EvoScientist: Towards Multi-Agent Evolving AI Scientists for Scientific Discovery

Anonymous ACL submission

Abstract

Large Language Models have enabled AI scientist systems that coordinate ideation, implementation, and evaluation, yet most systems remain static pipelines whose strategies do not improve from accumulated successes and failures. We propose **EvoScientist**, a self-evolving multi-agent framework that enables ideation–experimentation co-evolution through persistent memory. EvoScientist consists of a Researcher Agent (*RA*) for idea generation, an Engineer Agent (*EA*) for experiment execution, and an Evolution Manager Agent (*EMA*) that distills interaction histories into two persistent memories: an ideation memory for promising and failed research directions, and an experimentation memory for reusable data processing and model training strategies. For each new task, the *RA* and *EA* retrieve relevant memory items to improve idea quality and execution reliability over time. Experiments show that EvoScientist generally outperforms 4 open-source and 3 commercial AI scientist systems in scientific idea generation, demonstrating enhanced novelty, practical feasibility, topical relevance, and conceptual clarity across both automated and human evaluations. EvoScientist also increases code execution success rates through experiment strategy evolution. These results show that prior agent interactions can be consolidated into reusable knowledge, thereby enhancing AI-driven scientific discovery.

1 Introduction

Scientific discovery is an iterative process of reading prior work, forming hypotheses, testing them, and refining future research decisions from empirical feedback (Langley, 1987; Klahr and Simon, 1999; Popper, 2005). Recent progress in Large Language Models (LLMs) has accelerated this process by enabling agentic systems for scientific ideation, literature review, experiment implementation, and analysis (Chen et al., 2025; Lu et al., 2024; Yamada et al., 2025; Tang et al., 2025). One line of work focuses on idea generation, where LLM agents propose, critique, and refine research hypotheses (Si et al., 2025; Gottweis et al., 2025; Li et al., 2024; Gao et al., 2025b; Qi et al., 2024; Su et al., 2025). For example, Virtual Scientist (Su et al., 2025) and Co-Scientist (Gottweis et al., 2025) use multi-agent collaboration to simulate scientific ideation. Another line develops end-to-end AI scientist systems that automate broader research workflows from ideation to experimental validation (Lu et al., 2024; Yamada et al., 2025; Schmidgall et al., 2025; Weng et al., 2025; Shao et al., 2025; InternAgent Team et al., 2025; Tang et al., 2025; Chen et al., 2026; Liu et al., 2026; Zhu et al., 2026), including AI Scientist-v2 (Yamada et al., 2025), AI-Researcher (Tang et al., 2025), and InternAgent (InternAgent Team et al., 2025).

However, existing AI scientist systems largely operate as static pipelines. Their agent roles, prompting strategies, and interaction patterns are usually fixed after deployment, while accumulated outcomes and failures are rarely transformed into reusable experience. As a result, these systems repeatedly explore failed directions, overlook promising alternatives, or waste resources on infeasible proposals. This gives rise to a key question:

> How can an AI scientist system improve its idea-generation and code-generation strategies by learning from prior successes and failures?

To answer this question, we propose **EvoScientist**, a self-evolving multi-agent framework for ideation–experimentation co-evolution. EvoScientist consists of a Researcher Agent (*RA*) for generating research ideas and proposals, an Engineer Agent (*EA*) for implementing and executing experiments, and an Evolution Manager Agent (*EMA*) that distills interaction histories into structured persistent memories. Concretely, the *EMA* maintains an *ideation memory* that records promising and failed research directions, and an *experimentation*

memory that stores reusable data processing and model training strategies. EvoScientist updates these memories through idea direction evolution, idea validation evolution, and experiment strategy evolution, which summarize high-quality ideas, record high-level infeasible directions, and extract reusable execution strategies from detailed code search trajectories. For each new task, the *RA* and *EA* retrieve relevant memory items before generating ideas or code, enabling the system to improve across tasks through its own multi-agent evolution.

We conduct experiments on scientific idea generation, code generation, and full-paper manuscript-level peer-review evaluation. EvoScientist outperforms 4 open-source and 3 commercial baselines in idea generation quality, measured by novelty, feasibility, relevance, and clarity under both automated and expert human evaluation. EvoScientist also achieves higher code execution success rates through iterative experiment strategy evolution. As a supplementary external case study, we further evaluate EvoScientist through independent human expert peer review at the 1st International Conference on AI Scientists [1] (Zhongguancun Academy, 2025). Notably, all six full papers generated by EvoScientist and submitted to the conference satisfied the acceptance criteria established by human reviewers. Our main contributions are as follows:

- We propose **EvoScientist**, a self-evolving multi-agent framework for ideation–experimentation co-evolution, with stage-aligned ideation and experimentation memories that reuse prior successes and failures.
- We introduce three multi-agent self-evolution mechanisms, namely idea direction evolution, idea validation evolution, and experiment strategy evolution, that improve both idea generation and experiment execution over time.
- We provide empirical evidence that EvoScientist generates higher-quality ideas and achieves higher code execution success rates compared to strong open-source and commercial baselines.

2 Related Work

2.1 AI Agents for Scientific Discovery

AI agents for scientific discovery have progressed from assisting isolated research tasks to coordinating larger portions of the research lifecycle (Chen et al., 2025). Early studies show that LLMs can support scientific ideation by proposing, evaluating, and refining research ideas (Si et al., 2025; Li et al., 2024; Gao et al., 2025b; Qi et al., 2024). A second line of work explicitly targets idea novelty and diversity through iterative planning, search, or novelty-opimized retrieval (Wang et al., 2024c; Hu et al., 2025). Related systems use literature analysis, retrieval, and multi-agent debate to generate or evaluate research ideas (O'Neill et al., 2025; Azher et al., 2025; Sanyal et al., 2025; Liu et al., 2025b; Gottweis et al., 2025). Among the baselines in our experiments, Virtual Scientist (Su et al., 2025) represents multi-agent scientific ideation through proposal, critique, and refinement.

Recent systems move beyond ideation toward end-to-end scientific discovery, covering literature review, experimental design, code implementation, data analysis, and manuscript preparation (Lu et al., 2024; Intology AI, 2025; Schmidgall et al., 2025; Chen et al., 2026; Liu et al., 2026; Yang et al., 2026; Zhu et al., 2026). AI Scientist-v2 (Yamada et al., 2025) improves research trajectories with agentic tree search, AI-Researcher (Tang et al., 2025) orchestrates a structured research workflow, and InternAgent (InternAgent Team et al., 2025) incorporates human expert feedback into autonomous research. Other work explores collaborative agents and long-horizon discovery settings (Schmidgall and Moor, 2025; Schmidgall et al., 2025; Weng et al., 2025; Shao et al., 2025). In contrast to these systems, which mainly improve performance through within-run search, debate, or workflow design, EvoScientist focuses on cross-task multi-agent evolution by converting interaction histories into persistent knowledge for scientific discovery.

2.2 Self-Evolving Agents

Self-evolving agents aim to improve their behavior from experience rather than relying only on fixed prompts or pre-specified policies (Fang et al., 2025; Gao et al., 2025a). A common direction is to equip agents with memory systems that store, retrieve, and consolidate information from past interactions (Chhikara et al., 2025; Wang et al., 2025b; Zhao et al., 2024). Another direction improves agents through adaptive tool use, enabling them to create, refine, and reuse tools across tasks (Qiu et al., 2025a; Qu et al., 2025; Wang et al., 2024a). Related work also studies feedback-based learning (Shinn et al., 2023), imitation from expert demonstrations (Zelikman et al., 2022), and population-based evolution (Zhang et al., 2025).

[1] Submissions remain anonymous throughout human review, and accepted papers do not disclose author identities.

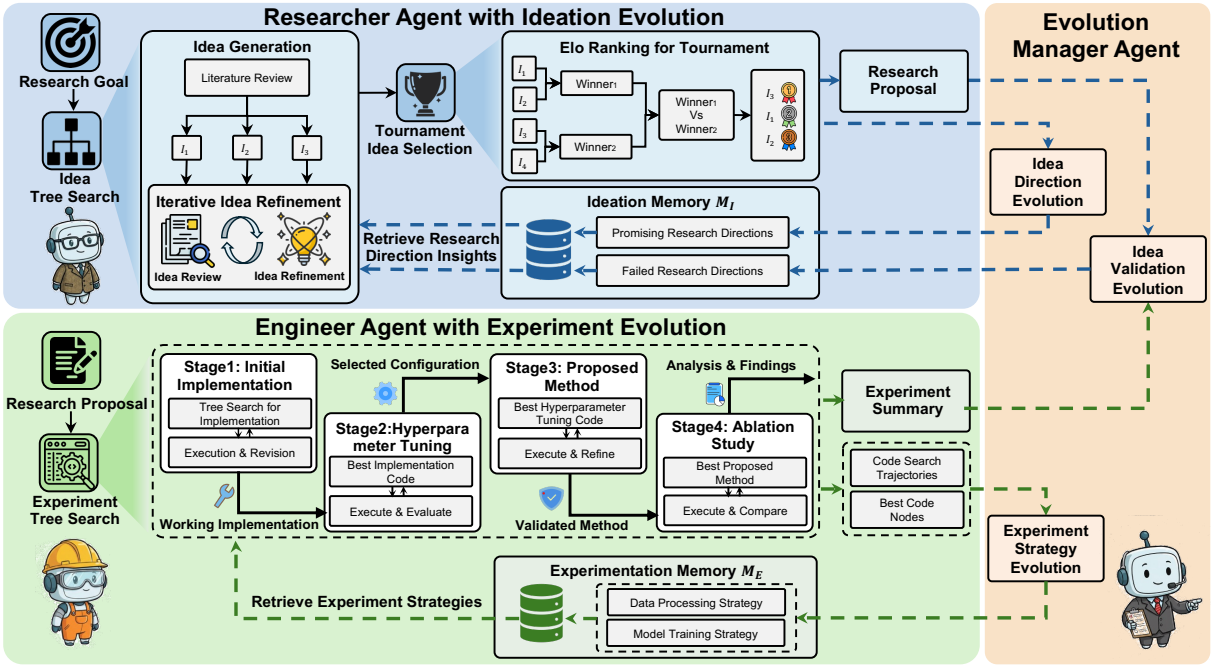

Figure 1: **Overview of the EvoScientist framework.** EvoScientist is a self-evolving multi-agent system for ideation–experimentation co-evolution. It consists of a researcher agent (RA), an engineer agent (EA), and an evolution manager agent (EMA). The EMA distills interaction histories into persistent ideation memory M_I and experimentation memory M_E, which are retrieved by the RA and EA to improve idea quality and execution success.

These techniques have shown promise in domains such as coding (Robeyns et al., 2025; Wang et al., 2025a), education (Liu et al., 2025a), and healthcare (Almansoori et al., 2025). However, most existing self-evolving agents are evaluated on single-stage or narrowly scoped tasks, and their evolution mechanisms are not explicitly designed for the multi-stage requirements of end-to-end scientific discovery. In contrast, EvoScientist evolves both ideation and experiment-execution strategies under one discovery objective, allowing accumulated successes and failures to guide future idea generation, validation, and code implementation.

3 Method

3.1 Problem Formulation

Following prior work (Weng et al., 2025; Tang et al., 2025; Shao et al., 2025), we formulate scientific discovery as a goal-driven and verifiable process that transforms a user goal G into a research proposal P and an execution report W. In this work, we focus on the ideation–experimentation co-evolution stage. The proposal is derived from an idea I, which specifies the method and experimental plan, while the execution report summarizes executable code C, logs, metrics, and failure diagnoses. The challenge is to improve both idea quality and execution reliability across tasks by learning from accumulated outcomes and failures.

3.2 Overall Framework

EvoScientist addresses this co-evolution task for a **user goal** G with three agents: a **researcher agent (RA)**, an **engineer agent (EA)**, and an **evolution manager agent (EMA)** (Figure 1). For a given G, the RA first retrieves goal-relevant direction knowledge from an ideation memory M_I, generates an idea I, and extends it into a full proposal P. Conditioned on P, the EA retrieves reusable execution strategies from an experimentation memory M_E, searches for executable code C, runs experiments, and produces a verifiable execution report W with outputs such as logs, metrics, and failure diagnoses. After the task is finished, the EMA summarizes the interaction histories to update M_I (promising and failed directions) and M_E (reusable execution strategies). For a new user goal G', the RA and EA retrieve the updated memories before generating I' and C', enabling cross-task multi-agent evolution.

3.3 Researcher Agent with Ideation Evolution

To enable multi-agent evolution in idea generation, EvoScientist equips the researcher agent with a persistent ideation memory M_I that records feasible directions and unsuccessful directions distilled from prior outcomes and failures.

Ideation Memory Retrieval. Given a user goal G,

the RA retrieves goal-relevant direction knowledge:

$$K_I = \text{Retrieve}_I(M_I, G), \tag{1}$$

where $\text{Retrieve}_I(\cdot)$ is implemented by embedding-based retrieval with cosine similarity, and we select the top-k_I most similar ideation memory items.

Idea Tree Search. Since the space of plausible ideas is large, the researcher agent performs a tree-structured propose–review–refine search grounded in literature review and retrieved memories. Specifically, each node in the search tree stores (i) an idea draft and (ii) its review feedback, and each expansion step uses the feedback to generate refined child ideas. The idea tree search generates a set of candidate ideas and their refinement signals:

$$\begin{aligned}&\{(I_1, \text{rev}_1), \ldots, (I_{N_I}, \text{rev}_{N_I})\}\\ &\quad = \text{IdeaTreeSearch}(G, L, K_I),\end{aligned} \tag{2}$$

where I_i is the i-th candidate idea, N_I is the maximum number of candidate ideas during tree search, rev_i stores the review feedback used in refinement, and L denotes the retrieved literature papers for G.

Tournament Idea Selection. EvoScientist adopts an Elo-based tournament (Elo, 1978), which produces stable rankings from noisy pairwise comparisons without requiring calibrated absolute scores. More specifically, the RA ranks candidate ideas on novelty, feasibility, relevance, and clarity:

$$\{r_1, \ldots, r_{N_I}\} = \text{EloRank}(I_{1:N_I}), \tag{3}$$

where r_i is the Elo score of idea I_i. We retain the top-3 ideas as a summary of promising directions:

$$\mathcal{I}_{\text{top}} = \text{Top-3}(\{(I_i, r_i)\}_{i=1}^{N_I}). \tag{4}$$

The RA then extends the top-1 idea into a proposal:

$$P = \text{Extend}(\text{Top-1}(\{(I_i, r_i)\}_{i=1}^{N_I})). \tag{5}$$

Here, the idea I includes a method description and an experimental plan, while the proposal P is a full version that contains background, related work, method, experimental plan, and expected results.

3.4 Engineer Agent with Experiment Evolution

To support multi-agent evolution in experiment execution, EvoScientist equips the engineer agent with a persistent experimentation memory M_E, which stores reusable data processing and model training strategies distilled from prior results and failures.

Experimentation Memory Retrieval. Given a proposal P, the engineer retrieves reusable execution strategies and augments the base prompt:

$$K_E = \text{Retrieve}_E(M_E, P), \tag{6}$$

where $\text{Retrieve}_E(\cdot)$ uses embedding-based retrieval with cosine similarity, selecting the top-k_E items.

Experiment Tree Search. Because the space of implementations and execution environments is large, the EA performs experiment tree search in four stages (initial implementation, hyperparameter tuning, proposed method, and ablation). At each stage $s \in \{1, 2, 3, 4\}$, it iteratively generates code, runs experiments, and records structured execution results; when execution fails, it diagnoses failures from logs and revises the code:

$$\begin{aligned}&\{(C_1^s, E_1^s), \ldots, (C_{N_E^s}^s, E_{N_E^s}^s)\}\\ &\quad = \text{ExperimentTreeSearch}(P, K_E),\end{aligned} \tag{7}$$

where C_j is j-th code at experiment stage s, N_E^s is the maximum number of attempts for stage s, and E_j^s is a structured execution record that includes run status, logs, and evaluation metrics. The best-performing code at each stage is selected as:

$$C_{best}^s = \text{argmax}_{j \in \{1, \ldots, N_E^s\}} \text{Top-1}(E_j^s), \tag{8}$$

We define the execution history for each stage s as:

$$H_E^s = \{(C_j^s, E_j^s)\}_{j=1}^{N_E^s}, \tag{9}$$

and summarize execution outcomes into an execution report aligned with the proposal:

$$W = \text{SummarizeExecution}(P, \{H_E^s\}_{s=1}^4). \tag{10}$$

3.5 Evolution Manager Agent

The evolution manager agent (EMA) converts interaction histories into reusable strategies so that the system can learn from outcomes and failures and improve both idea generation and experiment execution across tasks. EvoScientist implements multi-agent evolution through three self-evolutions: idea direction evolution, idea validation evolution, and experiment strategy evolution. Crucially, idea validation evolution closes a cross-agent loop: failures encountered by the EA during experiment tree search are surfaced through the EMA back into the ideation memory M_I, so that proposals similar to previously infeasible ones are deprioritized in future runs of the RA. Execution feedback thus shapes future ideation as well as future execution.

Idea Direction Evolution. To accumulate reusable feasible directions, the EMA summarizes promising research directions from top-ranked ideas $\mathcal{I}_{\text{top}}$:

$$F_I^{IDE} = \text{IDE}(G, \mathcal{I}_{\text{top}}). \quad (11)$$

where $\text{IDE}(\cdot)$ is an LLM-based summarizer. We use the summary to update the ideation memory:

$$M_I \leftarrow \text{Update}_I(M_I, F_I^{IDE}). \quad (12)$$

Idea Validation Evolution. To record unsuccessful directions, the EMA analyzes the execution report W for the selected proposal P. If the engineer cannot produce any code within the pre-defined budget (rule-based), we treat the proposal as failed. Otherwise, when experiments are complete, the EMA compares the proposed method against baselines based on W and uses an LLM-based analysis to judge whether the proposal fails (e.g., the proposed method performs worse than the baseline):

$$F_I^{IVE} = \text{IVE}(P, W), \quad (13)$$

where $\text{IVE}(\cdot)$ is an LLM-based validator. We use the analysis to update the ideation memory:

$$M_I \leftarrow \text{Update}_I(M_I, F_I^{IVE}). \quad (14)$$

Experiment Strategy Evolution. To improve execution reliability, the EMA distills reusable execution strategies from the EA's code search trajectories. The EMA summarizes reusable experiment strategies from both the best codes and full trajectories in experiments:

$$F_E = \text{ESE}(P, \{H_E^s\}_{s=1}^4), \quad (15)$$

where $\text{ESE}(\cdot)$ is an LLM-based extractor. In this evolution, the EMA uses a prompt to jointly summarize (i) a **data processing strategy** and (ii) a **model training strategy**, and finally writes the resulting strategies into the experimentation memory:

$$M_E \leftarrow \text{Update}_E(M_E, F_E). \quad (16)$$

4 Experimental Setup

We consider the following three research questions: **(RQ1)** How effectively does EvoScientist generate high-quality scientific ideas in terms of novelty, feasibility, relevance, and clarity? **(RQ2)** How reliable is EvoScientist in generating and executing experimental code, as measured by execution success rates? **(RQ3)** Does the idea evolution mechanism contribute to idea quality improvement?

4.1 Baselines

To comprehensively evaluate EvoScientist, we compare it with representative open-source systems (i.e. Virtual Scientist (Su et al., 2025), AI-Researcher (Tang et al., 2025), InternAgent (InternAgent Team et al., 2025), and AI Scientist-v2 (Yamada et al., 2025)) and commercial systems (i.e. Hypogenic (Hypogenic, 2026), Novix (Novix, 2026), and K-Dense (K-Dense, 2026)). Detailed baseline descriptions are provided in Appendix B.

4.2 Evaluation Setup

To comprehensively evaluate EvoScientist, we construct a multi-level evaluation set spanning three tasks: idea generation, code implementation, and manuscript-level peer-review evaluation. For **idea generation**, we curate 30 research queries from experienced AI researchers, covering diverse contemporary AI topics. Idea quality is evaluated through pairwise comparisons by an LLM judge and expert human evaluators (Liu et al., 2025b; Su et al., 2025), with each idea scored on novelty, feasibility, relevance, and clarity, and aggregated into win/tie/lose outcomes. We use GPT-5.4 as the automatic judge to reduce self-preference bias (Wataoka et al., 2024), and follow prior LLM-as-a-judge practice (Zheng et al., 2023) by swapping candidate order to mitigate positional bias (Ko et al., 2020). For **code implementation**, each research proposal serves as the input, and performance is measured by execution success rate, defined as the proportion of trials in which the generated code runs successfully in a sandboxed environment and produces valid outputs (Lu et al., 2024; Yamada et al., 2025). For **manuscript-level peer-review evaluation**, we select 6 research ideas, develop them into full manuscripts, and submit them to ICAIS 2025 (Zhongguancun Academy, 2025) for peer review by human reviewers. Following prior work (Yamada et al., 2025), manuscript writing uses the same paper-writing module as AI Scientist-v2, as our evolution mechanism focuses on ideation and experiment execution rather than scientific writing. See Appendix A for dataset details, and Appendix D for evaluation rubrics.

4.3 Implementation Details

Our implementation uses the Semantic Scholar API [2] for literature retrieval, Gemini-2.5-Pro for scientific idea generation and manuscript author-

[2] https://api.semanticscholar.org/api-docs/

Method	Novelty			Feasibility			Relevance			Clarity			
	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	Avg. gap
Open-source Systems													
EvoScientist vs Virtual Scientist	**93.33***	3.34	3.33	**93.33***	6.67	0.00	**96.67***	3.33	0.00	**100.00***	0.00	0.00	**+95.00**
EvoScientist vs AI-Researcher	**93.33***	6.67	0.00	**83.33***	13.34	3.33	**90.00***	3.33	6.67	**90.00***	3.33	6.67	**+85.00**
EvoScientist vs InternAgent	**83.33***	16.67	0.00	**86.67***	10.00	3.33	**76.67***	20.00	3.33	**90.00***	6.67	3.33	**+81.67**
EvoScientist vs AI Scientist-v2	**50.00***	33.33	16.67	**66.67***	6.66	26.67	**30.00***	53.33	16.67	**43.33***	43.34	13.33	**+29.17**
Commercial Systems													
EvoScientist vs Hypogenic	**76.67***	13.33	10.00	**86.67***	6.66	6.67	**66.67***	26.66	6.67	**96.67***	0.00	3.33	**+75.00**
EvoScientist vs Novix	**86.67***	3.33	10.00	**50.00***	6.67	43.33	**50.00***	30.00	20.00	**66.67***	20.00	13.33	**+41.67**
EvoScientist vs K-Dense	**76.67***	10.00	13.33	**63.33***	0.00	36.67	**50.00***	30.00	20.00	**76.67***	10.00	13.33	**+45.83**

Table 1: Comparison of EvoScientist with baseline systems on scientific idea generation, evaluated by GPT-5.4. Scores marked with $*$ indicate that EvoScientist significantly outperforms the baseline ($p < 0.05$, sign. test).

Method	Novelty			Feasibility			Relevance			Clarity			
	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	Avg. gap
Open-source Systems													
EvoScientist vs InternAgent	**66.67***	23.33	10.00	**96.67***	3.33	0.00	**90.00***	0.00	10.00	**93.33***	6.67	0.00	**+81.67**
EvoScientist vs AI Scientist-v2	**73.33***	10.00	16.67	**50.00***	16.67	33.33	**43.33***	50.00	6.67	**53.33***	20.00	26.67	**+34.16**
Commercial Systems													
EvoScientist vs Novix	**93.33***	0.00	6.67	**56.67***	6.66	36.67	**36.67***	60.00	3.33	**73.33***	10.00	16.67	**+49.17**
EvoScientist vs K-Dense	**96.67***	3.33	0.00	**53.33***	26.67	20.00	**40.00***	43.33	16.67	**53.34***	43.33	3.33	**+50.84**

Table 2: Comparison of EvoScientist with baseline systems on idea generation, evaluated by human experts. Scores marked with $*$ indicate that EvoScientist significantly outperforms the baseline ($p < 0.05$, sign. test).

ing, Claude-4.5-Haiku for code generation, and the `mxbai-embed-large` embedding model (Lee et al., 2024) via Ollama (Ollama, 2026) for memory indexing and retrieval. For idea generation, we set the ideation retrieval top-k_I to 2, maintain at most $N_I = 21$ candidate ideas during idea tree search, and use 3 parallel workers. Each refinement trajectory starts from an initial idea and runs for at most two refinement rounds. Candidate ideas are ranked by an in-house Elo tournament with initial rating 1200, fixed $K = 32$, and full round-robin pairwise comparisons. For experiment execution, we set the experimentation retrieval top-k_E to 1, use 4 parallel workers, and allow 20, 12, 12, and 18 attempts for the four-stage execution pipeline, respectively. All experiments are conducted under consistent task settings and evaluation protocols across EvoScientist and baseline systems. Detailed implementation settings are provided in Appendix C.

5 Experimental Results

5.1 Idea Generation Performance (RQ1)

Automatic evaluation. Table 1 presents pairwise comparison results for scientific idea generation, evaluated by an LLM judge (GPT-5.4). EvoScientist is compared against both open-source and commercial AI scientist systems across four dimensions: **Novelty**, **Feasibility**, **Relevance**, and **Clarity**. These results demonstrate the effectiveness of EvoScientist's idea generation capability under automated evaluation, with three key observations:

- **EvoScientist achieves a poisitive average gap against all compared baselines.** Table 1 shows that the Avg. gap is positive across all comparisons, indicating an aggregate advantage across novelty, feasibility, relevance, and clarity. In particular, the Avg. gap ranges from +29.17 to +95.00 against open-source baselines and from +41.67 to +75.00 against commercial baselines.

- **EvoScientist achieves stronger novelty and feasibility than all compared baselines.** As shown in Table 1, EvoScientist wins more frequently than open-source and commercial baselines on both dimensions. This aligns with the memory-driven multi-agent evolution design: the EMA distills outcomes and failures into ideation memory, which the RA retrieves and incorporates into subsequent prompts, improving the novelty and feasibility of generated ideas over time.

- **EvoScientist improves relevance and clarity against all compared baselines via idea tree search and tournament selection.** Table 1 shows strong performance on relevance, with the largest gaps appearing in clarity. This pattern reflects two complementary design choices: the propose–review–refine idea tree search generates candidate ideas with explicit critique signals, and the subsequent Elo-based tournament ranks candidates along all four quality dimensions.

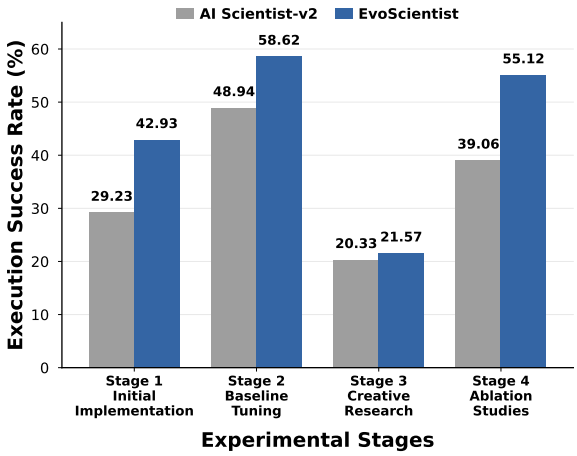

Figure 2: Mean execution success rate by experiment stage, comparing EvoScientist with AI Scientist-v2.

Human evaluation. Beyond LLM judgement, human evaluation offers a more reliable assessment of idea quality. Table 2 reports pairwise results from expert human judges, focusing on baselines that performed strongly under automated evaluation.

- **EvoScientist outperforms strong baselines on novelty and feasibility under human evaluation.** As shown in Table 2, expert judges prefer EvoScientist over both open-source and commercial baselines on novelty and feasibility, with win rates consistently exceeding lose rates. Averaged across four representative comparisons (InternAgent, AI Scientist-v2, Novix, and K-Dense), EvoScientist achieves a novelty win rate of 82.50% and a feasibility win rate of 64.17%.
- **EvoScientist remains competitive on relevance and shows stronger gains on clarity under human evaluation.** As shown in Table 2, relevance exhibits a higher tie rate, particularly against commercial baselines, suggesting that topical alignment can be subtle and evaluator-dependent. Nevertheless, EvoScientist's win rates on relevance consistently exceed its lose rates, while its clarity wins are more pronounced. This is consistent with the idea tree search and Elo-based tournament selection, which favor ideas with clearer presentation, making differences easier to assess even when relevance judgments converge.

5.2 Code Generation Performance (RQ2)

We evaluate code generation using the execution success rate, comparing EvoScientist with AI Scientist-v2 in Figure 2. We treat execution success rate as a *reliability indicator*: it captures whether generated code runs and produces valid outputs, but it does not by itself certify that the executed code semantically implements the intended method. We discuss this scope in Limitations and treat it as a deliberate next step rather than a current claim. Two observations stand out:

- **EvoScientist achieves higher execution success rates than AI Scientist-v2 across all four experiment stages.** Averaged across stages, EvoScientist achieves an execution success rate of 44.56%, compared with 34.39% for AI Scientist-v2, corresponding to a 10.17-point absolute improvement. This gain is consistent with EvoScientist's experiment strategy evolution: the EMA distills outcomes and failures from code-generation trajectories into the experimentation memory, which the engineer agent retrieves and incorporates into subsequent prompts to produce more robust and consistently reliable implementations.
- **EvoScientist provides broad gains, especially in stages with transferable execution patterns.** EvoScientist improves over AI Scientist-v2 by 13.70 points in Stage 1, 9.68 points in Stage 2, 1.24 points in Stage 3, and 16.06 points in Stage 4. The largest gains appear in implementation setup, baseline tuning, and ablation studies, where successful execution patterns transfer more directly across proposals. Stage 3 also shows a measurable improvement despite requiring proposal-specific creative method implementation, suggesting that experimentation memory can still provide useful guidance in the most customized stage of the research workflow.

5.3 Ablation Studies (RQ3)

Table 3 compares EvoScientist with several ablative variants: (i) **-IDE**: we remove idea direction evolution, (ii) **-IVE**: we remove idea validation evolution, and (iii) **-all**: we remove all idea evolution. We highlight three findings:

- **Removing idea direction evolution reduces both novelty and feasibility.** The **-IDE** variant loses to EvoScientist more frequently on novelty (Lose: 76.67%) and feasibility (Lose: 53.33%). This suggests that evolving and reusing direction-level insights helps the RA generate ideas that are both more original and more practically grounded across both evaluation criteria.
- **Removing idea validation evolution disproportionately harms feasibility.** The **-IVE** variant loses to EvoScientist on feasibility in 63.33% of comparisons, showcasing a larger degradation than on other dimensions. This suggests that

Method	Novelty			Feasibility			Relevance			Clarity			
	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	Avg. gap
-IDE vs EvoScientist	10.00	13.33	76.67	20.00	26.67	53.33	20.00	43.33	36.67	20.00	46.67	33.33	-32.50
-IVE vs EvoScientist	43.33	13.34	43.33	10.00	26.67	63.33	26.67	40.00	33.33	13.33	50.00	36.67	-20.83
-all vs EvoScientist	10.00	10.00	80.00	0.00	16.67	83.33	20.00	43.33	36.67	26.67	36.66	36.67	-45.00

Table 3: Ablation study on scientific idea generation, evaluated by GPT-5.4.

validation-driven evolution is essential for filtering out experimentally infeasible directions.

- **Removing all idea evolution causes substantial drops in novelty and feasibility, but smaller changes in relevance and clarity.** The **-all** variant loses heavily on novelty (Lose: 80.00%) and feasibility (Lose: 83.33%). In contrast, differences are less pronounced on relevance and clarity, where many comparisons result in ties. This pattern indicates that idea evolution has its strongest effect on novelty and feasibility, with smaller effects on surface-level relevance and linguistic clarity in generated research ideas.

5.4 Manuscript-Level Peer-Review Case Study

To evaluate EvoScientist under an externally graded manuscript-level protocol that does not depend on our own evaluation pipeline, we deployed EvoScientist to generate six complete research papers, which were subsequently submitted to the ICAIS 2025 (Zhongguancun Academy, 2025). The venue reviewed submissions under its standard protocol, independent of the evaluation procedures used elsewhere in this paper. We chose an external venue rather than rerunning our own LLM judge on full manuscripts so that the headline "acceptance" signal is not produced by the same evaluation infrastructure that grades our idea-quality results. Although ICAIS 2025 is a first-edition venue, this provides a third line of evidence decoupled from the LLM and human evaluations in §5.1. The AI Scientist Track received 82 submissions and accepted 26 papers, corresponding to an acceptance rate of 31.71%. All six papers generated by EvoScientist were accepted, as shown in Table 4.

Beyond review scores, we analyze peer review feedback to characterize EvoScientist's behavior under peer review. We synthesize meta-reviews from all six papers, revealing a consistent profile of strengths and limitations that align with our memory-driven multi-agent evolution framework.

- **EvoScientist demonstrates strength in methodological novelty.** Reviewers consistently highlighted the novelty of EvoScientist's submissions, along with their relevance and clarity. This reflects the RA's use of ideation memory, where strategies distilled from prior outcomes and failures are retrieved and incorporated into subsequent high-quality research proposals.
- **EvoScientist delivers robust experimental validation and execution.** Four of the six papers received explicit praise for their "comprehensive and sound experimental design" or "solid empirical evidence". This reflects the EA's use of experimentation memory, which stores reusable execution lessons, such as debugging patterns and validated implementations, and enables their systematic reuse across tasks.
- **EvoScientist can further benefit from stronger theoretical analysis and formal grounding.** A recurring critique concerned the lack of deeper theoretical formalization beyond empirical results. Because EvoScientist prioritizes testable proposals and experimental evidence through outcome-driven evolution, the system does not consistently abstract findings into formal theoretical frameworks. This defines a clear handover point: EvoScientist delivers the empirical "what", while the theoretical "why" remains a direction for human researchers to investigate further.

See Appendix E for all case studies and feedback.

6 Conclusions

In this paper, we introduced EvoScientist, a self-evolving multi-agent framework that coordinates researcher, engineer, and evolution manager agents with persistent ideation and experimentation memories. Experiments on idea generation, code generation, and manuscript-level peer-review evaluation across diverse computational research settings show that EvoScientist improves idea quality and execution reliability compared with open-source and commercial baselines. Together, these findings suggest that stage-aligned, memory-driven co-evolution offers a practical path toward AI scientist systems that compound learning across tasks while remaining grounded in verifiable evidence rather than treating each research run as an isolated trial.

Limitations

EvoScientist is mainly evaluated on computational research tasks, where hypotheses can be tested through simulation, code execution, and peer-review-style assessment. This scope is consistent with recent AI scientist systems and surveys that emphasize verifiable computational workflows as an important first step toward automated scientific discovery (Lu et al., 2024; Yamada et al., 2025; Schmidgall et al., 2025; Chen et al., 2025). In future work, we plan to extend EvoScientist to broader scientific domains, especially those that require physical experimentation, such as materials science and drug discovery, by incorporating laboratory constraints, safety requirements, and domain-specific measurement protocols beyond code execution.

Ethical Considerations

EvoScientist is designed to assist human-led scientific discovery rather than replace expert judgment. Autonomous research agents can increase productivity, but they also raise concerns about trust calibration, accountability, and over-reliance on automated outputs (Dzindolet et al., 2003; Gottweis et al., 2025; Shao et al., 2025). Therefore, EvoScientist outputs should be treated as research candidates: domain experts should verify literature claims, experimental protocols, code, data use, and conclusions before dissemination. Because LLM-based systems can inherit corpus-level and instruction-induced biases (Schramowski et al., 2022; Itzhak et al., 2024; Schick et al., 2021), deployments should include provenance logging, memory inspection, and safeguards against storing private data, unsafe procedural details, or unverified claims as reusable knowledge. These requirements do not reduce the empirical value of EvoScientist; rather, they clarify the intended use setting in which the system accelerates ideation and experimentation while human researchers remain responsible for validation, attribution, and final scientific judgment.

References

Mohammad Almansoori, Komal Kumar, and Hisham Cholakkal. 2025. Medagentsim: Self-evolving multi-agent simulations for realistic clinical interactions. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 362–372. Springer.

Ibrahim Al Azher, Miftahul Jannat Mokarrama, Zhishuai Guo, Sagnik Ray Choudhury, and Hamed Alhoori. 2025. Futuregen: Llm-rag approach to generate the future work of scientific article. *arXiv preprint arXiv:2503.16561*.

Guoxin Chen, Jie Chen, Lei Chen, Jiale Zhao, Fanzhe Meng, Wayne Xin Zhao, Ruihua Song, Cheng Chen, Ji-Rong Wen, and Kai Jia. 2026. Toward autonomous long-horizon engineering for ml research. *arXiv preprint arXiv:2604.13018*.

Qiguang Chen, Mingda Yang, Libo Qin, Jinhao Liu, Zheng Yan, Jiannan Guan, Dengyun Peng, Yiyan Ji, Hanjing Li, Mengkang Hu, and 1 others. 2025. Ai4research: A survey of artificial intelligence for scientific research. *arXiv preprint arXiv:2507.01903*.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. In *Proceedings of the 28th European Conference on Artificial Intelligence (ECAI)*.

Mary T. Dzindolet, Scott A. Peterson, Regina A. Pomranky, Linda G. Pierce, and Hall P. Beck. 2003. The role of trust in automation reliance. *International Journal of Human-Computer Studies*, 58(6):697–718.

Arpad E. Elo. 1978. *The Rating of Chessplayers, Past and Present*. Arco Pub., New York.

Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, and 1 others. 2025. A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems. *arXiv preprint arXiv:2508.07407*.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, and 1 others. 2025a. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*.

Xian Gao, Zongyun Zhang, Mingye Xie, Ting Liu, and Yuzhuo Fu. 2025b. Graph of ai ideas: Leveraging knowledge graphs and llms for ai research idea generation. *arXiv preprint arXiv:2503.08549*.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, and 1 others. 2025. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*.

Xiang Hu, Hongyu Fu, Jinge Wang, Yifeng Wang, Zhikun Li, Renjun Xu, Yu Lu, Yaochu Jin, Lili Pan, and Zhenzhong Lan. 2025. NOVA: An iterative planning framework for enhancing scientific innovation with large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21330–21359, Vienna, Austria. Association for Computational Linguistics.

Hypogenic. 2026. Hypogenic: An AI research acceleration platform. https://hypogenic.ai/. Accessed: 2026-05-08.

InternAgent Team, Bo Zhang, Shiyang Feng, Xiangchao Yan, Jiakang Yuan, Zhiyin Yu, Xiaohan He, S ongtao Huang, Shaowei Hou, Zheng Nie, Zhilong Wang, Jinyao Liu, Runmin Ma, Tianshuo Peng, Peng Ye, Shufei Zhou, Dongzhan an d Zhang, Xiaosong Wang, Yilan Zhang, Meng Li, and 5 others. 2025. InternAgent: When agent becomes the scientist—building closed-loop system from hypothesis to verification. *arXiv preprint arXiv:2505.16938*. Originally released as "NovelSeek".

Intology AI. 2025. Zochi technical report. Technical Report arXiv:2503.10617, Intology.

Itay Itzhak, Gabriel Stanovsky, Nir Rosenfeld, and Yonatan Belinkov. 2024. Instructed to bias: Instruction-tuned language models exhibit emergent cognitive bias. *TACL*, 12:771–785.

K-Dense. 2026. K-dense: An agentic AI platform for scientific research. https://www.k-dense.ai/. Accessed: 2026-05-08.

David Klahr and Herbert A Simon. 1999. Studies of scientific discovery: Complementary approaches and convergent findings. *Psychological Bulletin*, 125(5):524.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. In *Proceedings of EMNLP*, pages 1109–1121.

Pat Langley. 1987. *Scientific discovery: Computational explorations of the creative processes*. MIT press.

Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. 2024. Open source strikes bread — new fluffy embeddings model. https://www.mixedbread.ai/blog/mxbai-embed-large-v1. Mixedbread.ai blog post; accessed: 2026-05-08.

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

Ben Liu, Jihai Zhang, Fangquan Lin, Xu Jia, and Min Peng. 2025a. One size doesn't fit all: A personalized conversational tutoring agent for mathematics instruction. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2401–2410.

Jiaqi Liu, Shi Qiu, Mairui Li, Bingzhou Li, Haonian Ji, Siwei Han, Xinyu Ye, Peng Xia, Zihan Dong, Congyu Zhang, and 1 others. 2026. Autoresearchclaw: Self-reinforcing autonomous research with human-ai collaboration. *arXiv preprint arXiv:2605.20025*.

Yujie Liu, Zonglin Yang, Tong Xie, Jinjie Ni, Ben Gao, Yuqiang Li, Shixiang Tang, Wanli Ouyang, Erik Cambria, and Dongzhan Zhou. 2025b. Researchbench: Benchmarking llms in scientific discovery via inspiration-based task decomposition. *arXiv preprint arXiv:2503.21248*.

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*.

Novix. 2026. Novix: The AI agentic co-scientist for scientific research. https://novix.science/. Accessed: 2026-05-08.

Ollama. 2026. Ollama: Get up and running with large language models locally. https://ollama.com/. Accessed: 2026-05-08.

Charles O'Neill, Tirthankar Ghosal, Roberta Răileanu, Mike Walmsley, Thang Bui, Kevin Schawinski, and Ioana Ciucă. 2025. Sparks of science: Hypothesis generation using structured paper data. *arXiv preprint arXiv:2504.12976*.

Karl Popper. 2005. *The logic of scientific discovery*. Routledge.

Biqing Qi, Kaiyan Zhang, Kai Tian, Haoxiang Li, Zhang-Ren Chen, Sihang Zeng, Ermo Hua, Hu Jinfang, and Bowen Zhou. 2024. Large language models as biomedical hypothesis generators: a comprehensive evaluation. *arXiv preprint arXiv:2407.08940*.

Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, and 1 others. 2025a. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*.

Yansheng Qiu, Haoquan Zhang, Zhaopan Xu, Ming Li, Diping Song, Zheng Wang, and Kaipeng Zhang. 2025b. Ai idea bench 2025: Ai research idea generation benchmark. *arXiv preprint arXiv:2504.14191*.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2025. From exploration to mastery: Enabling llms to master tools via self-driven interactions. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*. Oral.

Maxime Robeyns, Martin Szummer, and Laurence Aitchison. 2025. A self-improving coding agent. *arXiv preprint arXiv:2504.15228*.

Aishik Sanyal, Samuel Schapiro, Sumuk Shashidhar, Royce Moon, Lav R. Varshney, and Dilek Hakkani-Tur. 2025. Spark: A system for scientifically creative idea generation. In *Proceedings of the 16th International Conference on Computational Creativity (ICCC)*.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Trans. Assoc. Comput. Linguistics*, 9:1408–1424.

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

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Moor, Michael a nd Liu, and Emad Barsoum. 2025. Agent laboratory: Using LLM agents as research assistants. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5977–6043.

Patrick Schramowski, Cigdem Turan, Nico Andersen, Constantin A Rothkopf, and Kristian Kersting. 2022. Large pre-trained language models contain human-like biases of what is right and wrong to do. *Nature Machine Intelligence*, 4(3):258–268.

Chenyang Shao, Dehao Huang, Yu Li, Keyu Zhao, Weiquan Lin, Yining Zhang, Qingbin Zeng, Zhiyu Chen, Tianxing Li, Yifei Huang, and 1 others. 2025. Omnisciently: Toward a co-evolving ecosystem of human and ai scientists. *arXiv preprint arXiv:2511.16931*.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 8634–8652.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2025. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, and 1 others. 2025. Paperbench: Evaluating ai's ability to replicate ai research. *arXiv preprint arXiv:2504.01848*.

Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, and 1 others. 2025. Many heads are better than one: Improved scientific idea generation by a llm-based multi-agent system. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28201–28240.

Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. 2025. AI-Researcher: Autonomous scientific innovation. In *Advances in Neural Information Processing Systems (NeurIPS)*. Spotlight.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024a. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research (TMLR)*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Li u, Tianyu Liu, and Zhifang Sui. 2024b. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2024c. SciMON: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–299, Bangkok, Thailand. Association for Computational Linguistics.

Yanlin Wang, Yanli Wang, Daya Guo, Jiachi Chen, Ruikai Zhang, Yuchi Ma, and Zibin Zheng. 2025a. Rlcoder: Reinforcement learning for repository-level code completion. In *Proceedings of the 47th IEEE/ACM International Conference on Software Engineering (ICSE)*.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2025b. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. Self-preference bias in llm-as-a-judge. *CoRR*, abs/2410.21819.

Yixuan Weng, Minjun Zhu, Qiujie Xie, Qiyao Sun, Zhen Lin, Sifan Liu, and Yue Zhang. 2025. Deepscientist: Advancing frontier-pushing scientific findings progressively. *arXiv preprint arXiv:2509.26603*.

Tianze Xu, Pengrui Lu, Lyumanshan Ye, Xiangkun Hu, and Pengfei Liu. 2025. Researcherbench: Evaluating deep ai research systems on the frontiers of scientific inquiry. *arXiv preprint arXiv:2507.16280*.

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*.

Ruofeng Yang, Yongcan Li, and Shuai Li. 2026. Aris: Autonomous research via adversarialmulti-agent collaboration. *arXiv preprint arXiv:2605.03042*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 15476–15488.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. 2025. Darwin godel machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22954*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings*

of the AAAI Conference on Artificial Intelligence, volume 38, pages 19632–19642.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023) Datasets and Benchmarks Track*.

Zhongguancun Academy. 2025. The 1st international conference on AI scientists (ICAIS 2025). https://icais.ai/. Accessed: 2026-05-08.

Xinyu Zhu, Yuzhu Cai, Zexi Liu, Bingyang Zheng, Cheng Wang, Rui Ye, Jiaao Chen, Hanrui Wang, Wei-Chen Wang, Yuzhi Zhang, and 1 others. 2026. Toward ultra-long-horizon agentic science: Cognitive accumulation for machine learning engineering. *arXiv preprint arXiv:2601.10402*.

Appendix Contents

A Research Queries **13**

B Baseline Systems **13**

C Implementation Details **13**
C.1 System Configuration 13
C.2 Hyperparameters 13
C.3 Prompts for Self-Evolution Operators 15

D Evaluation Protocol **15**
D.1 Pairwise LLM Judge 15
D.2 Human Evaluation 15
D.3 LLM-Human Agreement 15

E Case Studies **15**
E.1 End-to-End Submission Overview 15
E.2 Adaptive Evidential Meta-Learning 15
E.3 Hierarchical Adaptive Normalization 18

Appendix

A Research Queries

We curate a set of 30 research queries solicited from experienced AI researchers. Before evaluation, we rewrite each query into a unified template for consistency. During evaluation, each query string is used verbatim as the user goal G.

The queries span many AI research areas (machine translation, speech recognition, software engineering, healthcare agents, literature review automation, text-to-SQL, information extraction, retrieval-augmented generation, multimodal architectures, model efficiency & deployment, data synthesis, safety, alignment); we list them in Figure 3.

B Baseline Systems

We compare EvoScientist with both open-source and commercial systems that represent strong baselines for autonomous scientific discovery. Where applicable, baseline methods are configured following the settings reported in their original papers or official documentation.

Open-source systems

- **Virtual Scientist** (Su et al., 2025) is an large language model (LLM)-based multi-agent system that simulates collaborative scientific ideation through proposal, critique, and refinement, representing a strong baseline for idea generation.
- **AI-Researcher** (Tang et al., 2025) is a fully autonomous research system that orchestrates the complete research pipeline, from literature review and hypothesis generation to experiment implementation and manuscript preparation, with minimal human intervention.
- **InternAgent** (InternAgent Team et al., 2025) is a closed-loop multi-agent framework for autonomous scientific research, emphasizing scalability, interactivity, and human-in-the-loop extensibility.
- **AI Scientist-v2** (Yamada et al., 2025) is an end-to-end agentic system that employs a progressive agentic tree-search methodology to autonomously generate hypotheses, design experiments, analyze data, and produce scientific manuscripts.

Commercial systems

- **Hypogenic** (Hypogenic, 2026) is a community-driven AI research acceleration platform that utilizes AI agents to assist in interdisciplinary scientific exploration, featuring a weekly competition where winning ideas are implemented by AI research agents.
- **Novix** (Novix, 2026) is an AI Agentic Co-scientist designed to accelerate the full research lifecycle, including idea generation, literature review, experimental design, and data analysis.
- **K-Dense** (K-Dense, 2026) is an agentic AI platform positioned as an intelligent task executor, supporting end-to-end automation from data processing to decision-making in multiple domains.

C Implementation Details

C.1 System Configuration

For the initial literature review phase, EvoScientist uses the Semantic Scholar API to retrieve relevant papers. Scientific idea generation is performed using Gemini-2.5-Pro, code generation is performed using Claude-4.5-Haiku, and end-to-end manuscript authoring is handled by Gemini-2.5-Pro. For memory indexing and retrieval, we use the `mxbai-embed-large` embedding model (Lee et al., 2024) via Ollama (Ollama, 2026).

C.2 Hyperparameters

For idea generation, we set the ideation retrieval top-k_I to 2 and use a maximum of $N_I = 21$ candidate ideas during idea tree search, with 3 parallel workers. Each refinement trajectory starts from an initial idea and runs for at most two refinement rounds, terminating early if idea generation or evaluation fails. Candidate selection uses an in-house Elo tournament over the generated ideas. All candidates start with rating 1200, and pairwise matches are generated as a full round-robin over all candidate pairs. Each pair is judged once by the LLM with a forced A/B winner prompt. We use a fixed Elo K-factor of 32 and the standard expected-score formula

$$E_A = \frac{1}{1 + 10^{(r_B - r_A)/400}}, \tag{17}$$

where r_A and r_B are the current ratings of ideas A and B. Ratings are updated with $r'_A = \mathrm{round}(r_A + 32(s_A - E_A))$ and analogously for B, where $s_A = 1$ if A wins and $s_A = 0$ otherwise. If a judge response is unparsable or does not specify either A or B as the winner, we treat the match as a draw for the rating update with $s_A = s_B = 0.5$. The

ID	Domain	Query
1	Machine translation	Generate a research proposal on low-resource machine translation.
2	Software engineering	Help me generate a research proposal on AI for software engineering.
3	LLM evaluation	Generate a research proposal on how to address the accuracy issues of automated evaluation for Large Language Models.
4	Healthcare agents	Help me generate a research proposal on AI agents in healthcare.
5	Literature review automation	Generate a research proposal on automated literature review generation.
6	Speech recognition	Generate a research proposal on low-resource speech recognition.
7	Model efficiency	Generate a research proposal on how to break the performance ceiling of small models.
8	AI agents	Generate a research proposal on complex reasoning for AI agents.
9	Model deployment	Generate a research proposal on solving quantization challenges for LLM deployment on edge devices.
10	Text-to-SQL	Help me generate a research proposal on Text-to-SQL.
11	LLM capabilities	Generate a research proposal on the decoupling of Large Language Model capabilities.
12	Factual consistency	Generate a research proposal on how to resolve factual consistency issues in Large Language Models.
13	Data synthesis	Generate a research proposal on data synthesis for code LLMs.
14	Inference efficiency	Generate a research proposal on solving inference latency issues in Large Language Models.
15	Machine translation	Generate a research proposal on debiasing in machine translation.
16	Knowledge injection	Generate a research proposal on addressing the challenges of knowledge injection in domain-specific Large Language Models.
17	Information extraction	Generate a research proposal on automated information extraction.
18	UX evaluation	Generate a research proposal on user experience evaluation for Large Language Models.
19	Evaluation leaderboards	Generate a research proposal on how to construct efficient leaderboards for LLM evaluation.
20	Text generation	Generate a research proposal on the evaluation of diversity in text generation.
21	Content detection	Generate a research proposal on how to address false positives in AI-generated content detection.
22	Code LLM security	Generate a research proposal on the security of code LLMs.
23	Code LLM evaluation	Generate a research proposal on how to build a deep evaluation framework for code LLMs.
24	RAG	Generate a research proposal on solving multi-hop reasoning challenges in Retrieval-Augmented Generation (RAG).
25	Multi-source reasoning	Generate a research proposal on complex reasoning with multi-source information.
26	Data filtering	Generate a research proposal on solving data filtering challenges for domain-specific models.
27	Long-context understanding	Generate a research proposal on how to enhance the long-context understanding of Large Language Models.
28	Alignment	Generate a research proposal on cross-cultural alignment for Large Language Models.
29	Alignment	Generate a research proposal on value alignment for AI agents.
30	Audio foundation models	Help me generate a research proposal on general-purpose audio foundation models.

Figure 3: Research queries used for evaluation.

tournament stops after all pairwise matches have been completed.

For experiment execution, we set the experimentation retrieval top-k_E to 1 and use 4 parallel workers. The maximum numbers of attempts at stages $s \in \{1, 2, 3, 4\}$ are $N_E^1 = 20$, $N_E^2 = 12$, $N_E^3 = 12$, and $N_E^4 = 18$.

C.3 Prompts for Self-Evolution Operators

We list the full prompts for the three self-evolution operators introduced in §3.5: idea generation evolution (Figure 4), idea validation evolution (Figure 5), and experiment strategy evolution (Figure 6).

D Evaluation Protocol

D.1 Pairwise LLM Judge

This subsection reports the prompt used to obtain pairwise LLM judgments for evaluating idea-generation outputs with *GPT-5.4*. Figures 7 – 9 presents the full prompt template adapted from Zheng et al. (2023), which assesses four dimensions: novelty, feasibility, relevance, and clarity. To mitigate positional bias (Ko et al., 2020; Wang et al., 2024b), we evaluate each answer pair twice by swapping their order in two independent runs.

D.2 Human Evaluation

For human evaluation, we recruited 3 annotators with PhD degrees in relevant AI domains to do human evaluation. To preserve the integrity of the comparison, annotators were blinded to system identities: each candidate pair was presented in randomized order without any indication of which system produced which idea, and the four-dimension rubric was applied independently per pair. Annotators were allowed to use the internet to verify literature-related claims when needed. The full instructions are shown in Figure 10.

D.3 LLM-Human Agreement

To validate the reliability of our LLM-based evaluation, we conducted a human evaluation study on a subset of 120 idea pairs (30 pairs × 4 comparison groups). Three expert annotators independently assessed each pair across four dimensions: Clarity, Novelty, Feasibility, and Relevance. We then computed the agreement rate between human judgments and LLM judgments. The results demonstrate strong alignment between LLM and human evaluators. Overall judgment agreement reached 90.0% (108/120). Among individual dimensions, Clarity achieved the highest agreement rate of 90.8% (109/120), followed by Novelty at 88.3% (106/120), Relevance at 84.2% (101/120), and Feasibility at 83.3% (100/120). The average agreement across all dimensions was 87.3% (524/600). These results are consistent with prior work (Qiu et al., 2025b; Starace et al., 2025; Xu et al., 2025) showing that well-prompted LLM judges can achieve agreement rates comparable to human-human agreement (typically 80-85%), validating the effectiveness of our automated evaluation framework.

E Case Studies

E.1 End-to-End Submission Overview

Table 4 summarizes the six accepted EvoScientist-generated papers at ICAIS 2025 (Zhongguancun Academy, 2025) (AI Scientist Track). For two of these papers we provide a deeper qualitative analysis below: an instance that primarily exercises *idea direction evolution* (§E.2) and one that primarily exercises *experiment strategy evolution* (§E.3).

E.2 Adaptive Evidential Meta-Learning

Design callback This case highlights the effectiveness of **idea direction evolution** within EvoScientist. The researcher agent repeatedly retrieved direction-level insights from **ideation memory**, enabling iterative refinement of a clinically meaningful problem formulation that balances personalization and uncertainty calibration. Reviewers emphasized the coherence and deployability of the proposed architecture, which aligns with EvoScientist's ability to reuse high-level design patterns distilled from prior outcomes rather than relying on isolated ideation.

Outcome signal Meta-review feedback indicates that the paper's primary strengths lie in the validity of its core contribution and its balance between methodological novelty and engineering practicality. At the same time, reviewer requests focused on clearer formalization, metric specification, and reproducibility details. This pattern is consistent with EvoScientist's design emphasis on generating empirically grounded and testable research artifacts, while leaving deeper theoretical formalization and documentation refinement as natural handover points for subsequent human-led iteration.

 # Role
You are an expert Machine Learning Research Engineer specialized in scientific idea summarization and technical writing for persistent memory.

 # Task
I will provide you with a user goal, a set of top-ranked candidate ideas (with ratings and refinement feedback), and optional task outcomes. Your goal is to distill reusable, goal-relevant **promising research directions** into a compact memory item that can be retrieved in future tasks.

 # Input Data
- **User Goal:** `{user_goal}`
- **Top-Ranked Ideas:** `{top_ranked_ideas}`
- Each idea may include: method sketch, experimental plan, key assumptions, and reviewer feedback.

 # Instructions
1. **DIRECTION SUMMARY**: Extract promising directions from top ranked ideas. For each direction, include:
   - A clear direction title (one line).
   - The core mechanism or hypothesis.
   - Why it is promising for user goal (novelty/feasibility/relevance/clarity).
   - Key requirements and assumptions (data, compute, tools, environment).
   - A minimal experimental plan that could validate the direction (2–4 steps).
2. **DEDUPLICATION**: If multiple ideas express the same direction, merge them and retain only the most actionable formulation.

 # Constraints
- **Be Precise**: Do not omit concrete assumptions, evaluation signals, or experimental steps.
- **Be Concise**: Use technical language. Avoid conversational filler.
- **Actionable Memory**: Write so a future researcher agent can directly reuse the direction.

 # Output Format
```
DIRECTION SUMMARY:
- Direction 1:  [Title]
  - Core idea:
  - Why promising for user goal:
  - Requirements/assumptions:
  - Minimal validation plan:
- Direction 2:  ...
```

Figure 4: Prompt for idea direction evolution (IDE).

Title	Result
Adaptive Evidential Meta-Learning with Hyper-Conditioned Priors for Calibrated ECG Personalisation	Accepted
Hierarchical Change Signature Analysis: A Framework for Online Discrimination of Incipient Faults and Benign Drifts in Industrial Time Series	Accepted
Robust Zero-Shot NER for Crises via Iterative Knowledge Distillation and Confidence-Gated Induction	Accepted
Adaptive Log Anomaly Detection through Data–Centric Drift Characterization and Policy-Driven Lifelong Learning	Accepted
ConFIT: A Robust Knowledge-Guided Contrastive Framework for Financial Extraction	Accepted
Hierarchical Adaptive Normalization: A Placement-Conditioned Cascade for Robust Wearable Activity Recognition	Accepted

Table 4: End-to-end scientific discovery performance at ICAIS 2025 (AI Scientist Track). All six EvoScientist-generated submissions were accepted; the track's overall acceptance rate was 31.71% (26/82).

 # Role
You are an expert Machine Learning Research Engineer specialized in scientific evaluation, failure analysis, and memory writing for self-improving agentic systems.

 # Task
I will provide you with a selected research proposal and its execution report produced by running experiments. Your goal is to determine whether the proposal should be recorded as a failed direction, and to extract reusable failure signals and validation conclusions as a memory item.

 # Input Data
- **Selected Proposal:** `{research_proposal}`
  - Includes: background, method, experimental plan, expected results.
- **Execution Report:** `{execution_report}` — Includes: run status, logs, metrics, environment notes, and failure diagnoses.

 # Instructions
1. **FAILURE CASE CHECK (RULE-BASED)**:
   - If `{execution_report_W}` indicates the engineer could not find any executable code within the pre-defined budget/rules, mark the proposal as **FAILED (NoExecutableWithinBudget)**.
2. **PERFORMANCE VALIDATION (RESULT-BASED)**:
   - If experiments completed, compare the proposed method against baselines using the metrics in execution report.
   - If the proposed method performs worse than the baseline(s) under the stated evaluation, mark as **FAILED (WorseThanBaseline)**.
   - Otherwise, mark as **NOT_FAILED (ValidatedOrInconclusive)**.
3. **FAILURE SIGNAL EXTRACTION** (always do this when FAILED, optional otherwise):
   - Summarize the primary failure reason(s) (implementation, environment, data, training instability, evaluation mismatch).
   - List concrete evidence from logs/metrics (error messages, metric deltas, resource limits, reproducibility issues).
   - Identify what specifically in research proposal likely caused the failure (assumption, method choice, missing detail).
4. **REUSABLE COUNTERMEASURES**:
   - Provide 3–6 actionable countermeasures that would prevent the same failure pattern in future proposals or executions.
5. **MEMORY WRITING**:
   - Write a compact memory item that can update ideation memory as an unsuccessful direction summary.

 # Constraints
- **Be Precise**: Quote or reference exact metric names, thresholds, and failure evidence.
- **Be Concise**: Use technical language. Avoid conversational filler.
- **Decision + Evidence**: The final judgment must be supported by explicit evidence.

 # Output Format
```
- Title:
- Summary:
- Trigger conditions:
- Do-not-repeat notes:
- Retrieval tags:
```

Figure 5: Prompt for idea validation evolution (IVE).

 # Role
You are an expert Machine Learning Research Engineer specialized in reproducibility and technical documentation.

 # Task
I will provide you with a research task description, a set of exploration trajectories, and the final high-performance code for the task. Your goal is to extract and summarize the technical essence of the winning implementation so that another engineer could reconstruct the pipeline with high fidelity.

 # Input Data
- **Research Task Description:** `{research_proposal}`
- **Trajectories & Final Code:** `{trajectories}`

 # Instructions
1. **DATA SUMMARY**: Focus on the pipeline from raw input to tensor. Include:
   - Data loading libraries and custom classes.
   - Exact preprocessing steps (scaling, normalization, encoding).
   - Data augmentation techniques and their specific parameters.
   - Train/Val/Test split ratios and sampling strategies.
2. **MODEL SUMMARY**: Focus on the architecture and the optimization loop. Include:
   - Specific model backbone and modifications.
   - All critical hyperparameters (learning rate, batch size, weight decay, dropout).
   - Loss functions, optimizers (e.g., AdamW, SGD), and learning rate schedulers.
   - Training duration (epochs/steps) and convergence criteria.

 # Constraints
- **Be Precise**: Do NOT omit numerical parameters or specific library functions.
- **Be Concise**: Use technical language. Avoid conversational filler.
- **Reproducibility**: Ensure the summary is "actionable" for a code engineer.

 # Output Format
```
DATA SUMMARY:
[Your detailed summary here]

MODEL SUMMARY:
[Your detailed summary here]
```

Figure 6: Prompt for experiment strategy evolution (ESE).

E.3 Hierarchical Adaptive Normalization

Design callback This case illustrates the role of **experiment memory** in stabilizing complex experimental pipelines. Execution failures and configuration-level issues encountered during early experimentation were summarized and reused, enabling the engineer agent to converge toward a robust, deployment-oriented implementation. Reviewer praise for comprehensive empirical coverage and low runtime overhead is consistent with EvoScientist's capacity to accumulate and reuse execution-level lessons rather than repeatedly rediscovering them.

Outcome signal Reviewer feedback also identified several internal-consistency and protocol-level issues, including ambiguities in stability gating, metric reporting, and baseline fairness. These critiques underscore the importance of strict consistency auditing and reproducibility-complete reporting in end-to-end scientific discovery. In this sense, the review outcome reflects both the strengths and current limits of EvoScientist: it is effective at producing empirically strong and practically relevant systems, while rigorous protocol alignment and documentation remain critical areas for further refinement.

Synthesis Taken together, these two cases suggest that EvoScientist's evolution mechanisms can generate high-value ideas and robust empirical pipelines that align with expert evaluation criteria. At the same time, they underscore that sustained reviewer confidence depends on internal consistency, protocol-faithful baseline design, and reproducibility-complete reporting, pointing to natural directions for further system-level refinement.

[System prompt]
You are an AI analysis engine specializing in the comparative evaluation of two technical ideas. Your task is to conduct an objective, in-depth, and multi-dimensional comparison based on the provided context, which includes a **user's research goal** and **two competing AI-generated ideas**.

[User prompt]
```
[Research Goal]
{question}
[The End of Research Goal]
[The Start of Assistant 1's Idea]
{answer_a}
[The End of Assistant 1's Idea]
[The Start of Assistant 2's Idea]
{answer_b}
[The End of Assistant 2's Idea]
```

Core Task & Evaluation Dimensions
We would like you to evaluate the performance of the two AI assistants. Please rate each idea on a scale of 1 to 10 for each of the four core dimensions: **Clarity, Novelty, Feasibility, and Relevance**.

Evaluation Dimensions & Scoring Rubric
Please strictly adhere to the following detailed 1-10 scoring rubric for each dimension.

Clarity Scoring Rubric (1-10 Scale)
- **10 (Crystal Clear & Actionable):** The idea is articulated with exceptional precision. The problem, solution, methodology, and expected outcomes are immediately understandable without any ambiguity. The logic is flawless and presented in a way that is directly actionable, *often providing concrete implementation paths, and complex technical terms are either avoided or clearly defined*.
- **9 (Logically Rigorous):** The idea is articulated with high precision. Core concepts and execution steps are clear. The internal logic is highly coherent.
- **8 (Well-Defined):** The main idea is clear and the overall logic is sound. Some minor implementation details might require clarification, but the core proposal is solid.
- **7 (Mostly Clear):** The overall idea is understandable, but several components lack detailed explanation, *or rely on specialized jargon without adequate definition*, requiring the reader to make some assumptions.
- **6 (Vague Core):** The general direction is understandable, but the core mechanism or key components of the idea are vaguely defined, *often relying on abstract labels or unexplained technical jargon rather than concrete steps*.
- **5 (Ambiguous Path):** The description of the implementation path is ambiguous, leaving significant questions about how the idea would be realized.
- **4. (High-Level Only):** The idea is presented only at a very high, abstract level, *lacking the necessary mechanisms or concrete steps* to assess its substance.
- **3 (Internally Inconsistent):** The description contains logical contradictions or inconsistencies that undermine its coherence.
- **2 (Chaotic):** The articulation is chaotic and poorly structured, making it extremely difficult to grasp the intended meaning.
- **1 (Incomprehensible):** The description is nonsensical, incomplete, or so confusing that it's impossible to understand the idea.

Novelty Scoring Rubric (1-10 Scale)
- **10 (Paradigm Shift):** A true 0-to-1 creation. The idea introduces a completely new concept, method, or technology that could create an entirely new field or render existing approaches obsolete.
- **9 (Field-Defining):** While not creating a field from scratch, the idea is disruptive enough to fundamentally alter the dominant research direction or establish a new, important sub-field.
- **8 (Significant Innovation):** A clever and non-obvious fusion or application of existing technologies that achieves results far superior to the state-of-the-art. It solves a known problem in a surprisingly effective way.
- **7 (Clever Improvement):** A significant and non-trivial improvement over existing solutions (a 1-to-1.5 improvement). The innovation is substantial but still follows a recognizable evolutionary path.
- **6 (Incremental Improvement):** A noticeable optimization or feature extension on existing solutions (a 1-to-1.1 improvement). The innovation is localized and offers a moderate benefit.
- **5 (Minor Tweak):** A small, obvious, or superficial modification to an existing idea or product.
- **4 (Rehash):** Combines existing ideas without adding any new insight or significant value.
- **3 (Known Solution):** The idea is a well-known concept or a direct reimplementation of a standard, existing solution.
- **2 (Outdated):** The idea proposes a solution that is less advanced than current standard practices.
- **1 (Lacks Originality):** A direct copy or imitation of an easily identifiable existing solution with no discernible innovation.

Figure 7: Pairwise LLM judge prompt — Part 1/3: Task description, and the Clarity & Novelty rubrics.

Feasibility Scoring Rubric (1-10 Scale)

- **10 (Exemplary Research Design):** The proposal demonstrates exceptional methodological rigor. The hypothesis-method-outcome chain is crystal clear and internally consistent. All assumptions are explicitly stated with theoretical or empirical justification. The experimental design includes comprehensive ablations, baselines, and clear success/failure criteria. Risks are acknowledged with concrete mitigation strategies. Resources are accessible.
- **9 (Strong Research Design):** The methodology is logically sound with well-justified assumptions. The experimental plan is comprehensive with appropriate baselines and evaluation metrics. Key risks are identified with reasonable mitigation approaches. Minor gaps exist but do not undermine the research validity.
- **8 (Solid Research Design):** The core methodology is internally consistent and the logical path is clear. Most assumptions are reasonable, though one or two may need empirical validation (which is acceptable—that's what research does). The proposal includes a reasonable experimental plan. Some risks may not be fully addressed but are not critical.
- **7 (Adequate Research Design):** The proposal is generally sound with a clear logical structure. The methodology is reasonable but relies on assumptions that the research itself aims to validate. The experimental plan is present but may lack some ablations or edge case considerations. Risk awareness is partial.
- **6 (Acceptable with Gaps):** The logical chain has some weak links but remains plausible. The methodology is reasonable but underspecified in places. Key assumptions are stated but justification is thin. Experimental design exists but lacks rigor (e.g., missing baselines or unclear metrics). Some obvious risks are not acknowledged.
- **5 (Questionable Rigor):** The proposal has noticeable logical gaps. The connection between method and expected outcome relies on optimistic assumptions without adequate justification. Experimental design is incomplete or poorly specified. Major risks are not addressed.
- **4 (Weak Methodology):** Core assumptions are poorly justified or internally inconsistent. The logical chain from method to outcome has significant gaps. The proposal lacks a clear experimental validation strategy. Critical risks are ignored.
- **3 (Flawed Design):** The proposal contains logical contradictions or demonstrates misunderstanding of key concepts. The methodology cannot plausibly achieve the stated goals. No meaningful experimental plan is provided.
- **2 (Severely Flawed):** The proposal is based on demonstrably incorrect assumptions or fundamental misunderstandings. The logical structure is incoherent. The approach has no realistic path to generating valid research findings.
- **1 (Fundamentally Unsound):** The proposal contradicts established scientific principles or makes claims that are logically impossible. The core idea is not a valid research question.

Relevance Scoring Rubric (1-10 Scale)

- **10 (Perfect Alignment):** The core methodology *perfectly* addresses the **Core Problem Domain** (e.g., 'heterogeneous data') *and* all **Key Mechanisms** (e.g., 'filtering', 'fusion') specified in the goal.
- **9 (Excellent Alignment):** The core methodology *directly* addresses the **Core Problem Domain** *and most* of the primary **Key Mechanisms**.
- **8 (Core Alignment):** The core methodology *directly* addresses the **Core Problem Domain** *and* at least *one* primary **Key Mechanism**.
- **7 (Mechanism Alignment):** The idea provides a strong solution for one or more **Key Mechanisms** (e.g., 'fusion'), but its method is *agnostic to or misinterprets* the specific **Core Problem Domain** (e.g., it solves for 'multi-source text' when 'heterogeneous' was required). *The score is capped at 7 for this domain mismatch.*
- **6 (Partial Alignment):** The idea is related to the research field but addresses only *secondary* mechanisms or provides a partial solution to the Core Problem Domain.
- **5 (Misaligned Scope):** The idea provides a solution for an *inappropriate scope* (e.g., it offers a **specific application** when a **general framework** was requested, or vice-versa). *This is a fundamental misalignment; the score is capped at 5.*
- **4 (Tangential):** The idea is related to the research field but only tangentially addresses the specific research goals. The outcomes might provide useful context but are not essential to the core objective.
- **3 (Critically Misaligned):** The idea is based on a fundamental misunderstanding of the research goal's domain or scope (e.g., solving for 'text' when 'video' was required).
- **2 (Weakly Related):** The link between the idea and the research goals is weak or superficial.
- **1 (Irrelevant / Out of Scope):** The idea has no discernible connection to the defined research goals.

Figure 8: Pairwise LLM judge prompt — Part 2/3: Feasibility & Relevance rubrics.

Analysis Content Instructions

Before generating the final scores, you must first generate the comparative analysis.

1. **Dimensional Comparison:** For each of the `..._analysis` fields (e.g., `Clarity_analysis`, `Novelty_analysis`), you must write a concise, **direct, side-by-side comparison** of both ideas. Do not just describe each idea in isolation. You must contrast their respective strengths and weaknesses *against each other* for that specific dimension. You must justify your scores by *strictly adhering to the following logic*:
   - **For `Clarity_analysis`:** You must distinguish between mere **Articulacy** (how well-written or academically sound the text is) and **Actionability** (whether the description provides concrete components and mechanisms for *how* to build it). *An idea that is unactionable due to reliance on unexplained, complex jargon should score lower than a simple but clear idea.* Prioritize actionability and reproducibility over academic eloquence.
   - **For `Novelty_analysis`:** You must differentiate between **Component Novelty** (i.e., proposing the use of a single new algorithm) and **Architectural Novelty** (i.e., proposing a new *framework* or a non-obvious *way of combining* existing technologies). A significant architectural innovation is generally more novel than simply applying a new component.
   - **For `Feasibility_analysis`:** You must write a direct side-by-side comparison evaluating the **research feasibility** of both ideas as research proposals. **CRITICAL: Do NOT penalize proposals for using novel/unproven methods or for having uncertain outcomes—that is the nature of research. Instead, evaluate methodological rigor.** Focus on: (1) **Methodological Soundness**: Is the hypothesis-method-outcome chain complete and internally consistent? Are assumptions explicitly stated and justified? (2) **Testability**: Can the core hypotheses be validated through well-defined experiments? Are success/failure criteria clear? (3) **Resource Accessibility**: Are required data, compute, and tools realistically obtainable? (4) **Risk Awareness**: Does the proposal acknowledge key uncertainties? Are there mitigation strategies, ablations, or fallback plans? You must explicitly compare which idea demonstrates stronger *research design rigor*, NOT which is more likely to succeed or uses more established methods.
   - **For `Relevance_analysis`:** This is critical. You must first identify and state the **Core Problem Domain** (e.g., 'heterogeneous data'), the **Key Mechanisms** (e.g., 'efficient fusion'), and the **Required Scope** (e.g., a 'general framework' vs. a 'specific application') from the `{question}`. Your analysis must then compare how well each idea addresses *all three*.
     - An idea that perfectly solves the mechanisms but misinterprets the core domain (e.g., confusing 'multi-source text' with 'heterogeneous') is *significantly less relevant* (see Rubric score **7**).
     - An idea that correctly addresses the domain and mechanisms but **fails to match the Required Scope** (e.g., providing a specific application for a general framework goal) is also *fundamentally less relevant* and must be scored accordingly (see Rubric score **5**).

Output Format

Strictly return a single JSON block as follows (do not add extra explanation):

```
{{
  "overall_comparison":{{
    "Clarity_analysis": "A direct comparison of Idea 1's clarity versus Idea 2's
         clarity, explaining the rationale for their respective scores.",
    "Novelty_analysis": "A direct comparison of Idea 1's novelty versus Idea 2's
         novelty, explaining the rationale for their respective scores.",
    "Feasibility_analysis": "A direct comparison of Idea 1's feasibility versus
        Idea 2's feasibility, explaining the rationale for their respective
        scores.",
    "Relevance_analysis": "A direct comparison of Idea 1's relevance versus Idea
         2's relevance, explaining the rationale for their respective scores."
  }},
  "assistant_1": {{
    "Clarity": 0,
    "Novelty": 0,
    "Feasibility": 0,
    "Relevance": 0
  }},
  "assistant_2": {{
    "Clarity": 0,
    "Novelty": 0,
    "Feasibility": 0,
    "Relevance": 0
  }}
}}
```

Figure 9: Pairwise LLM judge prompt — Part 3/3: Analysis content instructions and JSON output format.

You'll be presented with a series of questions (research goals). For each question, two answers will be provided. Your task is to read both answers carefully and decide which one you believe is better.

When judging, consider:

- **Clarity:** It examines whether the answer is clear and actionable (not just well-written), with concrete steps/mechanisms rather than vague jargon.
- **Novelty:** It examines whether the answer proposes a non-obvious idea or framework (new combinations/frameworks typically count more than swapping a single component).
- **Feasibility:** It examines whether the answer is a rigorous, testable research proposal (clear hypothesis→method→outcome chain, experiments/baselines/ablations, realistic resources, risk awareness). Do not penalize uncertainty or use of novel methods by itself.
- **Relevance:** It examines whether the answer matches the research goal's core problem domain, required mechanisms, and required scope (general framework vs specific application).

Question: `{question}`
Answer A: `{answer_a}`
Answer B: `{answer_b}`

Based on the question, comparing these two answers, in terms of clarity, novelty, feasibility, and relevance, respectively. Give the win-tie-lose of Answer A compared to Answer B in each of the four aspects.

Definitions (A compared to B):

- **win**: Answer A is better than Answer B on this aspect.
- **tie**: Answer A and Answer B are roughly equal on this aspect.
- **lose**: Answer A is worse than Answer B on this aspect.

Output (no extra text):

```
Clarity: win/tie/lose
Novelty: win/tie/lose
Feasibility: win/tie/lose
Relevance: win/tie/lose
```

Figure 10: Instructions for human evaluation of idea generation.