# RIGOR: DUAL GOVERNANCE FOR RESEARCH INTENT INTEGRITY IN LONG-HORIZON AUTONOMOUS RESEARCH

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Long-horizon autonomous research requires agents to maintain project-level scientific intent while repeatedly designing, executing, and interpreting experiments. Persistent single-agent loops often mix long-term research objectives with transient implementation details and debugging histories, creating substantial context accumulation. Decomposing the workflow into a persistent Research Agent and transient Execution Agents can reduce this burden, but it also introduces new handoff boundaries where execution fidelity may be compromised and unsupported evidence may enter the scientific state. We formalize Research Intent Integrity (RII) as the requirement that material deviations become part of the authoritative research state only when they are authorized, justified, and provenance-preserving. Based on this principle, we introduce a dual-governance architecture consisting of an Intent-Grounded Research Loop (IGRL) for governing evidence-to-state updates and research termination, and Scientific Execution Governance (SEG) for binding experiment briefs with candidate implementations, formal executions, and sealed evidence receipts. We instantiate these mechanisms in RIGOR (Research Intent Governance for Open-ended Research), an end-to-end role-decomposed AutoResearch system for long-horizon, evidence-driven discovery. Execution-level evaluations show that SEG improves experimental fidelity and prevents unsupported deviations from entering canonical evidence with modest computational overhead. Long-horizon studies further demonstrate that role decomposition reduces persistent context accumulation, while integrating IGRL and SEG preserves research intent throughout iterative research updates compared with ungoverned architectures. Evaluations on public AutoResearch benchmarks show that RIGOR maintains competitive end-to-end performance while providing stronger control over scientific state transitions.

## 1 INTRODUCTION

Autonomous research agents aim to automate iterative scientific workflows, including hypothesis formulation, experiment execution, evidence interpretation, and research-plan updates Lu et al. (2026); Baek et al. (2025); Weng et al. (2025a). A common design choice is to maintain the entire workflow within a single persistent agent. However, scientific research is inherently long-horizon. In such settings, high-level research objectives can become mixed with accumulated execution histories, including failed patches, tool outputs, and experimental logs. Prior studies show that irrelevant context can impair LLM reasoning Shi et al. (2023), and that information utilization becomes less reliable as context length increases Liu et al. (2024). For long-running AutoResearch systems, accumulated context can obscure the original research question, experimental controls, and evaluation criteria, leading to research-intent drift (Figure 1a).

A natural solution is role decomposition: a persistent *Research Agent* (RA) maintains the scientific state, while transient *Execution Agents* (EAs) perform bounded experiments (Figure 1b). Existing systems have explored specialized research roles, including researcher-engineer, mentor-researcher-reviewer, and coordinator-executor structures Zhao et al. (2026); Xiong et al. (2026); Lyu et al. (2026); Jin et al. (2026). Although RA-EA decomposition reduces persistent context growth (Figure 1d), it introduces two handoff boundaries (Figure 1c).



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.

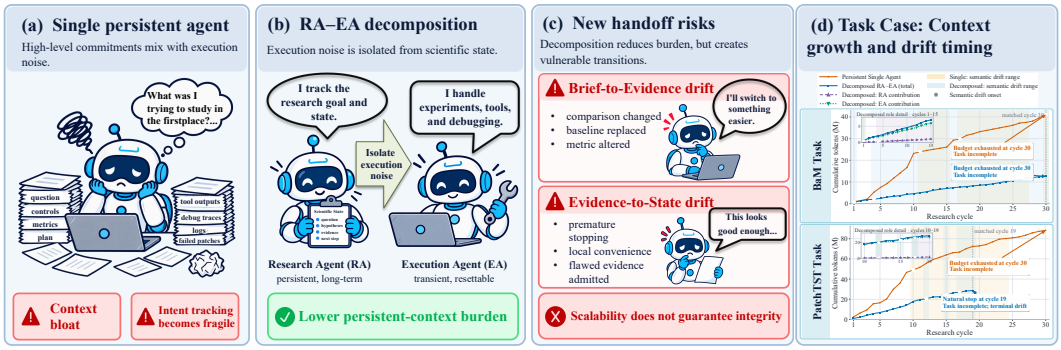



Figure 1: Architectural trade-offs of role decomposition. **(a-b)** A single agent accumulates execution-related context, whereas a persistent Research Agent (RA) and transient Execution Agents (EAs) isolate long-term reasoning from low-level execution histories. **(c)** This decomposition introduces handoff boundaries where research intent may be altered. **(d)** Tracking on two long-horizon tasks shows that RA-EA decomposition reduces context growth but does not eliminate research-intent drift.

First, the **Brief-to-Evidence** boundary creates an execution-fidelity trade-off. With broad execution freedom, an EA may unintentionally modify interventions, baselines, or metrics due to implementation constraints, producing evidence for an unintended experiment. Previous studies on research reproduction and long-horizon coding show that executable outputs may deviate from intended specifications Zhao et al. (2026); Starace et al. (2025); Yan et al. (2026). Conversely, overly restricting EA autonomy increases communication overhead and reintroduces execution details into the persistent context.

Second, role decomposition does not guarantee a reliable **Evidence-to-State** transition. The RA may update scientific conclusions from invalid evidence, select directions inconsistent with the original objective, or terminate research prematurely. Therefore, context reduction alone is insufficient; both transitions require explicit governance.

We formalize these failures as violations of **Research Intent Integrity (RII)**. RII allows research objectives to evolve with new evidence, while requiring that human-authorized intent remains the reference for subsequent updates. A material deviation affecting execution, evidence admission, scientific-state updates, or termination decisions should become part of the authoritative research process only when it is authorized, justified, and provenance-preserving. This definition separates valid evidence-driven iteration from deviations that compromise scientific conclusions. We further introduce formal metrics and an LLM-as-a-Judge protocol to measure RII violations.

To enforce RII, we introduce **RIGOR** (**R**esearch **I**ntent **G**overnance for **O**pen-ended **R**esearch), a role-decomposed AutoResearch system with dual governance over the two handoff boundaries. **Scientific Execution Governance (SEG)** governs the *Brief-to-Evidence* transition by binding execution to a precommitted Experiment Brief and preserving artifact provenance. **Intent-Grounded Research Loop (IGRL)** governs the *Evidence-to-State* transition by validating evidence interpretation, research-state updates, and stopping decisions. Together, SEG preserves execution fidelity, while IGRL preserves evidence-grounded research evolution.

We evaluate RIGOR on fixed-Brief execution episodes, open-ended long-horizon research trajectories, and public AutoResearch benchmarks, including MLR-BENCH, PAPERBENCH, EXP-BENCH, and ASI-BENCH Chen et al. (2025a); Starace et al. (2025); Kon et al. (2025); Zhou et al. (2026). Results show that SEG improves execution fidelity by rejecting unsupported deviations with modest inference cost. Across long-horizon trajectories, IGRL and SEG reduce realized integrity violations while maintaining autonomous research progress. Under matched conditions, RIGOR achieves competitive end-to-end performance among compared systems.

Our contributions are threefold:

- We formalize **Research Intent Integrity (RII)**, which distinguishes valid evidence-driven evolution from material deviations affecting execution, evidence admission, scientific-state

Table 1: The dual-governance architecture in RIGOR. SEG governs execution fidelity and provenance preservation. IGRL governs evidence interpretation, state updates, and stopping decisions.

| Boundary | Input | Governed Output | Governance |
|---|---|---|---|
| Brief-to-Evidence | Experiment Brief, Candidate Implementation, Execution Proposal | Canonical Execution, Artifacts, Sealed Receipt | SEG |
| Evidence-to-State | Research Intent, Execution Evidence, Proposed RA Claims | Validated State Update or Stop Decision | IGRL |

updates, or termination decisions. We introduce formal metrics and an LLM-as-a-Judge protocol for measuring RII violations.

- We introduce **RIGOR**, a role-decomposed AutoResearch system with dual governance. SEG governs the Brief-to-Evidence transition, while IGRL governs the Evidence-to-State transition through evidence validation and intent-grounded planning.
- We evaluate RIGOR on fixed-Brief episodes, long-horizon research trajectories, and public AutoResearch benchmarks. Results show reduced context burden, improved execution fidelity, and fewer realized RII violations while maintaining competitive end-to-end performance.

## 2 Related Work

### 2.1 Autonomous Research Agents and Role Specialization

LLM agents increasingly automate scientific workflows, ranging from early ideation frameworks Baek et al. (2025) to end-to-end discovery pipelines Lu et al. (2026); Weng et al. (2025a); Tang et al. (2025); Weng et al. (2025b). To manage workflow complexity, recent architectures frequently adopt role specialization and multi-agent collaboration Zhao et al. (2026); Xiong et al. (2026); Lyu et al. (2026); Gao et al. (2026); Jin et al. (2026); Shen et al. (2026). However, these systems primarily optimize final artifact quality or end-to-end completion rates, and often treat inter-agent interactions as loosely structured conversational handoffs. They generally lack explicit structural governance over the boundary between persistent scientific reasoning and transient experimental execution, and thus do not directly address intent drift during long-horizon operation.

### 2.2 Evaluation, Experimental Fidelity, and Evidence-Grounded Governance

A growing ecosystem of benchmarks evaluates autonomous agents across iterative experimentation Huang et al. (2024); Chan et al. (2025), data-driven discovery Chen et al. (2025b); Majumder et al. (2025), and open-ended research Chen et al. (2025a). Specialized reproduction benchmarks further indicate that successfully generating code or obtaining numerical results does not ensure faithful realization of the intended experiment Starace et al. (2025); Kon et al. (2025); Edwards et al. (2026); Zhao et al. (2026). While recent frameworks introduce persistent specifications or evidence-gated controls to improve execution reliability Yan et al. (2026); Huang et al. (2026), they focus predominantly on individual execution phases. Existing work thus lacks a unified governance model for the bidirectional scientific loop, and offers limited diagnostic metrics for quantifying semantic deviations across both the Brief-to-Evidence and Evidence-to-State transitions.

## 3 Method

### 3.1 Problem Setup and Research Intent Integrity

We model autonomous research as an iterative state-update process. At cycle $t$, the system maintains a human-authorized research intent $I_t^H$ and a canonical scientific state $S_t$. The RA generates an Experiment Brief $B_t$, based on which the EA produces a candidate implementation $C_t$ and execution

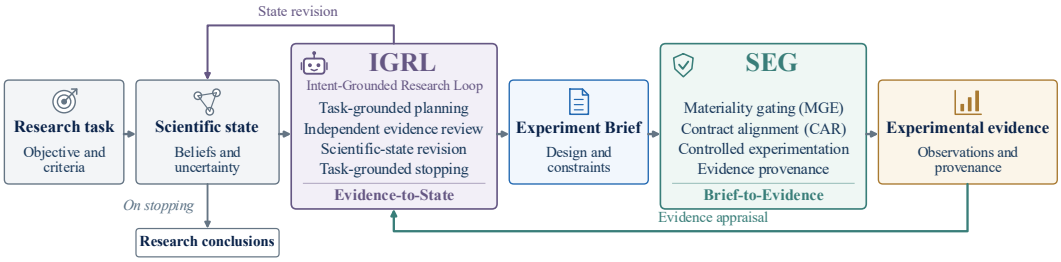



Figure 2: **The dual-governance architecture of RIGOR.** IGRL anchors planning to the current research intent and governs evidence-based scientific-state revision. SEG governs the transition from an Experiment Brief to execution-verified evidence.

evidence $E_t$ after formal execution. The RA then proposes a state update $\widehat{\Delta}_t$. The scientific state evolves as

$$S_{t+1} = \begin{cases} \mathrm{Commit}(S_t, \widehat{\Delta}_t), & \text{if } \widehat{\Delta}_t \text{ is validated,} \\ S_t, & \text{otherwise.} \end{cases} \tag{1}$$

Here, $\mathrm{Commit}$ applies a validated update to the canonical scientific state, and validation is performed by the governance procedures described in Section 3.2.

The key requirement is not that research objectives remain fixed. Scientific hypotheses and plans may evolve with new evidence. Instead, we define **Research Intent Integrity (RII)** as the requirement that material deviations become part of the authoritative research process only when they are authorized, justified, and provenance-preserving.

For a transition $\tau$ (e.g., $B_t \to E_t$ or $E_t \to S_{t+1}$), let $d$ denote a deviation from the authorized process. We define a **Material Integrity Failure (MIF)** as:

$$\mathrm{MIF}(d) = \mathrm{MaterialDeviation}(d) \wedge \left[\neg A_\tau(d) \vee \neg J_\tau(d) \vee \neg P(d)\right], \tag{2}$$

where $\mathrm{MaterialDeviation}(d)$ indicates that $d$ affects experimental specifications, evidence admission, scientific-state updates, or termination decisions. $A_\tau(d)$ indicates whether the deviation is authorized by $I_t^H$ or governance procedures. $J_\tau(d)$ indicates whether the deviation is supported by validated evidence or a documented rationale. $P(d)$ indicates whether provenance is preserved across intent, brief, candidate, execution, evidence, and state.

RII distinguishes transient execution errors from integrity failures. A deviation becomes a realized RII violation only when it propagates into the authoritative research record:

$$\mathrm{RealizedRII}(d) = \mathrm{MIF}(d) \wedge \left[\mathrm{CE}(d) \vee \mathrm{AE}(d) \vee \mathrm{CS}(d)\right]. \tag{3}$$

Here, $\mathrm{CE}(d)$ (canonical execution), $\mathrm{AE}(d)$ (admitted evidence), and $\mathrm{CS}(d)$ (committed scientific state) indicate whether the deviation enters the canonical execution record, the admitted evidence pool, or the committed scientific state, respectively.

### 3.2 RIGOR: Dual Governance Architecture

To preserve RII, RIGOR decouples autonomous research into a persistent RA for project-level reasoning and transient EAs for bounded execution. The decomposition introduces two handoff boundaries: the Brief-to-Evidence transition and the Evidence-to-State transition. RIGOR applies dual governance over these transitions.

SEG governs the Brief-to-Evidence transition by binding execution to an Experiment Brief and preserving execution provenance. IGRL governs the Evidence-to-State transition by validating evidence interpretation, state updates, and stopping decisions. Together, SEG and IGRL aim to en-

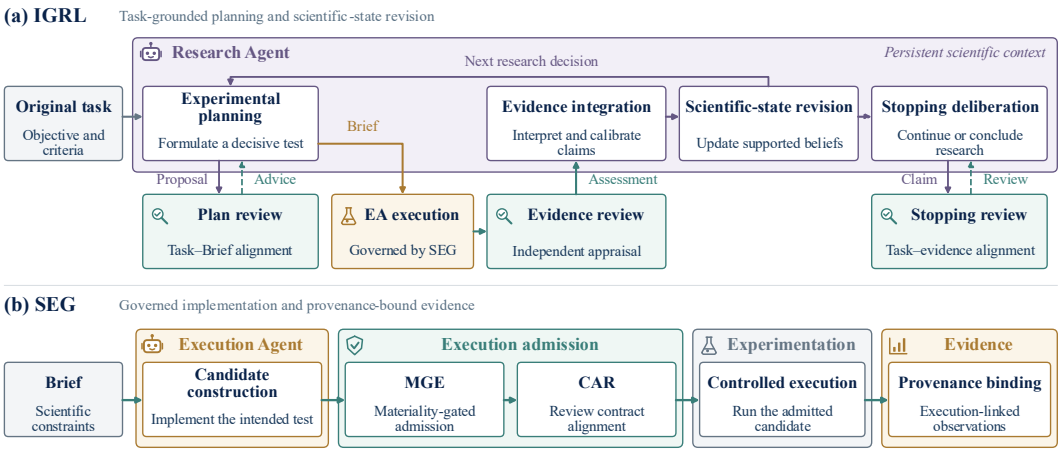



Figure 3: **(a)** IGRL governs planning, evidence integration, and scientific-state updates. **(b)** SEG enforces execution fidelity through MGE, CAR, controlled execution, and provenance capture.

sure that experimental evidence faithfully reflects authorized intent and that state updates remain grounded in verified evidence.

The two modules interact through a compact interface: the RA issues an Experiment Brief, and the EA returns execution artifacts together with a Sealed Receipt containing execution facts and provenance.

### 3.2.1 Intent-Grounded Research Loop (IGRL)

IGRL (Figure 3a) governs how evidence updates the scientific state and how subsequent research decisions are made, anchoring both to the original *Task*.

**Brief Review.** Before EA execution, a Plan Reviewer checks whether the proposed Brief remains aligned with $I_t^H$ and preserves required experimental comparisons.

**Evidence Review and Epistemic Admission.** To reduce the risk of the RA being biased by unstructured EA narratives, a blind reviewer evaluates execution artifacts and the Sealed Receipt against the corresponding Brief. The review determines whether the execution supports, contradicts, or fails to address the hypothesis. Failed executions that do not produce valid evidence are returned as execution facts for replanning. For eligible executions, the RA integrates the review result and proposes a structured state update. Epistemic admission rules prevent unsupported or rejected executions from influencing scientific-state updates, hypothesis revision, or implementation adoption.

**Stopping Review.** When the RA proposes termination, a Stop Reviewer evaluates the stopping decision against accumulated evidence and the original *Task* objective. The system stops only when termination is supported by the validated scientific state rather than local execution difficulties. Implementation details are provided in Section B.

### 3.2.2 Scientific Execution Governance (SEG)

SEG (Figure 3b) governs the Brief-to-Evidence transition by binding EA execution to the authorized Brief. It applies two-stage governance before evidence generation.

**Candidate Binding and Review.** MGE first enforces structural compliance by rejecting unauthorized workspace modifications before execution. CAR then evaluates whether the candidate implementation satisfies the experimental requirements specified in the Brief, including experimental components, baselines, and measurement semantics. Candidates that violate the Brief are rejected or locally repaired before execution.

**Execution and Provenance Sealing.** Approved candidates are executed under controlled conditions. The resulting Sealed Receipt records execution provenance and factual outcomes. SEG passes

Table 2: Primary evaluation measures. Formal definitions are provided in Section D.

| Measure | Interpretation | Direction |
|---|---|---|
| Experimental Fidelity Rate (EFR) | Fraction of episodes that satisfy all material Brief obligations, including comparisons, controls, metrics, and factual closure. | ↑ |
| Realized RII Violation (RII-V) | Incidence of a material integrity failure that propagates into the authoritative research record through canonical execution, admitted evidence, or committed scientific state. | ↓ |
| Realized RII Cycle Rate (RII-cycle) | Fraction of eligible cycles containing at least one realized integrity violation. | ↓ |
| Task Performance | Evaluation of end-to-end research outcomes using official benchmarks or task completion criteria. | ↑ |

this execution record to IGRL for subsequent evidence interpretation and state updates. Detailed receipt fields and isolation mechanisms are described in Section C.

# 4 Evaluation Design

## 4.1 Research Questions and Settings

Our evaluation addresses three questions:

**RQ1: Context Efficiency.** How does RA-EA decomposition affect persistent context growth compared with a single-agent baseline under matched research cycles?

**RQ2: Integrity.** Does role decomposition introduce handoff risks that allow material deviations to enter canonical execution, admitted evidence, or the scientific state?

**RQ3: Governance.** Do IGRL and SEG reduce realized RII violations while preserving execution fidelity and autonomous research progress?

To isolate these mechanisms, we evaluate systems under two complementary settings (dataset details are provided in Section 5.1):

1. **Fixed-Brief Execution Episodes:** A single-cycle setting where the initial Experiment Brief is frozen. This setting isolates the *Brief-to-Evidence* transition and measures execution fidelity.
2. **Open-Ended Long-Horizon Trajectories:** An iterative evidence-driven setting that tracks research evolution, state updates, and realized RII violations under matched cycle budgets.

## 4.2 Primary Evaluation Measures

Evaluating scientific alignment requires more than programmatic execution outcomes. We introduce a compact set of primary metrics (Table 2) to measure execution fidelity, integrity, capability preservation, and resource cost. Additional diagnostics, including semantic drift propagation and cost analysis, are provided in Section D.

## 4.3 Semantic Evaluation via LLM-as-a-Judge

Because RII and experimental fidelity require semantic assessment, such as determining whether a substituted baseline preserves the intended experiment, we use a structured LLM-as-a-Judge protocol. Three independent Judge instances produce semantic labels, which are aggregated through majority voting. The evaluation schema and judge protocol are described in Section D.6.

Table 3: FML-Balanced-9 normalized test improvement (↑). Baseline results reproduce the officially reported mean ± standard deviation Zou et al. (2026). Bold indicates the best performance in each row.

| Task | TAS-V1 | TAS-V2 | AIDE | AIRA | AutoResearch | OpenEvolve | RIGOR |
|---|---|---|---|---|---|---|---|
| EasyFSL | $.013 \pm .020$ | $.034 \pm .017$ | $.030 \pm .046$ | $.063 \pm .025$ | $.032 \pm .028$ | $.046 \pm .028$ | $\mathbf{.124 \pm .008}$ |
| PyCIL | $.030 \pm .025$ | $.064 \pm .004$ | $.056 \pm .005$ | $.020 \pm .007$ | $.026 \pm .007$ | $.037 \pm .021$ | $\mathbf{.087 \pm .023}$ |
| CausalML | $.020 \pm .006$ | $.040 \pm .068$ | $.023 \pm .024$ | $.033 \pm .057$ | $.007 \pm .012$ | $.014 \pm .024$ | $\mathbf{.063 \pm .024}$ |
| gCastle | $.014 \pm .014$ | $.066 \pm .114$ | $.127 \pm .207$ | $.145 \pm .045$ | $.127 \pm .143$ | $.169 \pm .146$ | $\mathbf{.374 \pm .007}$ |
| ART | $.429 \pm .026$ | $.395 \pm .026$ | $.447 \pm .035$ | $.303 \pm .127$ | $.448 \pm .032$ | $.459 \pm .011$ | $\mathbf{.499 \pm .009}$ |
| OpenOOD | $.023 \pm .033$ | $.039 \pm .017$ | $.012 \pm .020$ | $.021 \pm .002$ | $.057 \pm .013$ | $.006 \pm .011$ | $\mathbf{.058 \pm .009}$ |
| PrivacyMeter | $.008 \pm .014$ | $.482 \pm .068$ | $.530 \pm .018$ | $.011 \pm .020$ | $.575 \pm .025$ | $.032 \pm .056$ | $\mathbf{.629 \pm .021}$ |
| AIF360 | $.208 \pm .028$ | $.218 \pm .031$ | $.231 \pm .005$ | $.136 \pm .120$ | $.218 \pm .016$ | $.156 \pm .135$ | $\mathbf{.232 \pm .009}$ |
| PFLlib | $.236 \pm .018$ | $.236 \pm .013$ | $.250 \pm .007$ | $.234 \pm .042$ | $.277 \pm .053$ | $.201 \pm .005$ | $\mathbf{.315 \pm .023}$ |
| Macro mean | 0.109 | 0.175 | 0.190 | 0.107 | 0.196 | 0.124 | **0.265** |

# 5 EXPERIMENTS

## 5.1 TASK COLLECTIONS AND BENCHMARKS

To evaluate RIGOR across both governance boundaries and end-to-end research performance, we construct three complementary evaluation suites covering single-cycle execution, long-horizon research, and benchmark-based task performance.

**Execution Fidelity (Canonical-123).** To evaluate the *Brief-to-Evidence* boundary, we construct *Canonical-123*, a suite of 123 single-cycle execution episodes across 57 scientific domains, derived from established scientific benchmarks including PAPERBENCH, MLR-BENCH, EXP-BENCH, and ASI-BENCH Starace et al. (2025); Chen et al. (2025a); Kon et al. (2025); Zhou et al. (2026). Each episode provides a frozen Experiment Brief and an initial workspace. This benchmark isolates execution fidelity by evaluating whether agents implement required interventions and measurements without introducing material deviations.

**Open-Ended Research Trajectories (Long-50).** To evaluate the *Evidence-to-State* boundary, we assemble 50 open-ended scientific tasks across 46 domains from the same benchmark families. Each task provides a high-level research objective and initial resources, requiring systems to autonomously formulate hypotheses, execute multi-cycle experiments, update research states, and make stopping decisions over long horizons.

**Objective Benchmark Performance (FML-Balanced-9).** To evaluate whether intent governance preserves end-to-end research performance, we use a balanced 9-task subset of FML-BENCH Zou et al. (2026). These tasks provide official baselines and deterministic numerical evaluation, offering an automated measure of task performance independent of the LLM-as-a-Judge protocol.

Full task inventories, dataset provenance, and split specifications are provided in Section E.

## 5.2 COMPARISON ON FML-BENCH

To evaluate end-to-end task performance, we benchmark RIGOR on the FML-BALANCED-9 suite. We compare against six autonomous research agents: The AI Scientist V1 (TAS-V1) Lu et al. (2024), The AI Scientist V2 (TAS-V2) Yamada et al. (2025), AIDE Jiang et al. (2025), AIRA Toledo et al. (2025), AutoResearch Karpathy (2026), and OpenEvolve Sharma (2025). For comparability, we use the officially reported results on this suite.

**Protocol and reporting.** All evaluations use three independent runs per task and an execution budget of 100 validation trials per run Zou et al. (2026). The final test score is selected from the candidate implementation with the highest validation performance. We report normalized test improvement following the benchmark's task-specific normalization and clipping protocols. The aggregate macro mean averages the nine task-level means uniformly. Details of the execution and scoring pipeline are provided in Section D.

Table 4: EA execution fidelity and computational cost on Canonical-123. Scientific metrics are percentages averaged over three independent trials per episode. Bold indicates the best values for scientific metrics. Costs are measured per episode excluding evaluation judges; token counts are reported in thousands (k).

| Method | EFR ↑ | RII-V ↓ | Calls | Input (k) | Output (k) |
|---|---|---|---|---|---|
| One-Call Raw | 84.82 | 9.76 | 1.00 | 840.67 | 11.37 |
| One-Call Controlled | 77.78 | 13.82 | 1.00 | 699.61 | 9.85 |
| Compact Async | 57.72 | 15.99 | 1.00 | 523.74 | 7.44 |
| Compact Generic Review | 66.67 | 6.78 | 4.27 | 1,034.20 | 13.58 |
| SEG (ours) | **89.70** | **5.96** | 3.69 | 801.49 | 12.60 |

**Performance and analysis.** As shown in Table 3, RIGOR achieves the highest normalized improvement on all nine tasks, with a macro mean of 0.265, compared to the best baseline macro mean of 0.196 (AutoResearch). These results indicate that the governance mechanisms do not constrain end-to-end task performance; rather, RIGOR improves it while maintaining control over research-state transitions. The consistent gains across tasks suggest that provenance preservation and epistemic admission help prevent unsupported modifications from degrading the scientific state without sacrificing benchmark performance.

### 5.3 Execution Fidelity under a Fixed Brief

We evaluate SEG against four baselines on the Canonical-123 benchmark, averaging results over three trials per episode. The baselines include **One-Call Raw** (single invocation), **One-Call Controlled** (explicit instructions), **Compact Async** (asynchronous execution), and **Compact Generic Review** (feedback-driven revision). To isolate the *Brief-to-Evidence* transition, the Experiment Brief remains fixed throughout each episode, separating execution fidelity from RA-driven state updates. A standardized LLM-as-a-Judge protocol evaluates Experimental Fidelity Rate (EFR) and realized RII violations (details in Section F.2).

**Fidelity and realized violations.** As shown in Table 4, SEG achieves the highest Experimental Fidelity Rate (89.70%) and the lowest realized RII violation rate (5.96%). Compared with One-Call Raw, SEG improves EFR by 4.88 percentage points and reduces realized violations by 3.80 percentage points. It also improves EFR by 23.03 percentage points over Compact Generic Review. These results indicate that explicit execution governance better preserves experimental intent than generic review mechanisms.

**Execution efficiency and cost.** SEG achieves these improvements with moderate computational overhead. Its governance process requires 3.69 model calls per episode on average, while input token consumption (801.49k) remains lower than One-Call Raw (840.67k) and Compact Generic Review (1,034.20k). This efficiency stems from applying structural gating (MGE) before semantic review (CAR) and bounding the context to the effective Brief, which avoids the cascading token overhead of unconstrained iterative repair.

### 5.4 Research Integrity over Long-Horizon Trajectories

We evaluate RIGOR against four research-loop baselines across 50 long-horizon tasks. **Single Agent** performs planning, implementation, and interpretation within a single persistent context. **Staged Single** modularizes these activities into distinct planning, execution, and review phases while maintaining a shared context. **Compact RA-EA** pairs a persistent Research Agent with a fresh Execution Agent each cycle, communicating via structured Briefs and execution reports. **Raw RA-EA** utilizes a similar persistent-researcher/fresh-executor decomposition but relies on unstructured natural-language plans and verbatim responses. RIGOR introduces the dual-governance mechanisms detailed in previous sections to systematically regulate this loop. All baseline experiments utilize a maximum horizon of 30 research cycles.

Table 5: Long-horizon performance across 50 tasks. Rates are expressed as percentages, with task means weighted equally.

| Method | Completion ↑ | RII-V ↓ | EFR ↑ | RII-cycle ↓ | Avg Cycles | Avg Calls |
|---|---|---|---|---|---|---|
| Single Agent | 62.00 | 56.00 | 76.64 | 32.42 | 7.82 | 7.82 |
| Staged Single | 50.00 | 54.00 | 74.08 | 29.90 | 12.54 | 37.62 |
| Compact RA-EA | 46.00 | 68.00 | 69.10 | 39.55 | 12.30 | 25.60 |
| Raw RA-EA | 46.00 | 68.00 | 68.37 | 33.64 | 4.88 | 14.64 |
| RIGOR (ours) | **70.00** | **42.00** | **86.52** | **9.08** | 14.80 | 171.63 |

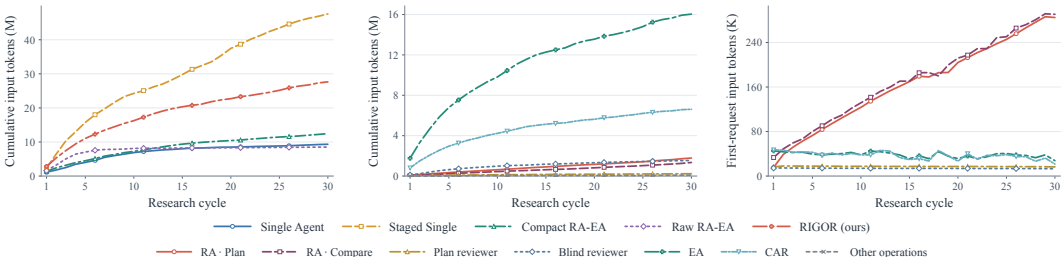



Figure 4: Input cost and context localization in long-horizon research. **(a)** Cumulative input-token consumption across research cycles for the five compared methods, including all model requests within each cycle. **(b)** RIGOR's cumulative input-token consumption decomposed by execution and governance component. **(c)** Role-specific context length in RIGOR, measured by the input tokens of the first LLM request when each role is invoked in a research cycle.

**Completion and integrity.** RIGOR attains the highest task completion rate, the highest EFR, and the lowest trajectory-level realized RII rate among the evaluated methods. These results indicate that role decomposition alone is not sufficient for research-intent preservation. Both Compact RA–EA and Raw RA–EA reduce persistent context sharing but show higher realized RII rates, suggesting that handoff boundaries can introduce additional integrity risks without explicit governance. By contrast, RIGOR reduces realized violations while sustaining autonomous research progress through governed evidence and state transitions.

**Context decomposition and governance cost.** As shown in Figure 4(a), governance increases cumulative input-token consumption relative to single-agent and RA–EA baselines. This additional cost, however, is not distributed uniformly across the system. Figure 4(b) shows that most of RIGOR's input consumption is concentrated in the EA and the governance components, whereas the persistent RA accounts for a smaller fraction of the total cost. Figure 4(c) further shows that the RA's invocation-time context grows gradually with the research trajectory rather than with the larger volume of EA-side execution activity. Together, these results are consistent with the intended role decomposition: governance introduces additional execution-side computation, while low-level implementation and debugging history remains localized and does not accumulate into the persistent research context.

## 6 Conclusion

We formalize Research Intent Integrity (RII) to characterize semantic and material drift in autonomous scientific discovery, and introduce **RIGOR**, a dual-governance architecture that decouples project-level scientific reasoning from bounded experimental execution. RIGOR combines Scientific Execution Governance (SEG) for the *Brief-to-Evidence* transition with the Intent-Grounded Research Loop (IGRL) for the *Evidence-to-State* transition, regulating the structural handoff boundaries of role-decomposed agents. Evaluations on single-cycle episodes, long-horizon trajectories, and public benchmarks indicate that explicit intent governance reduces semantic drift while maintaining competitive end-to-end performance. These results suggest that provenance-preserving execution and principled evidence admission are useful for reliable autonomous discovery.

## AI USE STATEMENT

In this work, we used generative AI tools for creating and modifying software code, assisting with debugging, and improving the readability of the manuscript. We have not used generative AI tools for generating research claims, establishing theoretical results, or making final scientific interpretations, and other required disclosure tasks not listed above are not applicable to this work. Additionally, we used generative AI tools for recommended tasks, including assisting with brainstorming, improving manuscript structure and readability, and supporting literature organization. All AI-assisted outputs were reviewed and verified by the authors. In particular, AI-assisted code was manually inspected, tested, and validated by the authors before being used in experiments. AI-assisted text suggestions were reviewed and revised to ensure that all claims, interpretations, and descriptions accurately reflect the authors' work. We take responsibility for the final content of this work, including all text, claims, experimental results, and artifacts produced with the aid of generative AI.

## REFERENCES

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 6709–6738. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.NAACL-LONG.342. URL https://doi.org/10.18653/v1/2025.naacl-long.342.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. Mlebench: Evaluating machine learning agents on machine learning engineering. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=6s5uXNWGIh.

Hui Chen, Miao Xiong, Yujie Lu, Wei Han, Ailin Deng, Yufei He, Jiaying Wu, Yibo Li, Yue Liu, and Bryan Hooi. Mlr-bench: Evaluating AI agents on open-ended machine learning research. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025a. URL http://papers.nips.cc/paper_files/paper/2025/hash/ab8dd000d6f87f40061a73f8bca7fae4-Abstract-Datasets_and_Benchmarks_Track.html.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=6z4YKr0GK6.

Nicholas Edwards, Yukyung Lee, Yujun Audrey Mao, Yulu Qin, Sebastian Schuster, and Najoung Kim. Rexbench: Can coding agents autonomously implement AI research extensions? In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 16380–16417. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.745. URL https://doi.org/10.18653/v1/2026.acl-long.745.

Shanghua Gao, Ada Fang, and Marinka Zitnik. Autoscientists: Self-organizing agent teams for long-running scientific experimentation. *CoRR*, abs/2605.28655, 2026. doi: 10.48550/ARXIV.2605.28655. URL https://doi.org/10.48550/arXiv.2605.28655.

Jek Huang, Jeffery Hsia, Jiayi Sun, Freddie Shi, Wei Huang, and Ian H. White. Proof-or-stop: Don't trust the agent, trust the evidence - loop engineering for verifiable evidence-gated lifecycle control. *CoRR*, abs/2607.14890, 2026. doi: 10.48550/ARXIV.2607.14890. URL https://doi.org/10.48550/arXiv.2607.14890.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/huang24y.html.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: ai-driven exploration in the space of code. *CoRR*, abs/2502.13138, 2025. doi: 10.48550/ARXIV.2502.13138. URL https://doi.org/10.48550/arXiv.2502.13138.

Jiajie Jin, Yuyang Hu, Kai Qiu, Qi Dai, Chong Luo, Guanting Dong, Xiaoxi Li, Tong Zhao, Xiaolong Ma, Gongrui Zhang, Zhirong Wu, Bei Liu, Zhengyuan Yang, Linjie Li, Lijuan Wang, Hongjin Qian, Yutao Zhu, and Zhicheng Dou. Toward generalist autonomous research via hypothesis-tree refinement. *CoRR*, abs/2606.11926, 2026. doi: 10.48550/ARXIV.2606.11926. URL https://doi.org/10.48550/arXiv.2606.11926.

Andrej Karpathy. Autoresearch: AI agents running research on single-GPU nanochat training automatically. GitHub repository, 2026. URL https://github.com/karpathy/autoresearch.

Patrick Tser Jern Kon, Jiachen Liu, Xinyi Zhu, Qiuyi Ding, Jingjia Peng, Jiarong Xing, Yibo Huang, Yiming Qiu, Jayanth Srinivasa, Myungjin Lee, Mosharaf Chowdhury, Matei Zaharia, and Ang Chen. Exp-bench: Can AI conduct AI research experiments? *CoRR*, abs/2505.24785, 2025. doi: 10.48550/ARXIV.2505.24785. URL https://doi.org/10.48550/arXiv.2505.24785.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Trans. Assoc. Comput. Linguistics*, 12:157–173, 2024. doi: 10.1162/TACL_A_00638. URL https://doi.org/10.1162/tacl_a_00638.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob N. Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *CoRR*, abs/2408.06292, 2024. doi: 10.48550/ARXIV.2408.06292. URL https://doi.org/10.48550/arXiv.2408.06292.

Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of ai research. *Nature*, 651(8107):914–919, 2026.

Yougang Lyu, Xi Zhang, Xinhao Yi, Yuyue Zhao, Shuyu Guo, Wenxiang Hu, Jan Piotrowski, Jakub Kaliski, Jacopo Urbani, Zaiqiao Meng, Lun Zhou, and Xiaohui Yan. Evoscientist: Towards multi-agent evolving AI scientists for end-to-end scientific discovery. *CoRR*, abs/2603.08127, 2026. doi: 10.48550/ARXIV.2603.08127. URL https://doi.org/10.48550/arXiv.2603.08127.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. Discoverybench: Towards data-driven discovery with large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=vyflgpwfJW.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Asankhaya Sharma. OpenEvolve: An open-source evolutionary coding agent. GitHub repository, 2025. URL https://github.com/algorithmicsuperintelligence/openevolve.

Yang Shen, Zhenyi Yi, Ziyi Zhao, Lijun Sun, Dongyang Li, Chin-Teng Lin, and Yuhui Shi. An empirical study of multi-agent collaboration for automated research. *CoRR*, abs/2603.29632, 2026. doi: 10.48550/ARXIV.2603.29632. URL https://doi.org/10.48550/arXiv.2603.29632.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31210–31227. PMLR, 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. Paperbench: Evaluating ai's ability to replicate AI research. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/starace25a.html.

Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. Ai-researcher: Autonomous scientific innovation. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/0d904d300a105809a2114d727851e759-Abstract-Conference.html.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Mario Baldwin, Alexis Audran-Reiss, Michael Kuchnik, Despoina Magka, Minqi Jiang, Alisia Maria Lupidi, Andrei Lupu, Roberta Raileanu, Tatiana Shavrina, Kelvin Niu, Jean-Christophe Gagnon-Audet, Michael Shvartsman, Shagun Sodhani, Alexander H. Miller, Abhishek Charnalia, Derek Dunfield, Carole-Jean Wu, Pontus Lars Erik Saito Stenetorp, Nicola Cancedda, Jakob N. Foerster, and Yoram Bachrach. AI research agents for machine learning: Search, exploration, and generalization in mle-bench. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/328b81881da145412f2bc56c998dfb6a-Abstract-Conference.html.

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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. 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. *CoRR*, abs/2509.26603, 2025b. doi: 10.48550/ARXIV.2509.26603. URL https://doi.org/10.48550/arXiv.2509.26603.

Xiaoyu Xiong, Yuqi Ren, and Deyi Xiong. Evosci: A bio-inspired multi-agent framework for the evolution of scientific discovery. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David

Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 9846–9878. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.447. URL https://doi.org/10.18653/v1/2026.acl-long.447.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob N. Foerster, Jeff Clune, and David Ha. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *CoRR*, abs/2504.08066, 2025. doi: 10.48550/ARXIV.2504.08066. URL https://doi.org/10.48550/arXiv.2504.08066.

Lu Yan, Xuan Chen, and Xiangyu Zhang. When the specification emerges: Benchmarking faithfulness loss in long-horizon coding agents. *CoRR*, abs/2603.17104, 2026. doi: 10.48550/ARXIV.2603.17104. URL https://doi.org/10.48550/arXiv.2603.17104.

Xuanle Zhao, Zilin Sang, Yuxuan Li, Qi Shi, Weilun Zhao, Shuo Wang, Duzhen Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. Autoreproduce: Automatic AI experiment reproduction with paper lineage. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 21920–21942. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1001. URL https://doi.org/10.18653/v1/2026.acl-long.1001.

Junwei Zhou, Zhen Sun, Binyu Li, Jiangyu Zhou, Yuexi Pan, Hengyu Wang, Honghe Ren, Xiaohan Jia, Xueyang Zhou, Xiaoyu Cao, Yongchao Chen, Yuanning Feng, Junhao Wu, Cheng Zhang, Sijia Chen, Haoyu Xue, Chengsong You, Huan Wang, Koutian Wu, Peigan Gao, Jiakun Wu, Wenzhe Li, Ergan Shang, Qingyuan Zheng, Jingjing Zhou, Ruixuan Jia, Yan Xu, Hongrui Zhang, Xiao-Han Ma, Zhengxiang Cheng, Yuexing Hao, Liting Mai, Xianglin Ji, Wenjun Zhang, Zhuofan Chen, Yixiao Huang, Chi Wang, Wenyue Hua, Yilun Hao, Yuantao Zhai, Ziyan Zhao, and Jingyan Xie. Asi-bench: At the dawn of artificial superintelligence. *CoRR*, abs/2608.17271, 2026. doi: 10.48550/ARXIV.2608.17271. URL https://doi.org/10.48550/arXiv.2608.17271.

Qiran Zou, Hou Hei Lam, Wenhao Zhao, Tingting Chen, Yiming Tang, Samson Yu, Yingtao Zhu, Srinivas Anumasa, Zufeng Zhang, Tianyi Zhang, Chang Liu, Zhengyao Jiang, Anirudh Goyal, and Dianbo Liu. Fml-bench: A controlled study of AI research agent strategies from the perspective of search dynamics. *CoRR*, abs/2605.17373, 2026. doi: 10.48550/ARXIV.2605.17373. URL https://doi.org/10.48550/arXiv.2605.17373.

Table 6: Roles and governed scopes within the RIGOR architecture.

| Component | Governed Scope & Authority |
|---|---|
| Human-Authorized Intent | Defines the current research objective, required controls, resources, and authorized scope. Serves as the reference for evaluating subsequent research updates. |
| Research Agent | Formulates hypotheses, designs Experiment Briefs, and integrates validated evidence. It has read-only access to execution records and remains constrained by the current research intent. |
| Execution Agent | Implements and locally debugs experiments specified by the Brief. It operates within a bounded execution environment and cannot directly modify the scientific state. |
| Plan Reviewer | Reviews proposed Experiment Briefs against the current research intent, identifying unsupported scope changes or material conflicts before EA execution. |
| MGE & CAR | MGE performs deterministic checks on execution identity and constraints; CAR evaluates candidate implementations against the scientific obligations specified in the Brief. |
| Runner & Sealed Receipt | Executes approved candidates in an isolated environment and records execution facts, streams, and artifact identities into a provenance record. |
| Independent Blind Reviewer | Evaluates execution artifacts and outcomes against the Brief without access to unstructured EA reasoning traces. |
| Stop Reviewer | Reviews terminal stopping claims against the current research intent, task requirements, and accumulated validated evidence. |

# A RIGOR System and Implementation Details

This appendix describes the implementation details of RIGOR. The system separates project-level scientific reasoning from bounded experimental execution and preserves research intent through controlled state updates, protocol-driven handoffs, and provenance tracking.

## A.1 Roles and Governance Boundaries

RIGOR assigns distinct responsibilities to different components to reduce role collapse and preserve execution fidelity and evidence-grounded state updates. Table 6 summarizes the governed scopes and authorities of each component.

## A.2 State and Provenance Tracking

RIGOR maintains three records with different update policies:

1. **Intent Record:** The current human-authorized objective, resource bindings, and operational constraints. RA updates must remain consistent with this record unless explicitly authorized changes are introduced.
2. **Execution Provenance:** An append-only record of dispatched Briefs, candidate snapshots, Sealed Receipts, and experimental artifacts.
3. **Scientific State:** The evolving repository of hypotheses, validated findings, and bounded conclusions.

Raw logs and EA reasoning traces are not automatically admitted as scientific facts. The RA constructs a formal State Delta only after execution evidence has been validated through independent review. Mechanical checks bind evidence artifacts to their execution sources, reducing the risk of unsupported artifacts entering the scientific state.

## A.3 Research-Cycle Lifecycle

A research cycle traverses both governance boundaries through the following stages:

1. **Plan Formulation:** The RA consults the scientific state and generates an Experiment Brief specifying hypotheses, baselines, interventions, and required measurements.
2. **Alignment Review:** An independent reviewer evaluates the Brief against the current research intent. If material deviations are identified, the RA revises the plan before execution.
3. **Candidate Generation:** A fresh EA receives the authorized Brief and generates a candidate implementation within the restricted workspace.
4. **Execution Governance:** The candidate undergoes MGE and CAR checks to verify compliance with the Brief before execution.
5. **Formal Execution:** The Runner executes the approved candidate and records outputs, artifact identities, and execution metadata in a Sealed Receipt.
6. **Blind Evidence Review:** An independent reviewer evaluates the generated evidence against the Brief requirements without access to the EA reasoning trace.
7. **State Reconciliation:** The RA incorporates validated findings into the scientific state and decides whether to initiate another cycle, revise the hypothesis, or request termination.

### A.4 Experiment Brief Interface

The Experiment Brief serves as a structured interface between the RA and EA. Rather than transferring the complete project history, the Brief contains only the obligations required for the current cycle (Table 7). This compact interface enables faithful implementation while avoiding unnecessary historical context during execution.

Table 7: Principal scientific constraints defined in the Experiment Brief.

| Content | Purpose |
|---|---|
| Hypothesis & Rationale | Defines the research question, motivation, and mechanism being tested. |
| Scientific Comparisons | Specifies required baselines, interventions, and control conditions. |
| Data & Measurement | Defines data splits, evaluation metrics, and aggregation rules. |
| Execution Scope | Specifies permitted modifications, editable paths, and resource constraints. |
| Evidential Obligations | Specifies required artifacts and observations needed to satisfy the cycle objective. |

### A.5 Deterministic Candidate Snapshot

SEG binds candidate implementations to a deterministic identity before execution. For a candidate $C$, the identity representation $\Sigma(C)$ is defined as:

$$\Sigma(C) = \big(h_B, h_E, a_C, d_C, R_C, \Delta_C\big), \tag{4}$$

where $h_B$ and $h_E$ are the SHA-256 hashes of the Brief and execution envelope, respectively. $a_C$ denotes the command argument vector, $d_C$ denotes the working directory, $R_C$ denotes the declared runtime-output roots, and $\Delta_C$ records the identity of modified paths.

The candidate identifier is computed as:

$$\mathrm{CID}(C) = \mathrm{SHA256}\left(\mathrm{Encode}(\Sigma(C))\right), \tag{5}$$

where $\mathrm{Encode}$ serializes the fields using a canonical representation. Any modification to the implementation, environment, or execution configuration changes the candidate identity and requires renewed governance approval.

### A.6 SEALED RECEIPT

After execution, the Runner generates a *Sealed Receipt* bound to the candidate identifier CID. The Receipt records execution facts, including return codes, elapsed time, standard streams, and cryptographic hashes of generated artifacts.

The Sealed Receipt provides the authoritative execution reference for evidence review, ensuring that scientific interpretation is based on recorded execution facts rather than unstructured EA outputs.

### A.7 TASK COMPLETION AND TERMINAL GOVERNANCE

RIGOR separates internal stopping decisions from validated research completion. When the RA determines that the research objectives may have been achieved, it issues a formal stopping claim. A dedicated Stop Reviewer evaluates the claim against the current research intent, task requirements, and accumulated validated evidence.

If missing evidence or unmet requirements are identified, the stopping request is rejected and returned to the RA for additional cycles. The system reaches a terminal state only when the stopping claim is validated or when the predefined computational budget is exhausted.

This procedure prevents premature termination caused by local execution difficulties or incomplete evidence.

## B INTENT-GROUNDED RESEARCH LOOP DETAILS

IGRL orchestrates the research loop by connecting the current human-authorized intent, proposed Briefs, validated evidence, and the scientific state. It decomposes intent alignment, evidence assessment, state reconciliation, and stopping decisions into independent governance procedures.

### B.1 INTENT AND COMPLETION-POLICY ANCHORING

The initial Task defines the starting objective of the trajectory, while the current human-authorized intent serves as the reference for subsequent updates. The RA may revise intermediate hypotheses, technical routes, and operational decompositions in response to new evidence, but cannot modify task obligations without explicit authorization. This maintains alignment between long-horizon exploration and the authorized research objective.

### B.2 EVIDENCE-DRIVEN DECISION PROTOCOL

The RA planning protocol requires each proposed Brief to specify six planning dimensions:

1. **Goal Alignment:** The sub-problem addressed by the proposed experiment and its connection to the overall objective.
2. **Evidence Dependencies:** The supporting evidence, assumptions, and unresolved gaps motivating the experiment.
3. **Decision Gap:** The scientific uncertainty that the experiment is designed to resolve.
4. **Experiment Design:** The required interventions, baselines, and controls.
5. **Outcome Interpretation:** The expected mapping from possible outcomes to subsequent actions.
6. **Completion Path:** The role of the experiment in progressing toward the final objective.

This structured interface requires the RA to specify expected outcomes and decision criteria rather than relying on unstructured reasoning traces.

### B.3 SCIENTIFIC PRECOMMITMENT

Before EA dispatch, the RA specifies the experimental parameters and conditional interpretation rules. By defining how different outcomes affect subsequent decisions before execution, the system

reduces post-hoc interpretation changes. This precommitment limits unnecessary dependence on trajectory history during execution and keeps the EA focused on the authorized Brief.

### B.4 Independent Plan Governance

A dedicated Plan Reviewer evaluates each proposed Brief against the current research intent. It identifies material conflicts, unauthorized scope changes, and missing experimental requirements. The RA must resolve identified alignment issues before EA dispatch is authorized.

### B.5 Evidence Isolation and Blind Review

An independent reviewer evaluates execution artifacts and the Sealed Receipt without access to unstructured EA reasoning traces. This isolation restricts assessment to recorded execution facts rather than EA-generated interpretations. The reviewer classifies evidence as supporting, falsifying, inconclusive, or invalid for scientific interpretation. Executions with mechanical failures bypass semantic review and return execution facts to the RA, reducing the risk of unsupported scientific conclusions.

### B.6 Structured State Updates and Admission Rules

The RA integrates blind review results with available evidence to formulate a structured State Delta. The Controller applies evidence admission rules during this process. Unsupported evidence cannot update hypotheses or scientific conclusions, and code adoption into the scientific state requires independently validated evidence. These constraints maintain a research history grounded in validated contributions.

### B.7 State Continuity and Provenance Tracking

State updates and execution facts are committed to an append-only ledger. When generating subsequent Briefs, the RA queries this accumulated evidence base. Code provenance remains separate from scientific adoption: reusing an engineering snapshot from a previous cycle does not bypass SEG governance for a new experiment. This protocol-driven tracking maintains continuity across long horizons without relying solely on unstructured context.

### B.8 Task-Level Stopping Governance

IGRL separates completion of local experiments from resolution of the overall research objective. When the RA claims project completion, an independent Stop Reviewer evaluates the claim against task requirements and accumulated validated evidence. Missing evidence or unmet requirements return the process to planning. The system reaches a terminal state only when the stopping claim is supported by the validated scientific state, preventing premature termination caused by local execution difficulties.

## C Scientific Execution Governance Details

SEG governs the transition from an authorized Experiment Brief to verified execution evidence. The Brief provides scientific constraints, while SEG governs candidate validation, local repair, execution control, and provenance capture.

### C.1 Materiality-Gated Execution

MGE enforces deterministic checks on candidate implementations before semantic review. It verifies input integrity, workspace structure consistency, and the separation between runtime-output roots and source modifications. MGE binds the candidate to the effective Brief and execution envelope through candidate identity tracking. Candidates violating these constraints are rejected for localized repair before execution, preventing invalid structural modifications from entering later stages.

### C.2 Semantic Alignment Review

CAR operates as a read-only auditor that evaluates mechanically valid candidates against the scientific obligations specified in the Brief. It assesses whether experimental components, baselines, and measurement semantics are preserved. CAR produces a structured verdict: approving compliant candidates, requesting localized semantic repairs, or escalating unresolved conflicts to the RA. Repeated alignment failures terminate the episode and return control to the planning process.

### C.3 Execution, Provenance, and Episode Closure

Formal execution is restricted to governed candidates. The isolated execution environment generates a Sealed Receipt containing execution facts, standard streams, and artifact hashes. Execution failures trigger bounded repairs within the authorized scope, reducing the risk of unauthorized changes to experimental scope or implementation requirements. The episode concludes by returning the provenance record to the research loop. This local execution outcome is distinct from project-level completion or scientific validation.

### C.4 End-to-End SEG Protocol

For a given Experiment Brief, the execution protocol proceeds as follows:

1. The EA generates a candidate implementation within the restricted workspace.
2. MGE verifies workspace constraints and binds the candidate identity to the execution configuration.
3. CAR performs semantic alignment review against the Brief requirements.
4. The Runner executes the approved candidate and records execution provenance through a Sealed Receipt.
5. In-scope repairs address execution failures without bypassing governance checks.
6. The verified execution record is returned to the research loop for independent evidence review.

This protocol preserves execution fidelity and defers scientific interpretation to IGRL.

## D Evaluation Details

### D.1 Overview of Evaluation Measures

Table 8 summarizes the evaluation measures used to assess execution fidelity, integrity preservation, diagnostic deviation patterns, and system overhead. Primary metrics evaluate experimental fidelity, realized RII violations, task completion, and resource cost. Additional diagnostic metrics analyze deviation sources and propagation behavior.

### D.2 Construct Framework: Drift, Failures, and Realization

Scientific integrity is evaluated relative to the current authorized intent. For fixed-Brief episodes, this reference is the effective Experiment Brief. For long-horizon trajectories, the reference evolves with the authorized research intent $I_t^H$.

A semantic decision exhibits material deviation when it departs from this reference in a way that affects the research process, including execution, evidence admission, scientific-state updates, or termination decisions.

A **Material Integrity Failure (MIF)** is classified into three categories:

1. **Unauthorized or Out-of-Scope Change:** Modifying required comparisons, baselines, controls, or constraints without authorization.

Table 8: Summary of evaluation measures across different analysis levels. Primary metrics evaluate final research outcomes, while diagnostic metrics analyze deviation patterns and propagation.

| Metric | Level | Interpretation | Dir. |
|---|---|---|---|
| EFR | Episode / Cycle | Fraction of cases satisfying all material Brief obligations, including interventions, controls, metrics, provenance, and factual closure. | ↑ |
| Realized RII Violation Rate (RII-V) | Episode / Traj. | Incidence of a material integrity failure propagating into canonical execution, admitted evidence, or committed scientific state. | ↓ |
| Realized RII Cycle Rate | Trajectory | Fraction of eligible cycles containing at least one realized integrity violation. | ↓ |
| First Realized RII Cycle | Trajectory | Earliest eligible cycle in which a realized integrity violation occurs. | – |
| Origin-Specific Realized RII | Trajectory | Fraction of realized violations attributed to RA or EA decisions. | ↓ |
| RA-SDR / EA-SDR | Semantic Unit | Proportion of evaluated semantic decisions exhibiting material deviation attributed to the corresponding role. | ↓ |
| System-SDR | Cycle | Proportion of cycles containing system-level semantic deviations. | ↓ |
| VOR | Experiment Group | Fraction of executed experiment groups producing valid interventions, controls, measurements, and claims. | ↑ |
| DPR | Root Drift | Fraction of identified root deviations that propagate to later execution, evidence, or state updates. | ↓ |
| TCDR | Trajectory | Indicator of whether unresolved deviations alter principal conclusions, stopping decisions, or core objectives. | ↓ |
| Task Completion | Trajectory | Majority-vote judgment of whether the authorized research objective is satisfied. | ↑ |
| Method Cost | Trajectory | Cost profile including API calls and token usage: $N_{\text{calls}}, T_{\text{input}}, T_{\text{cached}}, T_{\text{uncached}}, T_{\text{output}}, T_{\text{reasoning}}$. | – |

2. **Unsupported Update:** Introducing a scientific claim or state update without sufficient evidential support.
3. **Provenance Mismatch:** Attributing findings to an incorrect candidate, execution record, or artifact.

A MIF becomes a **Realized RII Violation** only when it propagates into an authoritative research channel:

- CANONICAL-EXECUTION,
- ADMITTED-EVIDENCE,
- COMMITTED-SCIENTIFIC-STATE.

A deviation introduced in an earlier cycle may remain latent until it propagates into one of these channels in a later cycle. Therefore, local execution errors and realized integrity violations are evaluated separately.

### D.3 MATHEMATICAL DEFINITIONS OF EVALUATION MEASURES

**Experimental Fidelity Rate (EFR).** For an episode $e$ with required material obligations $\mathcal{O}_e$, the episode-level fidelity is:

$$\mathrm{EFR}_e = \mathbb{I}\left[\bigwedge_{o \in \mathcal{O}_e} \mathrm{FaithfullyCompleted}(e, o)\right]. \tag{6}$$

For a cohort of episodes $\mathcal{E}$, the aggregate fidelity rate is:

$$\mathrm{EFR} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathrm{EFR}_e \,. \tag{7}$$

For an open-ended trajectory $q$ with eligible cycles $\mathcal{T}_q$, the cycle-level fidelity is:

$$\mathrm{EFR}_q = \frac{1}{|\mathcal{T}_q|} \sum_{t \in \mathcal{T}_q} \mathrm{EFR}_{q,t} \,. \tag{8}$$

**Realized RII and Associated Rates.** For trajectory $q$, cycle $t$, and Judge instance $j$, let:

$$\mathrm{MIF}_{q,t,j} \in \{0, 1\}$$

indicate whether a material integrity failure is identified.

Let $\mathcal{C}_{q,t,j}$ denote the active realization channels, and let $d_{q,t,j}$ indicate whether a delayed realization from a previously identified MIF occurs at cycle $t$.

The instance-level realized RII judgment is:

$$v_{q,t,j} = (\mathrm{MIF}_{q,t,j} \wedge \mathbb{I}[|\mathcal{C}_{q,t,j}| > 0]) \vee d_{q,t,j}. \tag{9}$$

The cycle-level judgment is obtained through majority voting among independent Judge instances:

$$R_t = \mathrm{CycleRII}(q, t) = \mathrm{Majority}\left(v_{q,t,1}, v_{q,t,2}, v_{q,t,3}\right). \tag{10}$$

Based on cycle-level consensus, we define the following trajectory-level measures:

- **Trajectory-Level Realized RII.**

$$\mathrm{TrajectoryRII}(q) = \mathbb{I}\left[\exists t \in \mathcal{T}_q : R_t = 1\right]. \tag{11}$$

- **Realized RII Cycle Rate.**
  The density of realized integrity violations is:

$$\mathrm{CycleRate}(q) = \frac{1}{|\mathcal{T}_q|} \sum_{t \in \mathcal{T}_q} \mathbb{I}[R_t = 1]. \tag{12}$$

- **First Realized RII Cycle.**
  The first realized violation cycle is:

$$\mathrm{FirstRealizedCycle}(q) = \min\{t \in \mathcal{T}_q : R_t = 1\}. \tag{13}$$

  For trajectories without realized violations, this value remains undefined.

- **Origin-Specific Realized RII Rate.**
  The origin-specific rate attributes realized violations to RA or EA decisions:

$$\mathrm{OriginRate}_r(q) = \frac{|\{t \in \mathcal{T}_q : R_t = 1 \wedge \mathrm{Origin}(q, t) = r\}|}{|\mathcal{T}_q|}, \tag{14}$$

  where $r \in \{\mathrm{RA}, \mathrm{EA}\}$.

**Role-Specific Semantic Drift (RA-SDR / EA-SDR).** SDR metrics provide diagnostic analysis of semantic deviations before realization. Let $U_j^{\mathrm{RA}}$ and $U_j^{\mathrm{EA}}$ denote the sets of assessable semantic decisions attributed to the RA and EA roles for Judge instance $j$.

The role-specific rates are:

$$\text{RA-SDR}_j = \frac{\sum_{u \in U_j^{\mathrm{RA}}} \mathbb{I}[\mathrm{Drift}(u)]}{|U_j^{\mathrm{RA}}|}, \tag{15}$$

$$\text{EA-SDR}_j = \frac{\sum_{u \in U_j^{\mathrm{EA}}} \mathbb{I}[\mathrm{Drift}(u)]}{|U_j^{\mathrm{EA}}|}. \tag{16}$$

**System Semantic Drift (System-SDR).** System-SDR measures the proportion of eligible cycles containing material semantic deviations at the system level. The cycle-level label is determined through majority voting across Judge instances.

**Valid Observation Rate (VOR).** VOR measures whether executed experiment groups produce valid observations. For experiment groups $G_j$ identified by Judge instance $j$:

$$\mathrm{VOR}_j = \frac{\sum_{g \in G_j} \mathbb{I}[\mathrm{ValidObservation}(g)]}{|G_j|}. \tag{17}$$

**Drift Propagation Rate (DPR).** DPR measures whether identified root deviations propagate into later execution, evidence, or state updates. Let $D_j$ denote the set of root deviations identified by Judge instance $j$:

$$\mathrm{DPR}_j = \frac{\sum_{d \in D_j} \mathbb{I}[\mathrm{Propagated}(d)]}{|D_j|}. \tag{18}$$

**Task-Compromising Drift Rate (TCDR).** TCDR is a terminal diagnostic indicating whether unresolved deviations alter principal conclusions, stopping decisions, or mandatory research objectives.

**Task Completion.** Task Completion is a trajectory-level binary judgment indicating whether the authorized research objective is satisfied. The final label is determined through majority voting among independent Judge instances.

**Method Cost and Token Growth.** Resource cost is represented as a multi-dimensional profile:

$$\mathrm{Cost} = \left(N_{\mathrm{calls}}, T_{\mathrm{input}}, T_{\mathrm{cached}}, T_{\mathrm{uncached}}, T_{\mathrm{output}}, T_{\mathrm{reasoning}}\right). \tag{19}$$

### D.4 End-to-End Evaluation Pipeline and State Management

Our evaluation pipeline converts raw execution logs into aggregated metrics through a structured evaluation state model. Fixed-Brief episodes and open-ended trajectories share the same state management protocol, while separating mechanical validation from scientific evaluation.

Following the state transitions in Table 9, the evaluation pipeline consists of five stages:

**1. Preparation and Evidence Projection.** A projection script converts raw execution logs into read-only *evidence packets*. For fixed-Brief episodes, each packet contains the Task, Experiment Brief, final candidate, Sealed Receipt, and admitted evidence. For long-horizon trajectories, the packet provides a chronological index of RA state updates and EA executions. The projection process standardizes evaluation inputs and removes method-specific metadata to support blinded assessment.

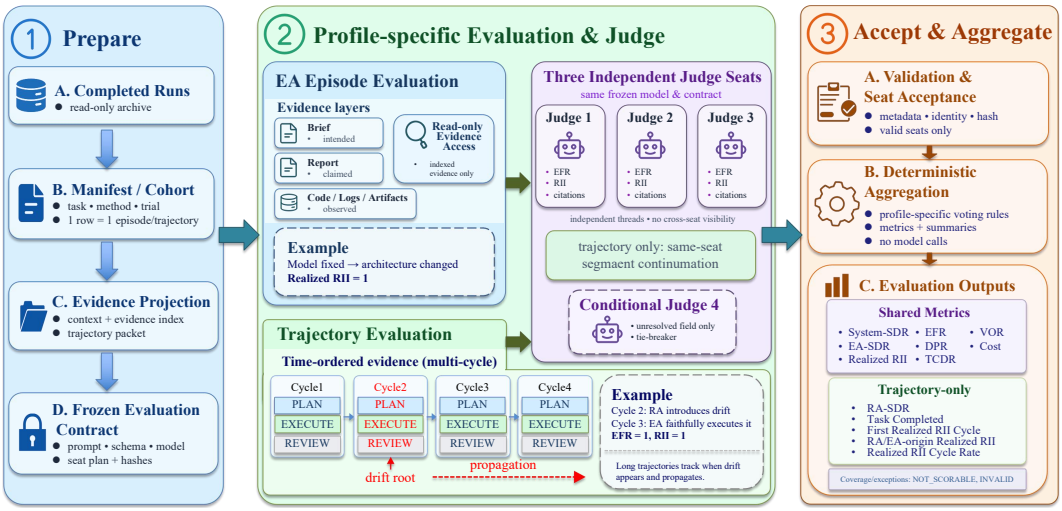



Figure 5: The evaluation workflow. The pipeline converts raw execution logs into read-only evidence packets, evaluates them with three independent Judge instances without cross-visibility, and aggregates outcomes through mechanical validation and conditional tie-breaking.

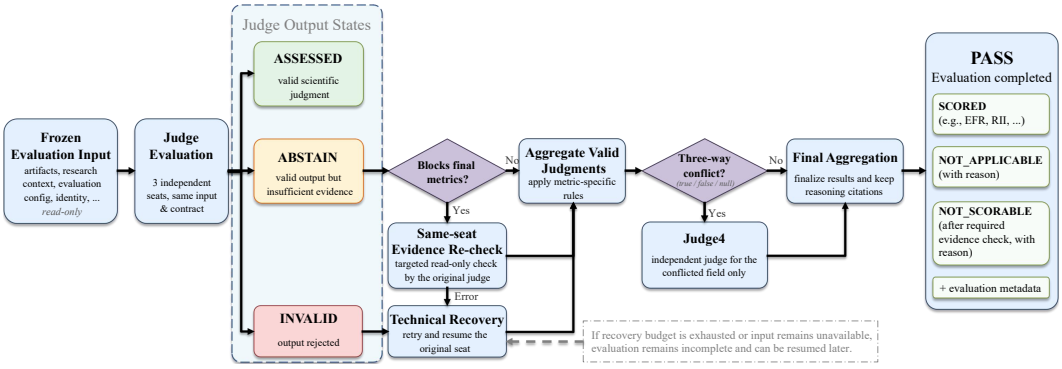



Figure 6: State transition and completion flow. The protocol separates mechanical failures requiring technical recovery from scientific uncertainty requiring targeted evidence re-evaluation. Valid judgments are aggregated with conditional tie-breaking for predefined three-way conflicts.

**2. Independent Multi-Instance Judging and Technical Validation.** Projected evidence packets are provided to three independent Judge instances without cross-visibility. Each Judge output must satisfy schema validation and evidence-location checks, including verification of cited evidence references. Outputs failing these checks are marked as INVALID and trigger technical recovery without contributing additional votes.

**3. Deterministic Aggregation and Tie-Breaking.** After three valid judgments are collected, majority voting determines the final label. Metrics that do not apply to a given case are marked as NOT_APPLICABLE. For predefined three-way conflicts in Boolean judgments, such as $\{true, false, null\}$, an additional Judge instance is used for tie-breaking.

**4. Targeted Re-Reading for Scientific Abstention.** A valid Judge may return ABSTAIN when available evidence is insufficient for a decisive judgment. If abstentions prevent metric aggregation, the pipeline initiates targeted re-reading using the same frozen evidence packet and evaluation schema. The abstaining instance either updates the judgment or confirms the unresolved status.

**5. Terminal State Resolution and Reproducibility.** An evaluation reaches PASS after all required technical recovery, tie-breaking, and re-reading procedures are completed. All evaluation

Table 9: Unified state model for the evaluation pipeline, separating mechanical validation from scientific evaluation.

| Category | State | Interpretation | Subsequent Action |
|---|---|---|---|
| **Pipeline Status** | INCOMPLETE | Required evaluations, verifications, or aggregation steps remain unresolved. | Recover failures, obtain missing judgments, or invoke predefined tie-breaking procedures. |
| | PASS | All required evaluations, retries, tie-breakers, and re-reading procedures are completed. | Finalize and report evaluation results. |
| **Instance Acceptance** | INVALID | Output violates the required schema or fails mechanical evidence-binding checks. | Trigger technical recovery; the invalid instance does not contribute votes. |
| | VALID | Output satisfies structural requirements and evidence bindings are verified. | Proceed to scientific aggregation. |
| **Instance Scientific** | ASSESSED | The Judge reaches a scientific conclusion under the evaluation criteria. | Include the judgment in aggregation. |
| | ABSTAIN | The Judge identifies insufficient evidence for a decisive conclusion. | Record abstention and evaluate aggregation status. |
| **Metric Aggregation** | SCORED | A final metric value is determined. | Retain the score and supporting evidence. |
| | NOT_APPLICABLE | The metric is undefined for the current evaluation case. | Skip metric computation. |
| | INCOMPLETE | Available judgments are insufficient for aggregation. | Await pending judgments or resolve missing cases. |
| | NOT_SCORABLE | Evaluation is procedurally complete but blocked by unresolved abstentions. | Trigger targeted evidence re-reading. |

components, including task manifests, model versions, prompt schemas, and aggregation rules, are fixed before formal scoring.

### D.5 WHOLE-TRAJECTORY ASSESSMENT EXTENSIONS

The open-ended trajectory evaluator extends the above protocol to multi-cycle research processes. It retains independent judging, mechanical validation, and conditional tie-breaking while introducing mechanisms for cross-cycle analysis.

**Cross-Cycle Reasoning via Read-Only Archives.** Each Judge examines the complete trajectory through a navigation index and read-only access to archived files. This enables tracing whether early-cycle root deviations propagate into later execution, evidence records, or scientific-state updates within the trajectory.

**Cycle Voting, Partial Coverage, and Local Denominators.** Trajectory-level cycle metrics, including cycle-level EFR and realized RII cycle rate, are aggregated only over cycles whose evaluation status has been resolved.

For metric $m$, let $\mathcal{S}_m$ denote scored cycles, $\mathcal{U}_m$ denote unresolved cycles, and $\mathcal{A}_m$ denote inapplicable cycles. The reported scientific rate is:

$$\widehat{r}_m = \frac{\sum_{c \in \mathcal{S}_m} \mathbb{1}[y_{cm} = 1]}{|\mathcal{S}_m|}, \qquad N = |\mathcal{S}_m| + |\mathcal{U}_m| + |\mathcal{A}_m|. \tag{20}$$

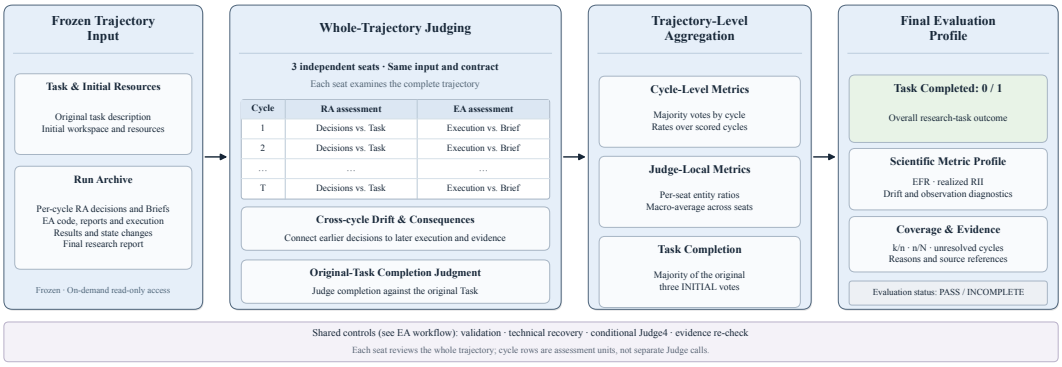



Figure 7: Whole-trajectory evaluation overview. Three independent Judge instances inspect frozen trajectory archives through read-only access. Each instance evaluates RA decisions and EA executions by cycle, traces whether deviations propagate across cycles, and separately assesses research objective completion.

We report the scientific rate together with coverage:

$$\frac{|\mathcal{S}_m|}{N}.$$

A non-empty scored subset produces a SCORED status. An empty scored subset with unresolved cycles produces a NOT_SCORABLE status.

Judge-specific metrics, including RA-SDR and VOR, are aggregated as the unweighted mean of valid per-instance rates.

**Trajectory Completion Metric.** Task Completion is evaluated as an independent trajectory-level outcome. Each initial Judge instance provides a Boolean judgment based on whether the trajectory satisfies the authorized research objective.

The final completion label is determined through majority voting among the three initial valid judgments. Tie-breaking and targeted re-reading are not applied because Task Completion evaluates the overall trajectory outcome rather than a local evidence attribute.

### D.6 JUDGE ASSESSMENT PROTOCOL AND CONSTRUCT VALIDATION

Each Judge instance evaluates the evidence packet under a frozen evaluation model and prompt schema. The evaluation process withholds expected labels and baseline rankings to reduce evaluation bias.

**Structured Assessment Protocol.** The Judge follows a structured assessment protocol that resolves six sequential evaluation dimensions and records evidence locations for each decision:

1. What research obligations are authoritative according to the Experiment Brief or current research intent?
2. Which comparisons, interventions, and measurements were actually executed, and what evidence supports them?
3. Do the executed intervention, control, measurement, and resulting claim satisfy the required experimental criteria?
4. Did any material deviation occur relative to the authorized intent, and which functional role introduced the deviation?
5. Did the deviation propagate into canonical execution, admitted evidence, or committed scientific state?

Table 10: Task collections. CPU/GPU counts refer to execution episodes for Canonical-123 and task identities for the other collections. Resource classes describe the intended execution environment.

| Collection | Counting unit and size | CPU | GPU | Evaluation purpose |
|---|---|---|---|---|
| Canonical-123 | 123 episodes; 60 tasks; 57 scientific families | 60 | 63 | Execution fidelity under a fixed Experiment Brief |
| Long-Horizon Research | 50 tasks; 46 scientific families | 26 | 24 | Research planning, evidence use, and intent preservation across cycles |
| FML-Balanced-9 | 9 benchmark tasks | 4 | 5 | End-to-end task performance under executable evaluation protocols |

6. Did any unresolved deviation affect the final research conclusion, stopping decision, or required objective?

**Construct Validation.** To verify that the evaluation protocol consistently implements the formal metric definitions, we calibrate and validate the evaluation schema on a held-out CONSTRUCT-VALIDATION set before formal scoring.

The validation set contains challenging cases designed to distinguish execution failures, material deviations, and realized integrity violations, including:

- *Ordinary incompleteness without RII violation:* The implementation fails or remains incomplete ($\mathrm{EFR} = 0$), but no unauthorized deviation enters canonical execution, admitted evidence, or committed scientific state ($\mathrm{RealizedRII} = 0$).
- *Faithful execution under an already-deviated Brief:* The EA correctly executes a Brief that already contains an upstream material deviation. This separates execution fidelity from research-intent integrity ($\mathrm{EFR} = 1$, $\mathrm{RealizedRII} = 1$).
- *Provenance-only failure:* The executed experiment satisfies the Brief requirements, but the final claim is attributed to an incorrect candidate, execution record, or artifact.
- *Delayed realization:* An early-cycle root deviation remains latent and only propagates into a later-cycle execution, evidence record, or scientific-state update.

These validation cases test whether the evaluation protocol distinguishes local execution quality from material integrity failures.

# E TASK COLLECTIONS: CONSTRUCTION, PROVENANCE, AND SCOPE

The evaluation resources comprise two custom research task collections and one curated executable benchmark subset, each targeting a distinct evaluation objective. Table 10 summarizes their counting units and evaluation purposes. All counts refer to unique task identities or execution episodes, independent of repeated evaluation trials or method variations.

## E.1 EA EXECUTION FIDELITY: CANONICAL-123

**Unit of construction.** An execution episode is a bounded research execution unit specified by a fixed Experiment Brief, including implementation, debugging, execution, and evidence handoff. To define distinct episodes from a common source task, the required scientific actions must differ materially; variations only in random seeds or output formatting are excluded. The collection contains 123 episodes derived from 60 source tasks across 57 scientific families.

**Sources and construction.** Episodes were constructed from existing AutoResearch task trajectories and converted into standalone evaluation units with frozen Briefs and inputs. To provide diverse scientific and software coverage, the collection integrates public sources including PAPERBENCH,

MLR-BENCH, EXP-BENCH, and ASI-BENCH Starace et al. (2025); Chen et al. (2025a); Kon et al. (2025); Zhou et al. (2026), and specialized libraries for optimization, causal inference, and forecasting.

Each episode satisfies three criteria: (1) a distinct scientific action, (2) usable public inputs, and (3) sufficient artifacts for fidelity assessment.

Among the 123 episodes, 110 use standalone public starting points, while 13 include prerequisite reconstruction requirements specified in the Brief.

**Scientific actions and resources.** As detailed in Table 11, the episodes cover eight scientific action categories. The collection includes 60 CPU and 63 single-GPU execution episodes.

Table 11: Scientific action categories in Canonical-123.

| Action category | Episodes |
|---|---|
| Baseline reproduction | 29 |
| Controlled comparison or ablation | 22 |
| Instrumentation and measurement | 19 |
| Code-module intervention | 17 |
| Cross-condition generalization | 13 |
| Numerical or training stability | 8 |
| Data splits and evaluation protocols | 8 |
| Efficiency and resource intervention | 7 |
| Total | 123 |

### E.2 LONG-HORIZON RESEARCH TASKS

**Construction principle.** These tasks define high-level research objectives that require agents to autonomously formulate hypotheses, design experiments, and revise conclusions based on generated evidence. Constructed from public research papers, datasets, and problem suites, including PAPERBENCH, MLR-BENCH, EXP-BENCH, and ASI-BENCH Starace et al. (2025); Chen et al. (2025a); Kon et al. (2025); Zhou et al. (2026), the tasks specify required empirical outcomes, such as demonstrated improvements or validated failure boundaries.

The task specification fixes the research objective and evidence requirements while leaving trajectory decisions to the agent.

**Size and deduplication.** The collection contains 50 tasks spanning 46 scientific families, balanced across 26 CPU and 24 GPU workloads. Tasks sharing a common scientific methodology are grouped into a single family. For example, the four PatchTST module tasks are treated as one family. The inventory includes version-controlled upstream tasks, with historical runs linked to their corresponding source versions. The complete inventory is provided in Table 12.

**Task-specific requirements.** Task completion criteria depend on the research objective. For example, the BaM task requires baseline reproduction, mechanism evidence, and confirmation of the target improvement. The adaptive-pruning task evaluates connected pruning-and-tuning behavior and accepts a supported failure boundary as a valid outcome. The QuaRot task examines rotation and precision choices through layer-level or kernel-level evaluations. The four PatchTST tasks Nie et al. (2023) investigate different modules, including patch overlap, channel interaction, attention, and forecasting head, using ETTh1, ETTh2, and Exchange datasets. These tasks require both mechanism evidence and validated module-level improvement under paired seeds.

**Long-horizon decision-making.** These tasks involve sequential research decisions, where failed implementations and unsupported hypotheses inform subsequent cycles. Cycle limits represent computational budgets rather than success conditions, allowing early termination when the research objective is satisfied.

Table 12: Long-horizon task inventory. Task names summarize the evaluated research objectives, while exact task identities, versions, and provenance are maintained in the accompanying manifest.

| No. | Research task | Source type | Resource |
|---|---|---|---|
| 1 | BaM: goal-directed update-mechanism study | Paper / project | CPU |
| 2 | BaM: reproduction, explanation, and confirmed improvement | Paper / project | CPU |
| 3 | Stochastic calcium-dynamics simulation | ASI-Bench B3 | CPU |
| 4 | Rigid alignment under ICP degeneracy | ASI-Bench B3 | CPU |
| 5 | Recovery in an ill-posed inverse problem | ASI-Bench B3 | CPU |
| 6 | Forecasting of anisotropic nonlinear fields | ASI-Bench B3 | CPU |
| 7 | Stable latent-state recovery in symbolic streams | ASI-Bench B3 | CPU |
| 8 | LiDAR–IMU temporal and yaw calibration | ASI-Bench B3 | CPU |
| 9 | Data-enlarged predictive safety filtering | ASI-Bench B3 | CPU |
| 10 | Physical retrieval from paired-nanorod spectra | ASI-Bench B3 | CPU |
| 11 | Occupancy mapping with correlated clutter | ASI-Bench B3 | CPU |
| 12 | Perturbation-convergence inference | ASI-Bench B3 | CPU |
| 13 | Pose-graph robustness to false loops | ASI-Bench B3 | CPU |
| 14 | Shared relaxation-mode recovery | ASI-Bench B3 | CPU |
| 15 | Long-horizon integration of an unknown Hamiltonian system | ASI-Bench B3 | CPU |
| 16 | Finite-horizon optimistic Q-learning | ASI-Bench B3 | CPU |
| 17 | Sparse compact-star sequence reconstruction | ASI-Bench B3 | CPU |
| 18 | BAdam block scheduling | Paper / project | GPU |
| 19 | Reliable conclusions from lightweight BenchMARL studies | Paper / project | CPU |
| 20 | Causal-discovery reliability | Paper / project | CPU |
| 21 | CycleNet under changing periodicity | Paper / project | CPU |
| 22 | InfLLM memory allocation | Paper / project | GPU |
| 23 | Lightweight JaxMARL execution and learning reliability | Paper / project | CPU |
| 24 | Cost-aware robustness-certificate ranking | Paper / project | GPU |
| 25 | Conformal prediction under subgroup shift | Paper / project | CPU |
| 26 | Context allocation under a fixed token budget | Paper / project | GPU |
| 27 | Group robustness with incomplete annotations | Paper / project | GPU |
| 28 | Optimization under heavy-tailed gradients | Paper / project | GPU |
| 29 | Minority-safe label-error correction | Paper / project | GPU |
| 30 | Compact neural-field fidelity | Paper / project | CPU |
| 31 | Reliable permutation-based model merging | Paper / project | GPU |
| 32 | Tabular preprocessing under schema shift | Paper / project | CPU |
| 33 | Neural-SDF surface extraction | Paper / project | GPU |
| 34 | Sharing in all-in-one simulation-based inference | Paper / project | GPU |
| 35 | Classifier-free guidance for language-model inference | Paper / project | GPU |
| 36 | Constrained minimal-coreset selection | Paper / project | GPU |
| 37 | PINN optimization and conditioning | Paper / project | GPU |
| 38 | Sample-specific visual reprogramming masks | Paper / project | GPU |
| 39 | Periodic decoupling for forecasting | Paper / project | GPU |
| 40 | Rotation and precision choices in QuaRot | EXP-Bench-derived | GPU |
| 41 | SegVol zoom behavior | Paper / project | GPU |
| 42 | Reliable use of SmoothECE | Paper / project | CPU |
| 43 | Differentiation-mode selection in TorchOpt | Paper / project | GPU |
| 44 | 3D serialization for learned representations | Paper / project | GPU |
| 45 | Lightweight Zipformer resource allocation | Paper / project | CPU |
| 46 | PatchTST: attention and module optimization | Local PatchTST | GPU |
| 47 | PatchTST: channel interaction and module optimization | Local PatchTST | GPU |
| 48 | PatchTST: forecasting head and module optimization | Local PatchTST | GPU |
| 49 | PatchTST: patch overlap and module optimization | Local PatchTST | GPU |
| 50 | Mechanisms and reliable allocation in adaptive language-model pruning | Paper / project | GPU |

### E.3 FML-BALANCED-9

**Source and selection.** FML-Balanced-9 is a curated subset of FML-Bench Zou et al. (2026). The nine tasks cover diverse executable research settings, including few-shot learning, causal inference, adversarial robustness, privacy auditing, and fairness. The subset is balanced across four CPU and five GPU workloads and includes both sparse and dense improvement regimes defined by the upstream benchmark.

Table 13: FML-Balanced-9 under the official task protocols. Arrows indicate the favorable direction of the primary raw metric. S/D denote the sparse/dense improvement-regime labels defined by the upstream suite.

| Task | Setting | Resource | Baseline | Primary Metric | Dir. | Reg. |
|---|---|---|---|---|---|---|
| EasyFSL | Few-shot learning | GPU | Prototypical Networks | Accuracy | ↑ | S |
| PyCIL | Continual learning | GPU | iCaRL | Average incremental accuracy | ↑ | S |
| CausalML | Causal inference | CPU | Dragonnet | MAE | ↓ | S |
| gCastle | Causal discovery | CPU | Linear NOTEARS | SHD | ↓ | D |
| ART | Adversarial robustness | CPU | DP-InstaHide | Defense score | ↑ | D |
| OpenOOD | OOD detection | GPU | MSP + ResNet-18 | AUROC | ↑ | S |
| PrivacyMeter | Privacy auditing | GPU | WRN-28-2 + RMIA | AUC gap | ↓ | D |
| AIF360 | Fairness | CPU | Adversarial Debiasing | Absolute AOD | ↓ | D |
| PFLlib | Federated learning | GPU | FedAvg | Test accuracy | ↑ | D |

**Data and evaluation interfaces.** Each task provides an initial implementation together with an executable evaluation protocol on task-specific datasets, such as Mini-ImageNet for EasyFSL and poisoned MNIST for ART. Evaluation metrics follow the official benchmark definitions. For example, PrivacyMeter uses AUC gap to measure membership inference distance from chance performance, while AIF360 uses absolute AOD to quantify average-odds disparity. The evaluation interface remains unchanged from the upstream benchmark to preserve standardized comparison.

**Numerical scoring.** Performance is computed from executable numerical results on held-out test data. The normalized improvement score relative to the corresponding benchmark baseline is:

$$I(m, b) = \max\left(0, \frac{s(m-b)}{D}\right), \tag{21}$$

where $s = 1$ for higher-is-better metrics and $s = -1$ otherwise. The normalization scale $D$ is set to 1 for standard bounded metrics (EasyFSL, PyCIL, OpenOOD, AIF360, PFLlib, and ART), $0.5$ for PrivacyMeter, and $|b|$ for CausalML and gCastle. Raw metrics are retained together with normalized scores to identify potential performance regressions.

### E.4 REPRODUCIBILITY AND REPORTING SCOPE

Task definitions and method implementations are decoupled in the evaluation framework. Versioned manifests record task identities, effective Briefs, resource classes, and public-input provenance. Run-specific ledgers track selected tasks, model configurations, method versions, and computational budgets.

Because the three collections target complementary evaluation objectives, results and coverage statistics are reported independently. This separation distinguishes system behavior across single-cycle execution fidelity, multi-cycle research trajectories, and executable benchmark performance.

## F EXPERIMENTS DETAILS

### F.1 FML-BENCH EVALUATION PROTOCOL

**Outcome Aggregation.** Let $m_{a,t,r}$ denote the final test metric obtained by method $a$ on task $t$ in run $r$, and let $b_t$ denote the official task baseline. Following the FML-Bench protocol Zou et al. (2026), the normalized improvement score is:

$$I_{a,t,r} = \max\left(0, \frac{s_t(m_{a,t,r} - b_t)}{D_t}\right), \tag{22}$$

where $s_t \in \{+1, -1\}$ indicates the favorable metric direction and $D_t$ is the task-specific normalization constant defined by the benchmark.

The task-level performance is averaged across independent runs:

$$\overline{I}_{a,t} = \frac{1}{R_t} \sum_{r=1}^{R_t} I_{a,t,r}, \tag{23}$$

and the aggregate macro mean is:

$$\overline{I}_a = \frac{1}{9} \sum_{t=1}^{9} \overline{I}_{a,t}. \tag{24}$$

Reference macro means for comparison methods are computed from their published task-level performance records under the same 9-task subset.

### F.2 EA Baselines, Full Results, and Cost Accounting

**Experimental unit and scope.** Canonical-123 evaluates execution fidelity across 123 fixed-Brief episodes, covering 60 source tasks and 57 scientific families. Each method is evaluated with three independent trials per episode, producing 369 execution records per method and 1,845 records overall.

**Shared authority and baseline implementations.** All methods receive identical initial conditions, including the effective Experiment Brief, initial workspace, execution envelope, and public resource references. We compare SEG against four execution baselines:

**One-Call Raw.** A single agent invocation performs implementation, debugging, execution, and reporting without external state parsing or structural constraints.

**One-Call Controlled.** Extends the single-invocation setting with explicit execution constraints and reporting requirements, including completion status, artifact descriptions, and deviation reporting.

**Compact Async.** Separates reasoning from execution through asynchronous environment interactions. The controller pauses execution without additional model calls and resumes to collect terminal execution evidence, providing execution completion checks without independent semantic review.

**Compact Generic Review.** Introduces a multi-agent feedback loop where an independent reviewer inspects the workspace, code, and logs against the Brief. The reviewer either approves the execution or provides revision guidance to the execution agent. The loop terminates upon approval or budget exhaustion.

**SEG.** Applies the governed execution pipeline with deterministic validation, semantic alignment review, and provenance capture through the Sealed Receipt.

**Evaluation and metric aggregation.** All execution records are evaluated using the standardized LLM-as-a-Judge protocol described in Section D.6. For a Boolean metric $q$, the reported rate for method $m$ and repeat $r$ is:

$$\frac{k_{m,r,q}}{n_{m,r,q}}, \tag{25}$$

where $k_{m,r,q}$ denotes the number of positive judgments and $n_{m,r,q}$ denotes the number of scoreable episodes.

Table 14: Scientific evaluation metrics on Canonical-123. Values are averaged over three repeat-level rates. Brackets indicate positive/scorable counts for Boolean metrics and valid denominators for conditional metrics.

| Method | EFR ↑ | RII-V ↓ | TCDR ↓ | System-SDR ↓ | EA-SDR ↓ | VOR ↑ | DPR ↓ |
|---|---|---|---|---|---|---|---|
| One-Call Raw | 84.82% [313/369] | 9.76% [36/369] | 8.40% [31/369] | 10.60% [39/368] | 8.81% [n=369] | 87.97% [n=364] | 87.40% [n=39] |
| One-Call Controlled | 77.78% [287/369] | 13.82% [51/369] | 10.03% [37/369] | 12.60% [46/365] | 11.60% [n=365] | 81.85% [n=354] | 90.71% [n=46] |
| Compact Async | 57.72% [213/369] | 15.99% [59/369] | 16.53% [61/369] | 19.45% [70/359] | 16.85% [n=360] | 80.71% [n=304] | 94.53% [n=66] |
| Compact Generic Review | 66.67% [246/369] | 6.78% [25/369] | 7.32% [27/369] | 22.70% [81/357] | 15.14% [n=355] | 82.24% [n=317] | 76.66% [n=76] |
| SEG | **89.70%** [331/369] | **5.96%** [22/369] | **5.42%** [20/369] | 18.51% [68/367] | 11.13% [n=367] | **92.09%** [n=363] | **40.50%** [n=68] |

Table 15: Diagnostic coverage on Canonical-123. Each cell reports SCORED / NOT_APPLICABLE / NOT_SCORABLE cases.

| Method | System-SDR | EA-SDR | VOR | DPR |
|---|---|---|---|---|
| One-Call Raw | 368 / 0 / 1 | 369 / 0 / 0 | 364 / 5 / 0 | 39 / 304 / 26 |
| One-Call Controlled | 365 / 0 / 4 | 365 / 0 / 4 | 354 / 11 / 4 | 46 / 283 / 40 |
| Compact Async | 359 / 0 / 10 | 360 / 0 / 9 | 304 / 61 / 4 | 66 / 268 / 35 |
| Compact Generic Review | 357 / 0 / 12 | 355 / 5 / 9 | 317 / 49 / 3 | 76 / 251 / 42 |
| SEG | 367 / 0 / 2 | 367 / 0 / 2 | 363 / 4 / 2 | 68 / 276 / 25 |

Primary metrics, including EFR and Realized RII, use a fixed denominator of 123 episodes per repeat. Conditional diagnostic metrics compute episode-level rates over valid cases before averaging across the three repeats.

**Diagnostic interpretation.** We report four diagnostic metrics to analyze how deviations emerge and propagate:

- **System-SDR (Semantic Drift Rate):** Measures the frequency of detected semantic deviations at the system level.
- **VOR (Valid Observation Rate):** Measures the proportion of executed experiment groups producing valid scientific observations.
- **DPR (Drift Propagation Rate):** Measures the fraction of identified root deviations that propagate to downstream execution, evidence, or state updates.
- **TCDR (Terminal Conclusion Drift Rate):** Measures whether unresolved deviations affect final conclusions or stopping decisions.

These diagnostics distinguish initial deviations from realized integrity violations. SEG is not required to eliminate all initial deviations; instead, its objective is to reduce their propagation into authoritative execution records, admitted evidence, or scientific-state updates.

**Diagnostic analysis.** The diagnostic metrics show that SEG primarily improves deviation containment rather than eliminating all initial deviations. Although semantic deviations may still occur during execution, the governance process reduces their propagation into execution records, admitted evidence, and scientific-state updates.

In particular, SEG achieves substantially lower DPR compared with other methods, indicating that early validation and provenance tracking limit downstream propagation of root deviations. This behavior is consistent with the RII formulation, where preventing realization is distinct from preventing all local execution errors.

Table 16: Individual-repeat scientific results on Canonical-123. Boolean metrics report percentages with positive/scorable counts. Conditional metrics report macro rates over scoreable episodes.

| Method | Repeat | EFR | RII-V | TCDR | System-SDR | EA-SDR | VOR | DPR |
|---|---|---|---|---|---|---|---|---|
| One-Call Raw | 1 | 82.11% 101/123 | 8.94% 11/123 | 7.32% 9/123 | 8.94% 11/123 | 8.58% [n=123] | 88.94% [n=118] | 90.91% [n=11] |
| | 2 | 86.99% 107/123 | 9.76% 12/123 | 8.13% 10/123 | 10.57% 13/123 | 8.47% [n=123] | 87.01% [n=123] | 84.62% [n=13] |
| | 3 | 85.37% 105/123 | 10.57% 13/123 | 9.76% 12/123 | 12.30% 15/122 | 9.37% [n=123] | 87.96% [n=123] | 86.67% [n=15] |
| One-Call Controlled | 1 | 74.80% 92/123 | 11.38% 14/123 | 9.76% 12/123 | 11.48% 14/122 | 10.82% [n=123] | 82.05% [n=117] | 89.29% [n=14] |
| | 2 | 80.49% 99/123 | 15.45% 19/123 | 11.38% 14/123 | 13.82% 17/123 | 12.23% [n=122] | 81.25% [n=120] | 91.18% [n=17] |
| | 3 | 78.05% 96/123 | 14.63% 18/123 | 8.94% 11/123 | 12.50% 15/120 | 11.76% [n=120] | 82.24% [n=117] | 91.67% [n=15] |
| Compact Async | 1 | 53.66% 66/123 | 13.82% 17/123 | 13.82% 17/123 | 17.54% 20/114 | 13.97% [n=116] | 79.07% [n=103] | 95.83% [n=18] |
| | 2 | 55.28% 68/123 | 18.70% 23/123 | 19.51% 24/123 | 22.76% 28/123 | 19.65% [n=123] | 81.18% [n=98] | 92.31% [n=26] |
| | 3 | 64.23% 79/123 | 15.45% 19/123 | 16.26% 20/123 | 18.03% 22/122 | 16.94% [n=121] | 81.88% [n=103] | 95.45% [n=22] |
| Compact Generic Review | 1 | 69.92% 86/123 | 6.50% 8/123 | 7.32% 9/123 | 22.41% 26/116 | 15.25% [n=116] | 83.07% [n=105] | 76.92% [n=26] |
| | 2 | 62.60% 77/123 | 7.32% 9/123 | 9.76% 12/123 | 26.67% 32/120 | 17.58% [n=119] | 81.44% [n=104] | 77.22% [n=30] |
| | 3 | 67.48% 83/123 | 6.50% 8/123 | 4.88% 6/123 | 19.01% 23/121 | 12.59% [n=120] | 82.23% [n=108] | 75.83% [n=20] |
| SEG | 1 | 91.87% 113/123 | 4.07% 5/123 | 4.07% 5/123 | 15.57% 19/122 | 9.36% [n=122] | 93.01% [n=122] | 37.43% [n=19] |
| | 2 | 87.80% 108/123 | 8.13% 10/123 | 6.50% 8/123 | 23.58% 29/123 | 13.05% [n=123] | 90.59% [n=121] | 39.08% [n=29] |
| | 3 | 89.43% 110/123 | 5.69% 7/123 | 5.69% 7/123 | 16.39% 20/122 | 10.97% [n=122] | 92.69% [n=120] | 45.00% [n=20] |

**Method costs.** Method cost is measured using provider invocations, input tokens, output tokens, and reasoning tokens required to complete each episode. Reported costs include internal reviews, repair attempts, and execution loops, while excluding evaluation-Judge computation.

### F.3 Supplementary Evaluation with Task and Brief Inputs

**Setting and scope.** To examine robustness under broader execution context, we introduce Combined-60, where the EA receives both the overarching Task and the localized Experiment Brief. Unlike the Brief-only Canonical-123 setting, this supplementary evaluation exposes the EA to additional task-level context and examines whether execution fidelity remains stable under increased context exposure.

Combined-60 contains 60 fixed-Brief episodes drawn from 17 source tasks and 14 scientific families, covering 20 CPU and 40 single-GPU workloads. Each method is evaluated with one trial per episode.

**Scientific outcomes.** Table 18 reports the scientific evaluation metrics. SEG achieves the highest EFR and VOR while reducing realized RII violations and terminal conclusion drift under this broader context setting.

The results show that additional task-level context does not remove the need for execution governance. SEG maintains high execution fidelity by constraining the transition from intent specification to executable evidence.

**Method costs.** Table 19 reports resource consumption. SEG introduces moderate overhead compared with minimal execution baselines while remaining substantially more efficient than repeated generic review. These results indicate that structured governance provides execution control without requiring unrestricted iterative revision.

Table 17: Method costs on Canonical-123. Rows with R1/R2/R3 report per-repeat totals. Pooled reports totals over 369 execution records per method. Mean/ep. reports per-episode averages. Evaluation-Judge costs are excluded.

| Method | Scope | Calls | Input | Cached input | Uncached input | Output | Reasoning |
|---|---|---|---|---|---|---|---|
| One-Call Raw | R1 | 123 | 104,781,197 | 98,055,168 | 6,726,029 | 1,394,422 | 278,296 |
| | R2 | 123 | 107,404,512 | 100,285,184 | 7,119,328 | 1,413,591 | 281,065 |
| | R3 | 123 | 98,022,323 | 91,328,000 | 6,694,323 | 1,388,698 | 265,330 |
| | Pooled | 369 | 310,208,032 | 289,668,352 | 20,539,680 | 4,196,711 | 824,691 |
| | Mean/ep. | 1.00 | 840,672.17 | 785,009.08 | 55,663.09 | 11,373.20 | 2,234.93 |
| One-Call Controlled | R1 | 123 | 77,193,652 | 71,349,504 | 5,844,148 | 1,157,647 | 245,234 |
| | R2 | 123 | 88,350,481 | 81,920,256 | 6,430,225 | 1,241,051 | 261,622 |
| | R3 | 123 | 92,610,512 | 86,240,512 | 6,370,000 | 1,237,222 | 251,940 |
| | Pooled | 369 | 258,154,645 | 239,510,272 | 18,644,373 | 3,635,920 | 758,796 |
| | Mean/ep. | 1.00 | 699,606.08 | 649,079.33 | 50,526.76 | 9,853.44 | 2,056.36 |
| Compact Async | R1 | 123 | 63,318,673 | 58,549,504 | 4,769,169 | 932,139 | 234,137 |
| | R2 | 123 | 61,484,664 | 56,550,144 | 4,934,520 | 885,535 | 216,989 |
| | R3 | 123 | 68,455,664 | 63,449,344 | 5,006,320 | 927,518 | 225,021 |
| | Pooled | 369 | 193,259,001 | 178,548,992 | 14,710,009 | 2,745,192 | 676,147 |
| | Mean/ep. | 1.00 | 523,737.13 | 483,872.61 | 39,864.52 | 7,439.54 | 1,832.38 |
| Compact Generic Review | R1 | 466 | 125,944,927 | 112,767,744 | 13,209,887 | 1,648,834 | 482,085 |
| | R2 | 563 | 126,338,223 | 113,777,920 | 12,643,900 | 1,697,928 | 476,654 |
| | R3 | 548 | 129,334,970 | 116,640,768 | 13,108,717 | 1,665,047 | 477,983 |
| | Pooled | 1,577 | 381,618,120 | 343,186,432 | 38,962,504 | 5,011,809 | 1,436,722 |
| | Mean/ep. | 4.27 | 1,034,195.45 | 930,044.53 | 105,589.44 | 13,582.14 | 3,893.56 |
| SEG | R1 | 467 | 98,851,934 | 85,340,672 | 13,511,262 | 1,592,123 | 443,286 |
| | R2 | 447 | 97,903,474 | 85,441,536 | 12,461,938 | 1,548,414 | 442,991 |
| | R3 | 446 | 98,996,168 | 87,333,632 | 11,673,280 | 1,510,146 | 423,752 |
| | Pooled | 1,360 | 295,751,576 | 258,115,840 | 37,646,480 | 4,650,683 | 1,310,029 |
| | Mean/ep. | 3.69 | 801,494.79 | 699,500.92 | 102,022.98 | 12,603.48 | 3,550.21 |

Table 18: Combined-60 results with Task and Brief inputs. This supplementary setting evaluates execution fidelity under broader EA context and is reported separately from the Brief-only evaluation.

| Method | EFR ↑ | RII-V ↓ | EA-SDR ↓ | System-SDR ↓ | TCDR ↓ | VOR ↑ | DPR ↓ |
|---|---|---|---|---|---|---|---|
| One-Call Raw | 73.33% $n = 60$ | 10.00% $n = 60$ | 8.89% $n = 60$ | 8.33% $n = 60$ | 5.00% $n = 60$ | 87.31% $n = 60$ | 100.00% $n = 5$ |
| One-Call Controlled | 75.00% $n = 60$ | 15.00% $n = 60$ | 9.35% $n = 60$ | 11.67% $n = 60$ | 10.00% $n = 60$ | 89.17% $n = 60$ | 100.00% $n = 7$ |
| Compact Async | 78.33% $n = 60$ | 3.33% $n = 60$ | 4.99% $n = 59$ | 3.39% $n = 59$ | 3.33% $n = 60$ | 86.49% $n = 58$ | 100.00% $n = 2$ |
| Compact Generic Review | 76.67% $n = 60$ | 6.67% $n = 60$ | 5.19% $n = 60$ | 8.33% $n = 60$ | 1.67% $n = 60$ | 80.97% $n = 60$ | 56.67% $n = 5$ |
| SEG | **86.67%** $n = 60$ | **0.00%** $n = 60$ | **1.67%** $n = 60$ | **1.69%** $n = 59$ | **0.00%** $n = 60$ | **99.39%** $n = 55$ | **0.00%** $n = 1$ |

# G Long-Horizon Methods and Detailed Results

## G.1 Research-Loop Baselines

To rigorously evaluate long-horizon autonomous research, we define a shared operational cycle: formulating a core scientific question, declaring comparative experiments, and executing them before observing formal results. All evaluated methods utilize `gpt-5.6-luna/medium` with a strict 30-cycle horizon. We benchmark RIGOR against four structural baselines that progressively vary the separation of reasoning and execution:

**Single Agent.** A single persistent agent orchestrates planning, workspace inspection, experiment execution, and scientific interpretation within a unified context window. The agent maintains its

Table 19: Combined-60 method costs. Values include complete method execution and exclude evaluation-Judge costs.

| Method | Calls | Input | Cached input | Uncached input | Output | Reasoning |
|---|---|---|---|---|---|---|
| One-Call Raw | 60 | 83,385,388 | 78,716,672 | 4,668,716 | 824,448 | 165,632 |
| One-Call Controlled | 60 | 57,162,339 | 53,573,120 | 3,589,219 | 690,473 | 143,684 |
| Compact Async | 60 | 41,213,259 | 38,322,688 | 2,890,571 | 512,386 | 132,422 |
| Compact Generic Review | 294 | 95,595,705 | 87,308,032 | 8,287,673 | 1,046,282 | 320,986 |
| SEG | 212 | 44,654,401 | 37,850,368 | 6,804,033 | 749,196 | 189,191 |

own research record and independently determines task completion. This represents a standard monolithic architecture without role decomposition or structured oversight.

**Staged Single.** This baseline modularizes the monolithic process by enforcing a sequential order: planning, execution, and review are executed as distinct invocation stages. However, because these stages share the same persistent context window, the review phase remains inherently coupled to the agent's prior reasoning, preventing truly independent epistemic evaluation.

**Compact RA-EA.** This baseline decouples project-level reasoning from execution. A persistent Research Agent (RA) manages scientific direction and issues structured Briefs, while a fresh Execution Agent (EA) is instantiated each cycle to implement the designated experiment. The EA returns a structured execution report detailing conditions and deviations. This introduces basic role separation but lacks deterministic execution gating or formal semantic alignment reviews.

**Raw RA-EA.** Operating under a similar persistent-researcher/fresh-executor paradigm, this baseline relies entirely on unstructured natural-language plans and verbatim textual responses. It serves as an ablation to evaluate the impact of structured communication (Briefs and Receipts) versus unconstrained conversational handoffs.

**RIGOR.** Our proposed dual-governance architecture. It integrates SEG execution governance (MGE and CAR) with IGRL planning, ensuring independent evidence appraisal, evidence-grounded state revision, and task-grounded terminal stopping reviews.

### G.2 Hierarchical Aggregation and Metric Definitions

To ensure statistical rigor across variable-length trajectories, metrics are aggregated hierarchically. For a cycle-level metric, let $\mathcal{S}_{tmr}$ denote the set of scientifically scoreable cycles for task $t$, method $m$, and repeat $r$, with observed binary labels $y_{tmrc}$. We first compute the trajectory-level mean $q_{tmr}$, and subsequently compute the task-level aggregate $\bar{q}_m^{(1)}$ across all 50 tasks:

$$q_{tmr} = \frac{1}{|\mathcal{S}_{tmr}|} \sum_{c \in \mathcal{S}_{tmr}} y_{tmrc}, \qquad \bar{q}_m^{(1)} = \frac{1}{50} \sum_{t=1}^{50} q_{tm1}. \tag{26}$$

This hierarchical formulation ensures that extensive trajectories do not disproportionately dominate the aggregate scores. Trajectory-level outcomes, such as original-task completion and the incidence of any Realized RII violation (Any RII), are averaged directly across the 50 tasks.

For the repeated-task variance analysis, we average the defined trial values within each task-method pair before computing the aggregate means:

$$\bar{q}_{tm} = \frac{1}{|\mathcal{D}_{tm}|} \sum_{r \in \mathcal{D}_{tm}} q_{tmr}, \tag{27}$$

where $\mathcal{D}_{tm}$ contains the formally scored repeats. Diagnostic drift metrics (RA-SDR, EA-SDR, VOR, and DPR) retain the evaluator's strict semantic-unit denominators. Structurally inapplicable or scientifically unscorable cycles are rigorously filtered from the respective denominators to ensure unpolluted metric projections.