IDEADEC: BENCHMARKING LONG-HORIZON SCIENTIFIC IDEA DECODING IN AI RESEARCH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

AI research agents increasingly participate in literature analysis, hypothesis formulation, experiment design, and proposal development, making the coherence of their evolving ideas consequential to downstream research. Yet existing evaluations largely score final outputs or isolated stages and therefore do not test whether agents sustain valid scientific commitments as evidence changes. We introduce IDEADEC, a benchmark for **long-horizon scientific idea decoding** under controlled evidence release. IdeaDec operationalizes this problem through 137 trajectories of 7–9 ordered checkpoints, each tracking a five-component state: hypothesis, method, metric, judgment, and plan. Three disjoint subsets evaluate *dependency maintenance*, *conflict-aware revision*, and *dynamic recovery*, while complementary metrics assess structural completion, success on a pre-specified target operation, and five-gate whole-trajectory semantic success. Across 18 fixed model configurations in the State-Aware setting, mean structural completion is 89.05% but mean semantic success is 51.95%, a gap of 37.10 percentage points; similar aggregate scores can also conceal distinct capability profiles. Recovery has the lowest observed subset rate for every configuration under the fixed benchmark distribution. These findings motivate evaluating scientific agents through state evolution rather than format compliance or endpoint outputs alone.

1 INTRODUCTION

AI Scientist systems increasingly integrate literature analysis, hypothesis generation, experiment design, execution, and reporting into automated research workflows (Boiko et al., 2023; Bran et al., 2024; Baek et al., 2025; Lu et al., 2024). Within this pipeline, idea formation is pivotal: the resulting hypotheses, methods, metrics, judgments, and plans determine what is tested and how subsequent evidence is interpreted. As these systems assume more of the research process, evaluating how ideas develop becomes essential for distinguishing coherent scientific reasoning from a merely plausible final report that may conceal inconsistent intermediate decisions.

Long-horizon settings change idea evaluation from scoring an isolated proposal to examining how scientific commitments evolve across interdependent decisions. Evidence arrives sequentially: an early assumption may constrain later checkpoints, a revision may alter downstream methods or metrics, and recovery may require preserving unaffected constraints. A final proposal can therefore appear plausible even when an intermediate decision conflicts with the evidence then available. Endpoint scores neither locate the first violation nor reveal whether subsequent updates propagate correctly across the complete sequence of decisions.

Existing idea-generation benchmarks assess novelty, feasibility, and evidential support (Guo et al., 2025; Qiu et al., 2025; Liu et al., 2025; Mo et al., 2026); scientific-agent benchmarks cover discovery, machine-learning experimentation, and research reproduction (Chen et al., 2025; Majumder et al., 2025; Chan et al., 2025; Starace et al., 2025; Liu et al., 2026a). Long-horizon studies examine extended reasoning, context management, and sustained execution (Luo et al., 2025; Zhao et al., 2026a; Li et al., 2026), while recent agents incorporate adaptive exploration and explicit execution states (Zhao et al., 2026b). Yet scientific idea evaluation lacks a long-horizon protocol that controls evidence release, tracks explicit state, and scores both designated transitions and the complete trajectory. It therefore remains difficult to determine whether agents preserve valid commitments, propagate revisions, and recover after new information invalidates the current direction.

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.

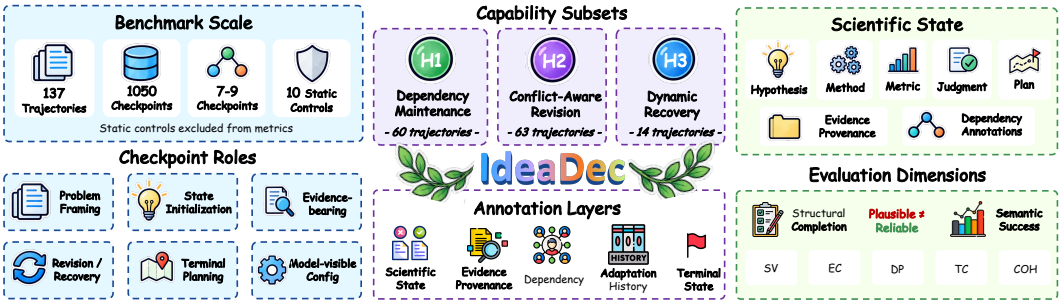


Figure 1: **Overview of IDEADEC.** Models update a five-field scientific state across 137 trajectories in three disjoint capability subsets. Evaluation reports structural completion, success at a pre-specified target operation, and whole-trajectory semantic success; 10 static controls are reported separately and excluded from these metrics.

To address these challenges, we introduce IDEADEC to evaluate scientific-state evolution under controlled evidence release. We define *scientific idea decoding* as translating sequentially released evidence into coherent updates to an explicit scientific state. At each checkpoint, a model combines permitted history, released evidence, and model-visible configuration to update five mutable components: hypothesis, method, metric, judgment, and plan. Future evidence, reference states, dependency annotations, and expected transitions remain hidden during generation.

Figure 1 summarizes the benchmark. IDEADEC contains 137 trajectories of 7–9 checkpoints, totaling 1,050 checkpoints. The trajectories form three disjoint capability subsets: Dependency Maintenance (H1), Conflict-Aware Revision (H2), and Dynamic Recovery (H3). One target checkpoint is fixed for each trajectory before inference. Our contributions are:

- **Long-Horizon Scientific Idea Decoding Benchmark.** We introduce IDEADEC for evaluating maintenance, revision, and recovery under ordered evidence release through source-grounded, dependency-governed trajectories.
- **Explicit Evaluation Contract.** We separate structural completion, target-operation success, and five-gate semantic success. The protocol carries forward the model's structurally valid state without reference repair, preventing evaluator annotations from altering execution.
- **Empirical Characterization.** Across 18 configurations in the State-Aware setting (M3), structural completion exceeds semantic success, and recovery is consistently lowest. The results motivate dependency tracking, revision propagation, and recovery mechanisms.

2 RELATED WORK

Scientific Idea Generation and Evaluation. Scientific-ideation benchmarks evaluate generated hypotheses or proposals for novelty, feasibility, and evidential support (Guo et al., 2025; Qiu et al., 2025; Liu et al., 2025; Si et al., 2025), building on advances in automated experimentation, modeling, and structure prediction (King et al., 2009; Schmidt & Lipson, 2009; Jumper et al., 2021). ProjectionBench studies progressive information disclosure; DriftBench evaluates multi-turn constraint adherence with human validation (Lew et al., 2026; Kruthof, 2026). These benchmarks capture idea quality or adaptation, but none jointly controls evidence release, tracks an explicit scientific state, and evaluates whole-trajectory semantic validity. IdeaDec combines a five-field state schema, a pre-specified target operation, and non-compensatory gates over the complete sequence.

Scientific Agents and Workflow Evaluation. Scientific agents integrate literature analysis, tool use, experimentation, and research reporting (Bran et al., 2024; Lu et al., 2024; Baek et al., 2025), while benchmarks cover discovery, machine-learning experimentation, research reproduction, and scientific tasks (Huang et al., 2024; Chen et al., 2025; Majumder et al., 2025; Chan et al., 2025; Starace et al., 2025; Liu et al., 2026a). AgentIdeaBench evaluates active exploration under matched settings (Mo et al., 2026). Many such evaluations emphasize open-ended execution and tool-mediated outcomes. IdeaDec instead fixes evidence release and disables tools and retrieval, holding the information schedule constant and separating state-update consistency from search strategy, tool quality, and dynamically acquired context.

Table 1: **Comparison with representative scientific-ideation and hypothesis-generation benchmarks.** ✓ denotes an explicitly represented feature: multiple criteria, dependent process operations, agent-directed information acquisition, ordered evidence release, or cross-decision dependency consistency. × denotes no such criterion. Scale units follow source reports and are not directly comparable.

Benchmark	Benchmark Design					Dataset Profile	
	Multi-Criteria	Process Tasks	Active Exploration	Sequential Evidence	Cross-Step Consistency	Source	Scale
Static or decomposed ideation benchmarks							
IdeaBench (Guo et al., 2025)	✓	✗	✗	✗	✗	Papers	2,374 papers
AI Idea Bench (Qiu et al., 2025)	✓	✗	✗	✗	✗	AI papers	3,495 papers
LiveIdeaBench (Ruan et al., 2026)	✓	✗	✗	✗	✗	Keywords	1,180 keywords
HypoBench (Liu et al., 2025)	✓	✗	✗	✗	✗	Data	12 domains / 194 datasets
ResearchBench (Liu et al., 2026b)	✗	✓	✗	✗	✗	Papers	1,386 papers / 12 disciplines
Ideation Arena (Chen et al., 2026)	✓	✗	✗	✗	✗	Literature	>6K comparisons
HypoArena (Zhong et al., 2026)	✓	✗	✗	✗	✗	Documents	988 cases
Sequential or interactive ideation benchmarks							
ProjectionBench (Lew et al., 2026)	✗	✓	✗	✓	✗	Papers	45 papers
DriftBench (Kruthof, 2026)	✓	✓	✗	✗	✓	Briefs	38 briefs / 2,146 runs
AgentIdeaBench (Mo et al., 2026)	✓	✓	✓	✗	✗	Literature	100 curated / 40 scored / 5 disciplines
IdeaDec (Ours)	✓	✓	✗	✓	✓	**Trajectories**	**137 traj. / 1,050 checkpoints**

Long-Horizon Agent Evaluation. Research on long-horizon agents examines planning, memory, tool use, and persistent execution (Yao et al., 2023; Shinn et al., 2023; Wang et al., 2023; Packer et al., 2023; Wang et al., 2024), with benchmarks spanning web, software, desktop, and extended research tasks (Liu et al., 2024; Zhou et al., 2024; Jimenez et al., 2024; Xie et al., 2024; Luo et al., 2025; Zhao et al., 2026a; Li et al., 2026). ScienceFlow incorporates recoverable executable states and adaptive redirection (Zhao et al., 2026b). IdeaDec instead evaluates scientific dependencies that persist across ordered evidence releases. Decisions are judged against contemporaneous evidence; one pre-specified target operation and whole-trajectory gates separate target success from cumulative consistency, making intermediate commitments an explicit object of evaluation. Table 1 compares these dimensions across representative scientific-ideation benchmarks.

3 IDEADEC

IDEADEC evaluates whether language models can coherently update scientific states as evidence is sequentially released. Each trajectory represents interdependent decisions over hypotheses, methods, metrics, judgments, and plans. Figure 2 summarizes benchmark construction, capability transitions, and evaluation.

3.1 SCIENTIFIC TRAJECTORY CONSTRUCTION

Each IdeaDec trajectory begins with a source-grounded seed containing an objective, initial evidence and hypothesis, a candidate method, and operational constraints. Its mutable state comprises five components: hypothesis, method, metric, judgment, and plan. Released evidence and model-visible configuration are inputs; adaptation and evidence identifiers are auxiliary outputs. Source provenance, dependency annotations, and reference transitions remain evaluator-only metadata identifying each finding's source, affected state components, and admissible changes.

Trajectory Execution. At each checkpoint, the model receives instructions, released evidence, any visible configuration, and permitted history, and must return seven fields. The M3 executor projects five state fields from the latest structurally valid output, even if scientifically incorrect. A missing or unparseable output leaves the projection unchanged; any returned raw response remains in history. Future evidence, reference states, dependency annotations, and expected transitions remain hidden; reference annotations never repair the state (Appendix D.3).

Dependency Annotation. The annotation protocol assigns each finding, its prerequisites, validity conditions, and affected downstream fields to an initial annotator. The item contract specifies retained and revised commitments and the scored target; a second annotator reviews dependency scope, admissible outcomes, and target placement before freezing. The protocol also requires stratified outputs to be scored independently by two evaluators for all five gates and the capability target,

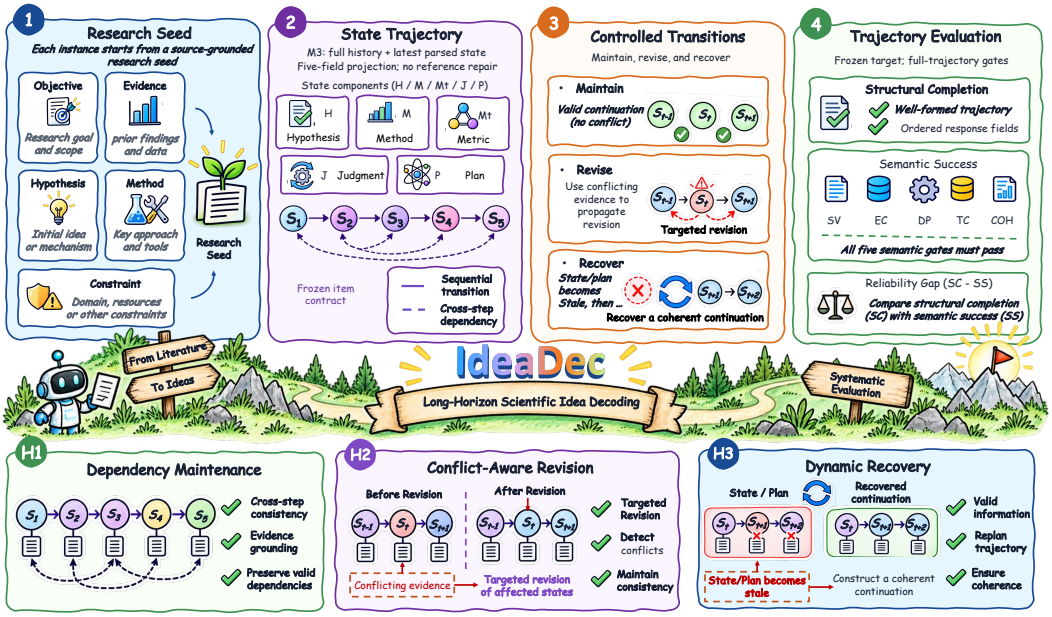


Figure 2: **Construction, execution, and evaluation of an IDEADEC trajectory.** A source-grounded seed and frozen item contract specify an ordered release schedule and one target checkpoint. Under the State-Aware setting (M3), the latest structurally valid five-field state is supplied for the next update without reference repair. Target-operation success is scored within one capability subset, whereas semantic success applies to the complete trajectory.

with first-pass decisions and adjudications linked in experimental records. Appendices A and E detail the frozen contract and accompanying evaluation records.

3.2 CAPABILITY-ORIENTED TASK DESIGN

IdeaDec assigns each trajectory to one of three parallel capability subsets: Dependency Maintenance, Conflict-Aware Revision, or Dynamic Recovery. H1–H3 are disjoint task categories, neither temporal stages nor ordered difficulty levels; maintenance, revision, and recovery operations may coexist within a trajectory under the shared execution protocol.

Dependency Maintenance (H1). H1 tests whether a model preserves dependencies that remain valid under newly released evidence. Valid hypotheses, constraints, and methodological commitments must continue to govern affected decisions; their scope changes only when the evidence warrants it. The target fails when a required dependency is omitted, contradicted without evidence, or not enforced in the returned state.

Conflict-Aware Revision (H2). H2 tests whether a model revises an active state when released evidence creates a material conflict. The target response must revise the affected component and propagate the change to every dependent state field. Acknowledging the conflict while retaining an incompatible decision in that response fails the target operation.

Dynamic Recovery (H3). H3 tests whether a model recovers a feasible continuation after released evidence or configuration invalidates the current direction. The target response must replace the invalidated continuation while preserving information and constraints that remain valid. Continuing the invalid path or discarding unaffected commitments fails the target. Any alternative continuation satisfying the evidence, objective, and remaining constraints is admissible.

3.3 TRAJECTORY-LEVEL EVALUATION

IdeaDec evaluates each trajectory against its frozen item contract, including its evidence-release schedule and admissible state transitions. Decisions are judged using evidence available at each checkpoint; later evidence does not alter the evaluation of a decision justified by the evidence then

available. Section 4.1 specifies aggregation, and Appendix B.2 provides gate definitions and adjudication cases.

Structural Completion. Structural Completion (SC) requires every checkpoint response to be present and parseable with the prescribed field set and order. It measures response coverage and serialization, independently of scientific validity.

Semantic Success. Semantic Success (SS) requires SC and all five non-compensatory gates: state validity (SV), evidence consistency (EC), dependency preservation (DP), transition correctness (TC), and terminal coherence (COH). All five gates apply to every capability subset, with each evaluated over its specified scope; TC covers all required operations, including the designated target. Later correction does not erase an earlier decisive violation.

Each trajectory has one target checkpoint fixed before inference. H1–H3 measure target-operation success within their respective subsets, using prior states, the evidence prefix, and affected fields in the target response. Missing or unparseable target responses fail. Target success is necessary but insufficient for SS: failures elsewhere can invalidate the trajectory without changing that target label. Later violations are assessed through whole-trajectory gates.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Dataset. The main evaluation comprises 137 IDEADEC trajectories and 1,050 checkpoints; 10 static controls are reported separately and excluded from the benchmark metrics.

H1, H2, and H3 contain 60, 63, and 14 trajectories with fixed horizons of 7, 8, and 9 checkpoints, totaling $60(7) + 63(8) + 14(9) = 1{,}050$ checkpoints (mean 7.66). All trajectories include problem framing, three state initializations, two evidence releases, and terminal planning; H2 adds revision, while H3 adds recovery and model-visible configuration. Table 2 accounts for every checkpoint. One target operation is fixed per trajectory before inference. Because subset content, horizon, and size differ, we treat H1–H3 as parallel capability diagnostics; Appendix A gives the construction contract.

Table 2: **Composition of IDEADEC.**

Category	Count (Ratio)
Global scale	
Trajectories	137
Checkpoints	1,050
Average horizon	7.66
Capability subsets	
H1: Dependency Maintenance	60 (43.8%)
H2: Conflict-Aware Revision	63 (46.0%)
H3: Dynamic Recovery	14 (10.2%)
Checkpoint roles	
Problem framing	137 (13.0%)
State initialization	411 (39.1%)
Evidence-bearing	274 (26.1%)
Revision / recovery	77 (7.3%)
Terminal planning	137 (13.0%)
Model-visible configuration	14 (1.3%)

Models. Table 3 reports 18 fixed model configurations, grouped as Open-source Models (10) and Closed-source Models (8). Each configuration is bound to its release or endpoint snapshot, date, weight and license status, access channel, limits, effective sampling parameters, and retry policy. Group and overall means weight configurations equally.

Evaluation protocol. At checkpoint t, the State-Aware setting (M3) supplies the task, released evidence, any model-visible configuration, full response history, and latest structurally valid five-field state, or the visible seed until one exists. Parseable states are carried forward even when semantically incorrect; future evidence and evaluator records remain hidden. Tools and retrieval are disabled; temperature is 0 and top-p is 1 where exposed, with provider defaults otherwise. Responses contain five state fields plus adaptation and evidence identifiers; M3 projects only the state fields. Appendices D and E.5 specify context and record requirements.

Metrics and aggregation. SC is the percentage of trajectories for which every checkpoint response is present and parseable with the prescribed fields and order; SS additionally requires all five semantic gates. The reported integer counts determine every displayed rate and derived quantity through the shared aggregation pipeline. The auxiliary summaries $RG = SC - SS$ and $S\text{-}Avg = (SC + SS)/2$ are interpreted jointly with SC and SS (Appendix B.1). H1–H3 report target success over 60, 63, and 14 items, so one label changes them by 1.67, 1.59, and 7.14 points.

Table 3: **Results on IDEADEC.** SC and SS are trajectory-level percentages; H1–H3 are target-operation percentages. S-Avg and H-Avg are percentage averages; RG and Drop are percentage-point differences. Bold and underline mark the highest and second-highest point estimates (lowest for RG; ties retained); Drop is not ranked.

Model	Reliability Evaluation				Capability Evaluation				
	SC	SS	RG	S-Avg	H1	H2	H3	Drop	H-Avg
Open-source Models									
DeepSeek-V4-Pro	89.78	40.15	49.64	64.96	48.33	38.10	14.29	34.05	33.57
Qwen3.8-2.4T-A95B	83.94	59.12	24.82	71.53	68.33	58.73	21.43	46.90	49.50
Moonshot Kimi K3	86.13	50.36	35.77	68.25	60.00	47.62	21.43	38.57	43.02
GLM-5.3	88.32	48.18	40.15	68.25	58.33	44.44	21.43	36.90	41.40
MiniMax-M3	86.13	44.53	41.61	65.33	55.00	41.27	14.29	40.71	36.85
Meta Llama 4 Maverick	89.78	66.42	<u>23.36</u>	78.10	71.67	68.25	**35.71**	35.95	58.54
Mistral Large 3	83.94	56.20	27.74	70.07	65.00	55.56	21.43	43.57	47.33
Gemma 4 31B IT	83.94	39.42	44.53	61.68	50.00	34.92	14.29	35.71	33.07
NVIDIA Nemotron 3 Super	83.94	44.53	39.42	64.23	55.00	39.68	21.43	33.57	38.70
Phi-4 Reasoning	81.75	35.04	46.72	58.39	45.00	30.16	14.29	30.71	29.81
Average	85.77	48.39	37.37	67.08	57.67	45.87	20.00	37.67	41.18
Closed-source Models									
Cohere Command A Reasoning	**97.08**	14.60	82.48	55.84	18.33	12.70	7.14	11.19	12.72
Gemini 3.7 Flash	91.97	48.91	43.07	70.44	60.00	44.44	21.43	38.57	41.96
Claude Sonnet 4.6	92.70	58.39	34.31	75.55	68.33	55.56	<u>28.57</u>	39.76	50.82
Gemini 3.1 Pro Preview	<u>94.16</u>	62.04	32.12	78.10	71.67	60.32	<u>28.57</u>	43.10	53.52
Qwen3.8-Max	83.94	65.69	**18.25**	74.82	70.00	<u>69.84</u>	<u>28.57</u>	41.43	56.14
Grok 4.6	**97.08**	**68.61**	28.47	**82.85**	70.00	**74.60**	**35.71**	34.29	**60.11**
Claude Opus 4.6	<u>94.16</u>	64.96	29.20	79.56	<u>75.00</u>	61.90	**35.71**	39.29	57.54
GPT-5.6 Sol	<u>94.16</u>	<u>67.88</u>	26.28	<u>81.02</u>	**78.33**	65.08	**35.71**	42.62	<u>59.71</u>
Average	93.16	56.39	36.77	74.77	63.96	55.56	27.68	36.28	49.06
Overall Average	89.05	51.95	37.10	70.50	60.46	50.18	23.41	37.05	44.68

$H\text{-}Avg = (H1 + H2 + H3)/3$ is an equal-weight macro-average; $Drop = \Delta_{13} = H1 - H3$ is a cross-subset contrast. We ask whether SC tracks SS, whether H-Avg masks capability structure, and how rate-based weakness differs from total failure burden. Rates derive from integer counts, with decimal half-up rounding only for display.

4.2 MAIN RESULTS

Large gaps separate structural and semantic success. Across 18 configurations in Table 3, mean SC is 89.05% and SS is 51.95%, yielding an RG of 37.10 percentage points. The closed-source group exceeds the open-source group by 7.39 and 7.99 percentage points in SC and SS, respectively, while its mean RG is 0.60 points smaller. Configuration-level RG ranges from 18.25 to 82.48 percentage points; Command A Reasoning reaches the upper endpoint despite pairing the highest SC (97.08%) with the lowest SS (14.60%). This separation shows that format compliance does not imply semantic reliability, while aggregate rates cannot localize the failing gate.

Top aggregate scores conceal different capability profiles. Grok 4.6 and GPT-5.6 Sol have similar H-Avg values of 60.11% and 59.71%, yet their success profiles differ. GPT-5.6 Sol passes five more H1 targets (47 versus 42), Grok 4.6 passes six more H2 targets (47 versus 41), and both pass 5/14 H3 targets. H-Avg alone is therefore insufficient to identify the underlying profile: a similar macro-average can encode different maintenance and revision behavior. Moreover, one H3 label changes H-Avg by 2.38 percentage points, over five times their 0.40-percentage-point separation. Their ordering is consequently descriptive rather than a stable distinction. Four configurations tie at the H3 maximum of 35.71% (5/14), whose Wilson 95% interval is approximately [16.3%, 61.2%].

H3 has the lowest reported subset rate. Mean success is 60.46% on H1, 50.18% on H2, and 23.41% on H3. H3 is lowest in all 18 rows, ranging from 7.14% to 35.71% across 14 items. This consistency identifies recovery as the benchmark's most persistent low-rate region under its fixed task distribution, but not as intrinsically harder because H1–H3 differ in content, horizon, and sample size. One label changes H3 by 7.14 percentage points, compared with 1.67 for H1 and 1.59 for H2;

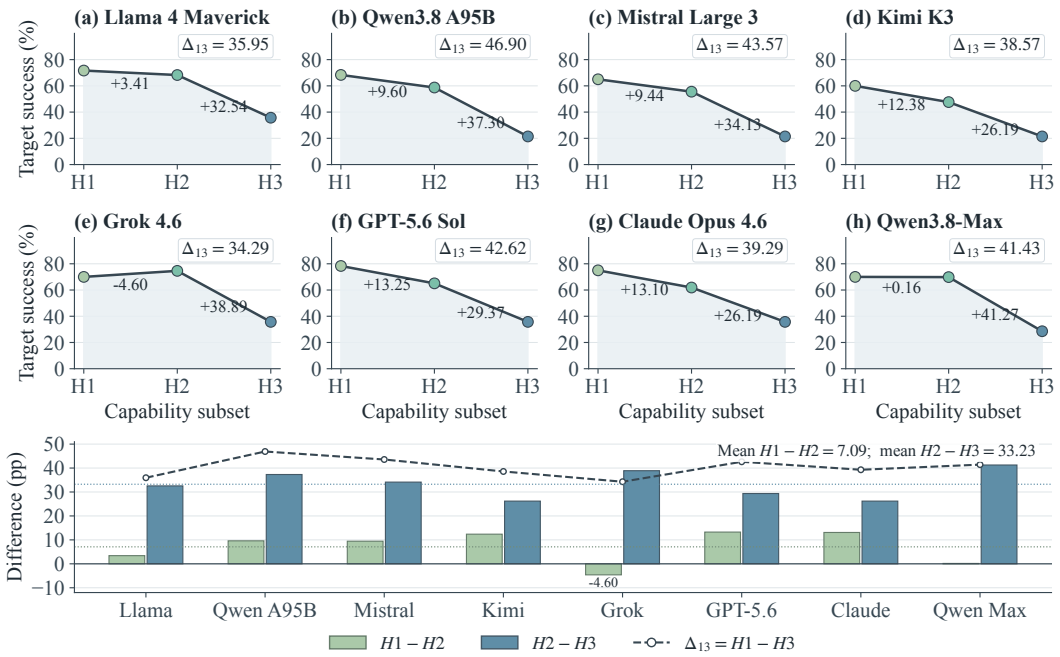


Figure 3: **Capability profiles and signed cross-subset differences.** Panels (a)–(d) show Open-source Models and (e)–(h) Closed-source Models. Segment labels and lower-panel summaries report $H1 - H2$, $H2 - H3$, and $\Delta_{13} = H1 - H3$ in percentage points, retaining negative values. H1–H3 are disjoint subsets; each group contributes its four highest H-Avg point estimates.

exact counts and the 5/14 Wilson interval are therefore essential. Attributing this gap to recovery capability itself requires matched variants rather than these marginals.

4.3 Capability-Specific Performance Profiles

Figure 3 shows the four highest H-Avg point estimates in each reported model group after full-table aggregation; Table 3 retains all 18 configurations. The connecting segments compare disjoint subset rates and therefore represent capability profiles rather than temporal trajectories. Across the eight selected rows, the mean signed differences are $H1 - H2 = 7.09$ points and $H2 - H3 = 33.23$ points. Seven rows have a positive H1–H2 difference, from Qwen3.8-Max's near tie of 0.16 points to 13.25 for GPT-5.6 Sol; Grok 4.6 is the sole reversal, with H2 exceeding H1 by 4.60 points. GPT-5.6 Sol passes 47/60 H1 and 41/63 H2 targets, whereas Grok 4.6 passes 42/60 and 47/63; both pass 5/14 H3 targets. The H2–H3 difference is positive in all eight rows, and four share the 5/14 H3 maximum. Because selection conditions on H-Avg, these patterns characterize the displayed high-scoring profiles and should not be interpreted as prevalence estimates over all 18 configurations. A descriptive equal-SC contrast further shows that structural completion does not determine the capability profile. Qwen3.8-Max, Gemma 4 31B IT, and Mistral Large 3 each attain 83.94% SC, but their SS rates range from 39.42% to 65.69% and their H-Avg values from 33.07% to 56.14%. Their subset rates also differ materially: H1 spans 50.00–70.00%, H2 spans 34.92–69.84%, and H3 spans 14.29–28.57%. Thus, identical observed SC constrains only structural completion; it neither implies semantic validity nor identifies how success is distributed across maintenance, revision, and recovery. Appendix B.3 formalizes the distinct aggregation and resolution of these measures.

4.4 Target-Operation Failure Composition

Table 4 expresses the same eight profiles as failed-target burden. Let $N_k \in \{60, 63, 14\}$ denote subset size and $p_{d,k}$ the target-success count for row d; then $f_{d,k} = N_k - p_{d,k}$, $F_d = \sum_k f_{d,k}$, and $w_{d,k} = 100 f_{d,k}/F_d$. Totals range from 43/137 (31.4%) for Grok 4.6 to 68/137 (49.6%) for Moonshot Kimi K3. Grok and GPT differ by one failure overall, yet their largest contributors differ: H1 supplies 18 for Grok, whereas H2 supplies 22 for GPT. Similar totals therefore do not imply similar capability deficits: Grok's burden is maintenance-heavy, whereas GPT's is revision-heavy,

Table 4: **Composition and burden of failed target operations.** Cells report n (within-row share %); failure rates use 137 targets. Dominant marks the largest count, Average Share the unweighted row mean, and bold row maxima. Rows are the four highest-H-Avg configurations per model group.

Model	Failed Target Operations: n (Share %)			Failure Burden		
	H1 Maintenance	H2 Revision	H3 Recovery	Total Failures	Failure Rate (%)	Dominant Subset
Meta Llama 4 Maverick	17 (37.0)	**20 (43.5)**	9 (19.6)	46	33.6	H2
Qwen3.8-2.4T-A95B	19 (33.9)	**26 (46.4)**	11 (19.6)	56	40.9	H2
Mistral Large 3	21 (35.0)	**28 (46.7)**	11 (18.3)	60	43.8	H2
Moonshot Kimi K3	24 (35.3)	**33 (48.5)**	11 (16.2)	68	49.6	H2
Grok 4.6	**18 (41.9)**	16 (37.2)	9 (20.9)	43	31.4	H1
GPT-5.6 Sol	13 (29.5)	**22 (50.0)**	9 (20.5)	44	32.1	H2
Claude Opus 4.6	15 (31.3)	**24 (50.0)**	9 (18.8)	48	35.0	H2
Qwen3.8-Max	18 (38.3)	**19 (40.4)**	10 (21.3)	47	34.3	H2
Average Share	35.3%	**45.3%**	19.4%	–	–	**H2**

a distinction hidden by the aggregate failure rate. H2 has the largest count in seven rows and a mean share of 45.3%, versus 35.3% for H1 and 19.4% for H3. Subset size explains why counts and rates imply different priorities: H3 has the lowest rate but contributes only 9–11 failures per row, whereas H2's 63 targets make it the largest aggregate burden. For these selected profiles, rate-based diagnosis prioritizes H3; volume-based remediation prioritizes H2. A fixed intervention budget would therefore target different subsets depending on whether the objective is worst-subset robustness or the number of corrected target operations.

4.5 Schematic State Evolution

Figure 4 illustrates two consequences of the evaluation protocol rather than an empirical error curve. Under M3, the latest structurally valid five-field state is projected into the next request; semantic correctness does not control this update, so a parseable but incorrect state can affect subsequent decisions (Appendix D.3). Error position is therefore consequential: an early projected error can alter several later decisions, whereas a terminal error has no downstream state update. Independently, a commitment can constrain later state components across multiple checkpoints. Failure to propagate a warranted revision may therefore leave individual responses locally plausible while rendering the trajectory globally inconsistent. Panel (a)'s six checkpoints and numerical scale are illustrative, whereas panel (b) represents the reach of cross-checkpoint dependencies. Evaluation is performed against the frozen release schedule and dependency contract: DP tests whether valid commitments remain operative, TC tests whether the required state operation is executed, and COH tests whether the terminal continuation is jointly compatible with retained evidence and constraints (Appendix B.2).

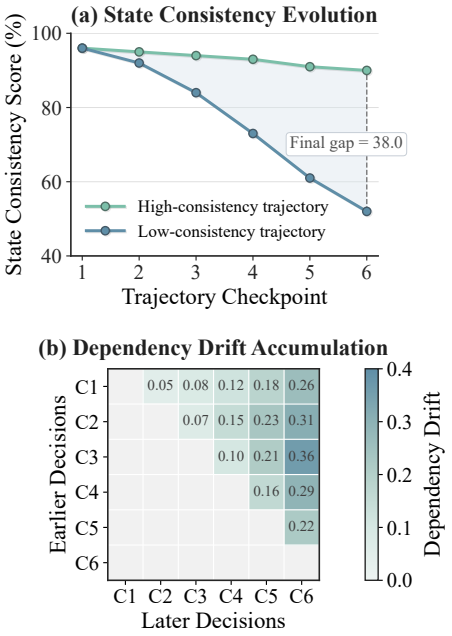


Figure 4: **Schematic state evolution.** (a) Consistency paths; (b) cross-checkpoint dependence. Values and axes are illustrative.

4.6 ILLUSTRATIVE STATE-EVOLUTION PATHS

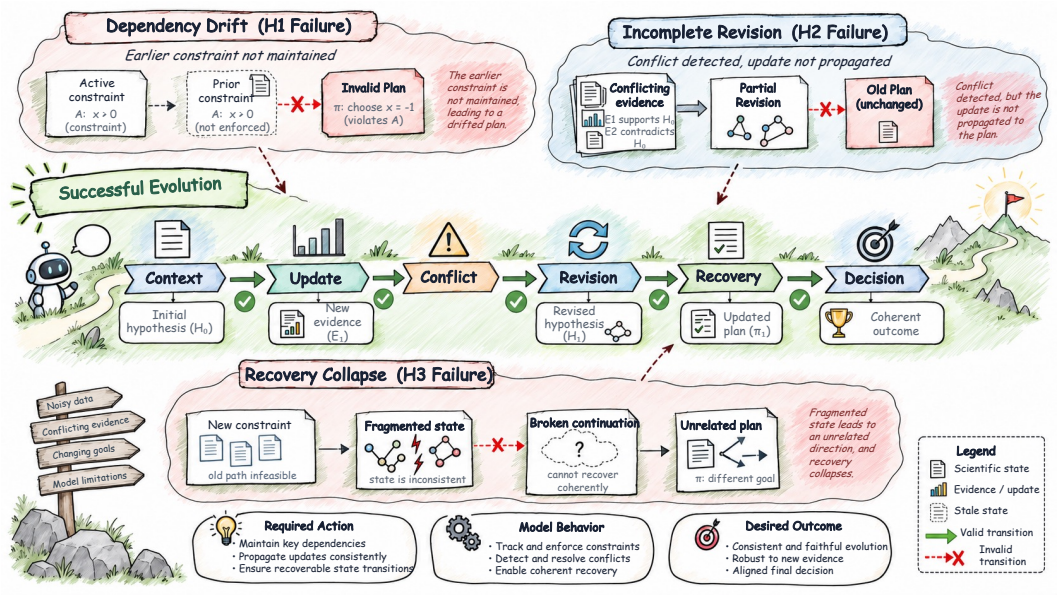


Figure 5: **Illustrative paths through scientific-state evolution.** The central path traces a coherent progression from context and new evidence through conflict, revision, recovery, and decision. The surrounding insets contrast dependency drift (H1), incomplete revision (H2), and recovery collapse (H3); green and red arrows denote valid and invalid transitions. All paths are schematic.

Figure 5 instantiates the target-level adjudication criteria around a shared schematic success path; it does not estimate failure prevalence. The H1 target fails when the still-valid constraint $x > 0$ ceases to govern the dependent plan, which selects $x = -1$. The H2 target fails when conflicting evidence prompts a hypothesis revision but not the corresponding plan update. The H3 target fails when a new constraint invalidates the active continuation and the returned sequence fragments the state before substituting an unrelated objective; it neither preserves valid information nor constructs a feasible continuation. The insets thus distinguish maintenance, revision propagation, and recovery by the relation between an update and its dependent decisions. These judgments concern targets; SS conjoins five gates over the trajectory. A designated-target failure entails TC and hence SS failure, but target success cannot ensure SS if another checkpoint or gate fails. Later valid recovery may thus pass H3 without repairing prior SS failure. Other gates require the full evidence prefix and item contract. Table 9 and Appendix C formalize these boundaries with domain-grounded examples.

5 CONCLUSION

IDEADEC reframes the evaluation of AI research agents from isolated-output scoring to tracing how an explicit scientific state evolves under controlled evidence release. By separating structural completion, target-operation success, and whole-trajectory semantic success, the benchmark distinguishes producing a parseable sequence from maintaining evidential and dependency coherence. The results reveal two complementary failure patterns: a substantial structure–semantics gap and capability profiles that diverge even when aggregate scores or structural completion are similar. Recovery has the lowest subset rate across all evaluated configurations, but this pattern describes the fixed benchmark distribution rather than establishing that recovery is intrinsically harder. Together, these findings support treating designated operations and complete trajectories as complementary units of evaluation. This framework provides a basis for developing research agents that explicitly preserve dependencies, propagate revisions, and reconstruct feasible plans as evidence changes.

AI USE STATEMENT

Generative AI tools aided and polished writing; drafted paper sections; supported literature retrieval and discovery; and assisted research ideation and execution through experimental-design feedback, code, figures, audits, and result interpretation. They generated neither synthetic datasets nor mathematical claims or proofs.

ETHICS STATEMENT

IDEADEC evaluates model-generated scientific-state updates in an offline research setting. Human review supports reference construction, semantic adjudication, and calibration. The release protocol requires item-level source provenance, transformation history, redistribution permissions, and sensitive-content screening. Results characterize scientific-state consistency under the stated protocol; they do not certify the safety of downstream scientific deployments.

REPRODUCIBILITY STATEMENT

The anonymized experimental supplement contains benchmark items, schemas and prompts, source and license manifests, configuration and run records, raw responses, parser outputs, and item-level scoring and review records. Appendix E.5 describes their joins and coverage checks. The manuscript source package provides the frozen count exports and scripts used to regenerate the reported tables, derived metrics, and quantitative figures.

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 *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6709–6738, 2025.

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

Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations*, 2025.

Zhiyu Chen, Keyu Zhao, Jigao Fu, Dong Liang, Yanbiao Wu, Jiaoyang Li, Haidong Xue, Xinhua Zeng, Yuanyi Zhen, Fengli Xu, et al. Ideation arena: Evaluating LLM-generated research ideas with battle-style human expert assessment. *arXiv preprint arXiv:2608.29696*, 2026.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, 2025.

Sikun Guo, Amir Hassan Shariatmadari, Guangzhi Xiong, Albert Huang, Myles Kim, Corey M Williams, Stefan Bekiranov, and Aidong Zhang. IdeaBench: Benchmarking large language models for research idea generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5888–5899, 2025.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309. PMLR, 2024.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024.

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

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

Garvin Kruthof. Models recall what they violate: Constraint adherence in multi-turn LLM ideation. *arXiv preprint arXiv:2604.28031*, 2026.

Andrew J Lew, Yuan Cao, and Markus J Buehler. ProjectionBench: Evaluating scientific hypothesis generation in LLMs under progressive information disclosure. *arXiv preprint arXiv:2605.30284*, 2026.

Yiwei Li, Wanli Yang, Hexiang Tan, Xiangzhou Huang, Zhengyu Chen, Ziran Li, Borun Chen, Shanglin Lei, Huaisheng Zhu, Hao Tian, et al. Beyond final scores: A systematic evaluation of agents for long-horizon AI research and development. *arXiv preprint arXiv:2608.13417*, 2026.

Haokun Liu, Sicong Huang, Jingyu Hu, Yangqiaoyu Zhou, and Chenhao Tan. HypoBench: Towards systematic and principled benchmarking for hypothesis generation. *arXiv preprint arXiv:2504.11524*, 2025.

Tianyu Liu, Allen Xin Wang, Antonia Panescu, Lisa Xinyi Chen, Wenxin Long, Xinyu Wei, Yueqian Jing, Ziyao Zeng, Jihang Chen, Sihan Jiang, et al. Benchmarking AI agents for addressing scientific challenges across scales. *arXiv preprint arXiv:2606.12736*, 2026a.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024.

Yujie Liu, Zonglin Yang, Tong Xie, Jinjie Ni, Ben Gao, Yuqiang Li, Shixiang Tang, Wanli Ouyang, Erik Cambria, and Dongzhan Zhou. ResearchBench: Benchmarking LLMs in scientific discovery via inspiration-based task decomposition. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13187–13207, 2026b.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Haotian Luo, Huaisong Zhang, Xuelin Zhang, Haoyu Wang, Zeyu Qin, Wenjie Lu, Guozheng Ma, Haiying He, Yingsha Xie, Qiyang Zhou, et al. UltraHorizon: Benchmarking agent capabilities in ultra long-horizon scenarios. *arXiv preprint arXiv:2509.21766*, 2025.

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 *International Conference on Learning Representations*, 2025.

Yunxiang Mo, Tianshi Zheng, Yisen Gao, Rui Wang, Newt Nguyen Kim Hue Nam, Kelvin Kiu Wai Tam, Jiaxin Bai, Yangqiu Song, Ginny Wong, and Simon See. AgentIdeaBench: Benchmarking scientific ideation in the agent era. *arXiv preprint arXiv:2609.07611*, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

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

Kai Ruan, Xuan Wang, Jixiang Hong, Peng Wang, Yang Liu, and Hao Sun. Evaluating LLMs' divergent thinking capabilities for scientific idea generation with minimal context. *Nature Communications*, 17(1):3625, 2026.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

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

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

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, et al. PaperBench: Evaluating AI's ability to replicate AI research. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56843–56873. PMLR, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Jiarui Zhao, Rongzhi Zhang, Lingchuan Liu, Hao Yang, Xunliang Cai, and Xi Su. LoHoSearch: Benchmarking long-horizon search agents beyond the human difficulty ceiling. *arXiv preprint arXiv:2606.12837*, 2026a.

Mingming Zhao, Jiqian Dong, Kangping Xu, Zadid Hasan, Chengrui Fan, Shan Jiang, Shuai Mao, Yating Ling, Linyi Zou, Tailin Zhou, et al. ScienceFlow: A long-horizon agent for ML research, scientific discovery and beyond. *arXiv preprint arXiv:2608.14354*, 2026b.

Tianyun Zhong, Wangyi Jiang, Wei Wang, Xuanang Chen, Yaojie Lu, Shiwei Ye, Yuzhen Shi, Boyu Yang, Jinghang Wang, Han Li, et al. Before the action: Benchmarking LLMs on prospective hypothesis discovery. *arXiv preprint arXiv:2607.15766*, 2026.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024.

A Benchmark Construction and Data Contract

A.1 Capability Assignments

An IdeaDec trajectory records an evolving scientific state with five mutable components: hypothesis, method, metric, judgment, and plan. The objective, released evidence, and operational constraints are inputs to the update process rather than components of the mutable state. Source provenance, dependency annotations, and reference transitions remain separate evaluation records.

The three capability subsets are assigned before inference and remain fixed across deployments. H1 tests whether a dependency remains enforced while it is valid; H2 tests whether a material conflict leads to substantive revision and propagation; H3 tests whether the model can replace an invalidated continuation while retaining information that remains valid. Table 5 states the target operation for each subset. Their rates form parallel capability diagnostics; content, size, and horizon remain properties of each stratum rather than an ordered difficulty scale.

Table 5: Capability assignments in the main evaluation. n is the number of trajectories and T the checkpoint horizon.

Subset	n	T	**Target operation**	**Decisive failure**
H1: Dependency Maintenance	60	7	Preserve every still-valid dependency; qualify its scope only when released evidence warrants the change.	A valid commitment disappears or ceases to constrain a dependent decision.
H2: Conflict-Aware Revision	63	8	Revise the component contradicted by new evidence and propagate the revision to affected decisions.	The conflict is acknowledged, but an incompatible state or dependent decision remains active.
H3: Dynamic Recovery	14	9	Identify the invalidated continuation, retain valid information, and construct a feasible alternative.	The response follows the invalid path or abandons constraints that remain valid.

A.2 Checkpoint Composition

Every trajectory contains one problem-framing checkpoint, three state-initialization checkpoints, two evidence-bearing checkpoints, and one terminal-planning checkpoint. H2 adds a revision checkpoint. H3 adds a recovery checkpoint and an ordered, model-visible configuration checkpoint. The configuration event belongs to the nine-checkpoint H3 sequence and requires a model output under the same execution protocol; it is not hidden annotation metadata.

Table 6: Checkpoint accounting by subset. PF denotes problem framing; Init., state initialization; Evid., evidence bearing; R/R, revision or recovery; Term., terminal planning; and Config., model-visible configuration.

Subset	n	T	**PF**	**Init.**	**Evid.**	**R/R**	**Term.**	**Config.**	**Total**
H1	60	7	60	180	120	0	60	0	420
H2	63	8	63	189	126	63	63	0	504
H3	14	9	14	42	28	14	14	14	126
Total	**137**	**7–9**	**137**	**411**	**274**	**77**	**137**	**14**	**1,050**

The row totals reproduce $60(7) + 63(8) + 14(9) = 1{,}050$ checkpoints and the mean horizon $1{,}050/137 = 7.66$. These equalities verify the aggregate accounting. The checkpoint registry joins every checkpoint to its trajectory, source, role, and release event.

A.3 Frozen Targets and Information Release

Each item contract binds a stable trajectory identifier to one subset, an ordered checkpoint sequence, an evidence-release schedule, and one target checkpoint selected before inference. For H1 the target is the designated maintenance event; for H2, the designated revision; for H3, the designated recovery. The same record specifies the affected dependencies, admissible outcomes, forbidden outcomes, and the target pass rule. This assignment remains identical across all configurations evaluated on the item.

At checkpoint t, the model sees the objective and constraints, all evidence and configuration released through t, the current instruction, and the history permitted by the selected context setting. Evidence

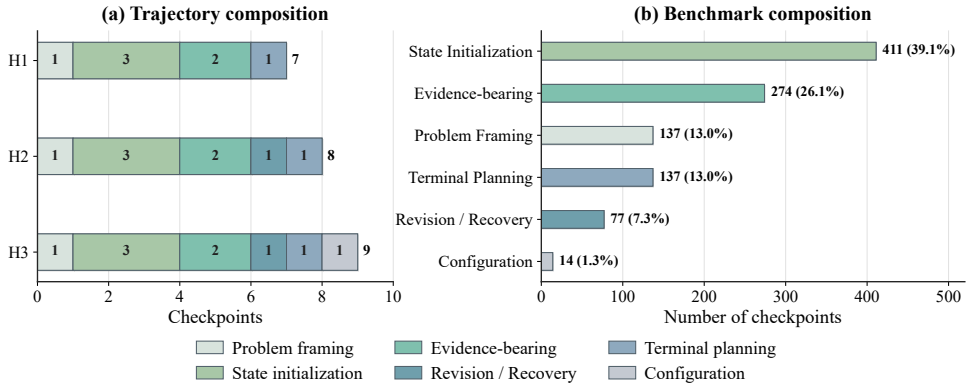


Figure 6: **Capability-specific checkpoint composition in IDEADEC.** The figure visualizes the mutually exclusive checkpoint roles audited numerically in Table 6. H2 adds revision, whereas H3 adds recovery and model-visible configuration.

Table 7: Minimum item contract. Generation fields and evaluator-only fields share a trajectory identifier but are projected into different views.

Record block	Contents	Use	Integrity condition
Identity	Trajectory identifier (ID), subset, horizon, freeze version	All stages	One immutable identity and one capability assignment
Checkpoint map	Ordered checkpoint IDs, roles, designated target	Prompt assembly; H scoring	Unique IDs, total order, target fixed before inference
State schema	Hypothesis, method, metric, judgment, plan	Parser; M3 projection	Accepted state comes from one structurally valid model response
Evidence map	Evidence IDs, visible text, release checkpoint, source join	Prompt assembly; EC scoring	Evidence is unavailable before its release checkpoint
Dependency contract	Target state, dependent states, validity interval, expected effect	DP and TC scoring	Every identifier resolves within the same trajectory
Reference transition	Admissible and forbidden outcomes with rationale	Semantic and H scoring	Correctness is constrained without prescribing exact wording

identifiers and their released text are visible. Future evidence, reference states, dependency annotations, expected transitions, and scoring labels are withheld. A dependency may point forward from an earlier state to a later decision, but its prerequisite state must already exist when that decision is made.

B Evaluation and Adjudication Protocol

B.1 Metric Definitions and Aggregation

For configuration d and trajectory i, let $c_{di} = 1$ when every required checkpoint response is present and parseable with the prescribed field set and order. For a complete trajectory, $q_{di,g} \in \{0, 1\}$ denotes the decision for semantic gate g in

$$\mathcal{G} = \{\mathrm{SV}, \mathrm{EC}, \mathrm{DP}, \mathrm{TC}, \mathrm{COH}\}.$$

Semantic success is a conjunction rather than a weighted score:

$$s_{di} = \begin{cases} \prod_{g \in \mathcal{G}} q_{di,g}, & c_{di} = 1, \\ 0, & c_{di} = 0. \end{cases} \qquad \mathrm{SC}_d = \frac{100}{137} \sum_i c_{di}, \qquad \mathrm{SS}_d = \frac{100}{137} \sum_i s_{di}.$$

Gates on an incomplete trajectory may remain unassessed. Assigning $s_{di} = 0$ in that case is an outcome rule, not a claim that five separate failures were adjudicated. The reliability gap is

$$\mathrm{RG}_d = \mathrm{SC}_d - \mathrm{SS}_d = \frac{100}{137} \sum_i (c_{di} - s_{di}),$$

so its denominator is all 137 trajectories, not only structurally complete ones.

Let $\mathcal{D}_k$ be subset k, with $N_k \in \{60, 63, 14\}$, and let τ_{di} indicate success at the item's target checkpoint. Then

$$H_{d,k} = \frac{100}{N_k} \sum_{i \in \mathcal{D}_k} \tau_{di}.$$

Transition Correctness includes that target operation; hence $s_{di} \leq \tau_{di}$. A missing or unparseable target response has $\tau_{di} = 0$, whereas a failure elsewhere in the trajectory can make $s_{di} = 0$ without changing a successful target label. The H rates therefore diagnose designated operations and do not partition SS.

Item-level decisions and count consistency. The display pipeline reads a frozen aggregate count export. At the stated denominators, SC, SS, and H1–H3 map to integer counts. If $C_d = \sum_i c_{di}$, $S_d = \sum_i s_{di}$, and $T_d = \sum_i \tau_{di}$, the non-compensatory metric contract implies

$$0 \leq S_d \leq \min(C_d, T_d) \leq 137.$$

The display pipeline checks this bound and reconstructs each base rate from its integer numerator and denominator. RG, S-Avg, Drop, and H-Avg are computed from those exact rates, with decimal half-up applied only to displayed values. The experimental audit joins each decision to its run and item records, while the display audit independently checks the resulting arithmetic.

B.2 Semantic Decisions

All five gates apply to every capability subset. A trajectory passes SS only if each gate passes; an especially strong hypothesis or plan cannot compensate for an evidence, dependency, transition, or terminal failure.

Evaluators may inspect the completed trajectory and its frozen item contract to resolve cross-checkpoint dependencies. The evidence decision at checkpoint t, however, is restricted to the prefix released by t. Later evidence cannot retrospectively invalidate a decision that was justified when made, and a later correction does not erase an earlier decisive violation from whole-trajectory SS.

Reference annotations define admissible outcomes rather than a canonical sentence. A different method or qualified judgment can pass when it respects the same evidence and transition constraints. Expressions of uncertainty are acceptable when warranted, but they cannot replace a required revision or recovery. Likewise, naming an operation in the adaptation field is insufficient unless the scientific state and its dependent decisions actually change.

B.3 Aggregation, Precision, and Interpretation

The auxiliary summaries are

$$S\text{-}Avg = (SC + SS)/2, \qquad H\text{-}Avg = (H1 + H2 + H3)/3, \qquad Drop = \Delta_{13} = H1 - H3.$$

H-Avg weights the three subset rates equally; pooled target success weights them by 60, 63, and 14. Drop is a signed cross-subset contrast. RG is interpreted jointly with SC and SS, since a low gap can accompany low completion. S-Avg is auxiliary; SS retains the five-gate conjunction.

Table 3 computes RG, S-Avg, Drop, H-Avg, and group and overall means from exact count-derived rates before decimal half-up display rounding. Table 4 reconstructs integer target successes $p_{d,k}$,

Table 8: Semantic rubric for a structurally complete trajectory. Decisions use only evidence admissible at the checkpoint under review.

Gate	Pass condition	Decisive failure	Scope
State Validity (SV)	Active hypotheses, methods, metrics, judgments, and plans are defensible, feasible, and mutually consistent.	An active component is unsupported, infeasible, or internally contradictory.	Each checkpoint
Evidence Consistency (EC)	Evidence is used with its correct direction, scope, and uncertainty; decisive evidence is incorporated.	Evidence is contradicted, overstated, ignored when decisive, or used before release.	Evidence prefix through t
Dependency Preservation (DP)	Still-valid dependencies remain effective; invalidated commitments cease to govern dependent decisions.	A required dependency disappears, or an invalidated commitment persists downstream.	Cross-checkpoint
Transition Correctness (TC)	Every required state operation, including the designated target, is carried out.	The event is acknowledged without the required maintenance, revision, or recovery.	Required transitions
Terminal Coherence (COH)	The final continuation agrees with the evidence, state components, and constraints that remain valid.	The terminal plan conflicts with retained evidence, state, or operational constraints.	Final trajectory

Table 9: Adjudication of common boundary cases. SC and SS are trajectory outcomes; the H effect applies only when the event occurs at the designated target.

Observed case	SC	Semantic and H decision	M3 state after the checkpoint
Missing response	0	SS = 0; target label = 0 when the target output is missing	Retain the latest structurally valid state
Unparseable response	0	SS = 0; target label = 0 when the target output is unparseable	Retain the latest structurally valid state; keep raw text in history
Parseable but scientifically wrong	1 if all responses parse	One or more gates fail; judge the target operation independently	Project the model's parsed, possibly incorrect, state
Later self-correction	No retroactive change	An earlier decisive gate failure remains; a later target can still pass	Continue from the latest structurally valid model state
Alternative valid solution	1 if all responses parse	May pass all gates and the target despite different wording	Project the valid alternative
Use of unreleased evidence	1 if all responses parse	EC fails; TC or the target may also fail according to the contract	No reference repair or hindsight correction

defines $f_{d,k} = N_k - p_{d,k}$, and reports

$$100 \frac{f_{d,k}}{\sum_{\ell=1}^{3} f_{d,\ell}}$$

beside each failed-target count. The same table adds total failed targets, total failure rate over 137 targets, and the dominant contributing subset. Its average row is the mean of unrounded within-row shares for the four highest-H-Avg records selected within each model group after aggregation. These arithmetic complements locate failed targets by subset; semantic error types and first failure checkpoints require separate trace-level labels.

One binary outcome changes SC or SS by $100/137 \approx 0.73$ percentage points. The corresponding steps for H1, H2, and H3 are 1.67, 1.59, and 7.14 percentage points; their effects on H-Avg are 0.56, 0.53, and 2.38 percentage points. Thus, 35.71% on H3 represents 5 of 14 targets. The Wilson interval reported in the main text summarizes denominator-limited binomial uncertainty for that proportion; it does not estimate variation across repeated executions. Configurations with the same H3 count are tied in the descriptive ranking.

C Worked Examples of Adjudication

The illustrative examples below are constructed to isolate the decision rule for each capability while holding the output schema fixed; they are not excerpts from evaluated model trajectories. Each specifies the available evidence, affected commitment, and required operation so that target and trajectory judgments can be read together.

H1: maintaining a scoped dependency. A treatment effect is supported only within a specified temperature range, and the planned experiment is constrained to that range. A later observation strengthens the claim under the same conditions. A valid update retains the scope in both the hypothesis and experimental plan. Retaining the qualified claim while proposing an unsupported temperature violates the dependency. At the designated maintenance checkpoint, DP and TC fail even if the local prose remains plausible.

H2: propagating a revision. A measurement method treats a signal as specific to the quantity of interest, after which new evidence establishes a material confound. A correct response revises the measurement claim and updates any metric interpretation, judgment, or plan that depended on specificity. Merely acknowledging the confound while continuing to use the unadjusted signal as direct evidence fails the target operation. An alternative measurement can pass without matching the reference wording if it addresses the same conflict and downstream consequences.

H3: recovering a feasible continuation. A released configuration update makes an instrument unavailable while leaving the research objective and earlier observations valid. Recovery requires identifying the affected steps, preserving usable evidence, and selecting a feasible alternative. Continuing to schedule the unavailable instrument fails TC; replacing the project with an unrelated direction also fails because valid constraints were discarded. If an earlier checkpoint contained a decisive semantic error, successful recovery can yield $\tau_{di} = 1$ on an H3 item while $s_{di} = 0$.

Reading local and trajectory judgments together. These cases pair local target labels with whole-trajectory outcomes. H scoring is restricted to the designated event, whereas SS preserves decisive violations across all checkpoints and gates.

D Prompting and Execution

D.1 Inference Configuration and Context Projection

The main results use the State-Aware setting (M3). Temperature is 0 and top-p is 1 where the provider exposes those controls; otherwise the provider default applies. External tools, retrieval, and web access are disabled. Reconstructing an execution requires a run record containing the endpoint or weight snapshot, execution date, requested and effective parameters, context and output limits, timeout behavior, and raw response. These fields distinguish the intended configuration from provider-side execution behavior.

Table 10: Information supplied by the four context settings. Every setting receives the initial research seed, current instruction, and evidence or configuration released through the current checkpoint.

Setting	Earlier model responses	Explicit state input
M0: Direct	None	None
M1: History	Immediately preceding response, or an explicit missing-event marker	None
M2: History-Aware	Complete preceding response history	None
M3: State-Aware	Complete preceding response history	Projection of the latest structurally valid model state; visible seed until one exists

Future checkpoints, unreleased evidence, reference states, dependency labels, and expected transitions are hidden under every setting. A missing checkpoint contributes an explicit event marker rather than fabricated response content; subsequent available responses remain in order. M0–M2 define controlled context variants, while M3 is the designated operating point for the benchmark results in Table 3.

D.2 Canonical Prompt Templates

The displays in this subsection separate invariant instructions from trajectory-specific and checkpoint-specific content. A complete execution record must preserve the serialized payload, message roles, token limits, and prompt version needed to reconstruct each request. For mutable API endpoints, these fields support request reconstruction rather than guaranteeing identical future outputs.

Table 11: Checkpoint prompt assembly. "Last" denotes the immediately preceding raw response and "Full" the complete prior response history. Evaluator-only metadata is excluded from every setting.

Prompt block	Source	Included settings	Update rule
Instruction prefix	Versioned prompt specification	M0–M3	Fixed across checkpoints and deployments
Initial research seed	Frozen task record	M0–M3	Same starting information in all settings
Checkpoint index and request	Checkpoint registry	M0–M3	Advances from t to $t+1$
Released evidence and configuration	Release schedule	M0–M3	Append only when the release checkpoint is reached
Permitted response history	Raw model responses	M1: Last; M2–M3: Full	Never repaired or replaced by reference text
Maintained scientific state	Latest valid model projection	M3 only	Updated only after a structurally valid response
Structured response contract	Versioned schema	M0–M3	Field set and order remain fixed
Reference state, dependencies, target label	Evaluator record	None	Forbidden in all generation payloads

Instruction prefix.

```
You are maintaining an explicit scientific research state
across an ordered sequence of checkpoints.

Use only the information included in the current request.
Treat RELEASED EVIDENCE and RELEASED CONFIGURATION as
available at this checkpoint.  Do not infer future evidence,
hidden dependency labels, expected transitions, or reference
answers.
```

```
Preserve state components that remain valid.  Change a
component only when available evidence justifies the change.
When evidence materially conflicts with the current state,
revise the affected component and its dependent decisions.
When the current continuation becomes infeasible, construct
a coherent alternative from information that remains valid.

Return one structured state record using the required field
names.  Cite only evidence identifiers supplied in the
request.
```

Checkpoint payload.

```
[INITIAL RESEARCH SEED]
[objective, constraints, and initial scientific state]

[CURRENT CHECKPOINT]
index:  [t] of [T]
instruction:  [checkpoint-specific request]

[RELEASED EVIDENCE AND CONFIGURATION THROUGH t]
[ordered identifiers and model-visible text]

[PERMITTED RESPONSE HISTORY]
[empty for M0; previous response for M1; full history for
M2/M3]

[MAINTAINED SCIENTIFIC STATE - M3 ONLY]
[projection from the latest structurally valid response]

[RESPONSE REQUIREMENT]
Update and return the complete structured state.
```

Structured response contract.

```
hypothesis:  [current claim and justified scope]
method:  [current procedure and affected design choices]
metric:  [current evaluation criterion]
judgment:  [assessment grounded in released evidence]
plan:  [next research action or terminal continuation]
adaptation:  [state operation and affected components]
evidence_ids:  [released identifiers used in this update]
```

Table 12: Normalized target-checkpoint projections. Capability names are editorial labels for this table and are not inserted into the model-visible payload.

Lens	Model-visible change	Normalized checkpoint request	Evaluator-only content excluded
H1	`E2` narrows the supported scope of an earlier finding without invalidating it	Integrate `E2`; return the complete state, retaining or qualifying prior components only as warranted	Designated dependency, expected maintenance decision, target label
H2	`E2` materially conflicts with an active assumption	Integrate `E2`; revise every state field whose content is no longer defensible	Affected-state map, admissible reference revisions, expected propagation
H3	A released configuration update makes the active continuation infeasible	Incorporate the update; return a coherent state and plan using information that remains valid	Invalidated-state map, admissible recovery paths, expected recovery operation

The visible change may expose a scientific tension, but the prompt does not name the capability subset, enumerate hidden dependencies, reveal the target label, or provide a reference answer. Released configuration remains visible at later checkpoints independently of M3 state projection.

D.3 Structured Output and M3 State Update

The first five response fields form the mutable scientific state. Adaptation and evidence identifiers support transition and provenance judgments but are not themselves state components.

Table 13: Structured response fields. Structural parsing checks the required serialization; semantic evaluation examines the scientific content after parsing.

Field	Required content	M3 role	Primary semantic use
hypothesis	Current claim, scope, and qualifications	Mutable state	SV, EC, DP
method	Current procedure, assumptions, and design choices	Mutable state	SV, DP, TC
metric	Evaluation criterion and its interpretation	Mutable state	SV, EC, TC
judgment	Assessment grounded in released evidence	Mutable state	SV, EC
plan	Next action or terminal continuation	Mutable state	DP, TC, COH
adaptation	Operation performed and components affected	Auxiliary transition record	TC and target scoring
evidence_ids	Released evidence used in the update	Auxiliary provenance link	EC and chronology

Let $y_{di,t}$ be the response at checkpoint t, $z_{di,t}$ the maintained state supplied to that checkpoint, and π the projection onto the five state fields. M3 evolves as

$$z_{di,1} = z_i^{\text{seed}}, \qquad z_{di,t+1} = \begin{cases} \pi(y_{di,t}), & y_{di,t} \text{ is structurally valid,} \\ z_{di,t}, & \text{otherwise.} \end{cases}$$

The initial seed is visible in every context setting; M3 additionally maintains the evolving projection. Structural validity controls this update, so a parseable but scientifically wrong state is carried forward. Missing or unparseable responses leave the state unchanged, including after consecutive failures. Reference annotations never repair the state.

Structural parsing applies to all seven response fields in the prescribed order; the projection π extracts only the first five. The parser specification must define the treatment of empty or duplicate fields, extra prose, code fences, Unicode, and truncation. Evidence-ID membership is evaluated separately: a structurally readable but unreleased identifier is an EC violation.

D.4 Execution Events

Neither semantic errors nor malformed output trigger regeneration under the stated protocol. Raw responses remain in the permitted history even when parsing fails. Missing responses, timeouts, and transport failures remain indexed rather than being removed from the denominator. A missing or unparseable required output sets $c_{di} = s_{di} = 0$ and sets $\tau_{di} = 0$ when it occurs at the target. If execution continues, M3 retains the previous structurally valid state.

The run ledger separates scientific content from interface behavior. For each expected checkpoint slot, it records deployment, trajectory, checkpoint, requested and effective parameters, prompt version, raw response, parse result, timeout or transport status, and attempt number. Stable identifiers join these events to item, decision, and deployment records so that structural and semantic outcomes can be recomputed.

E Reproducibility Records and Artifact Scope

E.1 Released Artifacts

The release separates the manuscript source archive from an anonymized experimental supplement. The source archive contains the LaTeX files, figure assets, frozen count exports, and scripts for regenerating the displayed rates, tables, and selected figures. The supplement contains benchmark items,

source manifests, frozen contracts, deployment and run ledgers, serialized requests, raw responses, parser outputs, and item-level decisions. Stable identifiers and cryptographic hashes join these artifacts, and the release pipeline reconciles the aggregate counts with the item-level decisions before display generation.

A versioned source manifest joins every trajectory to its seed, released evidence, and frozen item contract. It records source identifiers, domains, dates, transformations, reuse across trajectories, and redistribution permissions. Construction checks enforce unique trajectory and checkpoint identifiers, valid source joins, chronological release order, and agreement with Table 6; exclusions, repairs, and replacements remain explicit events rather than silent changes to the evaluation population.

E.2 Annotation and Adjudication Records

The evaluation protocol separates reference construction from model-output scoring. Its construction record assigns the evidence scope, dependency graph, admissible transitions, forbidden outcomes, and target rule to an initial annotator and independent source-based review to a second annotator before freezing the item contract. Its scoring record requires independent first-pass decisions for SV, EC, DP, TC, COH, and the capability target, followed by adjudication under a common rubric.

The experimental supplement preserves evaluator instructions, calibration materials, both first-pass decisions, explicit unassessed states, rationales, adjudicated labels, scorer versions, and review provenance. These records permit agreement to be measured before consensus and allow SS and H1–H3 to be recomputed directly from decisions.

E.3 Information-Isolation Evidence

The stipulated generation rule excludes future evidence, reference states, hidden dependencies, expected transitions, and evaluator outputs from model-visible requests. Evidence that this rule was enforced requires each serialized request to be joined to its release schedule and checked for prematurely released identifiers, text, or evaluator-only fields. A suitable automated isolation audit stores the result beside the request hash. Source-level deduplication and dated provenance metadata address benchmark overlap separately from generation-time isolation and from broader questions about model pretraining.

The experimental supplement stores serialized requests, release schedules, and per-request isolation records. Automated checks compare every request with its permitted information prefix and record the result beside the request hash.

E.4 Comparison Integrity

The experimental audit joins items, run events, and scoring decisions, while the manuscript audit checks count-derived rates, means, point-estimate emphasis, Table 4, figure inputs, prose macros, references, and layout. Group means weight reported rows equally. Weight availability and licensing describe a particular release, whereas serving channel describes access; an application programming interface (API) endpoint alone determines neither property.

Table 3 reports M3 as the sole operating point for all displayed model comparisons. M0–M2 define context variants but no outcomes for them are reported; every cross-configuration comparison therefore shares one context policy.

E.5 Reproducibility Artifacts

Table 14 specifies four linked record groups for reproducing construction, execution, scoring, and aggregation. Stable identifiers and cryptographic hashes connect the groups without mixing validation records into the main M3 denominators.

A complete pass over 18 configurations comprises $18 \times 137 = 2{,}466$ configuration–trajectory records and $18 \times 1{,}050 = 18{,}900$ expected checkpoint slots. Failed or unexecuted slots must remain explicit, while additional transport attempts receive separate event rows. Coverage checks should reconcile these products with the event and decision ledgers before result export.

Table 14: Records in the anonymized experimental supplement. Stable identifiers support joins across all four groups.

Record group	Essential contents	Reproduction function
Items and sources	Trajectory IDs; subset and horizon; checkpoint roles; released evidence and configuration; source transformations, permissions, and frozen item contracts	Rebuild the evaluation population and admissible information prefix
Deployments and events	Deployment ID, snapshot or endpoint, date, budgets, requested and effective parameters, serialized prompts, raw responses, attempts, and failures	Reconstruct requests and account for every expected checkpoint slot
Decisions	Deployment–trajectory keys; first-pass and adjudicated gate and target labels; rationales; scorer version; review provenance	Recompute SS, RG, H1–H3, agreement, and paired comparisons
Executable specifications	Schema, parser, prompt, and scorer versions; aggregation code; environment; manifest; cryptographic hashes	Reproduce parsing, metric computation, generated displays, and release-integrity checks

Within the source archive, `scripts/export_chapter4.py` generates result rows and prose macros from the frozen aggregate count export, and `scripts/render_chapter4.py` uses the same target counts for Figure 3. The source archive and anonymized experimental supplement form two linked release components: the former regenerates the manuscript displays, while the latter carries their item-level experimental records.

F Discussion and Scope

F.1 Discussion

IdeaDec treats scientific reasoning as a sequence of state updates rather than independent answers. A response can be locally plausible yet violate an earlier commitment, fail to propagate a revision, or continue from a state that new information has made infeasible. SC records whether the required structured sequence is complete, whereas SS additionally requires all five semantic conditions. Their difference therefore identifies structurally complete but semantically unsuccessful trajectories; it is interpreted with both component rates because a small gap can also accompany low completion.

H1–H3 provide a narrower view by isolating one maintenance, revision, or recovery target per trajectory. Because the subsets are disjoint and have different denominators, their rates describe capability profiles rather than successive difficulty levels. M3 supplies full response history and the latest structurally valid state projection, including an incorrect one.

F.2 Validation and Scope

Statistical resolution and matched controls. The primary evaluation contains 137 trajectories, including 14 H3 items, and these denominators determine the resolution of its point estimates. The item-level ledger preserves configuration–trajectory pairings and source-group identifiers. The paper reports point estimates and a Wilson interval for the sparse H3 maximum, but does not treat cross-subset marginals as matched causal estimates. A controlled capability comparison would require source-matched variants that hold horizon, domain, evidence length, target position, and output budget fixed across maintenance, revision, and recovery. The observed H1–H3 contrasts therefore describe the benchmark's fixed, heterogeneous subsets.

Semantic adjudication and human calibration. The five gates encode distinct requirements, and one error may violate more than one. Independent first-pass labels are retained before adjudication so that agreement can be distinguished from consensus. The benchmark metrics use adjudicated labels; evaluator agreement, gate-wise disagreement, and task-solving human performance remain separate calibration analyses rather than model-comparison metrics.

Execution and model identity. Each reported row has an immutable deployment identifier linked to the evaluated snapshot or endpoint, serialized requests, raw responses, and effective parameters. The deployment registry records release identity, license status, serving provenance, and repeated-name disambiguation; the aggregate export retains the corresponding row identifier and display name.

Scope of the experimental setting. External tools, retrieval, and web access are disabled under the reported protocol, making the supplied evidence schedule the basis for evaluation. The ten static controls are evaluated separately and excluded from SC, SS, RG, and H1–H3. Intermediate-state correction replays are outside this evaluation. Such a replay would need to fix the observed prefix, identify whether the intervention changes state, history, or both, and distinguish checkpoint-available repairs from reference-assisted ones; regardless of the intervention, the original trajectory would retain any earlier decisive violation in its SS label.