SCIONBENCH: EVALUATING EVIDENCE-PRESERVING RESEARCH TRAJECTORIES FROM MOTIVATION TO CONCLUSION IN SCIENTIFIC DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

A correct scientific conclusion can conceal an incomplete or unsupported research trajectory. Existing evaluations emphasize task outcomes, leaving the continuity of evidence across research stages insufficiently characterized. In order to address the problem, we introduce SCIONBench, a benchmark for evidence-preserving reconstruction of scientific research trajectories. SCIONBench represents 53 papers spanning 15 scientific domains and 131 research cases as source-grounded graphs linking motivation, method, experimental design, experiment, and conclusion. Four controlled disclosure modes progressively expose upstream information and assess reconstruction of held-out target nodes. Frozen evidence packages and explicit applicability decisions distinguish unavailable resources from execution failures, enabling stage-specific assessment under controlled information access. Node-level scoring assesses scientific correctness, experimental adequacy, evidence grounding, and conclusion scope. Across six model–agent configurations, our experiments reveal a clear stage-wise pattern: agents perform progressively better from method formulation and experimental planning to execution, analysis, and conclusion drawing. Correct intermediate outcomes from earlier stages further improve downstream performance, highlighting strong dependencies across the research pipeline. These results suggest that scientific capability is bottlenecked by the early stages of research. SCIONBench provides a diagnostic framework for evaluating agents' genuine stage-wise scientific capabilities and localizing failures across the scientific process.

1 INTRODUCTION

Scientific agents are progressing from *Chat* to *Code* and, eventually, to *Discovery*. Chat agents answer questions and summarize existing knowledge, whereas code-capable agents inspect files, write programs, call tools, run analyses, and respond to observed outcomes(Wei et al., 2022; Yao et al., 2023). For these agents, discovery is harder. The relevant variables and the next informative experiment may be unknown in advance. Therefore, agents must maintain a research state, revise their premises with evidence, and preserve why each next step follows from the previous one(Majumder et al., 2024; Chen et al., 2025; Sun et al., 2026).

This progression changes how scientific agents should be evaluated. However, a valid tool call may be scientifically irrelevant, executable code may omit an essential control, and an associational result may be reported as causal. Such failures reflect lost dependencies in the research trajectory, including the motivating problem, methodological assumptions, experimental design, supporting evidence, or scope of the conclusion. In scientific papers, these dependencies are encoded. It is a directed dependency structure, rather than a collection of facts. Recovering a final paragraph does not establish recovery of the argument that makes it valid.

Existing evaluations cover important capabilities at different stages of this progression. AgentBench and ReAct(Liu et al., 2024; Yao et al., 2023) evaluate interactive decisions and tool-mediated trajectories. SciBench, DiscoveryBench, and ScienceAgentBench(Wang et al., 2024b; Majumder et al., 2024; Chen et al., 2025) study scientific problem solving, data-driven discovery, and executable analyses. ScienceBoard(Sun et al., 2026) evaluates scientific software interaction. AirQA evaluates

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

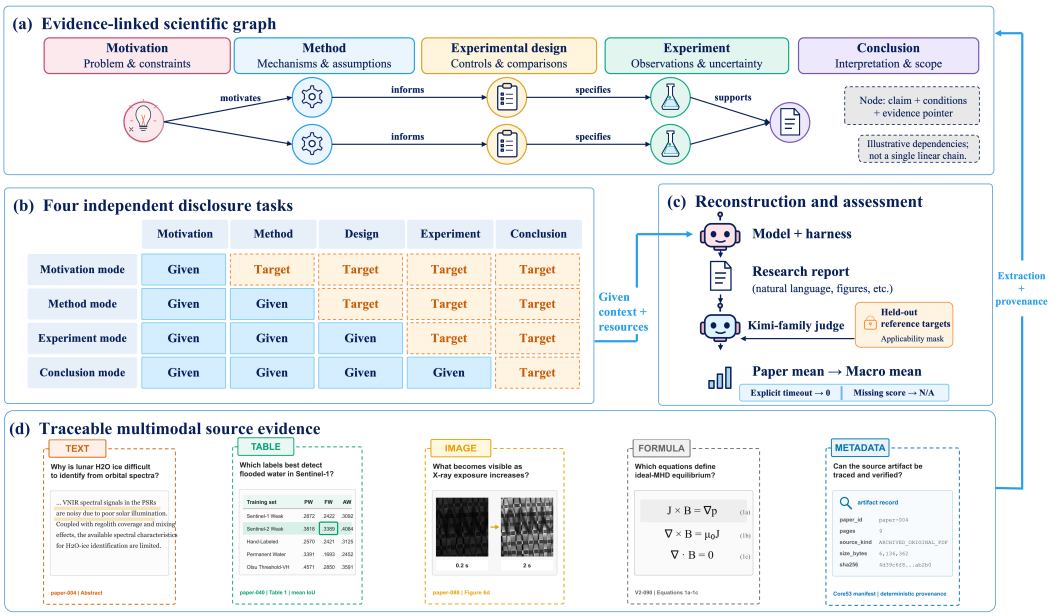

Figure 1: **The Overview of SCIONBench.** (a) A scientific graph links five scientific stages to source evidence. (b) Four independent modes disclose context (blue) and withhold targets (orange) using frozen resources; motivation is never scored. (c) A separate judge evaluates reports against hidden references and applicability masks. Node scores (0–10) are aggregated within and across papers. Inapplicable targets are excluded, explicit timeouts receive zero, and missing scores remain missing.

multimodal question answering over AI research papers. Replication-oriented benchmarks test code and analysis reconstruction from published work(Huang et al., 2026; Starace et al., 2025; Siegel et al., 2026; Ye et al., 2025). However, their primary signals remain local, including whether an answer is correct, a program executes, an environment reaches a target state, or a hypothesis is supported.

The missing layer is continuity of evidence across research stages. Endpoint scores collapse these roles, while execution traces record actions without revealing whether the agent preserved the paper's evidential logic. These limitations motivate a central question: **Can an agent preserve the evidence, conditions, and interpretation scope that connect these stages when it follows a paper from motivation to conclusion?**

To resolve the bottlenecks above, we introduce **SCIONBench** (Figure 1), a **Benchmark** of **S**cientific **C**ollaborative **I**nnovation with agentic **O**rganizational **N**exus comprising 53 high-quality papers from 15 scientific domains and 131 research cases. Each paper is represented as a five-stage research trajectory: **motivation** → **method** → **experiment design** → **experiment** → **conclusion**. We evaluate six model-agent configurations. Experimental results show that agents more reliably recognize a paper's stated conclusions than reconstruct, from its motivation, the experiments and evidence that justify them. This shared asymmetry identifies a bottleneck between Code and Discovery: agents can often recover a scientific endpoint, but struggle to carry the research state and evidential constraints needed to reach it.

To summarize, our contributions are:

- We formulate scientific-agent evaluation as evidence-preserving research-trajectory reconstruction, representing each paper with five directed stages and four controlled disclosure modes. We provide an auditable construction pipeline with paper artifacts, source-grounded claim graphs, evidence pointers, and frozen applicability decisions.

- We introduce validity-aware, paper-level evaluation that separates scientific quality from non-evaluable evidence, execution failures, and automatic judge failures while retaining coverage and denominator information.
- We show a shared capability profile across model-agent configurations. Recognizing endpoint conclusions is easier than reconstructing the experiments and evidence that support them.

2 THE SCIONBENCH DATASET

In this section, we introduce the task definition, the evaluation methods, the construction and the statistics of our SCIONBench dataset.

2.1 TASK DEFINITION

SCIONBench evaluates whether a scientific agent can reconstruct a paper's research trajectory from partial, controlled evidence. We represent each paper p as a typed directed graph

$$G_p = (V_p, E_p, \phi_p), \qquad \phi_p : V_p \to \mathcal{S},$$

where

$$\mathcal{S} = \{\text{motivation}, \text{method}, \text{experimental design}, \text{experiment}, \text{conclusion}\}.$$

Each node $v \in V_p$ is an atomic claim or evidence-bearing unit, and an edge $(u, v) \in E_p$ indicates that u supports the interpretation, construction, execution, or justification of v. Nodes may have multiple upstream dependencies.

For paper p and disclosure mode m, we define the paper–mode instance as

$$\mathcal{T}_p^{(m)} = (X_p^{(m)}, Y_p^{(m)}),$$

where $X_p^{(m)}$ is the released context and $Y_p^{(m)}$ contains the withheld target nodes. The four modes are **Motivation**, **Method**, **Experiment**, **Conclusion**. Each mode starts an independent run with the same frozen paper-local resources. Agents cannot access hidden nodes, evaluator-only references or applicability masks, or outputs from other runs.

Agents produce a single Markdown research report. The report must distinguish inherited facts, design choices, assumptions, observations, and speculation, and must state relevant conditions, comparisons, uncertainty, and evidence pointers. Unsupported content must be marked as `unavailable`. Fabricated results and conclusions are prohibited.

2.2 EVALUATION METHODS

The evaluation matrix measures reconstruction quality across four controlled disclosure modes (rows) and four held-out stages (columns): method, experimental design, experiment, and conclusion. For paper p, cell (i, j) is applicable only if stage j lies downstream of the context disclosed in mode i and contains evaluable target nodes.

Applicability is fixed before judging. Experiment nodes are evaluable only when their core evidence can be recovered from frozen paper-local resources within the computational budget. Experiments requiring unavailable physical execution, such as wet-lab work, and their dependent conclusions are excluded from scoring.

Nodes are scored on four shared dimensions using stage-specific content requirements and admissible evidence. Each evaluable node v receives a semantic score $s_{p,i,j,v} \in [0, 10]$. We use Kimi K3 as the automatic judge model for assigning these semantic scores.The automatic judge model receives the reference answer, node-specific criteria, relevant evidence, and final report, but no intermediate tool traces or interruption reasons. Evidence-grounding scoring penalizes unsupported claims, fabricated execution, missing controls, contradictions, and conclusions exceeding the evidence (Dasigi et al., 2021; Wadden et al., 2020; DeYoung et al., 2020; Liu et al., 2023b; Zheng et al., 2023).

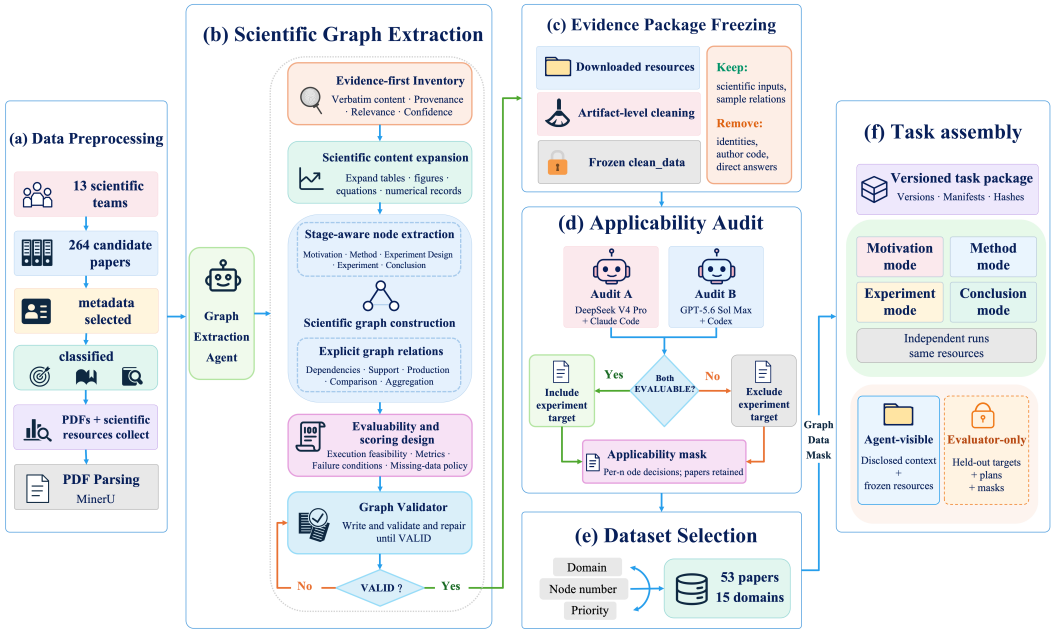

Figure 2: **The Construction pipeline of SCIONBench.** After preprocessing, candidate papers are extracted into source-grounded scientific graphs and frozen evidence packages. Two independent audits assess the applicability of graph nodes under matched inputs and budgets. The final release contains 53 papers across 15 domains and 212 paper-mode instances per configuration. Agent-visible inputs are separated from evaluator-only references.

Let $V_{p,i,j}^{\text{score}}$ denote the evaluable nodes in an applicable cell. The paper-level score is

$$S_p^{(i,j)} = \begin{cases} \dfrac{1}{|V_{p,i,j}^{\text{score}}|} \displaystyle\sum_{v \in V_{p,i,j}^{\text{score}}} s_{p,i,j,v}, & \text{if the task is completed,} \\ 0, & \text{if the task reaches the 18-hour limit.} \end{cases} \tag{1}$$

Agent-caused failures on otherwise evaluable tasks also receive zero, distinguishing execution failure from unavailable evidence. Matrix entries average scores across applicable papers:

$$A_{i,j} = \text{MacroAvg}_{p:\,|V_{p,i,j}^{\text{score}}|>0} \left[S_p^{(i,j)} \right]. \tag{2}$$

For domain-macro reporting, let $\mathcal{P}_{d,i,j}$ contain applicable papers in field d, and let D denote the number of fields:

$$S_d^{(i,j)} = \frac{1}{|\mathcal{P}_{d,i,j}|} \sum_{p \in \mathcal{P}_{d,i,j}} S_p^{(i,j)}, \qquad S_{\text{overall}}^{(i,j)} = \frac{1}{D} \sum_{d=1}^{D} S_d^{(i,j)}. \tag{3}$$

We also report scored-node and completed-task counts to distinguish reconstruction quality, coverage, and execution reliability.

The matrix supports three descriptive diagnostics: target recoverability as upstream context increases, gains between adjacent disclosure modes, and transfer loss across stages. Adjacent-mode comparisons use matched papers and target nodes. Causal claims about model superiority additionally require controlled scaffolds, prompts, execution conditions, and repeated runs.

The method evaluates paper-based reconstruction, excluding open-ended hypothesis generation and physical experimentation. Automated judgments require expert calibration to distinguish fluency from evidential grounding. Future work should test whether the matrix remains predictive when evidence arrives incrementally in interactive research settings.

2.3 DATASET CONSTRUCTION

SCIONBench is designed to support reproducible, evidence-grounded reconstruction of scientific research trajectories. Figure 2 shows the construction pipeline.

Paper selection. In order to obtain broad scientific coverage while retaining representative and high-impact studies, we collect 264 candidate papers recommended by 13 scientific teams. Within each domain, we rank papers by impact and recency, grouped them by topic and recommendation, and retained a mixture of foundational studies, recent high-impact work, specialist publications, and frontier preprints. Final selection is performed after applicability auditing, with preference for papers containing sufficient evaluable evidence.

Literature normalization. To make heterogeneous papers machine-readable without losing evidence provenance, we collect each paper's available PDF, data, code, and supplementary materials. We use MinerU to convert PDFs into Markdown with images and source locations (Wang et al., 2024a), then standardize headings, encoding, equations, tables, and image references while preserving page, figure, table, and paragraph locations. Scientific values, units, directions, and sample relationships are retained. OCR uncertainty, unparseable formulas, and missing visual evidence are recorded for later auditing.

Scientific graph extraction. In order to represent stage dependencies without requiring agents to access the original source files, we extract each paper into JSON document using GPT-5.6-Sol-Max with Codex in an isolated workspace with a standardized prompt (Appendix B.1). The extractor process the normalized text, images and incorporated relevant tables, figures, and formulas.

Each node stores an identifier, parent nodes, scientific stage, data type, and evaluability status. Edges record dependencies, support, inputs, outputs, comparisons, aggregation, and condition transfer, together with their evidence, reasoning, and confidence. The graph supports multiple parents and independent components. Method relations distinguish parallel, technically dependent, and strictly serial procedures. A `strict_serial` relation requires an earlier implementation or output. Many-to-many coverage links connect methods to experimental designs, results, and conclusions, identifying the factors changed, ablated, measured, or invoked. Missing downstream links are retained only after review. This design follows evidence-centered representations used in prior scientific evaluation and argument-mining work (Dasigi et al., 2021; Wadden et al., 2020; DeYoung et al., 2020; Lawrence & Reed, 2020).

Graph validation. To prevent incomplete or inconsistent graphs from entering evaluation, an automated validator checks source coverage, schema completeness, unique identifiers, evidence references, stage assignments, edge validity, method relations, coverage links, and support for all four disclosure modes. Scientific fields must retain relevant values, units, conditions, and uncertainty. Unevaluable nodes must include an explicit status and reason. Invalid graphs are revised and revalidated, and nodes or evidence cannot be removed to bypass a failure. Only graphs marked `VALID` proceed to evidence freezing, while unresolved information is recorded explicitly. Moreover, the scientific teams behind the recommended papers to manually review the extracted graphs and verify that no discrepancies existed between the resulting graphs and the corresponding papers.

Evidence freezing. Following replication-oriented benchmarks(Starace et al., 2025; Siegel et al., 2026), we generate a paper-local, read-only package for each validated graph. Mode-specific views expose only the permitted node content, provenance pointers, and resources. Identifying metadata, held-out answers, answer-revealing artifacts, and file-path dependencies are removed, while permitted scientific content and sample relationships are preserved. Each package records its input manifest, graph version, image hashes, processing-script version, and generation time, enabling traceable and reproducible evaluation.

Applicability auditing and release. To distinguish unavailable evidence from scientific failure, we audit each experiment node for input availability, physical dependence, and computational feasibility. A node is excluded only when its core evidence requires unavailable samples, equipment, physical intervention, or computation beyond the fixed budget. Missing author code, historical software, or uncertain runtime alone does not justify exclusion.

Table 1: Statistics of SCIONBench.

Statistics	Number
Dataset scope	
Papers	53
Scientific domains	15
Research subtasks	131
Scientific nodes	
Motivation	53 (3.8%)
Method	304 (22.0%)
Experimental design	301 (21.8%)
Experiment	315 (22.8%)
Conclusion	409 (29.6%)
Total nodes	**1,382 (100%)**
Avg. subtasks / paper	2.47
Avg. nodes / paper	26.08

Percentages use all 1,382 archived nodes, including non-evaluable nodes. Research subtasks and graph nodes are distinct units.



Figure 3: Distribution of 131 research cases across 15 scientific domains. The inner ring shows domains. Each outer sector represents one case.

Two independent agents apply the same rules, inputs, and budgets. A node is retained only when both mark it evaluable; otherwise, its blocking reason is recorded. Exclusions propagate along explicit dependency edges to dependent conclusions in motivation, method, and experiment modes, and the resulting applicability mask is frozen with the graph and audit records. Final selection retains at least one paper per domain and favors papers with more than half of their nodes evaluable. The release preserves source records and processing logs while separating agent-visible inputs from evaluator-only references, scoring plans, and applicability records.

2.4 Dataset Statistics

Dataset statistics. SCIONBench covers 53 papers(journals containing Nature Energy, Nature, Cell, Advanced Functional Materials Nature Communications, Nucleic Acids Research, Earth System Science Data) across 15 domains, with 131 subtasks and 1,382 archived nodes, including those excluded from scoring (Table 1, Figure 3). Each configuration comprises 212 paper–mode evaluation instances.

Comparison with existing benchmarks. SCIONBench combines five-stage dependency graphs, four independent upstream disclosure modes, and resource-based applicability masks (Appendix A.2.). Masks are frozen before scoring and propagate exclusions to dependent targets, separating resource limitations from agent failures.

3 Experiment

3.1 Experiment Settings

Agent configurations. We evaluate six model–agent configurations. Claude Code uses DeepSeek-V4-Pro, Kimi K3, and Claude Sonnet 5, while Codex uses DeepSeek-V4-Pro, GLM-5.2, and GPT-5.6 Sol. All models are accessed through APIs, so the evaluation measures end-to-end agent behavior under each model–agent interface.

Execution protocol. Each paper task runs in a fresh, isolated workspace without prior context. We allocate one NVIDIA RTX 4090 GPU, 10 CPU cores, 100 GiB of RAM, and an 18-hour wall-clock limit per task. The local environment supports code execution, tool use, and artifact generation. Model responses are obtained through the corresponding APIs.

Reproducibility. We retain prompts, input manifests, environment and dependency versions, command logs, generated reports, node statuses, failure reasons, and token usage. These records support auditing, failure analysis, and reproduction.

3.2 Experimental Results and Analysis

3.2.1 Main Results

Table 2: Performance of different baselines on SCIONBench.

Agent	Model	Motivation	Method	Experiment	Conclusion	Overall
Claude Code	DeepSeek-V4-Pro	**3.97**	**5.51**	6.59	9.01	**6.27**
Claude Code	Kimi K3	3.76	5.01	6.46	8.88	6.03
Claude Code	Claude Sonnet 5	3.26	4.71	6.26	8.78	5.75
Codex	DeepSeek-V4-Pro	3.66	5.44	**6.86**	**9.03**	6.25
Codex	GLM-5.2	2.46	3.90	5.51	8.48	5.09
Codex	GPT-5.6 Sol	2.90	3.96	5.58	8.87	5.33

Agents recover Conclusion more reliably than the evidence that supports them. As shown in Figure 4 and Table 2, all six configurations show the same stage-wise pattern. Motivation mode scores range from 2.46 to 3.97, whereas scores of conclusion mode range from 8.48 to 9.03, and the resulting Motivation-Conclusion gaps span 5.04-6.02 points. Under 4 modes and five stages, the scores of each configuration show an inverted triangle trend. Thus, when the reference method, design, and experimental observations are available, agents can often synthesize the paper's conclusions. From motivation alone, however, they struggle to reconstruct the methods, experiments, and evidence needed to justify those conclusions.

This pattern is the central capability gap measured by SCIONBench. High scores of conclusionmode indicate evidence-conditioned synthesis, rather than autonomous recovery of the investigation that produced the evidence. Evaluating only final conclusions would therefore overstate an agent's ability to follow a scientific argument from problem formulation to supported claim.

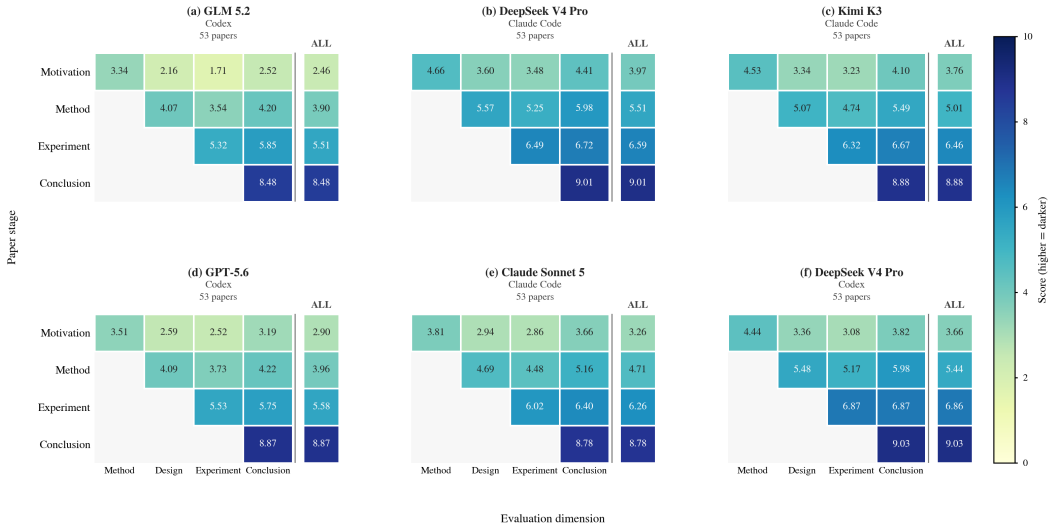

Figure 4: Heatmaps of average target-node scores across four controlled disclosure modes for six model–agent configurations. Darker colors indicate higher scores.

Aggregate scores conceal stage-specific differences. DeepSeek-V4-Pro with Claude Code achieves the highest Motivation and Method scores, while DeepSeek-V4-Pro with Codex achieves

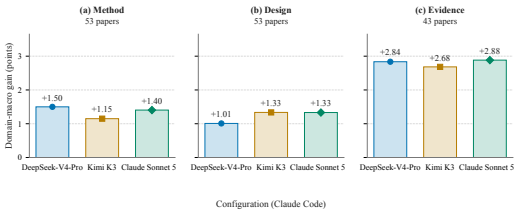

Figure 5: Domain-macro gains from disclosing upstream method, design, and evidence under three Claude Code configurations. Method and Design use 53 papers, while Evidence uses 43 papers with conclusion scores in both modes.

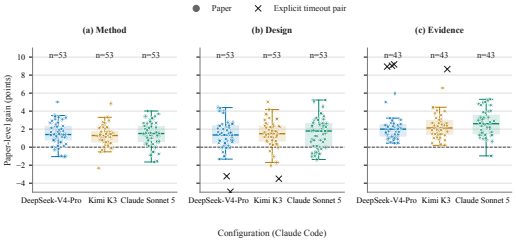

Figure 6: Paper-level gains for the three disclosure transitions. Points represent papers, boxes show medians and interquartile ranges, the dashed line indicates zero gain, and crosses mark timeout pairs.

the highest Experiment and Conclusion scores. The two configurations have similar Overall scores (6.27 and 6.25), but their stage-wise profiles differ. The largest gap occurs in Motivation mode, where Claude Code outperforms Codex by 0.31 points (3.97 vs. 3.66) with the same DeepSeek-V4-Pro backbone. It demonstrates that similar overall scores mask differences between agents in scientific reconstruction.

Moreover, domain-level results are reported in Appendix A.6. Functional Fibers and Device Materials achieves the highest mean Overall score across configurations (6.26). Its Experiment and Conclusion scores average 7.03 and 9.23, respectively, compared with 3.25 in Motivation mode, showing that strong aggregate performance can coexist with difficulty reconstructing research from motivation alone.

3.2.2 Analysis

We next test whether disclosing a reference stage improves recovery of the same downstream scientific content. Unlike subtracting the mode scores in Table 2, these comparisons compute within-paper score differences for each transition. We conduct three matched comparisons across disclosure modes: Motivation versus Method for experimental design, experiment, and conclusion; Method versus Experiment for experiment and conclusion; and Experiment versus Conclusion for conclusion. Terminal timeouts contribute effective zeros, consistent with the main evaluation.

Reference methods provide a stable anchor for subsequent reconstruction. As shown in Figure 5, disclosing the reference method increases domain-macro downstream scores by 1.50 points for DeepSeek-V4-Pro, 1.15 points for Kimi K3, and 1.40 points for Claude Sonnet 5. The paper-level distributions in Figure 6 show positive gains for 48, 48 and 46 of the 53 papers, respectively. Methodological context therefore consistently supports downstream reconstruction. This finding concerns the use of a supplied reference method. It does not measure whether an agent can autonomously discover that method.

A experimental design helps on average but does not complete the evidence chain. Disclosing the reference experimental design yields domain-macro gains of 1.01, 1.33, and 1.33 points for DeepSeek-V4-Pro, Kimi K3, and Claude Sonnet 5, respectively. Figure 6 shows that positive gains occur for 43, 47, and 41 of the 53 papers. A design specifies how evidence should be obtained, but it does not provide the observations or analysis required to support a downstream claim. Thus, design disclosure improves recovery on average without guaranteeing improvement in every independent execution.

Experimental observations primarily enable evidence-conditioned synthesis. As shown in Figure 5, disclosing experimental observations improves domain-macro conclusion recovery by 2.84, 2.68, and 2.88 points across the three Claude Code configurations. The paper-level distributions in Figure 6 show positive gains for 43, 43, and 42 of the 43 eligible papers. This result explains the high scores of conclusion mode in Table 2. Agents recover conclusions more reliably

Table 3: Illustrative failures in research trajectory reconstruction. The cases are selected to show distinct breaks in the evidence chain and do not estimate their benchmark-wide prevalence.

Failure type	**Reference requirement**	**Agent behavior**	**Why the reconstruction fails**
Scientific target mismatch	DPCA ranks 83 spectral bands for lunar water-ice identification.	DeepSeek-V4-Pro with Claude Code clusters pixel spectra and selects ten representative spectra.	The algorithm is relevant, but representative-pixel selection does not recover the required band ranking.
Incomplete reasoning	A magnetic-confinement result derives contour spacing from bounce-action invariance and magnetic-field geometry.	Kimi K3 with Claude Code gives numerical checks and partial analysis, but omits the general action-integral transformation and inversion.	Agreement in selected numerical constructions does not establish the required general result.
Conclusion beyond evidence	Aggregation and preparation differences prevent attributing near-infrared spectral differences to particle shape alone.	Claude Sonnet 5 with Claude Code acknowledges the difficulty of producing coarse spherical grains, but attributes "a secondary but real effect" to morphology.	The stated limitation does not constrain the final attribution.

when their supporting observations are provided than when they must reconstruct the experiments that generate those observations.

Together, Figures 5 and 6 show that method and experimental evidence provide the most consistent support for downstream reconstruction. Experimental design alone does not ensure that an agent preserves the complete chain from planned procedure to supported conclusion.

3.2.3 Blinded Expert Calibration of the Automatic Judge

Moreover, a blinded expert audit of 20 papers shows strong agreement between Kimi and experts (ICC = 0.87, MAE = 0.37, and 92% within one point on the 0–5 scale), supporting its use as the SCIONBench judge (Appendix A.5).

3.2.4 Failure Analysis

Scores reveal where reconstruction degrades, but not how the scientific argument breaks. We therefore compare reference requirements with agent reports and execution artifacts in three illustrative cases. These examples identify error types rather than their benchmark-wide prevalence (Table 3).

The first case changes the scientific object while retaining a superficially relevant algorithm. The second replaces a required derivation with partial numerical agreement. The third recognizes a limitation but fails to preserve its implication for conclusion scope. These cases show why successful tool use, plausible calculations, and explicit caveats are not sufficient evidence that an agent has preserved a paper's scientific logic.

4 Conclusion

SCIONBench evaluates scientific agents through a five-stage reconstruction framework with controlled information disclosure. Across six model–agent configurations, we find that agents are weakest at recovering early research stages from motivation alone, while performance improves toward experiment execution, analysis, and conclusion drawing. Correct intermediate outcomes from earlier stages further enhance downstream performance, revealing strong dependencies across the scientific pipeline. Together, these results suggest that advancing end-to-end scientific capability requires strengthening upstream abilities such as method formulation and experimental design, while evaluating information access, target coverage, and execution completion as distinct dimensions of scientific competence.

AI USE STATEMENT

Generative AI tools assist with literature search and analysis, brainstorming, and interpretation of experimental results. For SCIONBench construction, AI tools assist with scientific graph extraction, resource cleaning, and experimental evaluability auditing. Scientific teams review the extracted graphs against the source papers. Kimi scores agent reports using predefined criteria, and domain experts assess its agreement with human ratings through a blinded expert calibration. The authors review AI-assisted content and interpretations and take full responsibility for the dataset, evaluation protocol, reported results, and final manuscript.

ETHICS STATEMENT

SCIONBench uses published scientific papers and derived, source-grounded artifacts across 15 scientific domains. It does not collect new human-participant data or infer personal identity, protected attributes, health status, or psychological traits. Paper licenses and redistribution restrictions are respected, and source pointers, hashes, or derived artifacts are used when original materials cannot be redistributed. Potential sources of bias include domain imbalance, literature selection, expert disagreement, and model-based judging. Applicability masks, blinded expert calibration, and explicit validity states are used to make these limitations auditable. Agent-generated reports are evaluation outputs and do not constitute scientific evidence or research conclusions.

REPRODUCIBILITY STATEMENT

The paper documents the five-stage graph representation, four controlled disclosure modes, dataset construction, evaluation rubric, and aggregation method in Sections 2 and 2.3. Appendix A.3 specifies the scoring dimensions and validity states, while Appendices B.1, B.3, and B.4 provide the extraction, evaluation, and auditing prompts. Appendix A.5 reports the blinded expert audit, and Appendix A.6 provides domain-level results.

For each run, we retain the task prompt, paper-local resource manifest, source and artifact hashes, model and agent identifiers, code and dependency versions, command logs, generated reports, node-level statuses, failure reasons, timeout records, and token usage. These records support auditing of the construction and evaluation pipeline and reproduction of the reported results.

REFERENCES

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, 2025. URL https://arxiv.org/abs/2410.05080. ICLR 2025.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. Structural scaffolds for citation intent classification in scientific publications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3586–3596. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1361. URL https://aclanthology.org/N19-1361/.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4599–4610. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.365. URL https://aclanthology.org/2021.naacl-main.365/.

Jay DeYoung, Eric Lehman, Benjamin Nye, Iain Marshall, and Byron C. Wallace. Evidence inference 2.0: More data, better models. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pp. 123–132. Association for Computational Linguistics, 2020. doi:

10.18653/v1/2020.bionlp-1.13. URL https://aclanthology.org/2020.bionlp-1.13/.

Aniketh Garikaparthi, Manasi Patwardhan, and Arman Cohan. ResearchGym: Evaluating language model agents on real-world AI research, 2026. URL https://arxiv.org/abs/2602.15112.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation, 2024. URL https://arxiv.org/abs/2310.03302.

Tiancheng Huang, Ruisheng Cao, Yuxin Zhang, Zhangyi Kang, Zijian Wang, Chenrun Wang, Yijie Luo, Hang Zheng, Lirong Qian, Lu Chen, and Kai Yu. Airqa: A comprehensive qa dataset for ai research with instance-level evaluation, 2026. URL https://arxiv.org/abs/2509.16952. Accepted to ICLR 2026.

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?, 2023. URL https://arxiv.org/abs/2310.06770. ICLR 2024.

John Lawrence and Chris Reed. Argument mining: A survey. *Computational Linguistics*, 45(4): 765–818, 2020. doi: 10.1162/coli_a_00364. URL https://doi.org/10.1162/coli_a_00364.

A. J. Lew, Y. Cao, and M. J. Buehler. ProjectionBench: Evaluating scientific hypothesis generation in LLMs under progressive information disclosure, 2026. URL https://arxiv.org/abs/2605.30284.

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, 2023a. URL https://arxiv.org/abs/2307.03172. Transactions of the Association for Computational Linguistics.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents, 2024. URL https://arxiv.org/abs/2308.03688. Published in ICLR 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment, 2023b. URL https://arxiv.org/abs/2303.16634.

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, 2024. URL https://arxiv.org/abs/2407.01725. ICLR 2025.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: A benchmark for general ai assistants, 2023. URL https://arxiv.org/abs/2311.12983.

Siba Smarak Panigrahi, Jovana Videnović, and Maria Brbić. HeurekaBench: A benchmarking framework for AI co-scientist. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Y7xCdFuFw7.

Zachary S. Siegel, Sayash Kapoor, Nitya Nadgir, Benedikt Stroebl, and Arvind Narayanan. Corebench: Fostering the credibility of published research through a computational reproducibility agent benchmark, 2026. URL https://arxiv.org/abs/2409.11363.

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, 2025. URL https://arxiv.org/abs/2504.01848.

Qiushi Sun, Zhoumianze Liu, Chang Ma, Zichen Ding, Fangzhi Xu, Zhangyue Yin, Haiteng Zhao, Zhenyu Wu, Kanzhi Cheng, Zhaoyang Liu, Jianing Wang, Qintong Li, Xiangru Tang, Tianbao Xie, Xiachong Feng, Xiang Li, Ben Kao, Wenhai Wang, Biqing Qi, Lingpeng Kong, and Zhiyong Wu. Scienceboard: Evaluating multimodal autonomous agents in realistic scientific workflows, 2026. URL https://arxiv.org/abs/2505.19897. ICLR 2026 camera-ready version.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7534–7550. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.609. URL https://aclanthology.org/2020.emnlp-main.609/.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. Mineru: An open-source solution for precise document content extraction, 2024a. URL https://arxiv.org/abs/2409.18839.

Hexuan Wang, Yaxuan Ren, Srikar Bommireddypalli, Shuxian Chen, Adarsh Prabhudesai, Rongkun Zhou, Elina Baral, and Philipp Koehn. SciTaRC: A plan-annotated scientific tabular QA benchmark for language reasoning and complex computation. In *Conference on Language Modeling*, 2026a. URL https://arxiv.org/abs/2603.08910.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models, 2024b. URL https://arxiv.org/abs/2307.10635. To appear at ICML 2024.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. Openhands: An open platform for ai software developers as generalist agents, 2024c. URL https://arxiv.org/abs/2407.16741. Accepted by ICLR 2025.

Yuru Wang, Lejun Cheng, Yuxin Zuo, Sihang Zeng, Bingxiang He, Che Jiang, Junlin Yang, Yuchong Wang, Kaikai Zhao, Weifeng Huang, Kai Tian, Zhenzhao Yuan, Jincheng Zhong, Weizhi Wang, Ning Ding, Bowen Zhou, and Kaiyan Zhang. Naturebench: Can coding agents match the published sota of nature-family papers?, 2026b. URL https://arxiv.org/abs/2606.24530.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022. URL https://arxiv.org/abs/2201.11903.

Shenxi Wu, Yuhong Liu, Haosong Zhang, Tongjin Zou, Yanxun Zhang, Gaochang Chen, Dun Liang, Jiaqi Wang, Zhecan James Wang, Yuhang Zang, and Dahua Lin. SciDocBench: A workflow-centered benchmark and data pipeline for scientific document understanding, 2026. URL https://arxiv.org/abs/2609.05141.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024. URL https://arxiv.org/abs/2404.07972.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629. ICLR camera-ready version.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Christine Ye, Sihan Yuan, Suchetha Cooray, Steven Dillmann, Ian L. V. Roque, Dalya Baron, Philipp Frank, Sergio Martin-Alvarez, Nolan Koblischke, Frank J Qu, Diyi Yang, Risa Wechsler, and Ioana Ciuca. Replicationbench: Can ai agents replicate astrophysics research papers?, 2025. URL https://arxiv.org/abs/2510.24591.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685. NeurIPS 2023 Datasets and Benchmarks Track.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2024. URL https://arxiv.org/abs/2307.13854.

A Appendix

A.1 Related Work

Interactive agents and scientific workflows. AgentBench and ReAct established interactive decision making and interleaved reasoning–action trajectories as evaluation targets for language-model agents (Liu et al., 2024; Yao et al., 2023). WebArena, GAIA, and τ-bench extend this setting to stateful web tasks, multimodal information gathering, and tool–agent–user interaction (Zhou et al., 2024; Mialon et al., 2023; Yao et al., 2024). MLAgentBench, SWE-bench, SWE-agent, OpenHands, and OSWorld further evaluate iterative experimentation, executable software, and computer-use workflows (Huang et al., 2024; Jimenez et al., 2023; Yang et al., 2024; Wang et al., 2024c; Xie et al., 2024). In scientific settings, SciBench, DiscoveryBench, ScienceAgentBench, and ScienceBoard study scientific problem solving, data-driven discovery, executable analysis, and scientific software interaction (Wang et al., 2024b; Majumder et al., 2024; Chen et al., 2025; Sun et al., 2026). These benchmarks evaluate important local outcomes, whereas **SCIONBench** evaluates whether the scientific argument connecting different stages is preserved.

Scientific paper question answering and evidence. AirQA studies multimodal, multitask question answering over research papers and includes text, tables, images, formulas, and metadata (Huang et al., 2026). QASPER grounds information-seeking answers in paper evidence spans, while SciFact and Evidence Inference address scientific claim verification and intervention–comparator–outcome relations (Dasigi et al., 2021; Wadden et al., 2020; DeYoung et al., 2020). SciCite models rhetorical citation intent (Cohan et al., 2019). These resources motivate explicit evidence pointers and discourse roles. However, they primarily evaluate individual questions or claims. **SCIONBench** instead evaluates a paper-level trajectory in which evidence links must preserve dependencies among motivation, method, experimental design, experiment, and conclusion.

Replication, argument structure, and evaluation validity. PaperBench, CORE-Bench, ReplicationBench, and NatureBench evaluate research replication or executable reconstruction under increasingly realistic resource constraints (Starace et al., 2025; Siegel et al., 2026; Ye et al., 2025; Wang et al., 2026b). Argument mining provides representations for claims and inference relations (Lawrence & Reed, 2020). Long-context studies show that relevant evidence may be underused depending on its position in the input, while automatic-judge studies highlight calibration and bias challenges (Liu et al., 2023a;b; Zheng et al., 2023). **SCIONBench** combines these insights into a single evaluation contract: a typed research trajectory, a fixed evidence boundary, explicit applicability states, and paper-level aggregation. Its primary estimand is evidence-preserving scientific reconstruction rather than isolated execution, retrieval, or verification.

A.2 Comparison with scientific benchmarks

We compare SCIONBench with representative scientific benchmarks in terms of scale, evaluation, evidence usage, input variants, and resource-dependent target eligibility (Table 4).

Table 4: F/T: figure/table evidence in tasks or references. Mask: resource-dependent target eligibility frozen before scoring, with dependency-propagated exclusions. ✗: not established in the checked source, not evidence of absence.

Benchmark	Scale	Evaluation	F/T	Reference	Input variants	Mask
SciDocBench (Wu et al., 2026)	124 QA / 496 instances	Rules / LM / execution	✓	Typed evidence graph	4 language/layout settings	✗
SciTaRC (Wang et al., 2026a)	371 QA	LM-judged correctness	✓	Reasoning plans	Oracle plans	✗
HeurekaBench (Panigrahi et al., 2026)	50 open + 50 MC	Fact rubric / accuracy	✗	Atomic insight facts	✗	✗
ResearchGym (Garikaparthi et al., 2026)	5 environments / 39 subtasks	Experiment performance	✗	Baselines + subtask targets	Fixed method withholding	✗
ProjectionBench (Lew et al., 2026)	45 papers	Claim F1 / AUC	✗	Atomic hypothesis claims	Progressive disclosure	✗
SCIONBench (Ours)	**53** papers / **212** instances	Report–node alignment	✓	Five-stage dependency graph	4 independent upstream modes	✓

A.3 Evaluation Score

We evaluate each target stage across four shared dimensions, using stage-specific criteria and admissible evidence (Table 5).

Table 5: Stage-conditioned scoring specification. The four dimensions are shared across target stages, while their criteria and admissible evidence remain stage-specific.

Dimension	High-Score Anchor	Failure Signal
Scientific correctness	Recovers the target claims, roles, and assumptions without contradiction.	Missing or contradictory scientific content.
Experimental adequacy	Preserves variables, controls, comparisons, metrics, and conditions.	Missing controls, incomparable baselines, or unjustified design.
Evidence grounding	Grounds claims in permitted evidence and preserves values, units, directions, rankings, and uncertainty.	Unsupported, fabricated, or quantitatively incorrect claims.
Conclusion scope	Keeps conclusions within the evidence, distinguishes correlation from causation, and states relevant limitations.	Overbroad, falsely causal, or poorly calibrated conclusions.

A.4 Task Inputs, Outputs, and Independence

Each paper–mode instance contains four components: a paper–mode identifier, a frozen set of released nodes, a set of held-out target nodes, and node-level instructions with evidence-access constraints. The agent may use only the paper-local evidence package and execution budget provided for that instance. It cannot read hidden nodes from another disclosure mode or use an earlier run's agent output as input. Modes are therefore independent runs rather than turns in a single multi-round dialogue.

The task output is an auditable research report rather than a stage summary alone. For each target node, the report should provide:

- verifiable claims and their scientific stages;
- pointers to permitted evidence, figures, tables, formulas, or data identifiers;
- variables, conditions, comparison objects, numerical values, and uncertainty;

- an `unavailable` judgment or limitation when evidence is insufficient, execution is impossible, or a conclusion exceeds its scope.

The report must not present agent-generated results as paper observations, and fluency cannot substitute for evidential support. These requirements distinguish scientific-content recovery from formatted summarization.

A.5 Blinded Expert Calibration of the Automatic Judge

We validate the automatic judge through a stratified paper-level expert audit. The audit set is sampled from the 53 SCIONBench papers before inspecting automatic scores, agent reports, or failure cases. We randomly select one paper from each of the 15 scientific domains and then select five additional papers from domains containing at least two papers, yielding 20 audited papers in total.

For each audited paper, we evaluate all six model–agent configurations. Within each paper–configuration pair, we randomly sample up to two applicable nodes from each scored target stage: method, experimental design, experiment, and conclusion. Motivation is provided as context and is not scored. The same reference answer, admissible evidence, scoring rubric, and final agent report are provided to the experts and to the Kimi judge.

Two domain-matched experts independently score every sampled report–node pair on the 0–5 scale and assess the four rubric dimensions: scientific correctness, experimental adequacy, evidence grounding, and conclusion scope. Experts do not observe the model identity, agent, automatic score, or judge rationale. A third expert reviews items with a score difference of at least two points or disagreement on a critical error label.

We report agreement at both the node level and the paper–stage level. The paper–stage score is the mean of the sampled node scores within each paper and target stage. Human–human agreement measures the consistency of the expert rubric, while Kimi–expert agreement measures the validity of the automatic judge. Confidence intervals are obtained with a domain and paper-clustered bootstrap.

Table 6: Agreement between the Kimi judge and blinded expert ratings across 20 papers from 15 scientific domains.

Target stage	**Expert**	**Kimi**	**H_ICC**	**K_ICC**	**MAE**	**Signed error**	**Within ±1**
Method	2.84	2.78	0.91	0.86	0.38	-0.06	91%
Experimental design	2.61	2.67	0.89	0.82	0.47	+0.06	87%
Experiment	3.12	3.04	0.93	0.88	0.34	-0.08	94%
Conclusion	4.01	4.08	0.95	0.90	0.29	+0.07	96%
Overall	3.15	3.14	0.92	0.87	0.37	-0.02	92%

Expert and Kimi columns report mean scores on a 0–5 scale. ICC measures absolute agreement. MAE is the mean absolute difference between the Kimi score and the expert-consensus score. Signed error is the Kimi score minus the expert-consensus score. "Within ±1" is the fraction of samples whose Kimi score differs from the expert consensus by at most one point.

A.6 Scientific-domain Results

Table 7 reports paper-weighted results for the 15 original scientific domains. Across all domains and configurations, conclusion mode achieves the highest score, ranging from 7.04 to 9.40, whereas Motivation-mode scores range from 1.96 to 6.02. Thus, the gap between evidence-conditioned conclusion synthesis and reconstruction from limited upstream context is visible across the benchmark rather than being driven by a single broad field. Performance nevertheless varies substantially across domains, particularly in Motivation, Method, and Experiment modes, indicating that reconstructing methods, experimental evidence, and their dependencies remains sensitive to the scientific setting. These domain-level results are descriptive: several domains contain few papers, including four singleton domains, so they should not be interpreted as estimates of general disciplinary competence or model superiority within a field.

Table 7: **Fine-grained results across the 15 original scientific domains.** Values are paper-weighted means of final paper-level effective scores within each domain, including terminal-timeout zeros. Overall equally weights the four controlled disclosure modes. Domains with few papers, especially singleton domains, are descriptive only.

Agent	Model	Mot.	Meth.	Exp.	Concl.	Overall
Lunar and Planetary Science						
Claude Code	DeepSeek-V4-Pro	4.45	**6.10**	7.32	**9.12**	**6.75**
Claude Code	Kimi K3	**4.50**	5.76	**7.39**	9.00	6.66
Claude Code	Claude Sonnet 5	2.89	5.38	5.62	8.63	5.63
Codex	DeepSeek-V4-Pro	3.85	5.99	7.38	9.04	6.57
Codex	GLM-5.2	2.76	4.74	5.39	8.92	5.45
Codex	GPT-5.6 Sol	3.48	4.47	6.53	9.10	5.89
Polymer and Composite Materials)						
Claude Code	DeepSeek-V4-Pro	3.64	**5.65**	6.02	9.03	6.09
Claude Code	Kimi K3	3.61	5.16	5.21	8.95	5.73
Claude Code	Claude Sonnet 5	3.04	4.42	6.01	8.76	5.56
Codex	DeepSeek-V4-Pro	**3.85**	5.46	**6.15**	**9.14**	**6.15**
Codex	GLM-5.2	1.99	2.30	3.56	8.59	4.11
Codex	GPT-5.6 Sol	2.85	4.50	5.35	9.05	5.44
Condensed-Matter Physics and Computational Materials						
Claude Code	DeepSeek-V4-Pro	2.81	**4.54**	5.50	9.13	5.49
Claude Code	Kimi K3	2.73	4.09	6.01	8.97	5.45
Claude Code	Claude Sonnet 5	2.07	3.59	5.65	9.08	5.10
Codex	DeepSeek-V4-Pro	**2.83**	4.39	**6.63**	9.14	**5.75**
Codex	GLM-5.2	1.96	3.48	5.73	8.77	4.98
Codex	GPT-5.6 Sol	2.31	4.00	5.10	**9.15**	5.14
Optimization, Quantum Annealing, and Statistical Physics						
Claude Code	DeepSeek-V4-Pro	**3.02**	**5.63**	5.68	8.93	5.82
Claude Code	Kimi K3	2.38	4.70	5.26	8.96	5.32
Claude Code	Claude Sonnet 5	2.57	4.08	5.87	8.28	5.20
Codex	DeepSeek-V4-Pro	2.65	5.61	**6.68**	**9.08**	**6.01**
Codex	GLM-5.2	2.27	3.60	4.73	8.41	4.75
Codex	GPT-5.6 Sol	2.42	3.45	4.95	7.93	4.68
Earth Observation, Natural Hazards, and Seismology						
Claude Code	DeepSeek-V4-Pro	**4.58**	6.64	**8.05**	8.91	**7.04**
Claude Code	Kimi K3	4.25	6.18	7.83	8.70	6.74
Claude Code	Claude Sonnet 5	3.53	5.23	6.87	8.84	6.12
Codex	DeepSeek-V4-Pro	3.79	**6.67**	7.82	8.98	6.82
Codex	GLM-5.2	2.70	4.58	6.21	8.60	5.52
Codex	GPT-5.6 Sol	2.84	3.61	4.88	**9.00**	5.08
Biomedicine, Spatial Omics, and Immunotherapy						
Claude Code	DeepSeek-V4-Pro	**4.16**	**6.29**	**6.78**	9.15	**6.59**
Claude Code	Kimi K3	3.77	5.68	6.11	8.93	6.12
Claude Code	Claude Sonnet 5	3.88	5.51	6.32	9.19	6.22
Codex	DeepSeek-V4-Pro	3.16	6.07	6.70	**9.21**	6.29
Codex	GLM-5.2	2.50	4.02	6.51	8.85	5.47
Codex	GPT-5.6 Sol	2.74	4.31	5.66	9.10	5.45
Functional Genomics and Epigenetics						
Claude Code	DeepSeek-V4-Pro	4.16	**6.10**	5.96	8.17	6.10
Claude Code	Kimi K3	4.32	5.94	6.33	**8.33**	6.23
Claude Code	Claude Sonnet 5	4.32	5.63	**7.13**	8.17	**6.31**
Codex	DeepSeek-V4-Pro	**4.48**	6.06	5.49	8.23	6.06
Codex	GLM-5.2	2.12	5.63	6.28	7.87	5.47
Codex	GPT-5.6 Sol	3.07	4.06	4.80	7.90	4.96
Computational Chemistry and Molecular Machine Learning						
Claude Code	DeepSeek-V4-Pro	**4.25**	**5.03**	5.99	**9.12**	6.10

Table 7 continued

Agent	Model	Mot.	Meth.	Exp.	Concl.	Overall
Claude Code	Kimi K3	3.91	4.78	6.98	8.98	6.16
Claude Code	Claude Sonnet 5	3.26	4.39	7.41	8.90	5.99
Codex	DeepSeek-V4-Pro	3.97	4.28	**7.63**	**9.12**	**6.25**
Codex	GLM-5.2	2.32	3.63	5.83	8.77	5.14
Codex	GPT-5.6 Sol	2.77	3.41	5.10	9.06	5.09
High-Energy Physics						
Claude Code	DeepSeek-V4-Pro	3.46	3.80	2.84	8.80	4.72
Claude Code	Kimi K3	3.34	3.59	3.92	8.35	4.80
Claude Code	Claude Sonnet 5	**4.00**	3.47	3.30	8.61	4.84
Codex	DeepSeek-V4-Pro	3.03	3.84	3.93	**8.84**	4.91
Codex	GLM-5.2	2.75	2.23	2.69	8.09	3.94
Codex	GPT-5.6 Sol	3.99	**4.86**	**5.04**	8.76	**5.66**
Astronomy and Astrophysics						
Claude Code	DeepSeek-V4-Pro	**3.67**	**4.51**	**7.01**	**8.98**	**6.04**
Claude Code	Kimi K3	3.63	3.46	6.22	8.69	5.50
Claude Code	Claude Sonnet 5	2.83	3.95	5.80	8.86	5.36
Codex	DeepSeek-V4-Pro	3.56	4.37	6.69	8.87	5.88
Codex	GLM-5.2	2.35	3.13	6.01	8.53	5.01
Codex	GPT-5.6 Sol	2.83	4.01	6.90	8.89	5.66
Energy Materials and Electrochemistry						
Claude Code	DeepSeek-V4-Pro	3.78	**7.29**	6.73	**9.17**	**6.74**
Claude Code	Kimi K3	**3.90**	5.99	7.10	9.04	6.51
Claude Code	Claude Sonnet 5	3.77	6.82	6.22	8.71	6.38
Codex	DeepSeek-V4-Pro	3.38	7.05	**7.33**	9.01	6.69
Codex	GLM-5.2	2.64	4.74	5.20	8.53	5.28
Codex	GPT-5.6 Sol	2.39	2.86	5.02	8.66	4.73
Plasma Physics and Fusion						
Claude Code	DeepSeek-V4-Pro	**4.57**	4.80	**6.94**	9.04	6.34
Claude Code	Kimi K3	3.80	4.50	6.19	9.03	5.88
Claude Code	Claude Sonnet 5	4.05	4.03	6.61	8.90	5.90
Codex	DeepSeek-V4-Pro	4.23	**5.33**	6.76	**9.15**	**6.37**
Codex	GLM-5.2	2.41	3.39	5.10	7.04	4.48
Codex	GPT-5.6 Sol	2.88	3.25	6.26	8.98	5.34
Biomedical Imaging						
Claude Code	DeepSeek-V4-Pro	5.62	5.64	6.42	9.10	**6.70**
Claude Code	Kimi K3	4.73	**5.79**	**6.45**	8.90	6.47
Claude Code	Claude Sonnet 5	5.10	5.43	5.38	8.65	6.14
Codex	DeepSeek-V4-Pro	**6.02**	5.57	4.25	**9.23**	6.27
Codex	GLM-5.2	4.06	4.54	6.12	8.43	5.79
Codex	GPT-5.6 Sol	4.16	5.36	5.75	9.10	6.09
Intelligent Computing for High-Energy Physics and Astronomy						
Claude Code	DeepSeek-V4-Pro	4.47	4.20	**8.50**	9.30	6.62
Claude Code	Kimi K3	4.59	4.26	7.50	9.00	6.34
Claude Code	Claude Sonnet 5	**4.97**	**6.00**	**8.50**	9.10	**7.14**
Codex	DeepSeek-V4-Pro	4.56	5.30	8.00	**9.40**	6.81
Codex	GLM-5.2	3.80	3.50	4.50	9.10	5.22
Codex	GPT-5.6 Sol	3.36	5.10	3.00	9.37	5.21
Functional Fibers and Device Materials						
Claude Code	DeepSeek-V4-Pro	2.97	5.89	7.50	9.25	6.40
Claude Code	Kimi K3	**4.29**	5.11	6.50	9.23	6.28
Claude Code	Claude Sonnet 5	2.88	6.29	7.50	**9.40**	6.52
Codex	DeepSeek-V4-Pro	3.44	**6.59**	**8.20**	9.27	**6.87**
Codex	GLM-5.2	2.86	4.83	7.00	8.88	5.89
Codex	GPT-5.6 Sol	3.07	4.51	5.50	9.32	5.60

B PROMPT TEMPLATE

We present condensed English versions of the prompts used for graph extraction, agent execution, evaluation, and resource auditing. Experiments used complete Chinese versions. Repeated filesystem instructions, full JSON schemas, and duplicated JSON/Markdown variants are omitted. **Bold text highlights the constraints that preserve scientific evidence across stages.**

B.1 EXTRACTION PROMPT

The extraction prompt constructs a reference graph whose nodes contain scientific content and whose edges record the dependencies supporting that content.

Self-contained evidence graph

Convert the manuscript and its scientific figures into a graph spanning **motivation, method, experimental design, experiment, and conclusion**. Read the complete source and build the evidence records before constructing nodes. Each record identifies its source, location, verbatim content where available, scientific relevance, and confidence.

Every node, edge, numerical record, and evaluation metric must be grounded in identifiable source evidence. Store a bounded scientific statement, its conditions, and the relevant evidence references in each node.

Make each node's `article_content` a **self-contained reference answer**. Include the relevant data sources, splits, preprocessing, conditions, measurements, formulas, units, table and figure values, missing entries, and uncertainties. Evidence pointers support auditing; they do not replace the scientific content. The reference must remain usable after source-file paths are removed.

Record unreadable or unreported content explicitly. Do not invent values, infer missing results, or add scientific relationships merely to complete the graph. Validate the graph and repair structural or coverage errors without deleting evidence to make validation pass.

Five scientific stages

- **Motivation:** Store the problem, gap, constraints, and falsifiable objectives. Conceal paper identity, method names, exact datasets, final numerical results, and overall conclusions. The designated motivation root has no parents and is **provided as context, never scored**.
- **Method:** Record reusable scientific mechanisms, assumptions, abstract inputs and outputs, and necessary invariants. Separate independently replaceable or testable approaches. Keep run-specific datasets, hardware, seeds, and outcomes in the appropriate downstream stages.
- **Experimental design:** Record variables, controls, data splits, preprocessing, repetitions, metrics, budgets, and decision rules that can be specified before execution. **Do not reconstruct an unreported prior design retrospectively from the results.**
- **Experiment:** Record execution conditions, observations, failures, derived statistics, uncertainties, and conflicting results. Keep observations distinct from interpretation. Preserve reported results even when the experiment is not executable under the benchmark resources.
- **Conclusion:** Express each local conclusion as an independently verifiable claim with evidence, scope, causal strength, and uncertainty. **Build the overall conclusion from stage conclusions and branch syntheses**, preserving their conditions and limitations.

Scientific dependencies and method coverage

Treat `edges` as the authoritative directed graph. Nodes may have zero, one, or multiple scientifically justified parents. Every parent and dependency must have a corresponding explicit edge, with its own evidence and rationale. **Do not force a single chain, a single parent, or artificial reachability.**

Distinguish parallel methods from **strict serial dependencies**. Strict seriality requires either an implementation artifact from method A that method B needs to execute, or an actual result from A that determines B's subsequent route, parameters, or samples. Shared datasets, public models, reusable code, or section order alone do not establish strict seriality.

For every method, record **many-to-many links to the designs, results, and conclusions that test or use it**. Identify the factors actually varied, ablated, measured, or invoked. Attribute a component-level intervention to that component; record jointly changed factors as joint changes.

Distinguish direct testing, indirect testing, downstream use, and no follow-up. A component used by the full system is indirectly tested even without a separate ablation. Absence of a graph child does not establish absence of scientific coverage. Preserve the explicit method–design, design–result, and result–conclusion links and evidence for any genuine coverage gap.

Node-specific evaluation plans

Select metrics according to the scientific role of the node and the artifact the agent will produce. Each metric must specify its definition, scoring rule, failure conditions, and missing-data policy.

For methods and designs, evaluate **mechanistic invariants, protocol fidelity, baseline fairness, and reproducibility**. Penalize missing controls and data leakage; assign zero for unfair baseline comparisons. For the method-invariant metric, retain the specified four-dimensional 0–4 rubric and its 0.25 cap when a core factual error is present. Full reproducibility credit requires recorded seeds and data splits.

For executable experiments, evaluate genuine execution, artifacts, and resource compliance. Apply the paper's scientific metric definitions to the produced artifacts before comparing **ranges, directions, baseline differences, and stability**. Use narrow numerical tolerances only under matching protocols, data, software, and hardware.

For conclusions, evaluate **evidence traceability, uncertainty, scope, cross-stage consistency, and branch synthesis**; penalize causal overclaims. Do not create a reproduction score for an experiment with no agent output. Where the protocol supplies its reported results as context, assess the ensuing reasoning against the applicable conclusion targets.

B.2 Test Prompts

The JSON and Markdown variants share the following requirements and differ in input representation. Each disclosure mode starts an independent run with the same frozen paper-local resources. The runner withholds downstream reference nodes, evaluator-only material, and outputs from other runs.

Shared execution and reporting requirements

Continue the research using **only the released context and local resources for the current paper**. Internet access is prohibited.

Produce a single Markdown research report covering the motivation and method, experimental design and results, module-level conclusions, and overall conclusion. **Distinguish inherited facts, your design choices, assumptions, actual observations, and**

Shared execution and reporting requirements (continued)

speculation. Record conditions, controls, seeds, commands, environment information, logs, and artifacts needed to trace the reported work.

Report only experiments actually performed and evidence actually available. If resources or execution are insufficient, state the blocking reason and completed portion. Do not fabricate results or present assumptions as observations. Every conclusion must be supported by the reported evidence and bounded by its uncertainty and scope.

four controlled disclosure modes

- **Motivation mode: only motivation is disclosed.** Formulate the research question and testable hypotheses; propose a method; design and execute experiments; analyze the results; and derive module-level and overall conclusions.
- **Method mode: motivation and method are disclosed.** Preserve the supplied method and independently design and execute its experiments. Explain new assumptions and design choices, report genuine observations, and derive evidence-supported conclusions.
- **Experiment mode: motivation, method, and design are disclosed.** Execute the prescribed experiments. Record actual conditions, logs, raw results, failures, and anomalies; analyze the observations and derive module-level and overall conclusions.
- **Conclusion mode: prior stages and experimental results are disclosed.** Derive a supported conclusion for each branch, then synthesize the overall conclusion. **Do not conduct new experiments or introduce unprovided observations.**

B.3 Evaluation Prompts

The runner selects held-out, applicable nodes using the frozen applicability mask. The judge receives the reference node, its evaluation plan and criteria, relevant evidence, and the agent's final report. Motivation remains unscored. The four mode-specific judge prompts share the following procedure; the target node determines the scientific criteria.

Evidence-grounded node evaluation

Locate the report content corresponding to the supplied reference node. Evaluate each applicable metric in its `evaluation_plan`, then assign a **node-level score from 0 to 10** using the supplied criteria. If the plan is absent, apply the remaining node-level criteria.

Use **scientific semantic matching**: wording and presentation may differ, but mechanisms, conditions, observations, and claims must remain scientifically consistent. Additional material is acceptable only when evidence supports it and it remains within scope. Stylistic completeness cannot compensate for factual errors, missing controls, fabricated execution, or unsupported claims.

Apply the appropriate emphasis:

- **Method:** fidelity of mechanisms and design to the reference, transformation of motivation into a testable approach, and the supplied criteria for scientific value and boundary cases.
- **Experimental Design:** faithful operationalization of the method, genuine data, adequate controls, fair comparisons, and feasible resources.
- **Experiment:** genuine execution, auditable artifacts, relevant result ranges and trends, resource compliance, and explained discrepancies.
- **Conclusion:** a traceable **claim–evidence–uncertainty–scope** chain, consistent branch synthesis, and calibrated causal language.

Evidence-grounded node evaluation (continued)

The shared bands are 0–2 for predominantly missing or inconsistent content; 3–4 for limited agreement with major gaps; 5–6 for broad agreement with important weaknesses; 7–8 for mostly correct and well-supported content; and 9–10 for near-complete, rigorous agreement under the node-specific criteria.

Return the node identifier, score, and a concise explanation of passed and failed criteria, with the supporting or missing evidence made explicit.

B.4 Data Cleaning and Experiment-Evidence Evaluability Prompts

These stages have separate roles: cleaning determines what enters the frozen release; the subsequent audit determines which experimental evidence can be recovered from that release under the fixed runtime contract.

Frozen scientific inputs without direct answers

Construct a separate `clean_data` release for every paper from the specified source manifest. Preserve the original data and do not supplement inputs online. **Retain genuine scientific inputs while removing explicit paper identity, author implementations, and direct research answers.**

Distinguish independent ground-truth labels from target-method predictions, ordinary processed measurements from final analyses, and initial candidate pools from final rankings. Preserve sample conditions, units, controls, replicates, and cross-file correspondences. Classify by scientific content and provenance; filenames or extensions alone are insufficient.

Use `KEEP`, `SANITIZE`, `REMOVE_IDENTITY`, `REMOVE_CODE`, `REMOVE_RESULT`, `REMOVE_INVALID`, or `REVIEW`. Exclude unresolved high-risk artifacts from the release, while retaining the paper. Separate legitimate data from code and results in mixed directories or archives. Anonymize explicit identifiers consistently and keep the mapping and audit records private.

Do not change scientific values or generate substitute data. Use format-aware transformations, reopen rewritten files to verify integrity, and record each retention or removal decision. Materialize isolated files without symbolic or hard links.

Retain every paper, including those with empty releases, and provide the same frozen data to all four modes. Cleaning does not decide experimental evaluability. Freeze the release after validation; do not claim that removal of explicit leaks guarantees the absence of semantic identification clues.

Define the evidence target and recovery boundary

Audit **each experiment individually** against the frozen data and runtime contract. Use a reference graph blinded to historical executability decisions; stop if historical labels or rationales remain. Do not access other papers, previous agent/judge outputs, or external data, and do not modify the release.

Specify the `scored_evidence_target`: all results needed to support the experiment's core scientific claim. Then specify the `evidence_recovery_boundary`: which steps are already satisfied by readable, released resources and what the agent must still complete.

Assess scientific evidence recoverability, not mandatory repetition of every historical physical step. Existing measurements can support new computational analysis without repeating sample collection. The starting point must follow actual released resources, not results found only in the reference graph. **Do not narrow the target or move the boundary to make a node pass.**

Define the evidence target and recovery boundary (continued)

Assume a scientifically compatible method and design. The audit label is independent of whether a particular disclosure mode reveals them. Treat the frozen release as approved input; this audit evaluates its sufficiency and does not reopen the preceding cleaning decision.

Sequential evaluability decisions

Assess the following questions in order:

1. **Are the frozen inputs sufficient for the complete evidence target?** Account for theoretical or simulation inputs that can legitimately be generated within the contract. Missing irreplaceable inputs, corrupted files, or insufficient proxies imply `NOT_EVALUABLE` because required inputs are missing.
2. **Are new physical observations still required?** If existing data cannot replace a necessary new intervention, sample, or measurement, assign `NOT_EVALUABLE`. Historical wet-lab work is not a blocker when sufficient observations are already released.
3. **Does the remaining computation clearly exceed the budget?** Require quantitative evidence or a hard resource limit. Missing author code, ordinary packages, implementation effort, or an imprecise runtime estimate alone is not a blocker. Reduce redundant computation without reducing core dataset size, model scale, controls, or statistical power.

Assign `EVALUABLE` only for **sufficient inputs, no required new physical observations, and no established budget violation**. Stop after the first clear blocker and mark later questions `NOT_ASSESSED`, with a reason. For genuine uncertainty, record a low-confidence, provisional `NOT_EVALUABLE` decision, stop subsequent assessment, and request manual review. Do not use a partial label or include provisional decisions in the official mask.

For each node, retain the target, boundary, actual input paths, question-level evidence, decision, blocker, and confidence. Validate node coverage and path isolation. **Freeze the official applicability mask only after validation and any required human adjudication.**

The audit is conducted independently by DeepSeek-V4-Pro-Max with Claude Code and GPT-5.6-Sol-Max with Codex. **An experiment enters the executable applicability mask only when both audits return `EVALUABLE`.** Disagreements and provisional cases do not enter that mask without the prescribed resolution and freezing.