Imitate-and-Evolve: AI Research Assistants for Long-Horizon Planning and Execution with Human-in-the-Loop

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 Imitate-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. Imitate-and-Evolve: AI Research Assistants for Long-Horizon Planning and Execution with Human-in-the-Loop. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 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 [22]. 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 [10]. 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 [14]. The recent emergence of large language models (LLMs) [1] has driven remarkable progress in processing, organizing, and generating scientific text [22]. Increasing efforts now leverage commercial LLMs to develop research agents that can propose novel research ideas [2, 24, 29], assist in designing and conducting experiments [5], or function as AI scientists capable of generating complete, open-ended scientific publications automatically [16, 27]. 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 [26].

Manuscript submitted to ACM

Recent progress has pushed the boundary to the point where a workshop paper written entirely by AI has been accepted through peer review [27]. However, as highlighted by human evaluations of generated manuscripts [27], 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 [**? ? ? ?**].

As a demonstration, we employ state-of-the-art research assistants, OpenAI's Deep Research [**?**] and Gemini Deep Research, to draft a paper on Diffusion models for data augmentation in EEG (electroencephalogram)-based emotion recognition. As shown in Fig. **??**, 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 [26] 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 [29] to improve research idea generation. These works

aim to enhance the novelty and diversity of artificial intelligence in creative research tasks. Si et al. [19] conducted a comprehensive human evaluation of idea generation by language models. Wang et al. [25] 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. [11] introduced a benchmark for evaluating large language models in coding solutions for machine learning problems. Wang et al. [24] proposed an approach that leverages large language models for scientific literature retrieval. The AI Scientist project [16] introduced a fully automated, prompt-driven research pipeline. Subsequently, the AI Scientist-v2 project [27] introduced a workshop-level automated scientific discovery method based on agentic tree search. More recently, Weng et al. [26] 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. [18] 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 [12]. For example, Plan-and-Solve [23] 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 [20] 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 [13] 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 [28] incorporates planning into the action space using tree-like planning. WebPilot [30] uses six different agents. AdaPlanner [21] employs InPlan and Out-of-Plan Refiners for replanning, and ADaPT [17] 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.

3 Methodology

In this section, we present our Imitation-and-Evolve approach in detail. The overall framework, illustrated in Fig. **??**, 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 Experiments

4.1 Experimental Settings

Following the recenttrends in using LLMs to judge the quality of out-put texts (especially in the setting of reference-freeevaluations) [15, 31], we use GPT-4 to judge the qual-ity of research ideas. Note that each of the problem,method, and experiment design is evaluated withfive different criteria. We ask the LLM-based evaluation model toeither rate the generated idea on a 5-point Likertscale for each criterion or perform pairwise compar-isons between two ideas from different models.

We compare AutoSurvey with surveys authored by human experts (collected from Arxiv) and naive RAG-based LLMs across 20 different computer science topics across 20 different topics in the field of LLMs (see Table 6). For the naive RAG-based LLMs, we begin with a title and a survey length requirement, then iteratively prompt the model to write the content until completion. Note that we also provide the model with the same number of reference papers with AutoSurvey.

We mainly use the GPT-4 [1] releasefrom Nov 06, 2023, as the basis for all models,which is, notably, reported to be trained with dataup to Apr 2023 (meanwhile, the papers used foridea generation appear after May 2023).

5 Experiments

5.1 Experimental Settings

We conduct experiments on two benchmark datasets, AutoSurvey [25] and SurveyEval [**?**], as well as a self-constructed benchmark dataset, TopSurvey. For hyperparameters, we set the maximum number of iterations for exploration, exploitation, and experience taskforces to 4. The hyperparameter θ is set to 500. We use GPT-4.1 as the LLM for both generation and evaluation. The evaluation comprises two categories of metrics. For citation quality, we adopt citation recall and citation precision as proposed by Wang et al. [25]: recall measures whether cited passages fully support all statements, while precision measures the proportion of relevant citations that support their corresponding statements. For content quality, following Wang et al. [25], we use coverage, structure, and relevance, each rated by LLMs on a 5-point scale. Coverage assesses the extent to which the survey addresses all relevant aspects of the topic; structure evaluates logical organization and coherence; and relevance measures alignment with the specified research topic. We do not filter out fractional scores, such as 4.5.

5.2 Comparision Experiments

We first evaluate our framework on the AutoSurvey [25] benchmark, following the protocols established by Wang et al. [25**?**]. We use the same 20 topics from diverse subfields of LLM research to generate survey articles for comparison. We compare our approach with three baselines: Naive RAG-based LLMs using retrieval-augmented generation, AutoSurvey [25], and SurveyX [**?**]. We use the official implementations of AutoSurvey[3] and SurveyX[4] to conduct experiments. Due to the lack of an online version of SurveyX, we conduct offline generation.

As shown in Table **??**, most algorithms exhibit reduced performance as survey length increases, particularly in recall, precision, structure, and relevance. This decline is most pronounced in the Naive RAG baseline, indicating that longer workflows lead to greater error accumulation and propagation. In contrast, coverage remains stable or improves with longer surveys, likely due to more comprehensive topic inclusion. State-of-the-art methods such as AutoSurvey and SurveyX continue to face challenges with citation precision, frequently generating statements unsupported by references. For content quality, structure consistently receives the lowest scores, reflecting disorganized article organization. Our proposed method remains robust to these issues and consistently achieves superior results across all metrics.

We further evaluate our framework on the SurveyEval benchmark [**?**], using the same protocols. SurveyEval is the first benchmark in computer science that pairs surveys with complete reference papers, comprising 384 arXiv cs.CL surveys citing over 26,000 references. Twenty topics were selected for testing based on reference completeness and reference list diversity. Experimental results are shown in Fig. **??**, with Coverage, Structure, and Relevance scores normalized to a 100-point scale.

In Fig. **??**, all methods achieve high coverage and relevance, indicating that LLMs like GPT-4.1 can generate comprehensive and relevant content. However, recall and precision remain low, reflecting poor reference retrieval and insufficient support for generated statements. As a result, state-of-the-art methods struggle with logical organization, leading to lower structure scores. Our method overcomes these issues, achieving the best performance across all metrics.

[3]https://github.com/AutoSurveys/AutoSurvey
[4]https://github.com/IAAR-Shanghai/SurveyX

Methods	Citation Quality		Content Quality			
	Rec. ↑	Pre. ↑	Cov. ↑	Str. ↑	Rel. ↑	Avg. ↑
w/o exploration	94.38	88.56	4.83	4.83	5.00	4.89
w/o exploitation	97.86	79.02	4.97	4.93	4.93	4.94
w/o experience	97.78	80.82	4.79	4.89	4.97	4.88
Proposed	**98.17**	**89.28**	**4.97**	**4.95**	**5.00**	**4.97**

Table 1. Ablation study of the proposed modules on the benchmark dataset.

Methods	Citation Quality		Content Quality			
	Recall ↑	Precision ↑	Coverage ↑	Structure ↑	Relevance ↑	Avg. ↑
Human	93.07	87.76	5.00	4.97	5.00	4.99
Naive RAG (2024)	64.57	61.89	4.29	3.58	4.67	4.18
AutoSurvey (2024)	70.03	71.66	4.68	4.67	4.87	4.74
SurveyX (2025)	75.51	77.90	4.71	4.84	4.93	4.83
Proposed	**86.63**	**81.98**	**4.85**	**4.90**	**5.00**	**4.92**

Table 2. Comparison of automatic survey generation methods at a survey length of 64k tokens on the new large-scale benchmark dataset. Higher scores indicate better performance.

Method	Iteration	Citation Quality		Content Quality			
		Rec. ↑	Pre. ↑	Cov. ↑	Str. ↑	Rel. ↑	Avg. ↑
Exper.	1	83.64	74.37	4.61	4.79	4.94	4.78
	2	84.99	78.19	4.77	4.83	4.98	4.86
	3	85.11	80.13	4.82	4.86	4.98	4.89
	4	86.63	**81.98**	**4.85**	**4.90**	**5.00**	**4.92**
	5	**86.71**	81.86	**4.85**	4.88	4.98	4.90
Explor.	1	82.84	81.26	4.71	4.78	**5.00**	4.83
	2	84.79	81.67	4.83	4.82	**5.00**	4.88
	3	85.69	81.97	**4.85**	4.84	**5.00**	4.90
	4	86.63	**81.98**	**4.85**	**4.90**	**5.00**	**4.92**
	5	**86.65**	81.92	**4.85**	4.87	**5.00**	4.91
Exploi.	1	86.31	72.68	4.81	4.87	4.91	4.86
	2	86.00	77.34	4.83	4.90	4.94	4.89
	3	**86.75**	79.10	4.84	4.89	4.98	4.90
	4	86.63	**81.98**	**4.85**	**4.90**	**5.00**	**4.92**
	5	86.47	80.27	4.84	4.89	5.00	4.91

Table 3. Sensitivity experiments across different iterations for the experience, exploration, and exploitation taskforces on the new large-scale benchmark.

5.3 Ablation Study

We further conduct an ablation study to evaluate the effectiveness of each component in our proposed framework. Experiments are performed on the same benchmark dataset as before, with all experimental settings unchanged and the survey length set to 8k tokens. We compare the full model with three variants. We use a single round of retrieval and

organization for the variant without the exploration taskforce to generate the overall outline. We draft the manuscript using only one round of extraction and writing for the variant without the exploitation taskforce. For the variant without the experience taskforce, each agent completes its assigned task, without revision.

Table 1 presents the results. Removing the exploration taskforce causes the largest drops in recall and structure, demonstrating its importance for citation recall and logical organization. Excluding the exploitation taskforce sharply reduces precision and relevance, confirming its role in improving citation precision and content relevance. Without the experience taskforce, all metrics decrease, especially coverage, highlighting its key role in ensuring comprehensive coverage through collaborative revision.

5.4 A New Large-Scale Benchmark

To further validate the effectiveness of our proposed framework, we construct a new large-scale benchmark dataset. This dataset consists of 195 topics from various subfields of computer science, nearly 10 times larger than previous benchmarks [25**?**]. To ensure topic quality, we collect peer-reviewed survey topics from top computer science conferences, rather than from preprint sources such as arXiv. Survey papers accepted only as abstracts are excluded. To prevent data leakage from LLM pretraining data, we include only survey papers published in 2023, 2024, and 2025. We employ PhD students in computer science to verify whether a paper qualifies as a survey and to collect the final set of 195 survey papers: 9 from AAAI, 34 from ACL, 46 from EMNLP, 3 from ICLR, 2 from ICML, 77 from IJCAI, and 24 from NAACL. Of these, 68 were published in 2023, 105 in 2024, and 22 in 2025.

We conduct experiments using the same settings as above, generating 64k-token literature reviews for evaluation. As shown in Table 2, compared with results on the existing benchmark in Table **??**, performance on the new large-scale benchmark drops significantly, particularly in recall and coverage. This may be due to the greater number and broader range of topics, which leads to some relevant literature not being retrieved, resulting in lower citation recall and coverage. Precision also drops slightly, likely because the models lack knowledge of the most recent two years. Therefore, compared to existing benchmarks, the new large-scale benchmark is more challenging and leads to more errors in generation. Despite this, our proposed method achieves over 80% in citation scores and an average content quality score of 4.92.

5.5 Sensitivity Analysis

To gain deeper insight into our framework, we conduct a sensitivity analysis to observe how self-improving iterations affect performance. We evaluate the experience, exploration, and exploitation taskforces across different numbers of iterations, employing a controlled variable approach: when varying one hyperparameter, all others are held constant as previously described.

The results are presented in Table 3. We find that the second and third iterations yield the most significant improvements. This indicates that errors can accumulate at various steps in the workflow, and iterative refinement effectively reduces intermediate errors, thereby enhancing the quality of the final results. Both the Experience and exploitation taskforces contribute most to faithfulness; from the first to the fourth iteration, the number of references supporting claims in the final output increases by a large margin. Additionally, the Experience and exploration taskforces contribute notably to coverage and structural quality, suggesting that a single revision is often insufficient for comprehensive improvement. However, the effect of additional iterations gradually diminishes. By the fifth iteration, we observe a slight decline, due to unnecessary modifications overwriting correct results and introducing new errors.

Cluster	Citation Quality		Content Quality		
	Recall ↑	Precision ↑	Coverage ↑	Structure ↑	Relevance ↑
1	83.61	**82.57**	4.67	4.78	4.93
2	84.54	75.12	4.75	4.85	4.85
3	72.29	76.11	4.59	4.88	4.87
4	**86.63**	81.98	**4.85**	**4.90**	**5.00**
5	81.37	69.77	4.66	4.74	4.73

Table 4. Real-world case study across different topics on 400 generated results. Higher scores indicate better performance.

5.6 Disscussions

We evaluated our framework in a real-world setting by deploying an online automatic literature review generation system[5], which has produced over 20,000 reviews. We randomly selected 400 reviews, embedded them with all-MiniLM-L6-v2, and applied K-means clustering to form five groups. Results show differences in citation and content quality among clusters. The best performance appears in computer science (cluster 4), while education and social sciences (cluster 3) perform worse, especially in citation quality. It may result from the search engine not including part of the relevant literature in the social sciences. We also measured system efficiency, and generating an 8,000-token review takes an average of 8.45 minutes.

6 Conclusion

In this work, we address the persistent challenge of error accumulation in long-horizon planning for automated scientific research generation. We propose the Imitate-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*.

[5] Anonymous for review; to be revealed upon acceptance.

[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] 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.

[11] 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.

[12] 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).

[13] 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*.

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

[15] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment. In *Empirical Methods in Natural Language Processing*. 2511–2522.

[16] 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).

[17] 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.

[18] 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.

[19] 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*.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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*.

[27] 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).

[28] 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*.

[29] 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.

[30] 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.

[31] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems* 36 (2023), 46595–46623.