the evolve agent determines whether to revise or rebut using a decision function

[2] https://openreview.net/

$C(G_l, C_l, E_l) \in \{0, 1\}$, where 0 indicates rebuttal and 1 indicates revision. The update to the plan before execution can be formalized as:

$$G_l^{(t+1)} = \begin{cases} \varnothing, & \text{if } C(G_l, C_l, E_l) = 0 \\ G_l^{(t)}, & \text{if } C(G_l, C_l, E_l) = 1 \end{cases} \tag{3}$$

where $\varnothing$ denotes an empty plan that does not need to be executed, indicating that the review point is skipped if the decision is to rebut. If the decision is to rebut, the agent skips to the next point. Otherwise, the agent proceeds to modify the relevant section identified by C_l. The revised section is denoted as $\hat{D}_i = \eta(D_i, G_l)$, where η is a modification function that incorporates the feedback G_l into the original section draft D_i.

After executing the plan, we evaluate the revised draft to assess its quality. Here, we prompt the LLM to generate feedback on both versions, putting the revised section $\hat{D}_i$ and the original D_i back into the manuscript, respectively. Then we use $\delta()$ to generate feedback on the two versions of the manuscript and extract a quality score, with the derived score of the corresponding version denoted as $\gamma(\cdot)$. The acceptance or rollback of the update can be formalized as:

$$D_i^{(t+1)} = \begin{cases} \hat{D}_i, & \text{if } \gamma(\hat{D}_i) \geq \gamma(D_i) \\ D_i, & \text{if } \gamma(\hat{D}_i) < \gamma(D_i) \end{cases} \tag{4}$$

where the revision is accepted only if $\gamma(\hat{D}_i) \geq \gamma(D_i)$, indicating a genuine improvement. If the quality does not improve, we roll back to the previous version, keeping D_i unchanged.

This iterative process continues for all review points $\{G_1, G_2, \ldots, G_L\}$, resulting in a revised draft. The entire revision cycle can be repeated with new rounds of review feedback and revisions.

4 Conclusion

In this work, we address the persistent challenge of error accumulation in long-horizon planning for automated scientific research generation. We propose the Explore-and-Evolve framework, which integrates imitation and evolve agents to continuously realign research plans with new observations, and introduce a novel human–computer interaction paradigm that empowers users to intervene at any stage. Experimental results on both public and self-constructed benchmarks demonstrate that our approach achieves superior reviewer scores compared to state-of-the-art methods.

For future work, we plan to conduct more comprehensive subjective experiments by involving PhD students in human-in-the-loop control and professors in review and evaluation. This will enable us to further assess and enhance the proposed framework in real-world research scenarios.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *ArXiv Preprint* (2023).

[2] Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2025. ResearchAgent: Iterative Research Idea Generation over Scientific Literature with Large Language Models. In *Nations of the Americas Chapter of the Association for Computational Linguistics*. 6709–6738.

[3] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *ArXiv Preprint* (2024).

[4] Yanan Chen, Ali Pesaranghader, Tanmana Sadhu, and Dong Hoon Yi. 2024. Can We Rely on LLM Agents to Draft Long-Horizon Plans? Let's Take TravelPlanner as an Example. *ArXiv Preprint* (2024).

[5] Jiangshu Du, Yibo Wang, Wenting Zhao, Zhongfen Deng, Shuaiqi Liu, Renze Lou, Henry Zou, Pranav Narayanan Venkit, Nan Zhang, Mukund Srinath, et al. 2024. LLMs Assist NLP Researchers: Critique Paper (Meta-) Reviewing. In *Empirical Methods in Natural Language Processing*. 5081–5099.

[6] Lutfi Eren Erdogan, Hiroki Furuta, Sehoon Kim, Nicholas Lee, Suhong Moon, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. 2025. Plan-and-Act: Improving Planning of Agents for Long-Horizon Tasks. In *International Conference on Machine Learning*.

[7] Lutfi Eren Erdogan, Hiroki Furuta, Sehoon Kim, Nicholas Lee, Suhong Moon, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. 2025. Plan-and-Act: Improving Planning of Agents for Long-Horizon Tasks. In *International Conference on Machine Learning*.

[8] Aniketh Garikaparthi, Manasi Patwardhan, Lovekesh Vig, and Arman Cohan. 2025. Iris: Interactive research ideation system for accelerating scientific discovery. In *Annual Meeting of the Association for Computational Linguistics*.

[9] Sikun Guo, Amir Hassan Shariatmadari, Guangzhi Xiong, Albert Huang, Myles Kim, Corey M Williams, Stefan Bekiranov, and Aidong Zhang. 2025. Ideabench: Benchmarking large language models for research idea generation. In *Knowledge Discovery and Data Mining*. 5888–5899.

[10] Fuguang Han and Zongzhang Zhang. 2023. Expert data augmentation in imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 16220–16221.

[11] Joey Hejna, Pieter Abbeel, and Lerrel Pinto. 2023. Improving long-horizon imitation through instruction prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 7857–7865.

[12] Tom Hope, Doug Downey, Daniel S Weld, Oren Etzioni, and Eric Horvitz. 2023. A computational inflection for scientific discovery. *Commun. ACM* 66, 8 (2023), 62–73.

[13] Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2024. MLAgentBench: evaluating language agents on machine learning experimentation. In *International Conference on Machine Learning*. 20271–20309.

[14] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of LLM agents: A survey. *ArXiv Preprint* (2024).

[15] Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. 2024. An llm compiler for parallel function calling. In *International Conference on Machine Learning*.

[16] Pat Langley. 1987. *Scientific discovery: Computational explorations of the creative processes.*

[17] 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* (2024).

[18] OpenAI. 2025. Introducing GPT-4.1 in the API. https://openai.com/index/gpt-4-1. Accessed: 2025-05-01.

[19] Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. 2024. ADaPT: As-Needed Decomposition and Planning with Language Models. In *Findings of the Association for Computational Linguistics*. 4226–4252.

[20] Kevin Pu, KJ Kevin Feng, Tovi Grossman, Tom Hope, Bhavana Dalvi Mishra, Matt Latzke, Jonathan Bragg, Joseph Chee Chang, and Pao Siangliulue. 2025. Ideasynth: Iterative research idea development through evolving and composing idea facets with literature-grounded feedback. In *Conference on Human Factors in Computing Systems*. 1–31.

[21] Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2024. Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers. In *International Conference on Learning Representations*.

[22] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. 2023. ProgPrompt: Generating Situated Robot Task Plans using Large Language Models. In *International Conference on Robotics and Automation*. 11523–11530.

[23] Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. 2023. Adaplanner: Adaptive planning from feedback with language models. *Advances in Neural Information Processing Systems* 36 (2023), 58202–58245.

[24] Arun Venkatraman, Martial Hebert, and J Bagnell. 2015. Improving multi-step prediction of learned time series models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[25] Guan Wang, Haoyi Niu, Jianxiong Li, Li Jiang, Jianming Hu, and Xianyuan Zhan. 2025. Are Expressive Models Truly Necessary for Offline RL?. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 21062–21070.

[26] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. 2023. Scientific discovery in the age of artificial intelligence. *Nature* 620, 7972 (2023), 47–60.

[27] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In *Annual Meeting of the Association for Computational Linguistics*. 2609–2634.

[28] Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2023. Scimon: Scientific inspiration machines optimized for novelty. In *Annual Meeting of the Association for Computational Linguistics*. 279–299.

[29] Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Qingsong Wen, Wei Ye, et al. 2024. Autosurvey: Large language models can automatically write surveys. *Advances in Neural Information Processing Systems* 37 (2024), 115119–115145.

[30] Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. 2025. CycleResearcher: Improving Automated Research via Automated Review. In *International Conference on Learning Representations*.

[31] 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* (2025).

[32] Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. 2024. AgentOccam: A Simple Yet Strong Baseline for LLM-Based Web Agents. In *International Conference on Learning Representations*.

[33] Zonglin Yang, Xinya Du, Junxian Li, Jie Zheng, Soujanya Poria, and Erik Cambria. 2024. Large Language Models for Automated Open-domain Scientific Hypotheses Discovery. In *Findings of the Association for Computational Linguistics*. 13545–13565.

[34] Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. 2025. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. In *AAAI Conference on Artificial Intelligence*, Vol. 39. 23378–23386.