AI SCIENTISTS NEED STRONGER VERIFICATION FRAMEWORK TO ADDRESS SYSTEMIC FRAGILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Over the past few years, the development of AI Scientists has accelerated rapidly, demonstrating growing potential for automatic research across multiple disciplines. Despite this progress, the ***capability boundaries of current AI Scientist systems*** remain insufficiently characterized, especially with respect to verification ability. In this paper, we argue that autonomous verification should be treated as a first-class challenge in the development and evaluation of AI Scientist systems. **Current AI Scientist systems exhibit clear verification shortcomings,** demanding more comprehensive strategies to address the systemic fragility of autonomous verification. We support our position with three levels of evidence. First, we review **coarse-grained evidence from existing benchmarks** on complex scientific tasks, which shows that frontier models remain unreliable when tasks require domain-specific reasoning. Second, we conduct an **output-level evaluation of 28 research papers** generated by five AI Scientist systems, revealing persistent weaknesses in experimental design and methodological clarity. Third, we introduce **DeepVerify-5K**, which contains **4,890** end-to-end scientific discovery trajectories. Each trajectory provides records of idea generation, implementation, execution, and iterative revision, enabling process-level analysis of how autonomous verification fails. Our analysis of DeepVerify-5K shows that **current AI Scientist systems often exhibit fragile verification behaviors**, including redundant code changes and entanglement with evaluation pipelines. We hope this position paper encourages the community to pay greater attention to the systemic fragility of autonomous verification and evaluate AI Scientist systems' capabilities in closer alignment with actual scientific scenarios.

1 INTRODUCTION

The automation of scientific discovery has long been one of humanity's deepest desires (Langley, 1987; King et al., 2009; Radensky et al., 2024). In recent years, with the advances in deep learning technology (LeCun et al., 2015), a range of **automated scientific tools has emerged**, leading to groundbreaking achievements in fields such as biomedicine (Yang et al., 2025c; Jumper et al., 2021), chemistry (Stokes et al., 2020), and materials science (Szymanski et al., 2023). However, these scientific tools still rely heavily on human involvement. Researchers must first formulate ideas to be tested, while AI is responsible for the labor-intensive tasks of verification and iterative search. Therefore, these intelligence systems cannot be considered as truly automated scientific agents.

Recent advances in large language models (LLM) have given rise to **AI Scientist** systems, pushing the automation of scientific research into a new stage, with AI taking the lead as the primary actor of scientific discovery, managing the entire workflow from idea generation to experiment execution (Lu et al., 2024; Weng et al., 2025a). As shown in Figure 1, we demonstrate the progress made by current AI Scientist systems. Recent studies have shown that research papers produced by AI Scientist have already reached the level of submissions to workshops at major machine learning conferences (Si et al., 2024; Yamada et al., 2025; Intology, 2025). Despite this substantial progress, **the capability boundaries of AI Scientist systems remain insufficiently characterized.** While prior work has discussed their strengths in idea generation (Si et al., 2024; 2025; Ma et al., 2026), far less attention has been paid to their *verification* capabilities, which is the critical ability to reliably implement, execute, and iteratively refine ideas into reproducible results.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

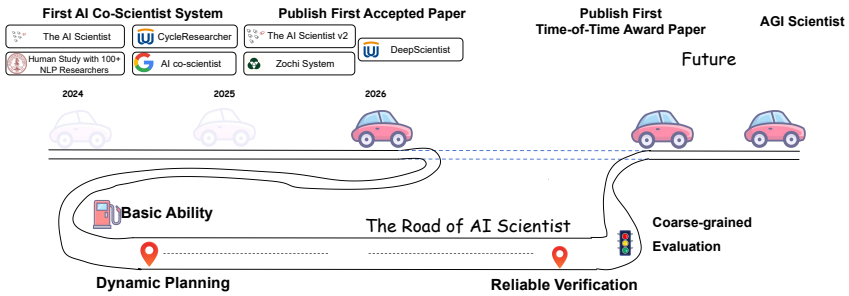


Figure 1: The roadmap of AI Scientist from 2024 to the future, highlighting key milestones and fundamental challenges that must be overcome to bridge the verification gap of AI Scientist.

In this position paper, we first propose **a conceptual framework that defines an AI Scientist** as an advanced end-to-end system capable of independently formulating scientific ideas and performing the verification process (Section 2). This definition aligns with current research progress (Lu et al., 2024; Weng et al., 2025a; Yamada et al., 2025), serving as the theoretical foundation for our position:

> **Current AI Scientist systems exhibit clear verification shortcomings, demanding more comprehensive strategies to address the systemic fragility of autonomous verification.**

We then defend our argument through three complementary sources of empirical evidence. First, we analyze **quantitative evidence from existing benchmarks** that evaluate LLMs' abilities on complex scientific tasks across diverse domains (Section 3.1). Although LLMs have shown strong potential in generating novel ideas (Si et al., 2024; Chai et al., 2024; Gottweis et al., 2025), **their performance remains far from reliable** when scientific tasks require faithful implementation and domain-specific reasoning. For instance, frontier models achieve only 16.90% on MLE-Bench (Chan et al., 2024) and 30.00% on CMT-Benchmark (Pan et al., 2026), suggesting that current systems still struggle in settings where scientific claims must be operationalized and tested. Beyond aggregate benchmark performance, we further provide a **fine-grained error attribution analysis** on LiveCodeBench. The failure decomposition shows that errors arise not only from wrong answers, but also from failures to form executable artifacts and from unstable verification outcomes, indicating **weaknesses in producing consistently verifiable solutions.**

Second, we conduct **a simulated peer-review experiment of AI-generated research papers** (Section 3.2). Using DeepReviewer-14B (Zhu et al., 2025b), we assess 28 publicly available papers generated by five advanced AI Scientist systems. The results show that current AI Scientist systems still lack the capabilities to conduct rigorous experiments and produce high-quality scientific papers.

Third, to examine **where** and **why** autonomous verification fails during the research process, we introduce **DeepVerify-5K**, a high-quality dataset containing **4,890 end-to-end scientific discovery trajectories** generated by a SoTA AI Scientist system (Weng et al., 2026). Each trajectory records idea generation, implementation, execution, and iterative revision, providing a process-level trace of autonomous verification (Appendix Figures 6 and 7). DeepVerify-5K focuses on computational research workflows, where current AI Scientist are most mature and complete verification traces can be collected and audited. Our analysis shows that **autonomous verification is often fragile and inefficient:** AI Scientist systems frequently introduce large and redundant code changes, modify evaluation pipelines instead of the core method, and fail due to unresolved execution errors.

This paper argues that autonomous verification should be treated as a major challenge in the development and evaluation of AI Scientist systems. We combine benchmark-level symptoms, output-level audits and process-level evidence from DeepVerify-5K to reveal that autonomous verification remains a central bottleneck. We hope this position paper will encourage the community to treat reliable verification as a foundational requirement for building trustworthy AI Scientist systems.

2 Definition of the AI Scientist

In this section, we first provide a discussion of the unique characteristics of the AI Scientist (Section 2.1). Building on this discussion, we then propose a conceptual framework that formally defines the scientific discovery process and AI Scientist (Section 2.2).

2.1 UNIQUE CHARACTERISTICS

Scientific tools represent specialized AI systems designed to accelerate individual stages of the scientific discovery process, operating within a paradigm where human researchers remain in the dominant position of scientific discovery. Such tools have demonstrated remarkable success across diverse scientific fields, including protein structure prediction (Jumper et al., 2021), antibiotic discovery through deep learning approaches (Stokes et al., 2020), and autonomous chemical research with large language models (Boiko et al., 2023).

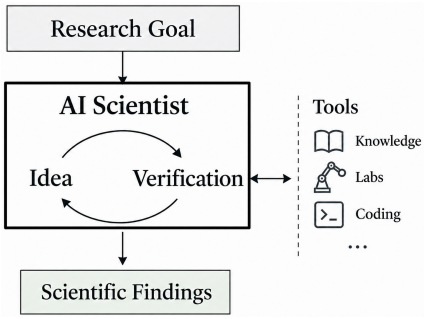


Figure 2: From scientific tools to AI Scientist systems.

AI Scientist systems instead represent a scientific paradigm shift from AI-assisted research to AI-driven exploration, where AI assumes the role of an autonomous agent capable of conducting end-to-end scientific research (Angello et al., 2024). As illustrated in Figure 2, while scientific tools operate under human supervision, receiving data as input and producing predictions as output, AI Scientist goes a step further by demonstrating autonomous scientific reasoning capabilities (Swanson et al., 2025). It receives human-defined research goals as input and engages in iterative, self-directed interactions with scientific tools to generate comprehensive solutions (Yamada et al., 2025).

2.2 CONCEPTUALIZED FRAMEWORK

First, we use the term **scientific discovery process** to refer to the end-to-end loop through which a scientific question is transformed into evidence-supported knowledge (Xie et al., 2025). This loop begins with a research goal and proceeds by formulating a scientific hypothesis, translating it into a testable procedure, executing that procedure, interpreting the resulting evidence, and then updating or rejecting the hypothesis accordingly.

We then define an **AI Scientist** as a fully autonomous scientific system capable of operating over the scientific discovery process. Unlike a general-purpose scientific tool, which typically supports a specific stage under human supervision, an AI Scientist is expected to generate scientific ideas autonomously (Weng et al., 2025b) and verify them through end-to-end execution. It synthesizes novel scientific ideas grounded in existing knowledge, then transforms them into executable artifacts, such as code, simulations, or machine-executable laboratory protocols (Boiko et al., 2023), and subsequently leverages automated experimental infrastructure, including coding agents (Yang et al., 2024b) or self-driving laboratories (Abolhasani & Kumacheva, 2023), to execute experiments and assess the efficacy of these ideas.

Definition: An AI Scientist is an autonomous system capable of formulating scientific ideas and executing the requisite verification procedures.

Under this framework, we define **verification** as the process-level capability that enables an AI Scientist to determine whether a scientific idea is supported by trustworthy evidence. Operationally, **verification consists of five interrelated stages:** *(1) implementation*, which transforms an idea into an executable artifact or testable procedure; *(2) execution*, which carries out that artifact under required experimental conditions; *(3) validation*, which checks whether the resulting evidence is faithful and whether the evaluation pipeline preserves the intended claim-evidence relation; *(4) iterative revision*, which diagnoses failures and corrects them through repeated refinement; and *(5) reproducible evidence*, which produces a trustworthy and reproducible conclusion or rejects the claim when verification fails.

Regarding scope, our definition conceptually includes executable verification procedures across scientific domains, including computational experiments, simulations, and machine-executable laboratory protocols. This scope allows us to analyze verification failures at the level of implementation, execution, validation, revision, and reproducibility.

Table 1: Benchmark evidence for verification-related challenges across scientific domains.

Benchmark	Task Description	Domains	LLM	Score
CMT-Benchmark (Pan et al., 2026)	Condensed-matter reasoning	Physics	GPT-5	30.00%
MLE-Bench (Chan et al., 2024)	Kaggle-style competitions	Computer Science	OpenAI o1-preview	16.90%
PaperBench (Starace et al., 2025)	Paper result reproduction	Computer Science	OpenAI o1-high	26.00%
LiveProteinBench (Rong et al., 2025)	Protein-science reasoning	Biology	GPT-5	60.65%
CORE-Bench (Siegel et al., 2024)	Paper result reproduction	Computer Science / Social Science / Medicine	GPT-4o	55.56%
ML-Dev-Bench (Padigela et al., 2025)	Applied Machine Learning tasks	Computer Science	Claude Sonnet 3.5	50.00%
AFMBench (Mandal et al., 2025)	AFM experiment execution	Materials / Biology	GPT-4o	65.00%
DSBench (Jing et al., 2024)	Data analysis and modeling	Data Science	Gemini-1.5-Pro	31.73%
LMR-BENCH (Yan et al., 2025)	Paper result reproduction	Computer Science	OpenAI o4-mini	42.90%
PRBench (Qiu et al., 2026)	Paper result reproduction	Physics	GPT-5.3-Codex	34.00%
SciReplicate-Bench (Xiang et al., 2025)	Algorithm reproduction	Computer Science	Claude Sonnet 3.7	39.00%
MaCBench (Alampara et al., 2025)	Scientific measurement interpretation	Chemistry / Materials	Claude Sonnet 3.5	43.00%
LabSafety Bench (Zhou et al., 2024)	Laboratory safety reasoning	Chemistry	GPT-4o	75.00%

3 Arguments for verification fragility

We argue that current AI Scientist systems exhibit clear verification shortcomings, demanding more comprehensive and robust strategies to address the systemic fragility of autonomous verification. To support this position, we present a broad analysis based on a comprehensive benchmark study spanning diverse domains (Section 3.1), a fine-grained error attribution analysis on LiveCodeBench, and a simulated peer-review assessment using an LLM-as-a-Judge methodology (Section 3.2).

3.1 Empirical analysis of verification capability

As the foundation models underlying AI Scientist systems, advanced LLMs have reached near-saturated performance on relatively short, well-specified code generation benchmarks (Liu et al., 2023; Yang et al., 2025a). **However, this proficiency fails to translate into real-world research settings,** where verification requires more than producing a runnable snippet of code.

Cross-domain benchmark evidence. Table 1 summarizes LLM agents' abilities on scientific benchmarks spanning diverse domains (Mandal et al., 2025; Chan et al., 2024; Zhou et al., 2024; Alampara et al., 2025; Jain et al., 2024). Although these benchmarks vary in task format and evaluation metrics, they reveal a common weakness:**when tasked with complex research workflows, the performance of SoTA LLMs drops dramatically.** Their failures often emerge when scientific tasks require them to convert high-level research goals into faithful executable procedures, obtain valid measurements, and draw evidence-grounded interpretations.

In computational research settings, benchmarks such as PaperBench (Starace et al., 2025), ML-Dev-Bench (Padigela et al., 2025), DSBench (Jing et al., 2024), and SciReplicate-Bench (Xiang et al., 2025) evaluate whether agents can reproduce results, develop machine-learning workflows, analyze data, or implement algorithms. As shown in Table 1, even strong frontier models achieve far-from-reliable scores of 26.00% on PaperBench and 55.56% on CORE-Bench. These results suggest that current research agents **struggle with implementing high-level research plans into functional artifacts that can be executed and evaluated.** They often fail to ensure implementation

correctness, control runtime behavior, match produced outputs to intended scientific claims, and iteratively optimize the pipeline when validation fails.

The same pattern also appears outside computer science benchmarks. In physics, CMT-Benchmark (Pan et al., 2026) and PRBench (Qiu et al., 2026) report only 30.00% and 34.00% performance, respectively, suggesting that models still **struggle with domain-specific assumptions and faithful reproduction of scientific claims.** In biology and materials science, LiveProtein-Bench (Rong et al., 2025) and AFMBench (Mandal et al., 2025) report stronger but still limited scores of 60.65% and 65.00%, showing that models remain unreliable when scientific reasoning must respect experimental procedures. In safety, LabSafety Bench (Zhou et al., 2024) reports 75.00% on laboratory safety reasoning, implying that even **safety-critical scientific judgment remains imperfect.** These results indicate that verification failures persist when models must reason under domain-specific assumptions, suggesting a broader weakness in scientific verification.

Fine-grained error attribution. Aggregate benchmark scores alone do not reveal where the verification process breaks down. To provide a more fine-grained attribution analysis, we further analyze three independent GPT-5.4 runs on the latest LiveCodeBench test set (v6, 175 instances). Across the three runs, pass@1 is 77.71%, 77.14%, and 73.71%, with a combined pass@1 of 76.57%. However, repeated-run outcomes also reveal a substantial verification instability: only 70.86% of problems are stably solved across runs, while 10.85% are unstable and 18.29% persistently fail. This indicates that even when a model appears strong under a single benchmark score, **its ability to produce consistently verifiable solutions remains limited.**

Table 2: Failure decomposition on LiveCodeBench.

Failure Category	Count	%
EMPTY_OR_EXTRACT	87	69.6
SEMANTIC-WA	26	20.8
EXEC-RUNTIME	10	8.0
EVAL-SENSITIVE	2	1.6
EXEC-TIMEOUT	0	0.0
RESOURCE / ENV	0	0.0

We then decompose failed run instances into artifact-formation, execution, semantic, and evaluator-sensitive failures. As shown in Table 2, the largest category is **EMPTY_OR_EXTRACT**, accounting for 69.6% of failed run instances. Manual inspection shows that these failures correspond to cases where the model fails to produce an executable code artifact. In such cases, verification fails before downstream execution or semantic validation. Semantic wrong-answer failures account for another 20.8%, showing that even when an executable artifact is produced, it may still **fail to implement the intended solution correctly.** By contrast, evaluator-sensitive failures are rare. Under stricter re-evaluation of all run instances from unstable and persistent-failure problems, only 2/153 cases flip from fail to pass, both from the same timeout-sensitive problem. This suggests that the observed failures cannot be explained away as benchmark noise or evaluator artifacts.

To sum up, this fine-grained decomposition strengthens our benchmark-level evidence for verification fragility. The failures appear across multiple stages of the verification, supporting our claim that current AI Scientist systems suffer from systemic weaknesses in autonomous verification.

3.2 LLM-AS-A-JUDGE REVEALS THE VERIFICATION WEAKNESSES

To examine whether verification weaknesses are visible in the final artifacts produced by current systems, **we employ a simulated peer review to assess the quality of scientific outputs** from current AI Scientist systems. We select 28 publicly available research papers generated independently by five different AI Scientist systems and utilize the powerful review model DeepReviewer-14B (Zhu et al., 2025b) to conduct systematic evaluation under unified standards.

We acknowledge that the public availability of these papers may introduce selection bias, as researchers may release better-performing outputs. Therefore, our evaluation should not be interpreted as a representative estimate of the average output quality of each system across all scenarios. Instead, **it serves as a complementary output-level audit:** if verification-related weaknesses remain visible even in publicly released AI-generated papers, this provides suggestive evidence that current AI Scientist systems still struggle to produce scientifically reliable final artifacts.

Quantitative Assessment Results. As Table 3 demonstrates, DeepReviewer-14B assigns generally low average scores to these AI-generated papers across multiple core dimensions. The Rating scale ranges from 1-10, where a score of 6 indicates acceptable quality, while Soundness, Presentation, and Contribution scores range from 1-4. The highest-rated system, Zochi (with a sample size of 2

Table 3: DeepReviewer-14B evaluation of AI-generated papers from various AI Scientist systems. Scores reflect averages across the 'Num' of available papers. Note: Publicly available papers may be curated and not fully representative of typical system output.

AI Scientist System	Num	Soundness↑	Presentation↑	Contribution↑	Decision↑	Rating↑	Percentile↑
HKUSD AI Researcher	7	1.75	1.46	1.57	0.0	2.57	3.43%
AI Scientist	10	2.08	1.80	1.75	0.0	3.35	8.22%
AI Scientist v2	3	1.67	1.50	1.50	0.0	2.33	2.04%
CycleResearcher-12B	6	2.25	1.75	2.13	0.0	3.75	16.88%
Zochi	2	2.38	2.38	2.25	0.0	4.63	29.96%

papers), achieves **an average rating of only 4.63**, while other systems score even lower, typically in the 2-3 point range. These quantitative scores reveal that current AI Scientist systems face **significant challenges in independently producing high-quality research papers.**

Table 4 shows that among the major defect categories identified by DeepReviewer-14B, "Experimental Weakness" appears across all 28 evaluated AI-generated papers, with a 100% occurrence rate. **This finding directly supports our positions regarding verification limitations**, particularly in experimental design, execution, and result analysis. The second and third most prevalent issues are "Methodological Unclarity/Flaws" (96.4%) and "Writing & Presentation Issues" (92.9%), which indirectly reflect AI Scientists' insufficient ability to clearly articulate and implement research plans. The prevalence of these defects suggests that **weaknesses in verification can propagate into final research outputs,** leaving them well below the standards required for reliable scientific work.

Table 4: Defect categories and their issues.

Defect Category	Number	Percentage
Experimental Weakness	28	100.0
Methodological Unclarity/Flaws	27	96.4
Writing & Presentation Issues	26	92.9
Novelty Concerns	25	89.3
Theoretical Weakness	24	85.7
Literature Review Deficiencies	22	78.6
Practicality & Robustness Gaps	21	75.0
Reproducibility Issues	20	71.4
Computational Cost Concerns	18	64.3
Component Analysis	16	57.1
Hyperparameter Analysis Lacking	16	57.1
Ethical Considerations Missing	3	10.7

4 Detailed analysis based on DeepVerify-5K

4.1 Dataset collection

During the data collection process of DeepVerify-5K, we use Gemini-2.5-Pro as the idea generator/evaluator and a powerful coding agent based on Claude-4-Opus as the verification tools. Specifically, we adopt DeepScientist (Weng et al., 2026), a SoTA AI Scientist system capable of conducting automated AI research, as our experimental platform. We run DeepScientist on three frontier AI research tasks: *Agent Failure Attribution, LLM Inference Acceleration, and AI Text Detection* (Appendix B). For each task, DeepScientist is instructed to autonomously generate candidate ideas, assess their quality, implement high-quality ideas into executable code, run the code via Claude Code, and propose new scientific ideas based on the experimental results. **Each iteration yields a complete trace of the scientific discovery process**, containing extensive verification details. These traces therefore serve as the empirical foundation for our analysis of AI Scientist verification capability. All models are deployed on two servers, each equipped with 8 NVIDIA H800 GPUs, **costing a month-long cycle of exploration, validation, and iteration.**

Table 5: Data statistics of DeepVerify-5K.

Task	#Traces	#Implemented
Agent Failure Attribution	1,077	196
LLM Inference Acceleration	1,332	312
AI Text Detection	2,481	601
Overall	**4,890**	**1,109**

As shown in Table 5, DeepVerify-5K contains **4,890** records of end-to-end scientific discovery trajectories. Each trace includes the **complete logs** for both idea generation and verification, along with the corresponding **idea artifacts** and the **code** produced from translating those ideas into executable implementations (Figure 7). During the discovery process, we further use Gemini-2.5-Pro to assess the generated ideas. Low-quality ideas are filtered out, resulting in a curated set of **1109** ideas that are then implemented and executed via Claude Code.

4.2 IN-DEPTH ANALYSIS OF THE VERIFICATION PROCESS

As depicted in Figure 3, the **autonomous verification process often demands substantial work** before an idea can be fully verified. The editing ranges from near-zero changes to large-scale revisions (exceeding 1k added lines), indicating that many idea-to-code translations require significant engineering to become executable. Crucially, after manual validation, we observe that **many of the added changes are not necessary for correct verification.** A considerable fraction of the newly introduced code appears redundant (e.g., duplicated utilities, unused branches, or modifications that do not affect the final evaluated outcome), suggesting that AI Scientist systems frequently **"edit to progress"** rather than "edit to verify". This redundancy provides concrete evidence that **verification is prone to unstable and inefficient iteration,** where execution feedback is poorly interpreted and translated into targeted fixes.

To characterize execution-level verification reliability, we also systematically analyze the verification logs in DeepVerify-5K and map diverse raw error messages into eight interpretable categories (ENV, DATA, EXEC, TOOL, RUNTIME, TEST, RESOURCE, OTHER; see Appendix for definitions). The overall error distribution (Figure 4a) indicates that autonomous verification is repeatedly challenged by **execution- and environment-related issues**. Across 1,109 verification traces, we observe 891 error occurrences. Crucially, the errors that persist and ultimately terminate verification are not representative of the full set of encountered failures. As shown in Figure 4b, terminal failures are driven primarily by **RUNTIME and EXEC errors**, and most notably by **RESOURCE constraints**, suggesting that current coding agents can often recover from relatively superficial issues (e.g., missing files or minor environment mismatches) but still struggle to robustly adapt to resource limitations (e.g., insufficient memory). In addition, non-trivial execution failures frequently demand deeper diagnosis and targeted remediation beyond the agents' current verification capabilities.

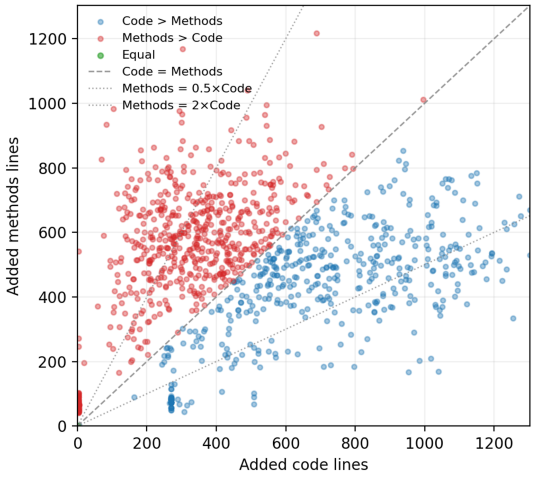
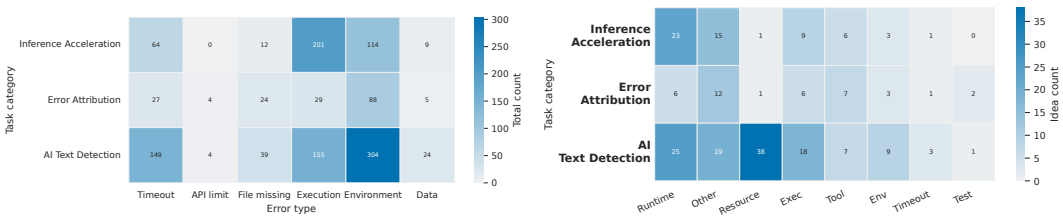


Figure 3: Implementation deviation from the provided baseline, measured by added code lines (x-axis) and method lines (y-axis).

Figure 5 reveals that the verification stage is entangled with broad edits across the surrounding experimental infrastructure. In particular, run scripts are edited almost universally, and tests are edited at high rates across all types, suggesting that a large fraction of trajectories involve modifying execution components rather than the core method implementation. More importantly, the Others category exposes the systemic fragility of autonomous verification. These traces rarely touch the core algorithm, yet they still heavily modify data pipeline / tests / scripts. This pattern is consistent with **non-faithful or shortcut verification behaviors**, where the agent achieves progress by tweaking experimental setup (e.g., data preprocessing or run-time configuration) rather than implementing the proposed algorithmic contribution. Similarly, the Eval-only group highlights another vulnerability where the system can **silently redefine success criteria.** These findings point to a systems implication that unconstrained write access enables AI Scientist systems to alter the evaluation metrics and execution scripts, thereby **undermining the trustworthiness of autonomous verification.**

4.3 DISCUSSION

In this section, we conduct an in-depth analysis of AI Scientist systems' verification capabilities using DeepVerify-5K. Our results show that autonomous verification often demands substantial edits to render ideas executable. However, manual inspection further reveals that many of these additions are unnecessary or redundant, resulting in unstable iteration. Moreover, verification often involves modifying surrounding infrastructure (scripts/tests/data pipelines) and is heavily bottlenecked by execution/runtime and resource failures, highlighting the fragility and low fidelity of unconstrained autonomous verification. These pieces of evidence support our position that current AI Scientist systems require more comprehensive and robust strategies to address the systemic fragility of



(a) Heatmap of the total error counts during verification across task categories. Counts include errors recovered by the coding agent and those that caused failure.

(b) Heatmap of terminal (unrecovered) errors that cause verification to fail, revealing the failure modes that remain unresolved by automated debugging and iteration.

Figure 4: Autonomous verification errors in DeepVerify-5K. Raw log messages are normalized into eight categories. The category definitions and full-resolution figures are provided in the Appendix.

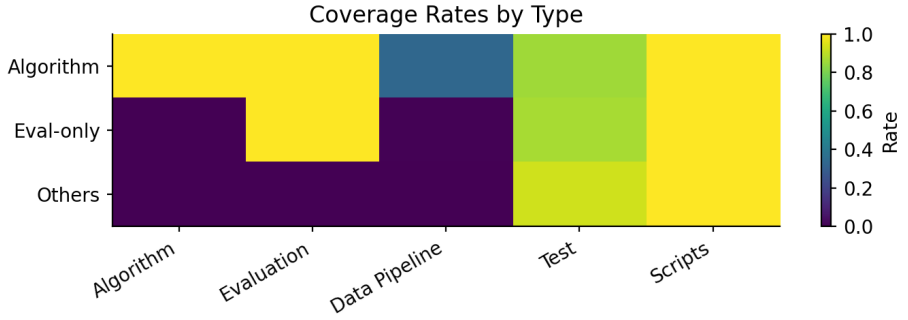


Figure 5: Heatmap of coverage rate indicating whether a given baseline was modified during verification. Rows group traces by the focus of modification: Algorithm (edits mainly target the method), Eval-only (edits mainly target evaluation), and Others (edits mainly target non-core aspects such as hyperparameters). Columns correspond to baseline components (e.g., core algorithm, evaluation module). Higher values indicate that component is frequently edited within that trace type.

autonomous verification. Notably, the end-to-end traces of DeepVerify-5K also make it a useful resource for probing AI Scientists' underlying mechanisms. For example, tracing how a scientific idea evolves across iterations or training process reward models tailored to scientific discovery. We consider these directions outside the scope of this paper and leave them to future work.

5 Future Directions

The preceding sections motivate a broader focus on the systemic fragility of autonomous verification and the need for more comprehensive and robust strategies. In this section, we outline feasible pathways to bridge the current verification capability gap of AI Scientists.

Built-in oversight and safety. While scaling laws for pre-training and post-training (Kaplan et al., 2020; Zhang et al., 2025) promise continued improvements, stronger base models alone are unlikely to eliminate verification failures in end-to-end scientific workflows. Instead, these improvements should be complemented by well-defined workflows (Li et al., 2024; Gu et al., 2024) that decompose research into inspectable intermediate steps, enabling targeted human or automated interventions (Fan et al., 2024; Arslan et al., 2024). Crucially, oversight and safety should be built into the verification process itself, not treated as a post hoc filter. This includes auditable logs, transparent tool use, and reproducible execution records, allowing inspection of why a system accepts or rejects a scientific claim (Zhu et al., 2025a; 2026). Such integrated oversight is essential for preventing benchmark "hacking" (Fu et al., 2025) and mitigating risks such as misleading reporting or improper data use, particularly in modular systems where failures often emerge from interactions among components.

The integration of modular architectures. Reliable verification in AI Scientist systems requires coordination among specialized agents responsible for literature grounding, experimental design, execution, and evaluation. Standardized interaction protocols, such as Model Context Protocol (MCP) (Yang et al., 2025b; Ray, 2025; Hou et al., 2025), provide an initial substrate for such coordination, and recent systems have begun to explore this paradigm. For example, Google's AI co-scientist (Gottweis et al., 2025) and Anthropic's multi-agent research system (Anthropic, 2025)

adopt orchestrator-based designs to decompose the research process. However, modularization introduces new sources of verification fragility. Heterogeneous agents may operate under inconsistent assumptions, propagate intermediate errors, and produce outputs that are difficult to reconcile into a coherent and auditable scientific trace. This calls for explicit interface contracts, shared state representations, cross-agent consistency checks, and mechanisms for failure attribution.

Finally, **training cost and efficiency.** A challenge for AI Scientist systems is the immense resource consumption of reinforcement learning (Cao et al., 2024; Oh et al., 2025). This concern is particularly acute in scientific discovery, where each action may require experiment reruns or long-horizon reasoning. Therefore, future work should prioritize feedback-efficient training. By allowing agents to receive cheaper and faster feedback on costly operations, such simulation-based methods may improve sampling efficiency and reduce wall-clock training time (Sun et al., 2025; Li et al., 2026).

6 Alternative views

Stronger autonomous verification may increase agency and safety risks. If an AI Scientist becomes better at end-to-end verification, it may also become better at executing harmful workflows (e.g., hazardous chemical synthesis) (Tang et al., 2025; Scheurer & Reuter, 2025). However, this does not remove the need for verification; rather, it changes what kind of verification is needed. Our position does not advocate maximal autonomy. Instead, we argue for safety-aware verification frameworks with auditable execution traces, transparent tool use, and structured validation protocols. In this view, verification is a mechanism for making any granted autonomy inspectable and interruptible.

Explicit research on scientific verification is unnecessary because verification may improve automatically as frontier models improve on general-purpose capabilities. We agree that such progress can reduce some implementation and execution errors (Jimenez et al., 2023; Yang et al., 2024a). However, scientific verification requires more than producing runnable artifacts (Canty et al., 2025). A system may execute code successfully while using invalid metrics, or overinterpreting weak evidence. Thus, stronger base models are helpful but not sufficient. AI Scientists still need mechanisms for tracing how evidence is produced, detecting non-faithful success, and explaining why a claim is accepted or rejected.

Discovery depends on a few genuine successes rather than uniformly correct execution across all attempts. We agree that discovery can be driven by rare successes (Weng et al., 2026). However, this argument makes verification more important. When true successes are rare, the key challenge is to distinguish credible findings from false positives caused by artifacts, data leakage, invalid baselines, or noisy measurements (Luo et al., 2025). Thus, the question is not whether every attempt succeeds, but whether AI Scientists can reliably identify which apparent successes deserve scientific trust.

AI Scientists may weaken human understanding, and falsification offers a stronger model for scientific verification. LLM-assisted workflows may outsource reasoning in ways that reduce human understanding (Rossi et al., 2026). Meanwhile, scientific inquiry emphasizes conjecture and refutation rather than confirmation alone (Popper, 2014), and recent agentic systems have explored hypothesis validation through sequential falsification (Huang et al., 2025). We view this as complementary to our position: trustworthy AI Scientists should expose reasoning, execution records, and evidential limits to human inspection, while verification should stress-test claims, seek counterevidence, and trigger revision rather than merely confirm that a pipeline runs.

7 Conclusion

This position paper aims to draw researchers' attention to the limitations of current AI Scientists, emphasizing the importance of enhancing verification capabilities to realize the trustworthy AI Scientist. To substantiate our position, we comprehensively reviewed and analyzed relevant benchmark and conducted simulated peer reviews of papers produced by existing AI Scientist systems to illustrate the current verification Gap. To further localize where and why verification fails, we introduced DeepVerify-5K, a high-quality dataset of 4,890 end-to-end scientific discovery trajectories. Analysis of DeepVerify-5K shows that autonomous verification is fragile and inefficient, supporting the arguments of this position paper. Finally, we explored and discussed potential directions for improving AI Scientists' verification capabilities.

AI USE STATEMENT

Generative AI tools were used in this work for language editing and coding assistance, as well as experimental components of the study. All AI-assisted analyses and code were reviewed and validated by the authors before being used in the paper. AI-generated outputs were not treated as ground truth without additional checks. Quantitative results were computed from the corresponding experimental artifacts and logs, and qualitative findings were manually inspected where appropriate. The authors take full responsibility for the final text, analyses, claims, and artifacts presented in this work.

ETHICS STATEMENT

This work studies the limitations of autonomous AI Scientist systems, with a focus on whether scientific claims can be supported by faithful and reproducible verification. The study does not involve human subjects or private personal data. DeepVerify-5K consists of model-generated scientific discovery trajectories collected from computational research tasks, and the AI-generated papers used in our output-level analysis are publicly available research artifacts.

Improving autonomous verification may also increase the ability of AI systems to execute complex scientific workflows, including potentially harmful ones. Our work does not advocate unrestricted scientific autonomy. Instead, we emphasize safety-aware verification with auditable execution traces and transparent tool use so that autonomous actions remain inspectable and interruptible. We also caution that automated evaluation, including LLM-based review, may introduce model-specific biases and should not be interpreted as a substitute for human scientific judgment.

REPRODUCIBILITY STATEMENT

We provide detailed information needed to reproduce our empirical analyses throughout the paper and supplementary materials. Section3 documents the benchmark results, LiveCodeBench repeated-run analysis, and the simulated peer-review protocol. Section 4 describes the construction of DeepVerify-5K, including the underlying research tasks, models, and verification process, together with the analyses performed on the collected trajectories. The appendix provides additional task descriptions, error-category definitions, examples of verification traces, and full-resolution analysis figures. We will publicly release DeepVerify-5K upon publication to facilitate independent verification and further research on autonomous scientific verification.

REFERENCES

Milad Abolhasani and Eugenia Kumacheva. The rise of self-driving labs in chemical and materials sciences. *Nature Synthesis*, 2(6):483–492, June 2023. ISSN 2731-0582. doi: 10.1038/s44160-022-00231-0. URL https://doi.org/10.1038/s44160-022-00231-0.

Nawaf Alampara, Mara Schilling-Wilhelmi, Martiño Ríos-García, Indrajeet Mandal, Pranav Khetarpal, Hargun Singh Grover, NM Anoop Krishnan, and Kevin Maik Jablonka. Probing the limitations of multimodal language models for chemistry and materials research. *Nature computational science*, 5(10):952–961, 2025.

Nicholas H. Angello, David M. Friday, Changhyun Hwang, Seungjoo Yi, Austin H. Cheng, Tiara C. Torres-Flores, Edward R. Jira, Wesley Wang, Alán Aspuru-Guzik, Martin D. Burke, Charles M. Schroeder, Ying Diao, and Nicholas E. Jackson. Closed-loop transfer enables artificial intelligence to yield chemical knowledge. *Nature*, 633(8029):351–358, September 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07892-1. URL https://doi.org/10.1038/s41586-024-07892-1.

Anthropic. How we built our multi-agent research system. https://www.anthropic.com/engineering/multi-agent-research-system, June 2025. Anthropic Engineering Blog.

Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. A survey on rag with llms. *Procedia Computer Science*, 246:3781–3790, 2024.

Yoshua Bengio, Michael Cohen, Damiano Fornasiere, Joumana Ghosn, Pietro Greiner, Matt MacDermott, Sören Mindermann, Adam Oberman, Jesse Richardson, Oliver Richardson, et al. Superintelligent agents pose catastrophic risks: Can scientist ai offer a safer path? *arXiv preprint arXiv:2502.15657*, 2025.

Federico Bianchi, Owen Queen, Nitya Thakkar, Eric Sun, and James Zou. Exploring the use of AI authors and reviewers at Agents4Science. *Nature Biotechnology*, 44(1):11–14, January 2026. ISSN 1546-1696. doi: 10.1038/s41587-025-02963-8. URL https://doi.org/10.1038/s41587-025-02963-8.

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

Richard B Canty, Jeffrey A Bennett, Keith A Brown, Tonio Buonassisi, Sergei V Kalinin, John R Kitchin, Benji Maruyama, Robert G Moore, Joshua Schrier, Martin Seifrid, et al. Science acceleration and accessibility with self-driving labs. *Nature Communications*, 16(1):3856, 2025.

Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

Miaosen Chai, Emily Herron, Erick Cervantes, and Tirthankar Ghosal. Exploring scientific hypothesis generation with mamba. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pp. 197–207, 2024.

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

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6491–6501, 2024.

Jiayi Fu, Xuandong Zhao, Chengyuan Yao, Heng Wang, Qi Han, and Yanghua Xiao. Reward shaping to mitigate reward hacking in rlhf. *arXiv preprint arXiv:2502.18770*, 2025.

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

Yang Gu, Hengyu You, Jian Cao, Muran Yu, Haoran Fan, and Shiyou Qian. Large language models for constructing and optimizing machine learning workflows: A survey. *arXiv preprint arXiv:2411.10478*, 2024.

Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. Model context protocol (mcp): Landscape, security threats, and future research directions. *arXiv preprint arXiv:2503.23278*, 2025.

Changwu Huang, Zeqi Zhang, Bifei Mao, and Xin Yao. An overview of artificial intelligence ethics. *IEEE Transactions on Artificial Intelligence*, 4(4):799–819, 2022.

Kexin Huang, Ying Jin, Ryan Li, Michael Y Li, Emmanuel Candès, and Jure Leskovec. Automated hypothesis validation with agentic sequential falsifications. *arXiv preprint arXiv:2502.09858*, 2025.

Intology. Zochi technical report. https://www.intology.ai/blog/zochi-tech-report, March 2025. Blog post. Updated May 27, 2025. Accessed May 7, 2026.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

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

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

Anna Jobin, Marcello Ienca, and Effy Vayena. The global landscape of ai ethics guidelines. *Nature machine intelligence*, 1(9):389–399, 2019.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Arif Ali Khan, Sher Badshah, Peng Liang, Muhammad Waseem, Bilal Khan, Aakash Ahmad, Mahdi Fahmideh, Mahmood Niazi, and Muhammad Azeem Akbar. Ethics of ai: A systematic literature review of principles and challenges. In *Proceedings of the 26th international conference on evaluation and assessment in software engineering*, pp. 383–392, 2022.

Ross D King, Jem Rowland, Stephen G Oliver, Michael Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pinar Pir, Larisa N Soldatova, et al. The automation of science. *Science*, 324(5923):85–89, 2009.

Keigo Kusumegi, Xinyu Yang, Paul Ginsparg, Mathijs de Vaan, Toby Stuart, and Yian Yin. Scientific production in the era of large language models. *Science*, 390(6779):1240–1243, 2025. doi: 10.1126/science.adw3000. URL https://www.science.org/doi/abs/10.1126/science.adw3000. _eprint: https://www.science.org/doi/pdf/10.1126/science.adw3000.

P Langley. *Scientific discovery: Computational explorations of the creative processes*. MIT Press, 1987.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Zelong Li, Shuyuan Xu, Kai Mei, Wenyue Hua, Balaji Rama, Om Raheja, Hao Wang, He Zhu, and Yongfeng Zhang. Autoflow: Automated workflow generation for large language model agents. *arXiv preprint arXiv:2407.12821*, 2024.

Zhuofeng Li, Dongfu Jiang, Xueguang Ma, Haoxiang Zhang, Ping Nie, Yuyu Zhang, Kai Zou, Jianwen Xie, Yu Zhang, and Wenhu Chen. Openresearcher: A fully open pipeline for long-horizon deep research trajectory synthesis. *arXiv preprint arXiv:2603.20278*, 2026.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. volume 36, pp. 21558–21572, 2023.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292v3*, 2024. URL https://www.arxiv.org/abs/2408.06292v3.

Ziming Luo, Atoosa Kasirzadeh, and Nihar B Shah. The more you automate, the less you see: Hidden pitfalls of ai scientist systems. *arXiv preprint arXiv:2509.08713*, 2025.

Ruiying Ma, Chieh-Jan Mike Liang, Yanjie Gao, and Francis Y. Yan. Metamuse: Algorithm generation via creative ideation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qJsShqQgWw.

Indrajeet Mandal, Jitendra Soni, Mohd Zaki, Morten M Smedskjaer, Katrin Wondraczek, Lothar Wondraczek, Nitya Nand Gosvami, and NM Anoop Krishnan. Evaluating large language model agents for automation of atomic force microscopy. *Nature Communications*, 16(1):9104, 2025.

Junhyuk Oh, Gregory Farquhar, Iurii Kemaev, Dan A. Calian, Matteo Hessel, Luisa Zintgraf, Satinder Singh, Hado van Hasselt, and David Silver. Discovering state-of-the-art reinforcement learning algorithms. *Nature*, 648(8093):312–319, December 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09761-x. URL https://doi.org/10.1038/s41586-025-09761-x.

Harshith Padigela, Chintan Shah, and Dinkar Juyal. Ml-dev-bench: Comparative analysis of ai agents on ml development workflows, 2025. URL https://arxiv.org/abs/2502.00964.

Haining Pan, James V Roggeveen, Erez Berg, Juan Felipe Carrasquilla Alvarez, Debanjan Chowdhury, Surya Ganguli, Federico Ghimenti, Juraj Hasik, Henry S. Hunt, Hong-Chen Jiang, Mason Kamb, Ying-Jer Kao, Ehsan Khatami, Michael J Lawler, Di Luo, Titus Neupert, Xiaoliang Qi, Michael Brenner, and Eun-Ah Kim. CMT-benchmark: A benchmark for condensed matter theory built by expert researchers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=pX6B28ynNh.

Karl Popper. *Conjectures and refutations: The growth of scientific knowledge*. routledge, 2014.

Shi Qiu, Junyi Deng, Yiwei Deng, Haoran Dong, Jieyu Fu, Mao Li, Zeyu Li, Zhaolong Zhang, Huiwen Zheng, Leidong Bao, et al. Prbench: End-to-end paper reproduction in physics research. *arXiv preprint arXiv:2603.27646*, 2026.

Marissa Radensky, Simra Shahid, Raymond Fok, Pao Siangliulue, Tom Hope, and Daniel S Weld. Scideator: Human-llm scientific idea generation grounded in research-paper facet recombination. *arXiv preprint arXiv:2409.14634*, 2024.

Partha Pratim Ray. A survey on model context protocol: Architecture, state-of-the-art, challenges and future directions. *Authorea Preprints*, 2025.

Dingyi Rong, Zijian Chen, Qi Jia, Kaiwei Zhang, Haotian Lu, Guangtao Zhai, and Ning Liu. Liveproteinbench: A contamination-free benchmark for assessing models' specialized capabilities in protein science. *arXiv preprint arXiv:2512.22257*, 2025.

Simone Rossi, Valter Fraccaro, and Riccardo Manzotti. The brain side of human-ai interactions in the long-term: the "3r principle". *npj Artificial Intelligence*, 2(1):15, 2026.

Christoph Scheurer and Karsten Reuter. Role of the human-in-the-loop in emerging self-driving laboratories for heterogeneous catalysis. *Nature Catalysis*, 8(1):13–19, 2025.

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

Chenglei Si, Tatsunori Hashimoto, and Diyi Yang. The ideation-execution gap: Execution outcomes of llm-generated versus human research ideas, 2025. URL https://arxiv.org/abs/2506.20803.

Zachary S Siegel, Sayash Kapoor, Nitya Nagdir, Benedikt Stroebl, and Arvind Narayanan. Corebench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *arXiv preprint arXiv:2409.11363*, 2024.

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

Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Fei Huang, and Yan Zhang. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025.

Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, and James Zou. The Virtual Lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 646(8085):716–723, October 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09442-9. URL https://doi.org/10.1038/s41586-025-09442-9.

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

Xiangru Tang, Qiao Jin, Kunlun Zhu, Tongxin Yuan, Yichi Zhang, Wangchunshu Zhou, Meng Qu, Yilun Zhao, Jian Tang, Zhuosheng Zhang, et al. Risks of ai scientists: prioritizing safeguarding over autonomy. *Nature Communications*, 16(1):8317, 2025.

Milena Tsvetkova, Taha Yasseri, Niccolo Pescetelli, and Tobias Werner. A new sociology of humans and machines. *Nature Human Behaviour*, 8(10):1864–1876, October 2024. ISSN 2397-3374. doi: 10.1038/s41562-024-02001-8. URL https://doi.org/10.1038/s41562-024-02001-8.

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

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. Cycleresearcher: Improving automated research via automated review. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=bjcsVLoHYs.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. CycleResearcher: Improving Automated Research via Automated Review. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=bjcsVLoHYs.

Yixuan Weng, Minjun Zhu, Qiujie Xie, QiYao Sun, Zhen Lin, Sifan Liu, and Yue Zhang. Deepscientist: Advancing frontier-pushing scientific findings progressively. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=cZFgsLq8Gs.

Yanzheng Xiang, Hanqi Yan, Shuyin Ouyang, Lin Gui, and Yulan He. Scireplicate-bench: Benchmarking LLMs in agent-driven algorithmic reproduction from research papers. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=8LoPjpvWde.

Qiujie Xie, Yixuan Weng, Minjun Zhu, Fuchen Shen, Shulin Huang, Zhen Lin, Jiahui Zhou, Zilan Mao, Zijie Yang, Linyi Yang, et al. How far are ai scientists from changing the world? *arXiv preprint arXiv:2507.23276*, 2025.

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

Shuo Yan, Ruochen Li, Ziming Luo, Zimu Wang, Daoyang Li, Liqiang Jing, Kaiyu He, Peilin Wu, Juntong Ni, George Michalopoulos, et al. Lmr-bench: Evaluating llm agent's ability on reproducing language modeling research. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6175–6197, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024a.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793v3*, 2024b. URL https://www.arxiv.org/abs/2405.15793v3.

Yingxuan Yang, Huacan Chai, Yuanyi Song, Siyuan Qi, Muning Wen, Ning Li, Junwei Liao, Haoyi Hu, Jianghao Lin, Gaowei Chang, et al. A survey of ai agent protocols. *arXiv preprint arXiv:2504.16736*, 2025b.

Zijie Yang, Yongjing Yin, Chaojun Kong, Tiange Chi, Wufan Tao, Yue Zhang, and Tian Xu. Shennongalpha: an ai-driven sharing and collaboration platform for intelligent curation, acquisition, and translation of natural medicinal material knowledge. *Cell Discovery*, 11(1):32, 2025c.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, et al. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*, 2025.

Yujun Zhou, Jingdong Yang, Yue Huang, Kehan Guo, Zoe Emory, Bikram Ghosh, Amita Bedar, Sujay Shekar, Zhenwen Liang, Pin-Yu Chen, et al. Labsafety bench: Benchmarking llms on safety issues in scientific labs. *arXiv preprint arXiv:2410.14182*, 2024.

Kunlun Zhu, Jiaxun Zhang, Ziheng Qi, Nuoxing Shang, Zijia Liu, Peixuan Han, Yue Su, Haofei Yu, and Jiaxuan You. Safescientist: Enhancing ai scientist safety for risk-aware scientific discovery. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2289–2317, 2025a.

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

Xiangyang Zhu, Yuan Tian, Qi Jia, Kaiwei Zhang, Zicheng Zhang, Chunyi Li, Kaiyuan Ji, Dongrui Liu, Zijian Chen, Lu Sun, et al. Safesci: Safety evaluation of large language models in science domains and beyond. *arXiv preprint arXiv:2603.01589*, 2026.

A Ethical Considerations

As autonomous research agents, AI Scientists lack moral constraints. They are incapable of making ethical judgments about the societal impact of their work, and they do not self-regulate based on potential risks associated with their findings (Bengio et al., 2025). As AI Scientists possess stronger capabilities in idea generation and experiment execution, their influence on scientific research and society could far surpass that of current LLMs and scientific tools (e.g., AutoSurvey (Wang et al., 2024)). In the absence of proper oversight, **AI Scientists may: (1) be misused**, overwhelming the peer review system, leading to a decline in overall research quality; **(2) enter unethical or dangerous research domains**, autonomously generating and publishing sensitive findings that accelerate the development of harmful technologies; **(3) weaken the quality of PhD training**, leading to a decline in human research standards and overall scientific literacy. To prevent the above situations, **AI Scientists are in urgent need of a comprehensive system for generation management and quality evaluation**, thus enabling behavior regulation within the human moral framework (Jobin et al., 2019). This system should include, but not be limited to, the following components:

(1) Implement measures to prevent AI-generated content from disrupting human review systems: Effective strategies should be adopted to ensure that AI-generated articles do not interfere with human peer-review systems while maintaining high standards of quality (Kusumegi et al., 2025). This includes establishing a centralized platform to archive scientific outputs generated by AI Scientists (Bianchi et al., 2026), developing automated detection systems to identify such content, and creating specialized evaluation tools (e.g., DeepReview (Zhu et al., 2025b)) to assess the quality of AI-generated research outputs. These tools should help identify and filter low-quality content, thereby reducing the burden on the peer review process. All AI-generated outputs must be transparently labeled and reviewed, including information on their origin, generation methods, and scientific tools.

(2) Establish boundaries and strengthen training programs: Implement clear boundaries between human-led and AI-led research processes to ensure that PhD students receive comprehensive training.

Key components of doctoral education(e.g., idea testing), should prioritize human involvement to maintain high standards of scientific literacy. Additionally, guidelines should be established to prevent over-reliance on AI Scientists in PhD training, ensuring that AI tools serve as supplements rather than substitutes in the educational process (Tsvetkova et al., 2024).

(3) Formulate an ethics and responsibility convention: A global convention should be established to define the ethical boundaries and risk management principles for AI-driven research (Huang et al., 2022). All researchers and institutions utilizing AI Scientists must fully disclose the generation process, algorithmic sources, training data, and potential societal risks of their findings. Additionally, a hybrid mechanism combining automated and human-in-the-loop review should be implemented for continuous ethical oversight and risk evaluation, ensuring that AI Scientist research activities remain within socially safe boundaries (Jobin et al., 2019; Khan et al., 2022). Furthermore, appropriate legislation should be developed to regulate AI Scientists by imposing strict limitations on their use for specific research purposes.

B DEEPVERIFY-5K

DeepVerify-5K is a large-scale dataset designed to empirically characterize verification behaviors in modern AI Scientist workflows under a realistic, up-to-date toolchain. We build the dataset on top of DeepScientist (Weng et al., 2026), a SoTA AI Scientist system for automated AI research. To enhance generality and reduce obsolescence, we use Gemini-2.5-Pro as the idea generator and evaluator, and a strong coding agent based on Claude-4-Opus as the primary verification tool. Concretely, we run DeepScientist on three frontier AI research tasks: **(1) Agent Failure Attribution**, where the goal is to analyze a failed episode in an LLM-based multi-agent system and pinpoint which agent and which step were decisively responsible for the failure. This task is an essential capability for debugging complex agent pipelines; **(2) LLM Inference Acceleration**, a highly optimized research area focused on maximizing throughput and reducing latency during LLM inference; and **(3) AI Text Detection**, a binary classification task that determines whether a given text potentially containing LLM-generated content and additional noise was produced by a human or an LLM.

For each task, DeepScientist is instructed to autonomously generate candidate ideas, assess their quality, implement high-quality ideas into executable code, run the code via Claude Code, and propose new scientific ideas based on the experimental results. Each iteration yields a complete trace of the scientific discovery process, capturing fine-grained verification details spanning planning, implementation, execution, and iterative revision. All models are deployed on two servers (each equipped with 8 NVIDIA H800 GPUs), costing a month-long cycle of exploration, validation, and iteration. As shown in Table 5, DeepVerify-5K contains 4,890 records of end-to-end scientific discovery trajectories. Each trace includes the complete logs for both idea generation and verification, along with the corresponding idea artifacts and the code produced from translating those ideas into executable implementations. We further present a concrete example trace in Figure 7 to illustrate what a single DeepVerify-5K data sample looks like in practice. During the discovery process, we use Gemini-2.5-Pro to assess generated ideas; low-quality ideas are filtered out, resulting in a curated set of 1,109 ideas that are subsequently implemented and executed via Claude Code. These traces collectively serve as the empirical foundation for our analysis of AI Scientist verification capability.

Scope of DeepVerify-5K. DeepVerify-5K focuses on computational AI research workflows rather than wet-lab or all-domain scientific discovery. We use it as a process-level diagnostic dataset because current AI Scientist systems are most mature in executable computational settings, where complete trajectories can be collected and audited. Therefore, our analysis of DeepVerify-5K should be interpreted as direct evidence for verification fragility in computational AI research, and as a methodological template for studying verification in broader scientific domains.

C TYPES OF ERRORS

We categorize failures into the following classes:

- **ENV** (*environment/dependency*): issues related to the runtime environment or dependencies, e.g., Conda setup, environment variables, path misconfiguration, or `ImportError`.

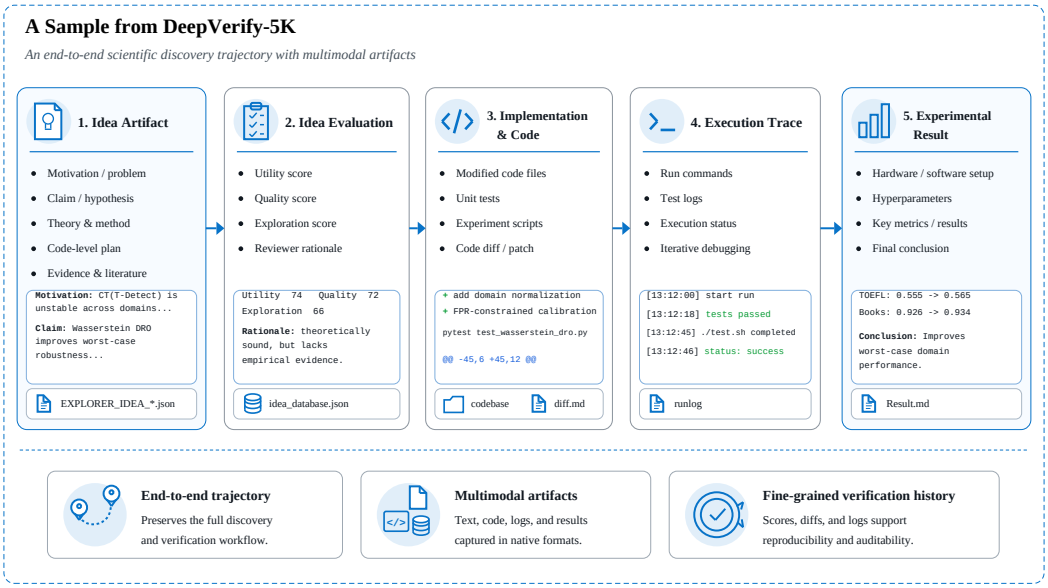


Figure 6: An example trace from DeepVerify-5K.

- **DATA** (*data/input*): issues with datasets or input files, e.g., missing data, malformed formats, or I/O/read failures.
- **EXEC** (*execution*): issues during execution, e.g., timeouts, abnormal process termination, or non-zero exit codes.
- **TOOL** (*tooling/config*): issues with scripts, configurations or paths, e.g., missing scripts or absent files.
- **RUNTIME** (*runtime exception*): runtime logic errors, e.g., `AssertionError`, `TypeError`.
- **TEST** (*test failure*): failures in testing, where assertions do not pass.
- **RESOURCE** (*resource limit*): resource-related issues, e.g., insufficient GPU memory or out-of-memory (OOM) errors.
- **OTHER** (*uncategorized*): errors outside the categories or cases where context is insufficient for reliable classification.

Workspace Sample:

[Idea content (EXPLORER_IDEA_2_IDEA-rwza.json)]
Motivation
The current CT(T-Detect) approach is unstable across domains/languages, with catastrophic OOD performance (e.g., Arabic 0.582 AUROC, TOEFL 0.555 AUROC). The root cause is average-risk training rather than worst-case robustness constraints.

Claim
Reformulating detector fusion as a Wasserstein DRO problem with domain-specific FPR constraints yields consistent cross-domain performance while enforcing strict FPR guarantees per context.

Theory / Method (core equations)
Wasserstein DRO worst-case objective:
\min_{f \in \mathcal{F}} \sup_{\mathbb{Q} \in \mathcal{U}_\rho(\mathbb{P})}
\mathbb{E}_{\mathbb{Q}}[\ell(f(\phi(X), \phi(C)), Y)]
Constraint: \mathbb{E}_{\mathbb{Q}_k}[\mathbb{I}[f(\phi(X), \phi(C)) = 1] \mid Y=0] \leq \epsilon,\ \forall k

Code-level plan (summary)
- Domain-aware normalization in `scripts/delegate_detector.py` with robust clipping
- Wasserstein DRO fusion via domain-weighted training
- Domain statistics collection during preparation
- Dynamic threshold calibration with per-domain FPR constraints

Relation to literature (summary)
- Saberi et al. (2024): Wasserstein DRO for gradual domain adaptation
- Imtiaz & Khan (2023): DRO for cross-domain performance constraints
- Gao & Kleywegt (2022): dual formulation for Wasserstein DRO

Evidence (original items)
1. Saberi et al. (2024) compatibility measure supports domain normalization
2. Imtiaz & Khan (2023) shows DRO can preserve cross-domain constraints
3. Gao & Kleywegt (2022) provides efficient Wasserstein DRO dual computation

[Review scores and rationale (idea_database.json)]
- Utility: **74**
- Quality: **72**
- Exploration: **66**
- Rationale (summary): targets a clear cross-domain failure, is theoretically sound, but lacks empirical evidence; expected improvement estimated at 10–15% AUROC, below higher-impact thresholds.

[Implementation process and changes (runlog + diff)]

Runlog highlights (IDEA-rwza.runlog)
- Start time: `[13:12:00]`
- Claude Code modifies code and runs tests
- Unit tests: `python -m pytest test/test_wasserstein_dro.py -v` (passed)
- Full experiment: `./test.sh` (completed)
[Target Value]: {target_value}

Major code changes (impl_IDEA-rwza-diff.md)
- `scripts/delegate_detector.py`: domain stats, Wasserstein normalization, FPR-constrained calibration
- `test/test_wasserstein_dro.py`: unit tests
- `test.sh`: experiment + logging script
- `Result.md`: experimental report

[Experimental setup and results (impl_IDEA-rwza/Result.md)]
Hardware
- CPU: AMD EPYC 7542
- GPU: 2×NVIDIA H100 80GB
- Memory: 503GB
- CUDA: 12.4

Software
- OS: Linux 5.14.0-427.13.1.el9_4.x86_64
- Python: 3.11.11
- PyTorch: 2.7.0
- Transformers: 4.53.1
- NumPy: 2.2.6

Core hyperparameters
- Wasserstein ρ=1.0
- FPR constraint ε=0.05
- DRO regularization λ=0.1

Key results (CT(t-detect))
- TOEFL: **0.555 → 0.565** (+0.010 AUROC)
- Books: **0.926 → 0.934** (+0.008 AUROC)
- News: **0.893 → 0.905** (+0.012 AUROC)
- Poetry: **0.898 → 0.909** (+0.011 AUROC)

Overall conclusion (Result.md summary)
Wasserstein DRO improves most domains while enforcing FPR constraints; the aggregated "ALL" metric slightly decreases, consistent with prioritizing worst-case domain performance.

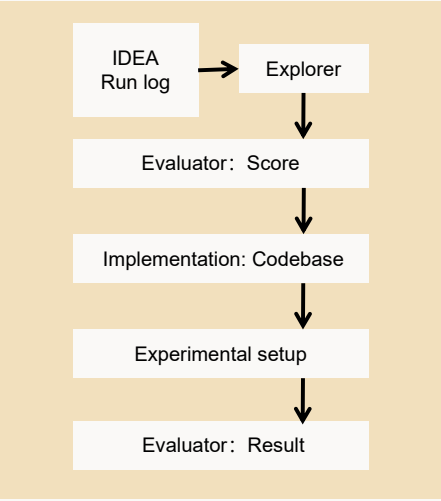


Figure 7: Another example trace from DeepVerify-5K.

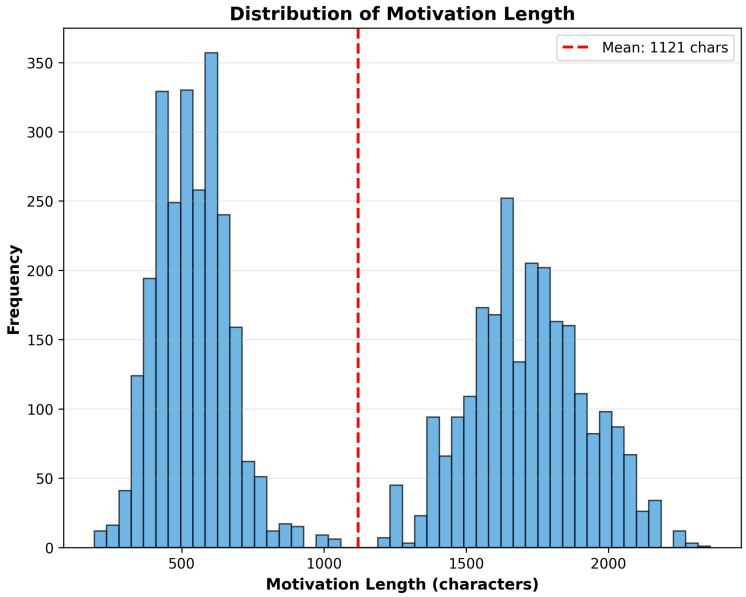


Figure 8: The motivation length distribution of ideas from DeepVerify-5K.

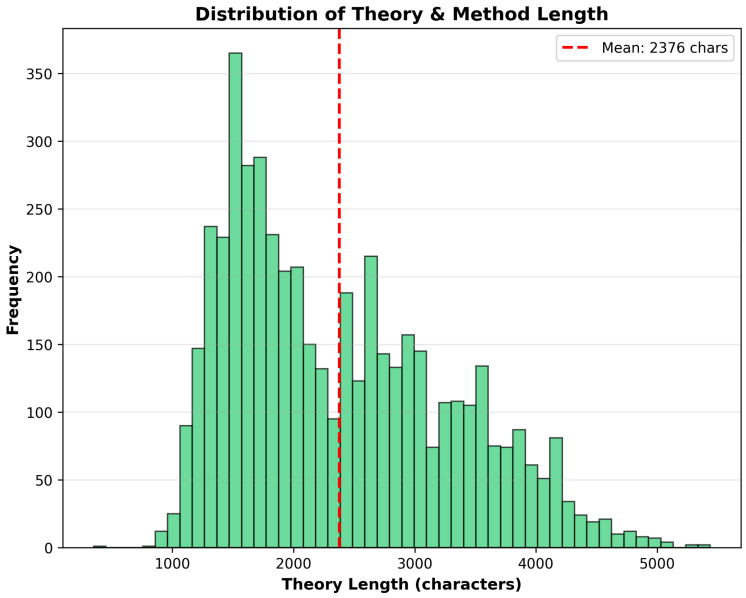


Figure 9: The theory and method length distribution of ideas from DeepVerify-5K.

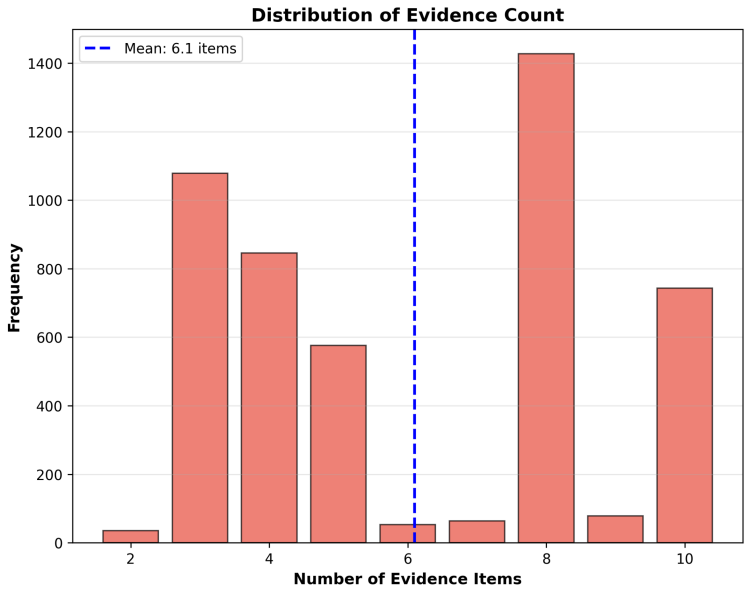


Figure 10: The evidence number of ideas from DeepVerify-5K.

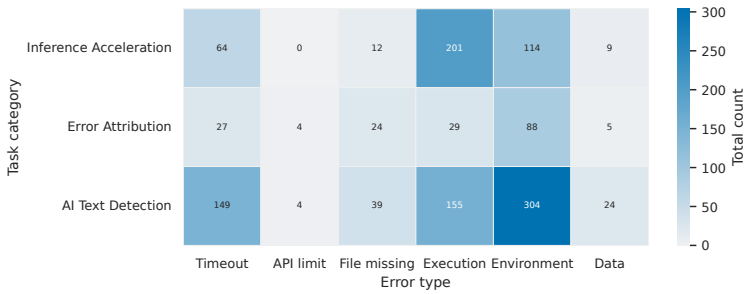


Figure 11: Verification error counts in DeepVerify-5K across task categories, including both recovered errors and those that cause failure. Raw log messages are normalized into eight categories.

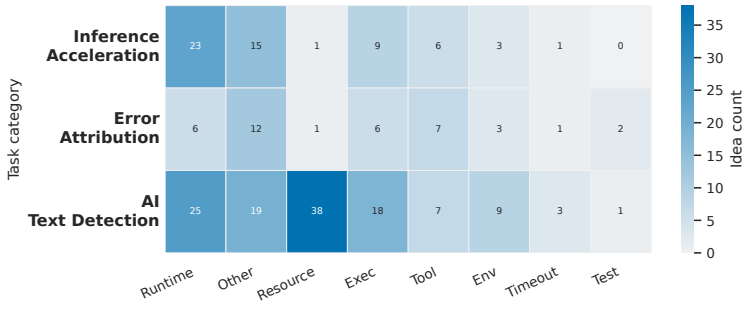


Figure 12: Terminal (unrecovered) verification errors in DeepVerify-5K that lead to failure, highlighting unresolved failure modes after automated debugging and iteration. Raw log messages are normalized into eight categories.