SCIRIGOR: EVALUATING OPEN-ENDED SCIENTIFIC ANALYSIS BEYOND FINAL SCORES

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific coding agents produce code, results, figures, and written findings within a single run. A correct final claim can nevertheless accompany contradictory analysis or incomplete evidence. We introduce **SCIRIGOR**, a benchmark that evaluates both claim correctness and support from the run's computations, figures, and analysis. It contains 100 cases from 27 articles across six domains and 17 subfields, with source code, inputs, results, and figures for verification. An evidence graph connects each claim to numerical results, plotted comparisons, and written interpretations. The evaluator checks these records against reference requirements and tests whether they support one another. Each claim's support is limited by its weakest required check, while approved alternatives accommodate valid methods and visual encodings. Across eight model–runner configurations, the evaluator identifies incomplete support in 64 of 357 submissions with perfect final-claim scores. The highest Path F1, which measures supported claims and finding coverage, is 84.4%; no configuration exceeds 40.8% Strict success, which requires all mandatory checks to pass.

1 INTRODUCTION

Scientific data analysis turns data into research findings through computation, visualization, and interpretation (Chen et al., 2025; Wang et al., 2026a;b). Agents can produce conflicting findings across these outputs within one run. Consider an agent comparing people named in Nature science news with first authors of Nature research papers, year by year from 2005 to 2020. It compares the supplied annual shares of names predicted as East Asian. In one Qwen3.8-Max submission, the results and plot show a lower news share in all 16 years. The written report says zero years, while the agent's final answer correctly states that the news share is lower in all 16 (Figure 1). The answer is correct, but the report contradicts it.

Existing benchmarks examine executable analyses, analytical decisions, and scientific deliverables. BLADE uses data-flow matching, DataSciBench uses executable task functions, and FrontierChallenge checks deliverables and cross-artifact consistency (Gu et al., 2024; Zhang et al., 2026; Su et al., 2026). Final-answer scoring cannot distinguish two submissions with the same answer when only one supplies complete supporting evidence. Contradictions make this gap visible; agreeing artifacts can also omit necessary evidence. We therefore evaluate each claim together with the evidence required to support it. The challenge is to identify the results, plotted comparisons, and report statements relevant to each claim, then check whether they provide full support.

SCIRIGOR evaluates claim correctness and evidence support together (Figure 2). Agents receive a question, analysis-ready data, and study context. They submit six artifacts: code, results, a figure, analysis, claims, and provenance. An evidence graph links each claim to specific computations, results, plotted marks, and report statements. The evaluator checks these records against reference requirements and tests their support relations. Full support requires every mandatory check to pass; approved alternatives allow different valid methods and visual encodings. Each claim's score reflects its verified support; failed checks locate the gaps.

Across eight model–runner configurations, the evaluator identifies incomplete support in 64 of 357 submissions with perfect final-claim scores. The highest complete success rate is 40.8% under the benchmark requirements.

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.

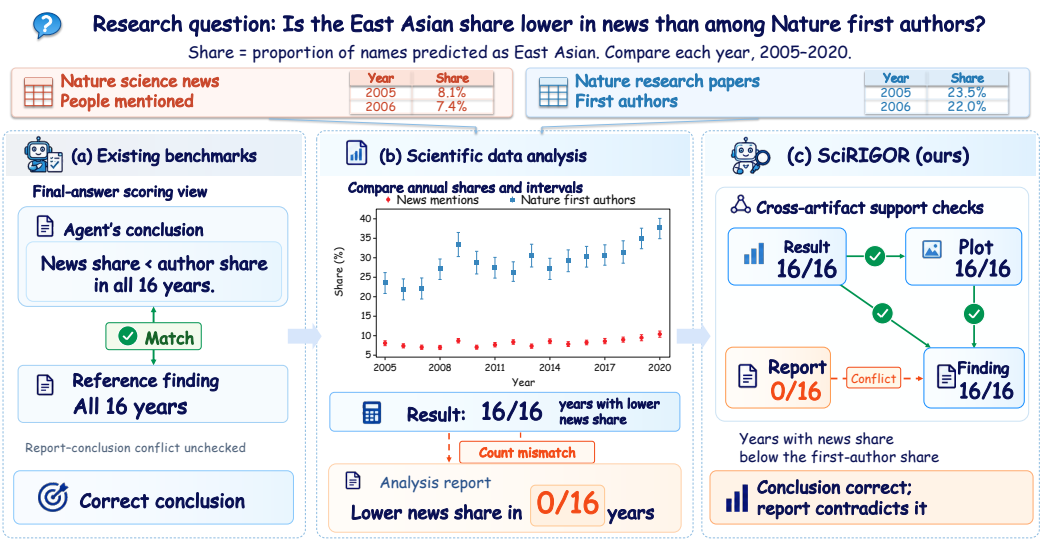


Figure 1: **A scientific data analysis can reach the correct conclusion with a contradictory report.** The agent compares annual name-origin shares: the results, plot, and final conclusion indicate a lower news share in all 16 years, but the report says zero. The left panel shows the final-answer scoring component of existing benchmarks. Name origin is predicted, not verified ethnicity (Appendix I.8).

We make three contributions:

- **A benchmark of complete scientific submissions.** We curate 100 cases from 27 articles across six domains, with analysis-ready inputs and verifiable reference findings. Each case passes two clean executions and human review.
- **A protocol for cross-artifact claim support.** We link each claim to its required evidence, check reference fidelity and local support, and score jointly required checks and approved alternatives.
- **An empirical analysis of missing and conflicting evidence.** We measure the gap between final-claim correctness and complete support, compare score-composition rules, and locate observed failures in the evidence graph.

2 Related Work

Scientific-agent benchmarks score different parts of a research workflow. ScienceAgentBench and NatureBench evaluate paper-derived scientific programs (Chen et al., 2025; Wang et al., 2026a); BLADE matches analytical decisions through data-flow graphs, and DataSciBench uses executable task functions (Gu et al., 2024; Zhang et al., 2026). DiscoveryBench and FIRE-Bench evaluate discoveries and research workflows (Majumder et al., 2025; Wang et al., 2026b). FrontierChallenge checks deliverables and cross-artifact consistency (Su et al., 2026). SciRIGOR evaluates each claim together with its required computational, visual, and textual evidence. It scores support from the same run by combining jointly required checks and approved alternatives.

Complementary work evaluates scientific visualizations, figure quality, and claim evidence (Ai et al., 2026; Deng et al., 2026; Yue et al., 2026). SCHEMA uses scientific concept graphs and counterfactual interventions to audit trajectory assertions (Feng et al., 2026). Our graph instead represents executed artifacts and their dependencies. We check newly generated results, figures, and claims against independently verified requirements and locate observed gaps in their support. These locations describe failed checks, not errors in the agent's internal reasoning. Table 1 summarizes benchmark coverage. Appendix N provides the extended discussion.

Table 1: Support coverage of scientific-agent benchmarks.

Benchmark	Code	Figure	Generated claim	Claim path	First failure
ScienceAgentBench (Chen et al., 2025)	✓	△	✗	✗	✗
BLADE (Gu et al., 2024)	✓	✗	✗	△	✗
DiscoveryBench (Majumder et al., 2025)	△	✗	✓	✗	✗
DataSciBench (Zhang et al., 2026)	✓	△	△	✗	✗
FIRE-Bench (Wang et al., 2026b)	✓	△	✓	✗	△
SciVisAgentBench (Ai et al., 2026)	✓	✓	✗	✗	✗
SciRigor	✓	✓	✓	✓	✓

Symbols: ✓ = required and directly evaluated; △ = partial coverage; ✗ = outside the primary scope. *Claim path*: verified computational and visual support from the same run. *First failure*: earliest observed unsupported relation.

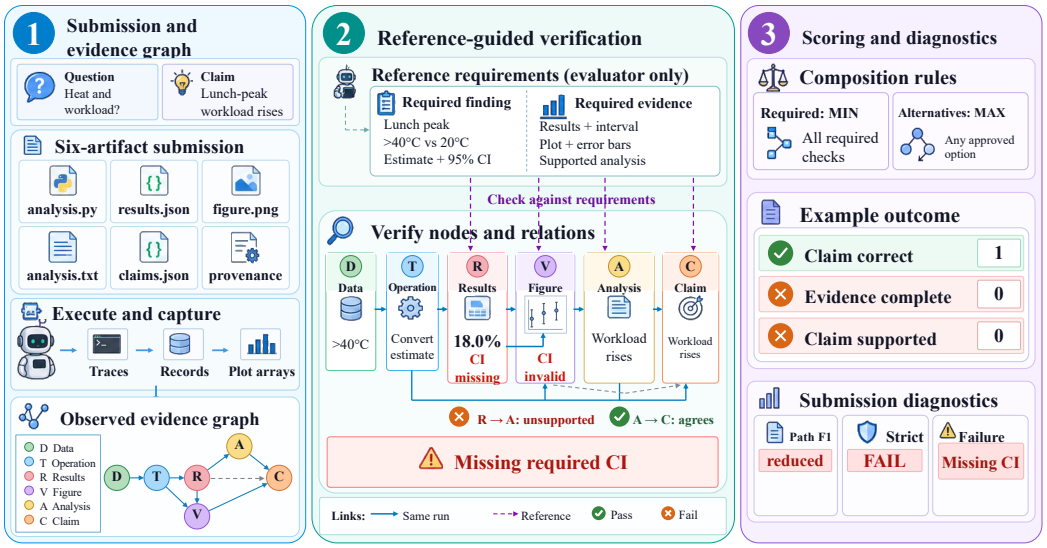


Figure 2: **A matching claim can still lack complete evidence support.** The three stages collect artifacts, verify evidence, and score claim support. The illustrative heat-workload case reports the correct 18.0% increase but lacks valid confidence-interval (CI) evidence, leaving support incomplete and causing Strict to fail. Selected relations are shown; Equation 2 specifies the full graph.

3 SciRigor

SciRigor asks whether the evidence produced in a run supports its reported findings. It links each claim to relevant records, checks their values and support relations, and combines those checks into a claim support score (Figure 2).

3.1 Task Formulation

Given a scientific question q, analysis-ready data $\mathcal{D}$, and study context κ, an agent $\mathcal{A}$ produces

$$\mathcal{A}(q, \mathcal{D}, \kappa) \longmapsto \widehat{\mathcal{Y}} = (\widehat{T}, \widehat{R}, \widehat{V}, \widehat{A}, \widehat{C}), \tag{1}$$

where $\widehat{T}$ is executable code, $\widehat{R}$ structured results, $\widehat{V}$ a visualization, $\widehat{A}$ textual analysis, and $\widehat{C}$ individual, separately checkable claims. Provenance $\widehat{P}$ records data sources and execution, completing the six-artifact submission (Table 7). Running the code must regenerate the results and figure; analysis and claims must refer to those same-run outputs. Success requires all requested findings and verified computational and visual support for every primary claim. Approved alternatives accommodate different valid methods and visual encodings.

3.2 From Artifacts to an Evidence Graph

Curators run the archived source code and define reference requirements for each finding. Only the evaluator has access to these requirements, which identify results by scientific meaning rather than variable names or file locations. For example, the heat-workload case compares lunch-peak workload above $40\,^{\circ}\mathrm{C}$ with the $20\,^{\circ}\mathrm{C}$ baseline. The reference reports an 18.0% increase and a 95% confidence interval of 12.6–23.6%. This is an association from processed regression outputs (Appendix I.2). A support set contains the checks that must all pass to support a finding. Separate sets describe approved alternatives. Curators verify each set against the source evidence.

The evaluator extracts individual records from the submitted files: operations (T), results (R), plotted marks (V), interpretations (A), and claims (C). With the selected input data D, their relations are:

$$D \to T \to R \to C, \qquad T \to V,\ R \to V,\ V \to C, \quad R \to A \to C. \tag{2}$$

The evaluator uses execution traces, regenerated outputs, plotting arrays, and reviewed text to verify the declared dependencies. In the heat example, it links the selected temperature bin to the percentage conversion, numerical result, plotted point and interval, and claim about that bin.

3.3 Artifact and Relation Verification

Verification separates two questions. **Reference fidelity** checks whether each record meets the reference requirements; **local support** checks whether linked records from the same run support one another. Node checks assess reference fidelity, and edge checks assess local support. Together they cover data selection, operation parameters, result values and uncertainty, plot groups, scales, and encodings, and each claim's direction, magnitude, population, and scope. Programs verify data sources, recompute values, and check units, plot bindings, and graph connections (Table 8). A fixed language-based matcher checks claim wording and scope. Its judgments cannot override numerical, structural, or provenance failures.

For example, the heat case requires applying $100[\exp(x) - 1]$ separately to the estimate and both confidence bounds. Converting only the estimate fails the reference uncertainty checks even if the claim reports 18.0% correctly. A plot that accurately displays incorrect bounds can pass the result-to-plot check while failing the reference uncertainty requirement. Local support therefore does not establish reference fidelity.

An LLM judge computes the Analysis score from coverage of required findings and factual accuracy. $R \to A$ checks whether results support the interpretation; $A \to C$ checks whether claims agree with it. Path scores include the applicable textual relations, but not the whole-report Analysis score. They therefore cover only the text linked to scored claims (Appendix J).

3.4 Claim-Path Scoring and Failure Localization

Each verified check has score $s(z) \in [0, 1]$. Programmatic checks return 1 for pass and 0 for fail; reviewed judgments of text meaning can receive graded scores. A check with unavailable observations remains unverified, with $[0, 1]$ bounds (Appendix R). For computational and visual support sets $\mathcal{P}_k(c)$ and claim correctness $F_C(c)$,

$$S_k(c) = \max_{p \in \mathcal{P}_k(c)} \min_{z \in p} s(z), \quad k \in \{\mathrm{comp}, \mathrm{vis}\}, \qquad S(c) = \min\{S_{\mathrm{comp}}(c), S_{\mathrm{vis}}(c), F_C(c)\}. \tag{3}$$

The minimum requires every check in a support set to pass; the maximum accepts any approved alternative that passes. For emitted primary claims $\mathcal{C}$ and required findings $\mathcal{G}$:

$$P = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} S(c), \qquad R = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} S(g), \qquad \text{Path F1} = \frac{2PR}{P + R}. \tag{4}$$

Here $S(g)$ is the support score for required finding g, computed by the same minimum/maximum rules. Unsupported claims reduce precision; missing required findings reduce recall. Zero total precision and recall gives F1 zero. **Strict** requires verified execution, full support for every primary claim and required finding, and all hard requirements. Artifact and edge scores remain separate diagnostics.

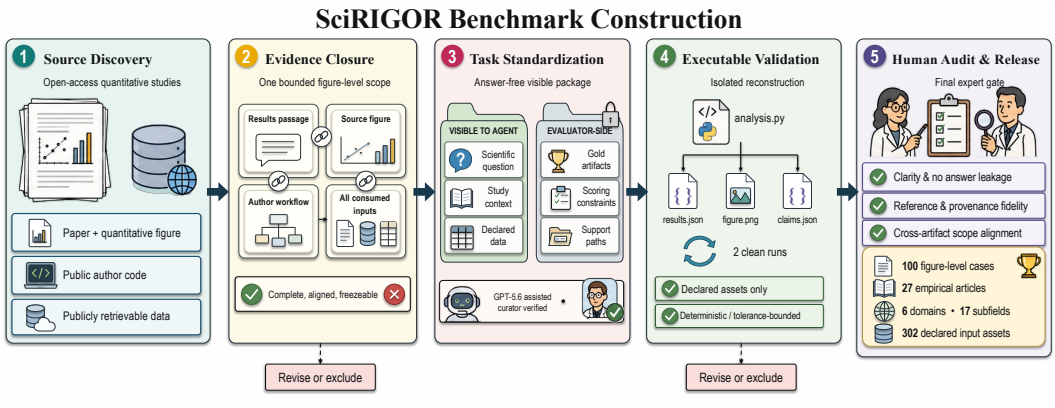


Figure 3: **Constructing cases with verifiable evidence.** Each case requires the selected figure's source code and all inputs, then passes two clean executions and human review. Agent-visible inputs are separated from reference artifacts and support rules.

The evaluator locates the earliest observed failed checks in the graph: submission/execution, Data→Transform, Transform→Result, Transform/Result→Visual, or analysis/claim checks. A failed path does not remove support if an approved alternative passes. Failure labels identify where observed support breaks down, without attributing the agent's underlying error. Unverified checks can leave earlier failures unobserved.

4 SciRigor Benchmark

SciRigor contains 100 scientific analysis cases from 27 empirical articles. Each case asks a research question about one figure or panel and links it to source code, inputs, numerical results, and reference findings. The question fixes the population, comparison, and measurement scope that all submitted artifacts must address. For example, a workload estimate, its confidence interval, and its visualization must describe the same temperature contrast. Figure 3 summarizes the construction process.

Source selection. We selected quantitative studies with accessible data, original analysis code, and a Results passage describing the target figure. Figure-level indexing linked source figures to quantitative statements, Methods sections, Data Availability statements, and repository identifiers. Curators located each figure's original source file, function, or notebook cell and verified every required input. Inputs could include raw observations, processed tables, model outputs, mappings, or configurations. Generic software, code for other panels, and independently written replacements did not satisfy this source requirement. We excluded missing dependencies, ambiguous operations, and substantial disagreement with published results. Visual references retain original figures or documented panel crops. Appendices E and F give the search and exclusion rules.

Task specification. Agents receive a neutral question, declared input files, and the scientific context needed to interpret measurements. The package stores the task in `input/task.json` and the available assets under `input/data/`. Curators checked the question's comparison groups and analysis unit, removing wording that disclosed target values or suggested the answer. The source paper, author code, reference outputs, and scoring rules remain hidden from the agent. This separation requires agents to generate findings from supplied data while giving the evaluator independent evidence for checking them.

Evidence requirements. Each reference finding specifies both the scientific statement and the evidence needed to support it. In the heat-workload case, this includes the temperature comparison, percentage change, confidence bounds, and their visual representation. A correct point estimate therefore cannot compensate for missing uncertainty. Computational and visual requirements remain linked to the same comparison, so separately plausible artifacts can still fail when they describe

different quantities. Approved alternatives specify acceptable ways to establish that support, as formalized in Section 3.

Reference reconstruction. GPT-5.6 assisted with indexing, task drafts, and proposed Python translations; curators checked these against the archived author workflows. Original Python operations were retained. Translations preserved data selection, transformations, estimators, scales, uncertainty calculations, and plot structure. The standardized program used only declared inputs, made no network calls, and regenerated results, figures, and claims. Figure checks compare plotted quantities, groups, units, and uncertainty, without requiring identical pixels. Methods outside the approved alternatives require expert review before acceptance and rescoring. Appendix G details standardization and the separation of visible and hidden files.

Verification and release. Each case passed two clean reconstruction runs with fixed inputs and software versions. Numerical results and plotted content had to match the source exactly or within specified tolerances. File-access traces verified that execution used only declared inputs. The release records paths and SHA-256 hashes for all 302 input files. Final human review checked that the question, source passage, code, outputs, and claims concerned the same analysis. Unresolved discrepancies led to revision or rejection. All 100 released cases record curator approval. The source registry retains the original code, software environment, execution logs, and comparison reports. Crop coordinates and hashes preserve the connection between panel references and source images. Appendix H describes these checks.

Coverage and complexity. The cases span six domains and 17 subfields: economics/policy (24 cases), computing/information (23), social/behavioral sciences (22), cognitive science (15), life/health sciences (9), and physical sciences (7). Assignments follow each study's scientific subject, rather than its statistical method or plot type. Cases contain 1–12 input files, 2–8 reference result entries, 3–6 visual requirements, and 1–5 reference claims. Their respective medians are 2, 4, 5, and 2. Methods include descriptive aggregation, regression, statistical inference, and trajectory analysis; a case may require several methods (Figure 4). Coverage reflects public code and data availability. Cases can share source articles, and these counts do not represent scientific practice as a whole. Appendix O provides the assignments.

5 EXPERIMENTS

Setup and metrics. We evaluate 800 submissions from eight model–runner configurations: five use the Codex runner and three use API tool loops. Each configuration contributes 100 submissions. Reasoning settings and recorded resource use differ across configurations (Appendix B). Missing artifacts and execution failures receive zero. Two cases have invalid reference findings; we exclude them from claim recall/F1 and complete-path metrics while retaining other applicable scores. We report artifact fidelity, Claim F1, local support, Path F1, and Strict, as defined in Section 3. Appendix L reports all diagnostics and denominators.

5.1 RQ1: CAN AGENTS PRODUCE EVIDENCE-GROUNDED SCIENTIFIC ANALYSES?

Table 2: Six-artifact results on SCIRIGOR (%; higher is better). Rows denote model–runner configurations (Appendix B). Blue shading is normalized within each row.

Model	Exec.	Result	Visual	Analysis	Claim F1	R→C	Path F1	Strict
GPT-5.5 (xhigh)	100.0	93.7	91.1	85.6	89.9	100.0	84.4	37.8
GPT-5.6 Sol (high)	100.0	90.9	90.9	85.1	89.8	99.9	79.0	40.8
GPT-5.6 Sol (low)	100.0	88.5	88.0	81.6	87.3	99.9	73.1	28.6
GPT-5.5 (high)	100.0	85.8	88.8	83.0	86.9	99.7	71.1	29.6
GPT-5.5 (low)	100.0	75.9	79.5	78.7	79.5	98.9	55.2	21.4
Kimi-K3 (low)	77.0	64.5	63.2	55.7	66.6	75.0	56.1	14.3
Qwen3.8-Max	77.0	49.1	59.9	50.6	58.7	72.1	36.5	11.2
Qwen3-Coder-30B-A3B	35.0	18.1	26.6	15.5	21.0	27.4	14.8	0.0

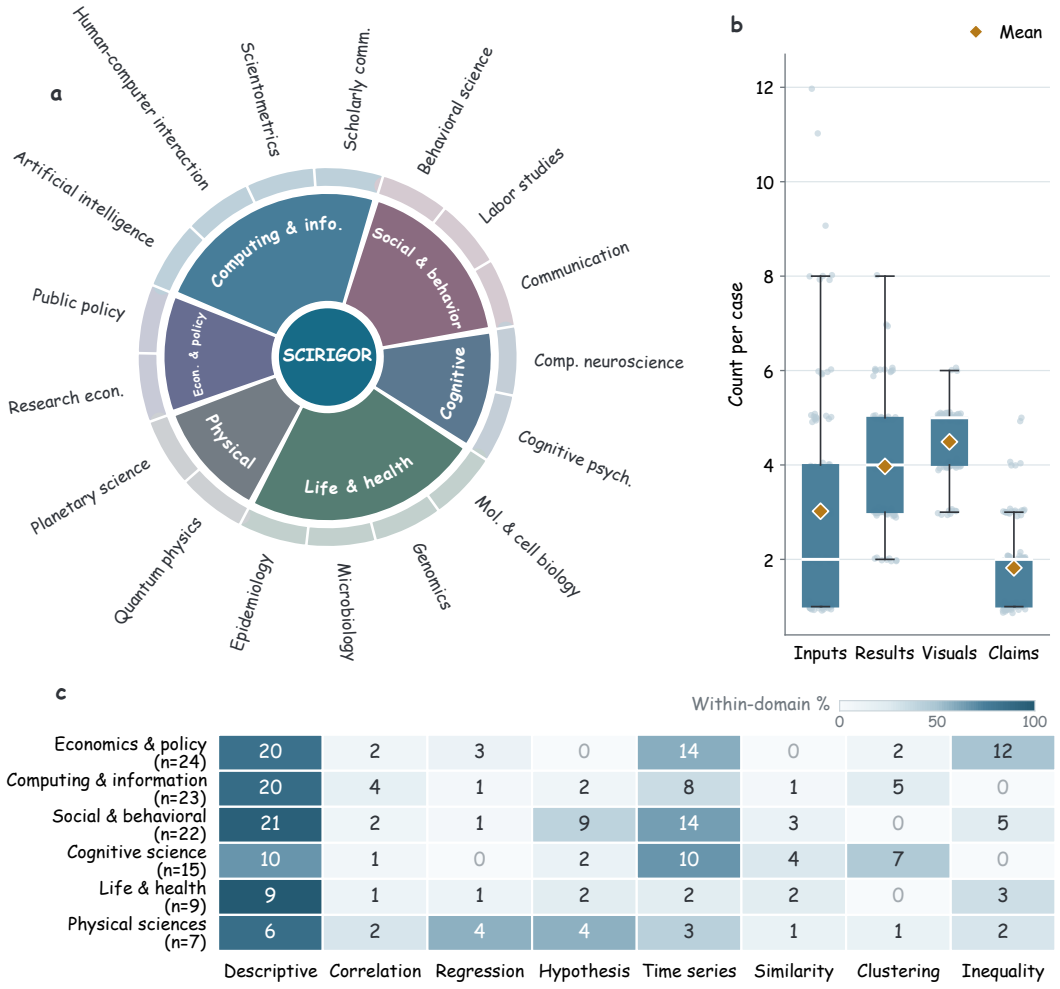


Figure 4: SCIRIGOR coverage and task complexity. (a) The 100 cases span six domains and 17 subfields. Subfields occupy equal angles; each domain spans its constituent subfields. (b) Counts of input files, reference result entries, plot requirements, and claims per case. (c) Cases by domain and method family; cases may use multiple methods. Cells show counts, with darker shading indicating higher within-domain percentages.

Internally consistent outputs can miss reference requirements. GPT-5.5 (low) obtains 98.9% $R \rightarrow C$ but only 75.9% Result and 79.5% Claim F1 (Table 2). Its claims largely agree with its own results, yet that agreement does not ensure correct values or coverage of required findings. Even GPT-5.5 (xhigh), with 93.7% Result and 91.1% Visual, reaches 84.4% Path F1 and 37.8% Strict. GPT-5.6 Sol (high) achieves the highest Strict, 40.8%, despite lower Path F1 (79.0%). Strict also depends on the number of mandatory requirements.

Execution creates a further gap across configurations. All five GPT configurations execute every case, compared with 77.0% for Kimi-K3 and Qwen3.8-Max, and 35.0% for Qwen3-Coder.

Performance also varies across the sampled domains (Figure 5): GPT-5.5 (xhigh) obtains 8.7% Strict in computing/information sciences versus 76.2% in social/behavioral sciences. Domain groups differ in task content and share source articles, so these are descriptive comparisons. Appendix A provides the full breakdown.

Resource use. GPT-5.6 Sol (high) achieves higher Strict success than GPT-5.5 (xhigh), with 27.4% fewer tokens and 21.9% less recorded time. GPT-5.5 (xhigh) leads on Path F1. Appendix C details the resource comparison and counting scopes.

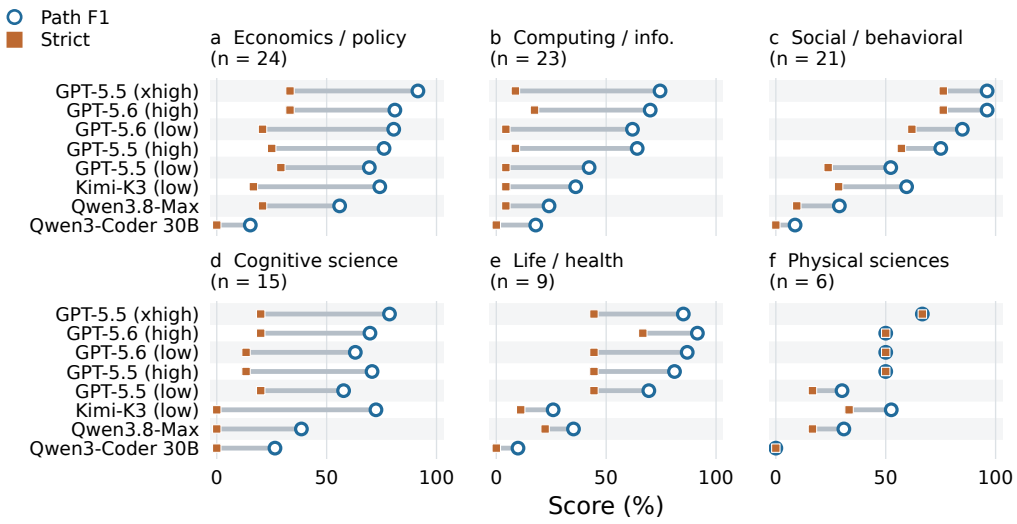


Figure 5: Path F1 (circles) and Strict (squares) across domains. Segments connect paired scores, not confidence intervals. The eight full-batch configurations share applicable references within each domain (n). GPT-5.6 denotes GPT-5.6 Sol; Table 4 reports all numerical results.

Table 3: Aggregation ablation (%; fixed checks). Counts: Appendix L.

Model	Claim F1	Flat mean	Flat min	Path F1
GPT-5.5 (xhigh)	89.9	98.0	84.2	84.4
GPT-5.6 Sol (high)	89.8	97.0	78.8	79.0
GPT-5.6 Sol (low)	87.3	96.8	72.9	73.1
GPT-5.5 (high)	86.9	95.8	70.8	71.1
GPT-5.5 (low)	79.5	92.1	54.9	55.2
Kimi-K3 (low)	85.8	96.3	73.9	74.3
Qwen3.8-Max	75.3	87.5	46.4	46.8
Qwen3-Coder-30B-A3B	61.4	88.2	41.3	42.2
Pooled	84.1	94.7	68.1	68.4

5.2 RQ2: What do evidence checks reveal beyond final-answer scores?

Perfect final-answer scores can hide incomplete evidence support. Among 357 submissions with perfect Claim F1, 64 (17.9%) have Path F1 below one and 177 (49.6%) fail Strict. These evaluator judgments cover missing and conflicting evidence under the benchmark requirements; configurations share cases.

To test whether averaging masks failed checks, we fix checks and their links to claims and findings (Table 3). *Flat mean* averages each claim's checks; *flat min* takes their minimum. Both retain precision/recall/F1 but treat required and alternative evidence alike. This tests aggregation rules, not other benchmarks' protocols.

Of 677 executed submissions with valid reference findings, 672 have known scores under all three rules; five have unobserved alternatives under the flat rules. Flat mean gives 94.7%, flat min 68.1%, and Path F1 68.4%. Requiring every mandatory check to pass accounts for most of the reduction from flat mean. Accepting approved alternatives recovers support in eight submissions, with a small overall effect relative to flat min.

What the additional checks reveal. The motivating submission reports lower proportions of predicted East Asian name origins in news mentions than among Nature first authors in all 16 years (Figure 1). Its final claim agrees, but the analysis reports zero years. Checking same-run artifacts exposes a contradiction that final-claim scoring misses.

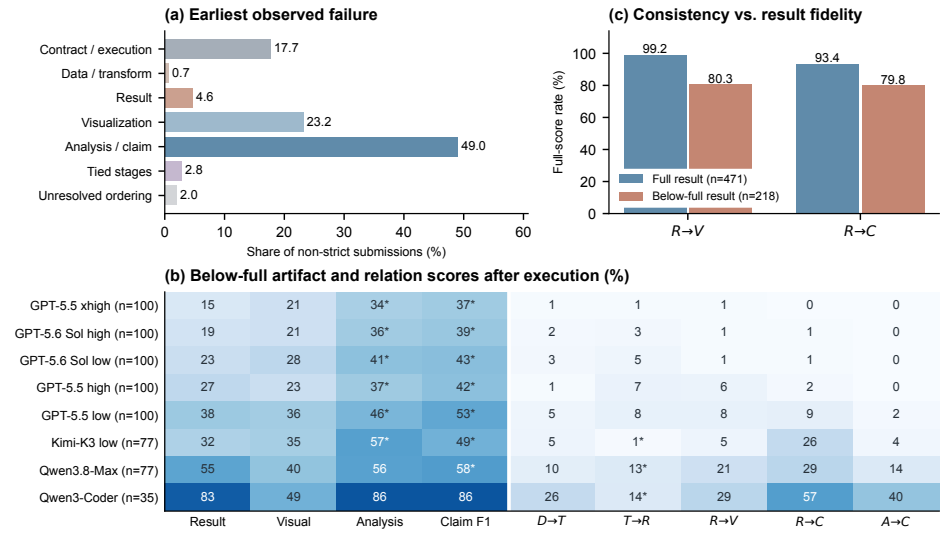


Figure 6: Six-artifact diagnostics. (a) Earliest observed failures among 604 non-strict submissions; ties and unverified ordering remain separate. (b) Below-full scores among executed submissions (counts in rows). Asterisks mark metric-specific exclusions or scores that cannot be determined. Both are omitted from cell denominators. (c) Local full-score rates across 689 executed submissions, grouped by Result fidelity. Appendix Q gives all counts and exclusions.

5.3 RQ3: Where does evidence support break down?

The batch contains 689 executed submissions. Excluding two cases with invalid reference findings leaves 784 Strict scores: 180 passes and 604 failures. Figure 6 locates failed checks and compares reference fidelity with local support.

Finding coverage and visual support dominate observed failures. Among 604 Strict failures, the earliest failed checks concern analysis/claims in 296 (49.0%), visualization in 140 (23.2%), and contract/execution in 107 (17.7%). Ties and unverified ordering remain separate (Figure 6a); these locations do not establish causes. Among 689 executed submissions, Result and Visual are below one in 31.6% and 29.6%, versus 10.9% for $R \to C$ (panel b). Among submissions with imperfect Result scores, 79.8% retain full $R \to C$ and 80.3% full $R \to V$ (panel c). Outputs can agree while missing reference requirements.

Expert agreement and evaluator failure detection. Two experts independently assessed support in 78 submissions, following PaperBench and FActScore (Starace et al., 2025; Min et al., 2023). They agreed on 97.65% of 980 result-to-claim and analysis-to-claim judgments ($\kappa = 0.798$). On 277 matched checks from the random sample, evaluator accuracy was 97.47%, with failure-detection precision, recall, and F1 of 100%, 30% (3/10), and 46.15%. The evaluator detected three contradictions but missed seven partially supported claims; the diagnostic sample added two misses. These involved unsupported details within otherwise supported claims. High expert agreement therefore does not establish reliable automated failure detection. The audit covers local support, not complete Path F1/Strict validation (Appendix K). Complexity analyses appear in Appendix P.

6 Conclusion

SciRigor connects scientific claims to the computations, results, figures, and analysis produced in the same run. Across eight configurations, 64 of 357 submissions with perfect final-claim scores have incomplete evidence support. Expert review confirms contradictions and identifies partially supported claims missed by automated checks. These findings motivate evaluating claim correctness and evidence support together, with claim-specific checks that locate missing or conflicting evidence.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and polishing, figure and table preparation, and consistency-checking code. The authors reviewed the AI-assisted materials and take responsibility for the final content.

ETHICS STATEMENT

SCIRIGOR evaluates scientific coding agents on specified analysis tasks. Passing these tasks does not certify autonomous scientific discovery or validate the source studies. Cases use publicly available articles, data, and author-released code. Public releases must preserve attribution and licensing terms. When redistribution is prohibited, releases should provide stable source references. The sampled disciplines reflect where code for individual figures and complete inputs are available. They do not represent all scientific practice. To help readers interpret scores within this scope, we report task requirements, output failures, and missing evidence alongside aggregate performance.

REPRODUCIBILITY STATEMENT

Every main-table score can be traced to individual submissions and evaluator decisions. The supplementary repository stores source locations, case file lists, input hashes, executable reference analyses, reference requirements, evaluator settings, and outputs for each case. It separates construction records from candidate execution records. Sections 3 and 4 specify the task interface, evidence graph, semantic matching, source selection, case construction, and validation. Section 5 reports the evaluation setup, metrics, and scope; the Appendix documents search and exclusion rules, reconstruction in clean environments, file integrity, domain assignments, and representative cases. Appendix K describes expert sampling, labeling rules, agreement, and the matching criteria for comparing expert and evaluator judgments. The validation summary gives sample sizes and confusion counts for the random and diagnostic groups separately.

REFERENCES

Kuangshi Ai, Haichao Miao, Kaiyuan Tang, Nathaniel Gorski, Jianxin Sun, Guoxi Liu, Helgi I. Ingolfsson, David Lenz, Hanqi Guo, Hongfeng Yu, Teja Leburu, Michael Molash, Bei Wang, Tom Peterka, Chaoli Wang, and Shusen Liu. Scivisagentbench: A benchmark for evaluating scientific data analysis and visualization agents. *IEEE Transactions on Visualization and Computer Graphics*, 2026. URL https://arxiv.org/abs/2603.29139. Accepted at IEEE VIS 2026.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f12b4df26344f3be803c06b555252efe-Abstract-Conference.html.

Zihan Deng, Chuanzhi Xu, Huiqi Liang, Haoyang Li, Xiaozhen Zhong, and Lequan Yu. Scifigqualbench: A benchmark for scientific figure quality assessment with full-manuscript context, 2026. URL https://arxiv.org/abs/2607.27084.

Xinshun Feng, Ziqi Miao, Lijun Li, and Jing Shao. Tracing the cascade: A topology-aware evaluation framework for scientific agent hallucinations, 2026. URL https://arxiv.org/abs/2608.00711.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, Yikun Zhang, Tianmai M. Zhang, Lanyi Zhu, Mike A. Merrill, Jeffrey Heer, and Tim Althoff. Blade: Benchmarking language model agents for data-driven science. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. URL https://aclanthology.org/2024.findings-emnlp.815/.

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. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/0d70af566e69f1dfb687791ecf955e28-Abstract-Conference.html.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56843–56873. PMLR, 2025. URL https://proceedings.mlr.press/v267/starace25a.html.

Liangcai Su, Zhaopeng Feng, Zhuo Chen, Zhen Zhang, Xiang Lin, Ruilin Li, Handuo Zhang, Ning Wang, Kailong Wen, Yueqi Guo, Feng Xing, Yiling Guo, Chenxiong Qian, Simon Shaolei Du, Lidong Bing, and Xinyu Wang. Frontierchallenge: Evaluating scientific workflow completion, 2026. URL https://arxiv.org/abs/2608.24979.

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?, 2026a. URL https://arxiv.org/abs/2606.24530.

Zhen Wang, Fan Bai, Zhongyan Luo, Jinyan Su, Kaiser Sun, Xinle Yu, Jieyuan Liu, Kun Zhou, Claire Cardie, Mark Dredze, Zhiting Hu, and Eric P. Xing. Fire-bench: Evaluating ai agents on the rediscovery of scientific insights. In *International Conference on Machine Learning*, 2026b. URL https://arxiv.org/abs/2602.02905.

Ling Yue, Chaoqian Ouyang, Hang Xu, Ruijun Huang, Yuchen Liu, Libin Zheng, Wei Liu, Shaowu Pan, Shimin Di, and Min-Ling Zhang. Factreview: Evidence-grounded peer review with execution-based claim verification, 2026. URL https://arxiv.org/abs/2604.04074.

Dan Zhang, Sining Zhoubian, Min Cai, Fengzu Li, Lekang Yang, Wei Wang, Tianjiao Dong, Ziniu Hu, Jie Tang, and Yisong Yue. Datascibench: An llm agent benchmark for data science. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. URL https://aclanthology.org/2026.findings-acl.181/.