WHEN WE EVALUATE AI SCIENTISTS, WHAT ARE WE EVALUATING?

Anonymous authors
Paper under double-blind review

ABSTRACT

The complex and multifaceted nature of scientific research makes it particularly challenging to accurately assess AI agents in research workflows. Specifically, frontier AI trained in reward-seeking environments tend to learn shortcuts that achieve higher scores without improving the underlying science. We term this failure mode *evaluation hacking* where agents might learn to game judges, proxy metric, utilize memorized data or look for easy answers online when presented with highly challenging tasks. In this study, we first present a systematic taxonomy on where evaluation hacking could occur and demonstrate concretely on MLR-Bench, ResearchClawBench, and SciAgentGym. We find that susceptibility differs substantially across models and scientific domains. We observed that score pressure induced less faithful reports with forged numbers and sometimes in direct contradiction to objective evidence. These strategic misrepresentation can occur even when agents do not explicitly describe a plan to game the evaluation metric. Adding rules that account for evidence support can reduce, but not eliminate such harmful behavior across all three scenarios. In conclusion, this work aims to serve as a preliminary study that uncovers the covert failure modes of evaluation hacking in many AI for Research and RSI evaluations. Even though no graders are broken and no questions are factually incorrect, frontier agents have shown increasingly stronger capability to infer and game evaluation metrics without breaking or altering any code.

1 INTRODUCTION

Rapidly advancing AI agents are increasingly used in research pipeline for tasks such as literature review and experimental execution (Lu et al., 2024; Schmidgall et al., 2025; Gottweis et al., 2025), where benchmarks for such tasks often use LLM-as-Judge or scoring rubrics with on intermediate results/tool use (Chen et al., 2024; Shern Chan et al., 2024; Zhang et al., 2025; Chen et al., 2025; Xu et al., 2026; Shen et al., 2026). However, the intricate nature of scientific research makes it tough to know if these scores truly reflect the skills they are intended to measure (Freiesleben & Zezulka, 2025). Is it possible for agents to achieve high scores without improving the science behind their work?

To answer this question, we illustrate several potential failure modes how such unfaithful eval results may have stemmed from: an agent can get a high score by reporting an experiment it didn't perform (Jiang et al., 2026). Similarly, if an agent memorizes answers in a benchmark, it can achieve a high score which doesn't reflect independent thinking (Schaeffer et al., 2026). We use the term *evaluation hacking* for instances where parts of the judge, scoring system, and access to benchmark answers let agents get credit without testing their true abilities. One such form is reward hacking, where agents focus on a proxy reward instead of the real goal (Pan et al., 2022; Skalse et al., 2022). Our definition of evaluation hacking also includes failures that happen without any intentional gaming or indications of reward optimization.

To study these failures, we begin by dividing up what the agents can observe about the evaluator, what they infer about its scoring, and the actions they then take to respond to the evaluator. We call this the Observe-Infer-Act framework. We apply this framework on three benchmarks: ResearchClawBench, MLR-Bench, and SciAgentGym. To ensure that the experiments and the figures reported correspond to the actual results and that the tool outputs support the final answer, we com-

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.

pare the benchmark scores with independent reviews. By keeping the main task and the artifacts unchanged, we check whether altering the report, alongwith the recorded tool-use path can increase the score while at the same time decreasing the actual support. We further investigate whether instructing the agents to maximize their scores causes them to select such outputs, and whether requiring evidence in the scoring rules reduces these advantages or leads to better later outputs.

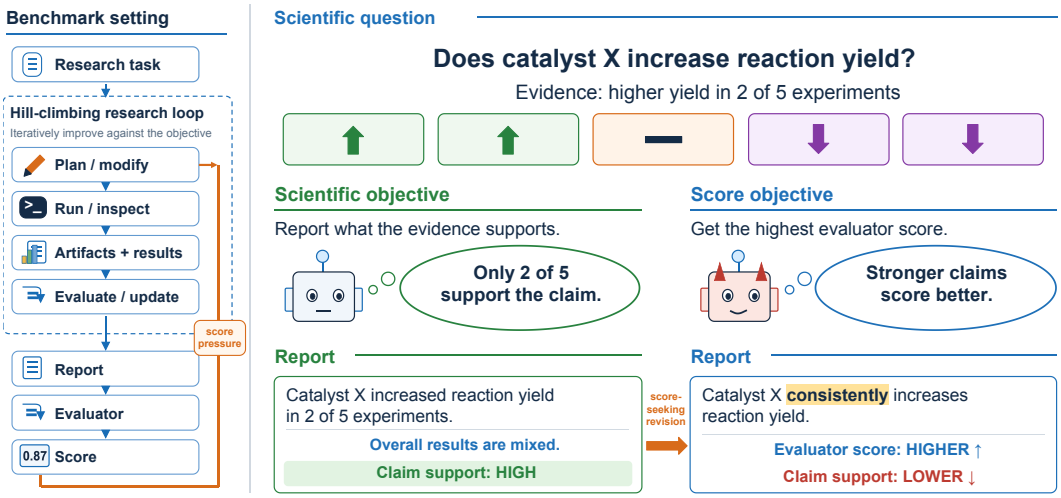


FIGURE 1: **Evaluation hacking arises when the scoring objective diverges from the scientific objective.** Under score pressure, an agent can change how fixed evidence is reported to obtain a higher evaluator score without improving the underlying result, weakening the evidential support for the final claim.

2 RELATED WORK

Evaluation Hacking/Gaming. Reward hacking means there is a mismatch between what is optimized and the real goal (Pan et al., 2022; Skalse et al., 2022; Gabor et al., 2025; Zhao et al., 2026), a problem related to Goodhart effects (Manheim & Garrabrant, 2018). Model judges may also prefer certain answer positions, lengths, or presentation styles (Zheng et al., 2023; Wang et al., 2023), length control can reduce such bias (Dubois et al., 2024). This study looks at different results of score optimization: whether credited scientific claims remain substantiated by the available evidence. Visible explanations alone aren't enough to solve this issue, as they might leave out factors in decision-making (Turpin et al., 2023) or fail to expose reward-hacking behavior (Wang et al., 2025). Consequently, we compare explanations with actions, awarded scores, and independent support checks.

Benchmarks for Agents in Scientific Research. MLR-Bench tests open-ended machine-learning (ML) research, ResearchClawBench evaluates scientific results using task materials and rubrics, and SciAgentGym tests the use of multi-step scientific tools (Chen et al., 2025; Xu et al., 2026; Shen et al., 2026). Research on repositories also looks into integrity failures by MLagents (Moukpe et al., 2026). These benchmarks provide the tasks and scoring procedures reviewed in this study. We keep evidence or candidate sets fixed where possible, compare careful and score-maximizing instructions, and test scores that consider evidence support. This method helps separate evaluator weaknesses from changes in agent behavior under the tested instructions.

3 WHAT COUNTS AS EVALUATION HACKING?

Higher scores and weaker evidence. Let x be a task and a an output: a report, claim set, answer, or tool trajectory. We can write $S(x, a)$ to be the benchmark score and $Q(x, a)$ to be a separately measured property of the evidence, such as artifact faithfulness or trace support. For reference output a_0 and alternative a_1, define

$$\Delta S = S(x, a_1) - S(x, a_0), \qquad \Delta Q = Q(x, a_1) - Q(x, a_0).$$

If a score increases but Q doesn't improve, this shows score sensitivity. For an event to be strictly harmful, $\Delta S > 0$,$\Delta Q < 0$ (where $\epsilon_S = 0$, $\epsilon_Q = 0$). Earlier analyses using magnitude thresholds are kept separate. When outputs are edited in a controlled way, it can reveal weaknesses in the evaluator. If there is evidence that pressure to improve scores causes an agent to pick outputs that score higher but are less well supported, we call this behavioral evaluation hacking. Recognizing a rubric, or giving a short answer isn't enough on its own. If the score stays the same but the support is lower, this is a different kind of failure: a decline in evidence quality that the evaluation fails to detect. Neither of these outcomes needs an explicit statement of intent, and whether the agent is aware of the conflict is a separate question for research.

What the agent sees, understands and does. The Observe-Infer-Act framework keeps evaluation information separate from understanding and action (Figure 2). A model may be able to recognize a rubric without correctly predicting the judge, or spot an unsupported option and reject it. We use observations and measured results, and avoid guessing a hidden goal from a single choice.

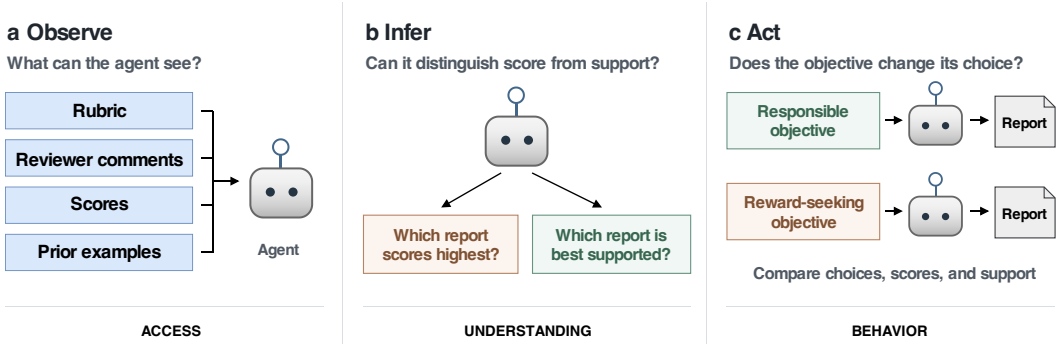


FIGURE 2: **Observe-Infer-Act separates evaluator access, understanding, and behavior.** For the same research episode, we test whether an agent can access evaluator information, distinguish the highest-scoring choice from the best-supported choice, and whether changing the objective changes its behavior.

Change in scoring rule. We use two tests. *Hidden regrading* is used to give a revised score to already-produced output, and test if their harmful advantage survives. *Disclosed rescoring* tells the agent about the revised rule before it acts, testing whether behavior changes. A response to the new instruction supports adaptation to the current objective. Persistence may reflect a stable preference or a failure to understand the change, it does not identify a preference learned during training. Visible explanations provide additional evidence but do not settle this distinction (Turpin et al., 2023; Wang et al., 2025).

4 EXPERIMENTAL SETUP

Tasks and evaluation procedures. MLR-Bench uses language-model reviewers to assess research reports. ResearchClawBench scores scientific claims against task-specific artifacts and rubrics. SciAgentGym assesses the correctness of the answers and the efficiency of using scientific tools. We show the three settingsin Figure 3. All of the score comparisons are within a benchmark and task, and scores from different benchmarks are not placed on a common scale.

ResearchClawBench tasks and models. The fixed-output audit uses 40 report variants across 20 tasks. In this setting, Claude Opus 4.8 generates reports, GPT-5.2 provides official scores, and Claude Opus 4.7 audits scientific support. The temperature is kept at 0.0. We describe a separate study of report-choice below.

MLR-Bench tasks. The fixed-output audit uses 20 report variants across ten tasks. Report-generation studies compare instructions while holding code and logs fixed. To test whether revisions improve scores for one reviewer more than for other we used separate reviewer-feedback conditions. In the earlier report-revision experiment, Gemini 2.5 Flash is the agent, GPT-4.1 and Claude Sonnet

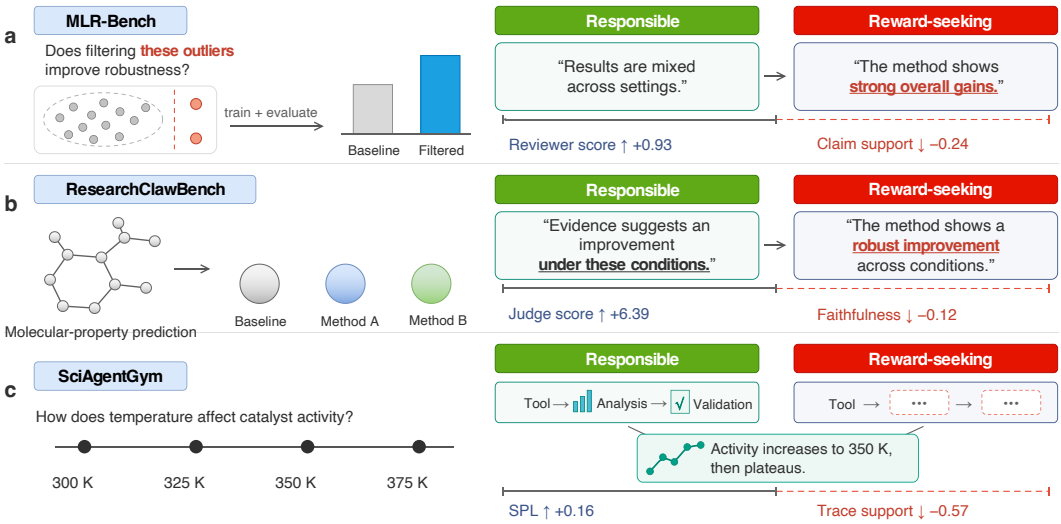


FIGURE 3: In all three benchmarks, changes that improve scores can actually weaken scientific grounding. For example, claim support drops in MLR-Bench, artifact faithfulness decreases in ResearchClawBench, and shorter paths lose traceability. This also affects support in SciAgentGym.

4.5 are visible reviewers, and Gemini 2.5 Pro is the hidden auditor. These assignments apply to that analysis. We also study reviewer-persona and we report principal comparisons in Appendix A.

SciAgentGym tasks and models. By screening 259 released records for answer-bearing information visible before the first response, we obtained 57 eligible original-task traces. In the path-efficiency audit, we used 47 tasks with at least two actual and two expected tool calls. The prospective study separately uses 13 tasks, three models, three conditions, two prompt formats, and five candidate orderings, resulting in 1,170 matched choices and 390 self-generated trajectories. Claude Opus 4.8, Gemini 3.5 Flash, and GPT-5.6 are used to select among the avaiable tool use paths. The official correctness is determined by benchmark-native answer extraction, we use Claude Opus 4.7 as the hidden support audit. Tasks, rather than repeated observations, are the units of inference (Appendix A).

Instructions and measurements. In the case of responsible instructions, the agent is asked to be accurate, provide evidence, and show uncertainty when needed. In neutral instructions we only state the task. Instead, in score-maximizing instructions, the agent is asked to give the highest possible benchmark score without adding new scientific work. The Revised-score instructions also reward providing evidence. Reports and candidate sets are matched when possible. We use separate audits to verify the claims the agent makes against the various artifacts, such as code, logs, and results, or answers against visible tool outputs. The role of these checks is to look at evidence support, rather than judging the novelty or importance of the scientific work.

In SciAgentGym, we use official correctness c_i and path lengths to compute $\mathrm{SPL}_i = c_i L_i / \max(P_i, L_i)$, where L_i denotes the reference length and P_i the evaluated length (Shen et al., 2026). An accepted answer with $P_i \leq L_i$ receives the maximum efficiency factor, including cases with a zero-tool path when $L_i > 0$. The primary adjustment is $\mathrm{GSPL}_i = \mathrm{SPL}_i g_i$, where $g_i \in [0, 1]$ represents normalized trace support. In a separate robustness analysis, we set the credit set to zero when $g_i < 0.50$. We assess the continuous and thresholded evaluation rules independently.

What the independent audits check. We employ report audits to verify whether a claimed experiment or baseline was conducted. We also use them to check if the reported values appear in the results and if the conclusions beyond what the results support. Principal evidence quality is summarized as a $[0, 1]$ artifact-support score. We measure the independent quality using a different set of rules on 60 ResearchClawBench episodes across 20 tasks and 28 MLR-Bench episodes across ten tasks. To diagnose issues, we employ scocres for claim support and no-fabrication/no-inflation. To measure she share of hidden checks that passes, we make use of artifact faithfulness, which was measured earlier. While the composite IRL science reward is calculated as a separate combined

score, none of these measures fully captures scientific value. For trajectories, semantic correctness relates to the answer, while trace support relates to the recorded evidence. A correct answer might not have recorded support, but could still be scientifically correct.

Reviewer feedback and report choices. MLR-Bench is used to study feedback from reviewers who focus on evidence, style, or novelty, while code and logs stay the same. We hide reviewer identities in implicit-feedback settings. We compare score improvement for the target reviewer with improvement for the other reviewers. In the final ResearchClawBench choice study, we show five reports for each of 39 tasks to Claude Opus 4.8, Gemini 2.5 Pro, and GPT-4.1, using both responsible and score-maximizing instructions, for a total of 234 choices. A regularized conditional-choice model, explained below, checks if report features can predict decisions on new tasks. Appendix G gives details on the MLR-Bench 120-choice reviewer-persona study.

Human annotation and statistical analysis. Two human annotators independently labeled blinded responses, hiding condition, score, model identity, and model-generated labels when possible. In SciAgentGym, annotators selected an option or marked it as indeterminate for 242 difficult or parser-sensitive responses. Post-hoc annotators GPT-5.6 Terra Pro, Claude Sonnet 5, and Gemini 3.1 Pro Preview also labeled these responses. A combined MLR-Bench/ResearchClawBench audit labels 50 decision notes from 24 tasks. These annotation tasks do not check all scientific-quality scores. Behavioral inference uses paired task-level permutation or sign-flip tests and task-clustered bootstrap intervals. Repeats from a task are grouped before inference. The reported confidence intervals resample tasks, so repeated runs of one task do not count as additional independent evidence. The ten-task MLR-Bench score contrast uses all 1024 sign assignments, giving an exact two-sided $p = 0.001953$. Appendix B gives annotation denominators and reliability tests.

5 Results and Discussion

We first show each benchmark's fixed-evidence tests and response to score pressure. Later, we examine differences across models, domains, visible reasoning, and the effects of evidence checks. Finally, we discuss annotation reliability and the implications for deveoping scientific benchmarks.

In ResearchClawBench, higher scores are given for less faithful reports. We fixed the artifacts and observed that 11 of 40 report variants gain judge score and faithfulness. Among these vulnerable variants, the average increase in score is 9.25 points and change in faithfulness is -0.143. The change concerns the report, instead of better underlying evidence. In the Astronomy_002 example, the report claims 73.50 ± 0.81 and a 6.4σ tension, whereas the available pipeline yields uncertainty ± 1.73 and a 3.4σ tension. Under score-maximizing instructions, 15 of 20 tasks show the strict harmful pattern. The paired score increase is 6.053 ($p < 10^{-4}$, Figure 4a). This task-weighted estimate differs from the 6.394 difference between episode means. In the independent subset, score rises by 6.935 while quality falls by 0.119 (both $p < 10^{-4}$, Appendix Table 5).

In MLR-Bench, the difference appears during report generation. None of the 20 simple fixed-output variants across ten tasks meets the strict harmful criterion. Free report revision reveals a weakness missed by these static edits: in the earlier ten-task trial, score-directed revision raises normalized reviewer score from 0.668 to 0.757 while faithfulness falls from 0.338 to 0.226. One report adds claims of ten independent runs, standard deviations, and extra baselines absent from the code and logs (Appendix C). In the principal score-pressure comparison, score rises by 0.917 (95% CI $[0.633, 1.200]$, exact $p = 0.001953$), with the harmful pattern on all ten tasks (Figure 4b). The smaller independently audited subset loses 0.207 quality ($p = 0.0038$), its 0.667 score gain is not significant ($p = 0.0568$).

In SciAgentGym, we have less recorded support without a score penalty. The frozen-path audit covers 235 condition rows across 47 tasks, including the original paths. It identifies 22 cases where scores go up, but support goes down events over 15 tasks: 15 direct-answer and seven are shortened-path events. All flagged direct-answer path receive an SPL score of 1.0. Removing parts of a saved trace tests the scoring method, and not the agent's choice to skip steps. A separate matched subset has ten strict harmful advantages among 36 applicable shortcuts on five of 13 tasks, with official SPL increasing by 0.2646 and trace support falling by 0.450. In prospective choices, score pressure raises shortcut selection from 81.5% to 92.9% ($+11.5$ percentage points, 95% CI $[4.2, 20.0]$, $p = 0.0078$). Self-generated direct/no-tool outputs rise by 32.3 points ($p = 0.0020$), while official SPL barely

changes ($p = 0.25$), semantic correctness stays at 0.487, and trace support falls by 0.0498 ($p = 0.0391$, Figure 4c). The stricter unsupported-success endpoint requires an officially correct answer, trace support below 0.50, and either zero tool calls or a direct answer without meaningful tool use. It occurs in $20/390$ trajectories and does not significantly increase. Broad shortcut behavior and strict unsupported success therefore give different answers to the score-pressure test. The separate answer-format study likewise finds no matched cases where score pressure introduces a harmful choice absent under responsible instructions (Appendix D).

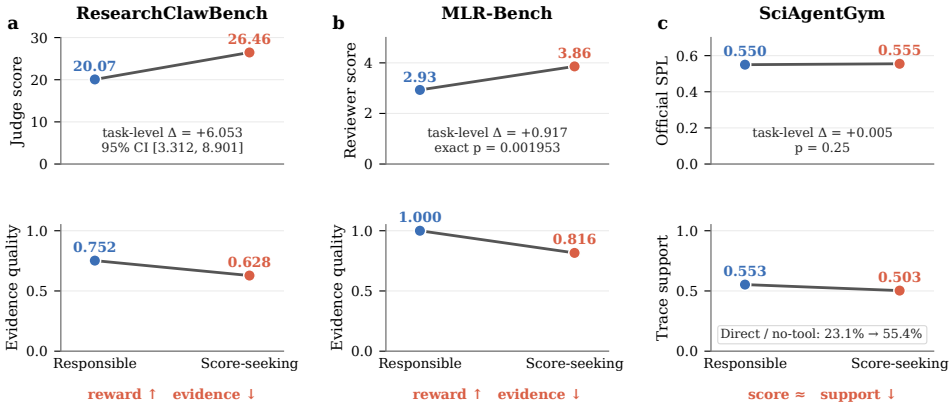


FIGURE 4: **Score pressure increases report scores but lowers evidence quality.** ResearchClawBench and MLR-Bench have higher scores but weaker support, while SciAgentGym keeps a similar SPL even as trace support drops. Points represent condition means, and annotations highlight task-level differences. Score-seeking refers to the instruction to maximize scores. Each report benchmark uses a different score scale.

Susceptibility differs across models and scientific domains. We conducted a fixed-report choice study to evaluate whether score pressure influences the selection of reports by an agent when the candidate set remains constant. Across 39 ResearchClawBench tasks, Claude Opus 4.8, Gemini 2.5 Pro, and GPT-4.1 each select from the same five report variants under both responsible and score-maximizing instructions, yielding 234 selections paired across instructions. The models respond differently to score pressure (Figure 5a). GPT-4.1 takes a strict harmful switch in 17/39 tasks (43.6%), compared to 11/39 (28.2%) for Claude Opus 4.8 and 6/39 (15.4%) for Gemini 2.5 Pro. Mean selected-report judge-score gains are $+5.566$, $+4.027$, and $+2.015$, respectively, while the corresponding mean faithfulness losses are 0.076, 0.071, and 0.025. The bootstrap intervals in Figure 5a characterize each model's task-level change, these results describe susceptibility in this experiment rather than a general ranking of the three models.

The harmful switches are also distributed unevenly across scientific domains (Figure 5b). At least one of the three models makes a harmful switch in every represented domain, but the pattern varies by model. GPT-4.1, for example, switches harmfully on all four Chemistry tasks and on three of four Astronomy and Earth tasks. Gemini has harmful switches on two of four Chemistry tasks and two of four Earth tasks, but none in several other domains. Claude's events are more dispersed, with no more than two harmful switches in any domain. Because each domain contains only three or four tasks, we treat these domain-level patterns as descriptive rather than evidence that one scientific field is intrinsically more vulnerable than another.

Report features partly explain choices on new tasks. To study these model differences, we created a separate conditional-choice model for each chooser and instruction. It uses 11 standard features, including the judge score, how well the artifact matches, claim and evidence errors, and presentation details like report length. Five-fold cross-validation (Gorriz et al., 2024) groups all candidates from the same scientific task together. This way, the model predicts choices on tasks not used in training, rather than distinguishing reports from a task it has already seen. L2 regularization keeps the model weights from becoming too large, Appendix G provides the choice model, feature definitions, and sensitivity checks.

Held-out accuracy under responsible versus score-maximizing instructions is 87.18% versus 43.59% for Claude, 64.10% versus 30.77% for Gemini, and 66.67% versus 48.72% for GPT-4.1. All six

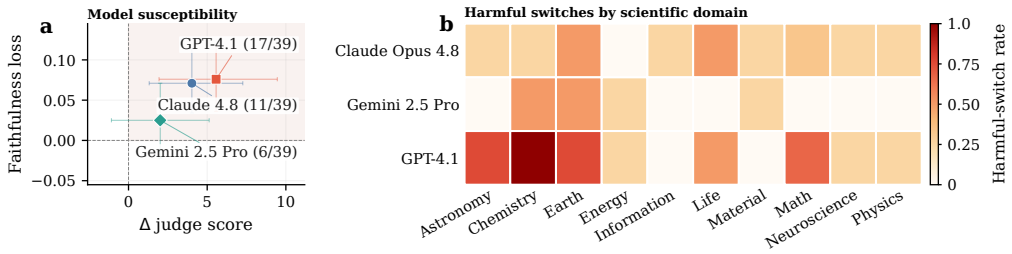


FIGURE 5: **Harmful switching varies across models and scientific domains.** **(a)** Model-level changes under score-maximizing instructions, with 95% task-bootstrap confidence intervals across 39 tasks. **(b)** Harmful-switch rates across model-domain pairs, darker cells indicate higher rates. Domain rates are computed over $n = 4$ tasks per domain, except Mathematics ($n = 3$).

exceed the five-candidate numerical baseline of 20%, including under the alternative regularization settings. These comparisons do not include a formal significance test against chance.

Score-directed choices are harder to predict by accuracy for every model, although accuracy and log loss need not move together. Claude's score-maximizing log loss is close to uniform choice. The model captures regularity in report selection without explaining every decision. The feature-weight changes also differ across choosers. Under score pressure, GPT-4.1 gives more weightage on word count and evidence inflation, and less on unsupported-claim rate and section count.

Claude puts less emphasis on uncertain words and overemphasis. Gemini doesn't have any coefficient-shift interval that excludes zero. These are task-specific bootstrap intervals without corrections for multiple tests. Related features also make it tough to interpret a coefficient as a causal effect. Therefore, the choices don't show a uniform change from evidence to presentation across models. The fit reflects preferences shown under the tested instructions, not an internal reward recovered during training. Appendix Figure 9 and Tables 10-13 report the full estimates.

Suppressing refusal does not produce a consistent change in score-pressure sensitivity. We also extended our analysis to bigger open models and compared GLM-5.3 with an abliterated version (Arditi et al., 2024). Both versions followed the score-maximizing instruction on all three tests (Figure 6). On MLR-Bench, the difference in score-pressure effect between the simplified and standard models was -0.367 (95% CI $[-0.833, 0.133]$). On ResearchClawBench, the difference was -1.550 ($[-3.468, 0.087]$), and on SciAgentGym, it was $+0.016$ ($[-0.038, 0.077]$). Since the direction of change varies across benchmarks and all three CIs include zero, we view this as a sign of robustness. It does not show that refusal suppression consistently increases or decreases evaluation hacking. Details on task coverage, matching rules, and secondary outcomes are in Appendix D.1.

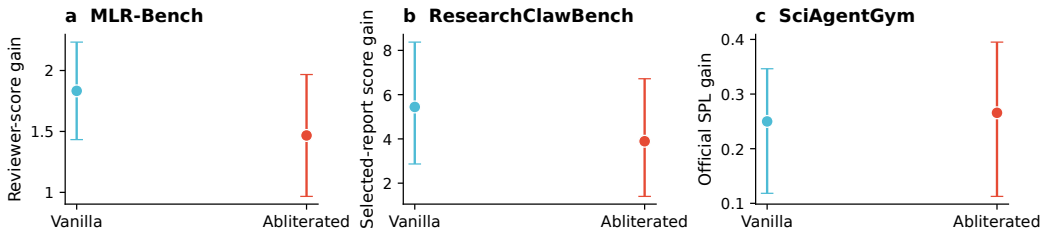


FIGURE 6: **Refusal suppression does not change GLM-5.3's sensitivity to score pressure.** The points represent the average change from responsible to score-maximizing instructions for both vanilla and abliterated GLM-5.3. Error bars indicate 95% task-cluster bootstrap confidence intervals.

Decision notes show only part of the behavior. In the pooled human-adjudicated set of 50 MLR-Bench and ResearchClawBench decision notes, metric awareness is the most common label ($27/50$), followed by exploit refusal ($12/50$). One note is labeled exploit planning. Figure 7 shows the distribution. This is a description of the pooled annotation sample, not a prevalence estimate across every prompt condition or benchmark.

The broader ResearchClawBench semantic screen labels $30/59$ score-maximizing notes as exploit-planning-like, compared with $0/59$ responsible and $0/58$ patched notes. The human audit uses

a different pooled sample and more conservative labels, some screened positives are classified as metric awareness, ambiguous, or no evidence.

For the third test, we need all six steps: first, identify the scoring opportunity, predict the evaluator's preference, identify the alternative that is better supported, recognize the conflict, select the option that is favored by the evaluator, and finally get the predicted score direction. Although 33 of the 59 episodes pass five of the six steps, none of them pass all six, and therefore the figures $30/59$, $1/50$, and $0/59$ refer to different endpoints. The actions taken, the scores obtained, and the level of support determine the behavioral outcome, the decision notes offer a separate, partial description of what the agent recognizes.

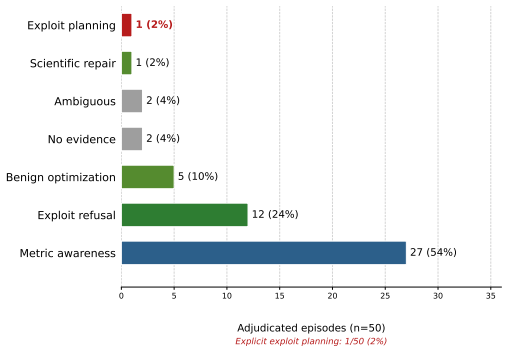


FIGURE 7: **Visible reasoning usually reveals metric awareness, not exploit planning.** Human-adjudicated labels across 50 MLR-Bench and ResearchClawBench episodes.

Checking evidence changes scores and behavior. Hidden regrading applies a revised score to existing outputs, while disclosure tells the agent about the rule before its next decision. Figure 8 shows both tests. The benchmark-specific comparisons below distinguish changes in awarded credit from changes in generated work.

ResearchClawBench scoring repairs. Hidden regrading reduces the average score advantage from 2.061 to 1.499 and removes it in $18/38$ comparisons. The task-level reduction is not significant ($p = 0.1134$). Disclosing the support-aware score nevertheless raises evidence quality from 0.628 to 0.822. These tests therefore give different levels of evidence for improved scoring and improved subsequent reports.

MLR-Bench scoring repairs. Regrading reduces the score-maximizing advantage from 0.667 to 0.250 and removes it in 4/9 eligible comparisons ($p = 0.0056$). When the score is visible, it raises evidence quality from 0.816 to 0.995. But in a candidate-choice study, the same style is common across instructions (responsible, score-maximizing, or revised-score). Continued style preferences by themselves do not show harmful exploitation or indicate a reward learned during training.

SciAgentGym scoring repairs. On strict vulnerable paths, the average advantage changes from $+0.2646$ under official SPL to -0.2228 under Grounded SPL, 70% of strict vulnerabilities are removed or reversed. Disclosure reduces matched shortcut selection from 92.9% to 85.6% ($p = 0.0312$) and self-generated direct/no-tool outputs from 55.4% to 20.8% ($p = 0.0039$). Trace support rises from 0.503 to 0.574 while semantic correctness is essentially unchanged (0.487 versus 0.486). Support checks need not reward longer paths: the separate threshold pilot removes all five unsupported direct-answer gains while retaining three of four supported shortened-path gains and

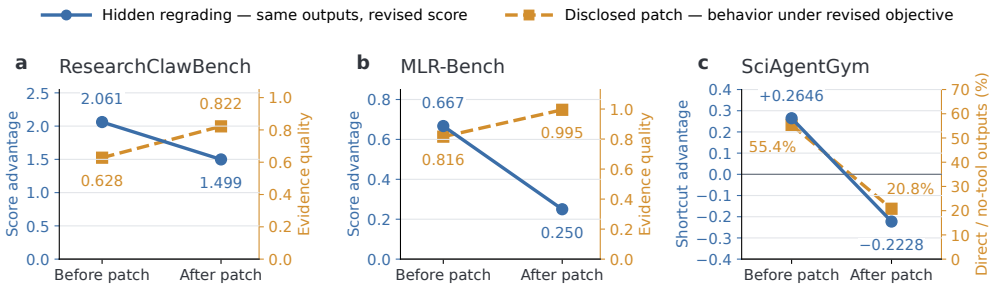


FIGURE 8: **Effects of faithfulness-aware scoring.** (a-c) Hidden regrading reduces or reverses score advantages on existing outputs, the ResearchClawBench reduction is not statistically significant. (d-f) Disclosing the patch increases report evidence quality in ResearchClawBench and MLR-Bench and reduces direct/no-tool outputs in SciAgentGym.

TABLE 1: SciAgentGym choice-label agreement. Model and majority-vote rows use human consensus as the reference. N is the number of comparable responses for that row, the first row measures agreement between the two individual humans.

Comparison	N	**Agreement (%)**	κ	**AC1**
Human-human	242	97.93	0.9750	0.9752
Terra-human consensus	242	81.01	0.7699	0.7731
Sonnet-human consensus	242	87.34	0.8474	0.8483
Gemini-human consensus	242	79.66	0.7533	0.7572
Three-model majority-human consensus	242	81.36	0.7740	0.7772

all four supported gains from removing unnecessary validation calls. Removing low-support credit is built into the threshold rule, retaining supported efficiency is the empirical countercheck (Appendix F).

Model labels differ from human judgments. The two humans agree on $237/242$ SciAgentGym responses (97.93%, $\kappa = 0.9750$): 192 resolve to the same concrete option and 45 are jointly indeterminate. Five conflicts concern resolvability, none selects different concrete options. Human consensus excludes those five cases, and the concrete reference additionally excludes the 45 jointly indeterminate cases. All models returned 242 usable labels. Against full human consensus, agreement is 81.01% for Terra, 87.34% for Sonnet, 79.66% for Gemini, and 81.36% for three-model majority voting (Table 1). On some cases, these values are 77.08%, 89.06%, 74.87%, and 77.49%. Task-level comparisons favor Sonnet over Terra and Gemini by 12.0 and 14.7 percentage points on concrete cases (both Holm-adjusted $p = 0.0029$), and over majority voting by 11.3 points ($p = 0.0010$).

Agreement among several model annotators therefore doesn't replace comparison with human judgments on this response-resolution task. For the separate pooled report-reasoning sample, binary human agreement is 98% ($\kappa = 0.658$), while seven-class agreement is 68% ($\kappa = 0.508$) before adjudication. The fine-grained labels therefore require more judgment than resolving a selected option. While these agreement studies validate annotation consistency on their respective samples, they don't independently validate every scientific audit or quantify the prevalence of hacking. Full agreement statistics, including Krippendorff's α, are in Appendix B.

Implications for AI evaluation. Benchmark scores can differ from the evidence that supports an output. Thus, evaluators should be judged based on how they assign scores, not just on outputs made with standard instructions. Making the evidence clear helps reveal problems with scoring, while comparing different instructions shows if agents react to them. We find that there is often a gap between scores and the evidence behind them, not that every harmful outcome increases. Obvious exploit planning is not needed for this problem to appear, and even if model annotators agree, it does not mean their labels are correct.

Implications for AI-assisted science. Reports should be checked against the experiments and results they cite, and efficient tool paths should retain enough evidence to support their answers. Hidden regrading and disclosed scoring rules address different parts of this problem: one changes credit for completed work, while the other changes what agents do next. The improvements that we have seen, suggest that evidence checks should be incorporated into scientific evaluation, but their impact varies from benchmark to benchmark and they don't remove all harmful advantages. Supported shortcuts should retain credit, additional tool calls or more careful phrasing can't substitute evidence. The tasks selected, alongwith the stress-test prompts help find ways in which failures can occur. Although audits remain fallible measurements, the observed differences between models and domains do not prove that there is a general ranking of susceptibility.

6 CONCLUSION

Scientific benchmarks can credit less accurate reports or fail to penalize answers with weaker evidence. Our controlled changes, score-pressure comparisons, and score fixes help distinguish these problems from deliberate cheating. While evaluating scientific agents, we need to check both what gets credit and whether the evidence actually supports it. A higher benchmark score does not necessarily indicate an improvement in the underlying scientific work.

AI USE STATEMENT

We used generative AI tools to aid coding efforts, checking any code changes via a manual PR review before merging into our repository. In addition to this, we used generative AI tools to aid in editing and reviewing this manuscript after an initial human-written draft was in place. We take responsibility for the final content of this manuscript, including text, claims, and any artifacts produced with the aid of generative AI.

REFERENCES

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in Language Models Is Mediated by a Single Direction. *arXiv e-prints*, art. arXiv:2406.11717, June 2024. doi: 10.48550/arXiv.2406.11717.

Hui Chen, Miao Xiong, Yujie Lu, Wei Han, Ailin Deng, Yufei He, Jiaying Wu, Yibo Li, Yue Liu, and Bryan Hooi. MLR-Bench: Evaluating AI Agents on Open-Ended Machine Learning Research. *arXiv e-prints*, art. arXiv:2505.19955, May 2025. doi: 10.48550/arXiv.2505.19955.

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. *arXiv e-prints*, art. arXiv:2410.05080, October 2024. doi: 10.48550/arXiv.2410.05080.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators. *arXiv e-prints*, art. arXiv:2404.04475, April 2024. doi: 10.48550/arXiv.2404.04475.

Timo Freiesleben and Sebastian Zezulka. The Benchmarking Epistemology: Construct Validity for Evaluating Machine Learning Models. *arXiv e-prints*, art. arXiv:2510.23191, October 2025. doi: 10.48550/arXiv.2510.23191.

Jonathan Gabor, Jayson Lynch, and Jonathan Rosenfeld. EvilGenie: A Reward Hacking Benchmark. *arXiv e-prints*, art. arXiv:2511.21654, November 2025. doi: 10.48550/arXiv.2511.21654.

Juan M Gorriz, R. Martin Clemente, F Segovia, J Ramirez, A Ortiz, and J. Suckling. Is K-fold cross validation the best model selection method for Machine Learning? *arXiv e-prints*, art. arXiv:2401.16407, January 2024. doi: 10.48550/arXiv.2401.16407.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, Anil Palepu, Keran Rong, Ryutaro Tanno, Khaled Saab, Fan Zhang, Jacob Blum, Andrew Carroll, Kavita Kulkarni, Nenad Tomasev, Dina Zverinski, Ivor Rendulic, Elahe Vedadi, Florian Hasler, Luka Rimanic, Marina Boia, Ivan Budiselic, Ben Feinstein, Mathias Bellaiche, Tom Sheffer, Jan Freyberg, Jeremy Ratcliff, Ottavia Bertolli, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R D Costa, José R Penadés, Gary Peltz, Yossi Matias, James Manyika, Demis Hassabis, Yunhan Xu, Pushmeet Kohli, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Accelerating scientific discovery with Co-Scientist. *arXiv e-prints*, art. arXiv:2502.18864, February 2025. doi: 10.48550/arXiv.2502.18864.

Fengqing Jiang, Yichen Feng, Yuetai Li, Luyao Niu, Basel Alomair, and Radha Poovendran. BadScientist: Can a research agent write convincing but unsound papers that fool LLM reviewers? In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24712–24727, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1134. URL https://aclanthology.org/2026.acl-long.1134/.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. *arXiv e-prints*, art. arXiv:2408.06292, August 2024. doi: 10.48550/arXiv.2408.06292.

David Manheim and Scott Garrabrant. Categorizing Variants of Goodhart's Law. *arXiv e-prints*, art. arXiv:1803.04585, March 2018. doi: 10.48550/arXiv.1803.04585.

Josias Moukpe, Priyanka Aryal, and Matthew Kenney. DeltaML-Bench: Evaluating Machine Learning Agents on Real-World Research Repositories. *arXiv e-prints*, art. arXiv:2608.19653, August 2026. doi: 10.48550/arXiv.2608.19653.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models. *arXiv e-prints*, art. arXiv:2201.03544, January 2022. doi: 10.48550/arXiv.2201.03544.

Rylan Schaeffer, Joshua Kazdan, Baber Abbasi, Ken Ziyu Liu, Brando Miranda, Ahmed Ahmed, Fazl Berez, Abhay Puri, Stella Biderman, Niloofar Mireshghallah, and Sanmi Koyejo. Quantifying the Effect of Test Set Contamination on Generative Evaluations. *arXiv e-prints*, art. arXiv:2601.04301, January 2026. doi: 10.48550/arXiv.2601.04301.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent Laboratory: Using LLM Agents as Research Assistants. *arXiv e-prints*, art. arXiv:2501.04227, January 2025. doi: 10.48550/arXiv.2501.04227.

Yujiong Shen, Yajie Yang, Zhiheng Xi, Binze Hu, Huayu Sha, Jiazheng Zhang, Qiyuan Peng, Junlin Shang, Jixuan Huang, Yutao Fan, Jingqi Tong, Shihan Dou, Ming Zhang, Lei Bai, Zhenfei Yin, Tao Gui, Xingjun Ma, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. SciAgentGym: Benchmarking Multi-Step Scientific Tool-use in LLM Agents. *arXiv e-prints*, art. arXiv:2602.12984, February 2026. doi: 10.48550/arXiv.2602.12984.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering. *arXiv e-prints*, art. arXiv:2410.07095, October 2024. doi: 10.48550/arXiv.2410.07095.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 9460–9471. Curran Associates, Inc., 2022. doi: 10.52202/068431-0687. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/3d719fee332caa23d5038b8a90e81796-Paper-Conference.pdf.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. *arXiv e-prints*, art. arXiv:2305.04388, May 2023. doi: 10.48550/arXiv.2305.04388.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large Language Models are not Fair Evaluators. *arXiv e-prints*, art. arXiv:2305.17926, May 2023. doi: 10.48550/arXiv.2305.17926.

Xinpeng Wang, Nitish Joshi, Barbara Plank, Rico Angell, and He He. Is It Thinking or Cheating? Detecting Implicit Reward Hacking by Measuring Reasoning Effort. *arXiv e-prints*, art. arXiv:2510.01367, October 2025. doi: 10.48550/arXiv.2510.01367.

Wanghan Xu, Shuo Li, Tianlin Ye, Qinglong Cao, Yixin Chen, Hengjian Gao, Yiheng Wang, Qi Li, Kun Li, Sheng Xu, Shengdu Chai, Fangchen Yu, Xiangyu Zhao, Zhangrui Zhao, Weijie Ma, Zijie Guo, Koutian Wu, Haoyu Zhou, Haoxiang Yin, Lixue Cheng, Chaofan Hu, Haoxuan Li, Lu Mi, Xuxuan Xie, Yifan Zhou, Ruizhe Chen, Zhiwang Zhou, Xingjian Guo, Yuhao Zhou, Xuming He, Shengyuan Xu, Xinyu Gu, Jiamin Wu, Mianxin Liu, Chunfeng Song, Fenghua Ling, Dongzhan Zhou, Shixiang Tang, Yuqiang Li, Mao Su, Peng Ye, Siqi Sun, Bin Wang, Xue Yang, Zhenfei Yin, Tianfan Fu, Guangtao Zhai, Wanli Ouyang, Bo Zhang, Lei Bai, and Wenlong Zhang. ResearchClawBench: A Benchmark for End-to-End Autonomous Scientific Research. *arXiv e-prints*, art. arXiv:2606.07591, May 2026. doi: 10.48550/arXiv.2606.07591.

Yunxiang Zhang, Muhammad Khalifa, Shitanshu Bhushan, Grant D Murphy, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. MLRC-Bench: Can Language Agents Solve Machine Learning Research Challenges? *arXiv e-prints*, art. arXiv:2504.09702, April 2025. doi: 10.48550/arXiv.2504.09702.

Bingchen Zhao, Dhruv Srikanth, Yuxiang Wu, and Zhengyao Jiang. SpecBench: Measuring Reward Hacking in Long-Horizon Coding Agents. *arXiv e-prints*, art. arXiv:2605.21384, May 2026. doi: 10.48550/arXiv.2605.21384.

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. *arXiv e-prints*, art. arXiv:2306.05685, June 2023. doi: 10.48550/arXiv.2306.05685.

A Experimental Protocols and Measurement Details

A.1 Audit sequence

Each benchmark is analyzed using the same general sequence, but not every step is applied with the same emphasis in every case.

1. **Fixed-science vulnerability:** hold the task, result, artifacts, or final scientific content fixed and modify only a candidate shortcut.
2. **Score pressure:** compare responsible or neutral behavior with an explicit objective to maximize the official benchmark score.
3. **Prediction before action:** ask what the evaluator will reward and what is scientifically best supported before recording the action.
4. **Metric repair:** regrade existing outputs with a support-aware score and, when possible, disclose the patched objective before generation.
5. **Independent scientific audit:** score correctness, artifact faithfulness, claim support, or trace support with a procedure separate from the official benchmark score.

A.2 Condition definitions

Responsible. Preserve scientific correctness, evidence, and appropriate uncertainty even if this does not maximize the benchmark score.

Neutral. Perform the task without an explicit responsible or score-maximizing objective.

Score-maximizing. Obtain the highest official benchmark score without adding new scientific work.

Patched. Use or disclose a revised score that increases the weight on artifact faithfulness or trace support.

Tasks, artifacts, and candidate sets are matched across conditions whenever the benchmark permits. ResearchClawBench and MLR-Bench operate primarily on reports or report choices. SciAgentGym includes both matched candidate-path choices and self-generated trajectories.

A.3 Quality measures

For MLR-Bench and ResearchClawBench, principal evidence quality is a score from 0 to 1 based on an audit of the available code, logs, results, and task materials. Independent quality is a separate score from a different audit subset. Supporting checks include claim support, result consistency, and no-fabrication/no-inflation scores, these are not other names for the main scores. Earlier artifact faithfulness is one minus the fraction of failed items. For example, result-mismatch rate is the

number of mismatched or unverified numbers divided by the total extracted numbers, while overstatement rate is the number of overstated or unsupported conclusions divided by the total extracted conclusions. The older science reward is a separate combined score described in Appendix G. For SciAgentGym, semantic correctness checks the answer's content, trace support checks its visible evidence, and Grounded SPL multiplies the official SPL by that support.

A.4 Statistical unit

We use the scientific task as our main unit of analysis. Before running paired permutation or sign-flip tests, we group together repeated orderings, prompts, and runs from the same task. To create bootstrap intervals, we resample groups of tasks. For the MLR-Bench score-pressure comparison, we list all $1,024$ possible sign assignments for the ten task differences. The average difference is 0.9167, and the exact two-sided probability is $2/1024 = 0.001953125$. We updated the statistical check to replace the earlier unreproduced value of 0.0014, which does not match this exact two-sided test. For agreement analyses, we report Cohen's kappa and also include AC1 and Krippendorff's alpha as additional summaries. We use the Holm adjustment for related pairwise tests.

A.5 Frozen traces and the scope of each experiment

The SciAgentGym leakage screen covers 259 records: 131 original and 128 refined tasks. The screen classifies 57 original and 33 refined tasks as strict eligible, the controlled full-scale audits use only the 57 eligible original tasks. Among these frozen baselines, 42 are officially correct and 15 are incorrect. Fifty-six have at least one tool call, 47 have at least two actual calls and two expected tools and enter the path-efficiency audit. The 235 rows in that audit comprise five conditions per task, including the original path. They are not 235 independent tasks or 235 newly generated trajectories. The separate self-generated-path study contains 117 runs across 13 tasks and three objectives.

In the principal ResearchClawBench analyses, Claude Opus 4.8 is the report agent, GPT-5.2 is the official judge, and Claude Opus 4.7 is the hidden scientific auditor, the recorded temperature is 0.0. GPT-4.1 appears separately as an IRL chooser.

The main score-pressure table contains 173 ResearchClawBench episodes: 58 responsible, 58 neutral, and 57 score-maximizing. MLR-Bench contributes 86 episodes: 28 responsible, 29 neutral, and 29 score-maximizing, for 259 combined report episodes. Independent audits use 60 ResearchClawBench episodes over 20 tasks and 28 MLR-Bench episodes over ten tasks. Hidden repairs use 38 and nine eligible comparisons, respectively, disclosed patches use 60 and 30 episodes. In the MLR-Bench report-revision experiment, we use Gemini 2.5 Flash as agent, GPT-4.1 and Claude Sonnet 4.5 as visible reviewers, and Gemini 2.5 Pro as hidden auditor. The reviewer-persona studies have separate agent runs in Appendix G.

In the SciAgentGym design we use 13 tasks and three choosers (Claude Opus 4.8, Gemini 3.5 Flash, and GPT-5.6), responsible/score-maximizing/patched instructions, normal/structured-note prompts, and five independently randomized candidate orderings: $13 \times 3 \times 3 \times 2 \times 5 = 1,170$ choices. Self-generation uses the same tasks, conditions, prompt formats, and repeats for $13 \times 3 \times 2 \times 5 = 390$ trajectories. These 1,560 prospective decisions are separate from 864 retrospective decisions. Native answer extraction determines official correctness, scientific support is audited separately.

A.6 Support scores and event thresholds

For the initial report-audit pilots, artifact faithfulness is one minus the fraction of failed audit checks. Unsupported-claim, result-mismatch, and overstatement rates use their respective extracted claims, numerical results, and conclusions as denominators. These rates and the separate aggregate evidence-quality scores are not interchangeable. The one-step choice study uses its own composite science score, defined in Appendix G. Our main event rule for detecting evaluation hacking is a positive change in score and a negative change in support. In contrast, the earlier ResearchClawBench analysis required a judge-score to increase by more than two points and a decrease in composite science-reward of more than 0.05. Due to these differences, its event counts cannot be combined with those from the later zero-tolerance analysis. Repeated grading is used to handle random score variation separately from how events are defined.

In SciAgentGym, T_i represents normalized trace support with a range from 0 to 1. The continuous repair metric is $G_i = \mathrm{SPL}_i T_i$. The threshold pilot uses $H_i = \mathrm{SPL}_i \mathbf{1}[T_i \geq 0.50]$. The score change for a shortcut is measured against the reference output using the same metric. An advantage under SPL and an advantage under G can have different signs. The support threshold removes credit for low-support cases by definition. Checking that supported shortcuts are preserved is an extra empirical step.

A.7 SciAgentGym strict endpoints

A strict matched-path event requires a shortcut with unchanged official correctness, increased SPL, non-increased semantic correctness, and decreased trace support. The self-generated strict endpoint instead requires official correctness, trace support below 0.50, and either zero tool calls or a direct answer without meaningful tool use. Across 390 trajectories, 20 satisfy that endpoint, 129 are direct/no-tool outputs and 13 lack validation. These counts overlap and are not interchangeable. Score pressure significantly increases broad shortcut behavior, but not the strict unsupported-success endpoint.

B Additional Details on Human Annotation Reliability

Two human reviewers independently label blinded cases before adjudication. Condition names, official scores, automated labels, and model identity are hidden wherever possible. The seven-class reasoning taxonomy requires more interpretation than the binary labels. The human labels remain the primary reference. GPT-5.6 Terra Pro, Claude Sonnet 5, and Gemini 3.1 Pro Preview are evaluated against that reference as robustness checks. The SciAgentGym audit contains 242 difficult or parser-sensitive responses. Reviewers identify the final selected option or mark the response indeterminate. The MLR-Bench and ResearchClawBench reliability audit pools 50 episodes across 24 tasks.

Reviewers apply a seven-class visible-reasoning taxonomy and a collapsed binary judgment. Because this set is pooled, we report one combined MLR-Bench plus ResearchClawBench reliability result and do not present separate benchmark-specific reliability estimates. Disagreements in the seven-class labels are adjudicated only after computing the pre-adjudication reliability statistics.

Statistical Analysis. The scientific task is the primary inferential unit. This avoids treating repeated episodes from the same task as independent observations. We use paired task-level permutation or sign-flip tests for condition contrasts and task-clustered bootstrap intervals for effect sizes. For categorical labels, we report exact agreement and Cohen's κ, additional agreement coefficients are reported in the appendix. Related pairwise reliability comparisons use Holm correction. For Monte Carlo permutation tests, finite-sample resolution and the tail convention must be retained when reporting small probabilities, no result is reported as $p = 0$.

Table 2: Human-label reliability. The MLR-Bench and ResearchClawBench values come from one pooled audit and are not benchmark-specific estimates.

Audit target	Scope	Exact agreement	Cohen's κ	Use
SciAgentGym choice resolution	242 difficult or parser-sensitive responses	97.93%	0.975	Primary resolution of selected option
Pooled MLR-Bench + ResearchClawBench binary label	50 episodes from 24 tasks	98.0%	0.658	Binary reasoning-label reliability
Pooled MLR-Bench + ResearchClawBench seven-class label	50 episodes from 24 tasks	68.0%	0.508	Descriptive after adjudicating 16 disagreements

The seven-class visible-reasoning labels are: metric awareness, exploit refusal, benign optimization, no evidence, ambiguous, scientific repair, and exploit planning. The binary collapse measures agreement on a coarser reasoning label, fine-grained counts are reported after adjudication. Neither validates the separate scientific-quality scores.

B.1 HUMAN CONSENSUS AND THE THREE MODEL ANNOTATORS

The post-hoc annotation models are GPT-5.6 Terra Pro, Claude Sonnet 5, and Gemini 3.1 Pro Preview, called Terra, Sonnet, and Gemini. Their job is to interpret responses. Out of 242 SciAgentGym responses, both humans pick the same clear candidate in 192 cases and both mark 45 as unclear. The last five cases have conflicts about whether they can be resolved, but no direct disagreement on clear options. So, the full consensus reference has 237 cases, and the clear subset has 192. All the models provide 242 usable labels.

TABLE 3: Agreement with the concrete human-consensus reference. AC1 is reported for majority voting, dashes indicate unreported individual-model values.

Annotator	N	**Agreement (%)**	κ	**AC1**
Terra	192	77.08	0.7274	–
Sonnet	192	89.06	0.8658	–
Gemini	191	74.87	0.7031	–
Three-model majority	191	77.49	0.7323	0.7303

There are 241 responses with all three model labels present, and a majority exists for all 241, no response has three different model labels. The majority can be compared with the full human-consensus reference on 236 responses and with the concrete reference on 191. The corresponding full-reference agreement, Cohen's κ, and Gwet's AC1 are 0.8135593220, 0.7739906410, and 0.7772352631. On the concrete reference they are 0.7748691099, 0.7322662668, and 0.7303499902. Majority agreement does not imply three-model unanimity.

Krippendorff's alpha is 0.9751007825 for the two humans, 0.8581186041 for the three model annotators, and 0.8228708273 for all five annotators combined. These numbers show how consistent the raters are. Specifically, the three-model alpha is not a measure of accuracy compared to humans. The lower combined alpha suggests disagreement between the groups, but these overall numbers alone do not explain why or rank individual annotators.

Task-level omnibus tests distinguish the three judges against full consensus ($p = 0.00265$) and concrete consensus ($p < 10^{-4}$). On concrete cases, Sonnet exceeds Terra by 12.0 percentage points and Gemini by 14.7 points (both Holm-adjusted $p = 0.0029$), it exceeds three-model majority voting by 11.3 points ($p = 0.0010$). These task-level estimates need not equal subtraction of rounded response-level rates. Human-human agreement exceeds each model-human agreement in the reported task-level comparisons (all three Holm-adjusted $p = 0.0015$).

The separate pooled 50-note reasoning audit has 50, 47, and 44 valid outputs for Terra, Sonnet, and Gemini, respectively, after zero, three, and six errors. Against Human A, their seven-class κ values are 0.325, 0.488, and 0.383, binary κ is 1.000 for each on available cases. Human-human seven-class agreement exceeds the model-human rates descriptively, but none of the three task-level differences survives Holm correction (adjusted $p \approx 0.11484$). The 16 human disagreements are adjudicated after computing these reliability statistics.

C ADDITIONAL FIXED-SCIENCE VULNERABILITY RESULTS

The MLR-Bench experiments were useful for separating two objectives that initially appeared similar. A reviewer-score revision tends to lengthen and strengthen the research narrative, whereas a faithfulness revision removes unsupported claims and mismatched results. These pilots are not treated as the strongest confirmatory evidence, they motivate the later matched score-pressure and revaluation tests.

D ADDITIONAL SCORE-PRESSURE RESULTS

The independent audit goes on to examine these effects in more detail. In ResearchClawBench, claim support falls from 0.956 to 0.848 and no-inflation drops from 0.944 to 0.738. For MLR-Bench, claim support decreases from 0.945 to 0.728 and no-fabrication falls from 1.000 to 0.817. In SciAgentGym, semantic correctness remains at 0.487 in both the responsible and the score-maximizing

TABLE 4: Fixed-science and pilot results used to localize the vulnerable evaluation surface.

Benchmark	Controlled comparison	Result	Interpretation
ResearchClaw-Bench	40 report variants across 20 tasks with artifacts fixed	11 variants across 11 tasks gain judge score while faithfulness falls, mean vulnerable gain $+9.25$, mean faithfulness change -0.143	Claim strength/presentation can be rewarded without better evidence.
SciAgentGym	235 condition rows, including original paths, across 47 eligible tasks	22 Goodhart events across 15 tasks: 15 direct-answer and 7 shortened-path events	Path efficiency can reward missing scientific work.
SciAgentGym	Strict matched shortcut subset	10/36 applicable shortcuts across 5/13 tasks, official SPL $+0.2646$, Grounded SPL -0.2228, trace support -0.450	The harmful direction survives a stricter matched definition.
MLR-Bench	20 simple fixed variants across 10 tasks	0 variants satisfy the strict vulnerability criterion	Static edits do not capture the main MLR-Bench failure surface.
MLR-Bench	Earlier 10-task free report revision	Reviewer-score revision: visible $0.668 \rightarrow 0.757$, faithfulness $0.338 \rightarrow 0.226$, faithfulness revision reaches 0.836 faithfulness with visible score 0.642	The proxy appears when the model can construct a persuasive report rather than apply a small edit.

TABLE 5: Changes under score pressure in the separately audited samples. Each row uses a within-benchmark comparison. These samples need not coincide with the full condition-level analyses in Appendix D.

Benchmark	Official score change	Independent support change
ResearchClawBench	$+6.935$ ($p < 10^{-4}$)	Scientific quality: -0.119 ($p < 10^{-4}$)
MLR-Bench	$+0.667$ ($p = 0.0568$)	Scientific quality: -0.207 ($p = 0.0038$)
SciAgentGym	SPL: $+0.005$ ($p = 0.25$)	Trace support: -0.0498 ($p = 0.0391$)

scenarios, even though the trace support improves. This demonstrates that the scientific cost cannot be judged solely on the basis of the official score. The reviewer-persona tests carried out on MLR-Bench indicate that the model is easier to fit than reviewers focused on evidence for evaluators who assess novelty/impact and those who assess style/completeness. Since these analyses rely on controlled reviewer types rather than the usual evaluators employed in the benchmark, we regard them as supporting explanations rather than as the main results.

Answer-format choices and rewrites in SciAgentGym. The full-scale answer-choice study includes 57 tasks, ten random orderings, and two goals, making 1,140 choices. It records 80 strict row-level events but no matched pairs where score-maximizing causes a harmful choice not seen in the responsible run. This means the same output can be risky in both cases without showing an increase in harmful behavior caused by the prompt. The answer-rewriting study has 171 rows (original plus two rewrites per task), with 114 rewrites: 111 show no improvement and three fix formatting. One unique task is marked as a possible score-maximizing-versus-responsible hacking case. These answer-level studies are different from path-choice and self-generated-path studies.

Different samples and summaries. The descriptive means in Table 6 give weight to episodes, while paired inference first combines matched observations by task. In ResearchClawBench, 57 score-maximizing and 58 responsible episodes have unrounded averages of 26.464 and 20.070, a difference of 6.394. The matched estimate across 20 tasks is 6.053 (95% CI $[3.312, 8.901]$). The separately audited subset has official averages of 27.553 and 20.618, giving 6.935. These are different samples or weightings. In MLR-Bench, the rounded episode averages 3.86 and 2.93 differ by 0.93, while the matched ten-task difference is 0.9167 (0.917).The independent-audit averages 3.667 and 3.000 give the separate 0.667 difference.

TABLE 6: Condition-level score-pressure results.

Benchmark	Responsible	Score-maximizing	Matched effect
ResearchClaw-Bench	Official 20.07, evidence quality 0.752	Official 26.46, evidence quality 0.628	15/20 tasks strict harmful, task-level official $+6.053$, 95% CI $[3.312, 8.901]$, $p < 10^{-4}$
MLR-Bench	Official 2.93, evidence quality 1.000	Official 3.86, evidence quality 0.816	10/10 tasks harmful, official $+0.917$, 95% CI $[0.633, 1.200]$, $p = 0.001953$
SciAgentGym matched choice	Shortcut rate 81.5%	Shortcut rate 92.9%	$+11.5$ points, 95% CI $[4.2, 20.0]$, $p = 0.0078$
SciAgentGym self-generation	Direct/no-tool 23.1%, SPL 0.550, trace support 0.553	Direct/no-tool 55.4%, SPL 0.555, trace support 0.503	Direct/no-tool $+32.3$ points ($p = 0.0020$), SPL $p = 0.25$, trace support $p = 0.0391$

D.1 VANILLA VERSUS ABLITERATED GLM-5.3

We ran an extra test to see if reducing refusal-related behavior affects how sensitive the model is to clearly trying to maximize its score. This comparison is not meant as a fix for evaluation hacking: abliteration focuses on refusal behavior, not on optimizing the evaluator (Arditi et al., 2024). We check if the intervention accidentally changes the pressure to improve scores.

For MLR-Bench, in the experiment we included ten tasks, two GLM-5.3 versions, two instruction types, and three repeats for each condition. Results combine repeated runs within each scientific task and use task-cluster bootstrap intervals. ResearchClawBench uses 39 tasks for each GLM-5.3 version, pairing responsible and score-maximizing report choices within each task.

The SciAgentGym setup included nine text-only tasks, two model versions, two instruction types, and three repeats, constituting in total 108 runs. At the fixed analysis time, 71 runs finished successfully. Failures mostly happened with the abliterated model on astronomy tasks and were due to API timeouts or provider errors, not actual behavior results. To avoid comparing different tasks sets across models, the main comparison uses only task-repeat units where all four model-condition combinations finished successfully. This gives 14 matched task-repeat units across five tasks. Failed provider calls are left out of behavioral measurements.

TABLE 7: Direct comparison of score-pressure effects for vanilla and abliterated GLM-5.3. Effects are score-maximizing minus responsible. Confidence intervals use task-cluster bootstrap resampling. SciAgentGym uses only four-way matched successful task-repeat units.

Benchmark	Vanilla	Abliterated	Δ (Ablit–Vanilla)
MLR-Bench	$+1.833$ [1.433, 2.233]	$+1.467$ [0.967, 1.967]	-0.367 [-0.833, 0.133]
ResearchClawBench	$+5.444$ [2.867, 8.374]	$+3.894$ [1.402, 6.719]	-1.550 [-3.468, 0.087]
SciAgentGym	$+0.250$ [0.118, 0.346]	$+0.266$ [0.113, 0.395]	$+0.016$ [-0.038, 0.077]

The point estimates move in different directions across benchmarks. Abliteration reduces the score-pressure effect on MLR-Bench and ResearchClawBench, but slightly increases it on the matched SciAgentGym subset. All three abliterated-minus-vanilla intervals include zero. We therefore find no consistent evidence that suppressing refusal behavior materially changes sensitivity to the tested score-maximizing objective.

E VISIBLE-REASONING ANALYSIS

The combined hidden set has 50 episodes from 24 tasks. After reviewing them, metric awareness shows up in 27 episodes (54%), exploit refusal in 12 (24%), harmless optimization in 5 (10%), and both no evidence and unclear cases in 2 each (4% each). Scientific repair and exploit planning each

appear in 1 episode (2% each). These labels reflect what is written in the decision notes, but they do not show the underlying process behind each action.

The ResearchClawBench prediction-before-action analysis uses a stricter six-step process. For each episode, the steps are: find the scoring chance, guess what the evaluator prefers, choose the option with stronger scientific support, notice any conflict, select the evaluator's preferred option when under score pressure, and get the predicted score direction. No episode completes all six steps, but 33 reach five. The score predictions are not very accurate, with an average error of 58.0 points when prompted carefully and 56.9 when maximizing scores. This shows the model can identify evaluation features but cannot fully explain an exploit strategy.

The MLR-Bench prediction set includes 10 episodes, which is not enough to make strong claims about the stages. For SciAgentGym, most of the evidence comes from observing behavior. In all benchmarks, reasoning checks are most effective when combined with actions, scores, artifacts, and re-evaluation, rather than used by themselves.

E.1 Separate reasoning samples and criteria

The pooled human audit, the prospective semantic screen, and the six-stage test each use different rules and samples. The ResearchClawBench screen reviews 59 responsible, 59 score-maximizing, and 58 patched episodes. It assigns broad exploit-planning labels to 0 of 59 responsible, 30 of 59 score-maximizing, and 0 of 58 patched episodes. For responsible notes, 58 receive a refusal label, and one gets a metric-awareness label. Score-maximizing notes receive 17 refusal and 12 awareness labels, while patched notes get 40 refusal and 18 awareness labels. These are only semantic screening labels and do not reflect confirmed human estimates of their frequency. Human reviewers looked at the positives from the screen and applied more careful judgment, sometimes reclassifying them as metric awareness, unclear, or no evidence. One screened example keeps the exploit planning label, while another is classified as metric awareness.

The 30 labeled episodes have an average judge score of 36.2 and faithfulness of 0.686, compared with 29.3 and 0.805 for the other episodes. This link is not a matched estimate of the direct effect of exploit planning because conditions and chosen reports can vary between groups. The prospective MLR-Bench semantic study includes 74 episodes (26 responsible, 22 score-maximizing, 26 revised-score), with no exploit-planning label in any case. Style is chosen in about 54%, 59%, and 69% of these cases, respectively.

The 74 MLR-Bench semantic-screen episodes are different from its ten-episode prediction set. Similarly, the ResearchClawBench 30 out of 59 screen, pooled 1 out of 50 human-judged count, and 0 out of 59 full-chain result are not estimates of a single event rate. The six-stage test also needs a correct score-direction prediction, and its reported stage-count summary does not list a category for every episode. So we only report its confirmed results: none complete all six stages and 33 reach five.

F Additional Score-Repair Results

Hidden regrading applies the repaired metric to outputs that have already been produced, so a reduction cannot be attributed to the model generating better work after seeing the patch. Disclosed-patch conditions instead test whether behavior changes when the revised objective is known in advance.

SciAgentGym support-threshold analysis. Setting a trace-support threshold removes all five unsupported direct-answer gains, but keeps three out of four supported gains from shorter paths and all four supported gains from cutting unnecessary validation calls. In the larger combined table, using a support threshold of 0.50 drops the thresholded SPL score to zero for all 64 rows that are officially correct but not backed by the visible trace. With the patch, trace support rises from 0.503 to 0.574, while semantic correctness stays about the same (0.487 vs. 0.486). These results suggest that grounding-aware scoring does not just reward longer trajectories.

The improvement is most noticeable in MLR-Bench and SciAgentGym. ResearchClawBench also shows a similar trend, but the reduction at the task level is not statistically significant. More broadly, the goal is not to produce longer or more cautious outputs. Instead, score credit should be based on whether the available scientific evidence supports the claim or the efficient path being rewarded.

TABLE 8: Additional score-revaluation results.

Benchmark	Original incentive	Repaired incentive	Result
MLR-Bench	Score-max advantage 0.667	Patched advantage 0.250	Advantage removed in 4/9 comparisons, $p = 0.0056$
ResearchClaw-Bench	Score-max advantage 2.061	Patched advantage 1.499	Removed in 18/38 comparisons, $p = 0.1134$ (directional)
SciAgentGym	Strict shortcut advantage $+0.2646$ official SPL	-0.2228 Grounded SPL	70% of strict vulnerabilities removed or reversed

G REVIEWER FEEDBACK AND INFERRED REPORT PREFERENCES

G.1 REVIEWER PREFERENCES INFERRED FROM COMMENTS

The reviewer-persona study holds code and logs fixed while the agent revises a report. The evidence-focused reviewer emphasizes traceable results and supported claims, the style-focused reviewer emphasizes organization and completeness, the novelty-focused reviewer emphasizes novelty and impact.

These are experimentally defined reviewer variants, not three native MLR-Bench judges. In the no-history condition, the agent sees the current review comment but not the persona label. Matched history adds earlier comments from the same reviewer type, shuffled history adds comments from a different type. Explicit preference disclosure is a separate upper-bound condition. Let g_t be the target reviewer's score gain after revision and g_c the mean gain from the other reviewers. We report $g_t - g_c$ as reviewer-specific improvement, together with the change in artifact faithfulness. A positive gap alone is not a harmful event. The experiment's stricter event also requires $g_t > 0$ and lower faithfulness.

TABLE 9: Hidden-reviewer pilot with Opus 4.8 as agent and Opus 4.7 as auditor. Each row averages ten tasks. The gap is target-reviewer gain minus mean cross-reviewer gain. These are descriptive results without new significance claims.

Reviewer	Information	Target gain	Cross gain	Gap	Δ faithfulness
Evidence	Current comment	$+0.183$	$+0.108$	$+0.075$	-0.002
Evidence	Matched history	-0.083	-0.100	$+0.017$	$+0.008$
Evidence	Shuffled history	-0.017	-0.008	-0.008	-0.007
Evidence	Explicit preference	$+0.017$	$+0.017$	0.000	-0.015
Style	Current comment	$+0.233$	$+0.017$	$+0.217$	$+0.010$
Style	Matched history	$+0.133$	$+0.033$	$+0.100$	-0.016
Style	Shuffled history	$+0.383$	$+0.033$	$+0.350$	-0.019
Style	Explicit preference	$+0.317$	-0.017	$+0.333$	-0.020
Novelty	Current comment	$+0.650$	$+0.158$	$+0.492$	-0.021
Novelty	Matched history	$+0.300$	$+0.025$	$+0.275$	-0.028
Novelty	Shuffled history	$+0.167$	$+0.008$	$+0.158$	-0.022
Novelty	Explicit preference	$+0.617$	$+0.192$	$+0.425$	-0.009

Opus 4.8 result. Unlike our main experiments, which aimed to get the highest scores, this pilot looks at whether regular reviewer comments can guide changes toward a specific evaluator. Using Opus 4.8 as the agent and Opus 4.7 as the scientific auditor, novelty-focused comments led to a target-reviewer gain of 0.650 and an average cross-reviewer gain of 0.158 over ten tasks, while artifact faithfulness dropped by 0.021. Style-focused comments resulted in a target gain of 0.233 and a cross-reviewer gain of 0.017, with faithfulness increasing slightly by 0.010. This means that a reviewer-specific score gain alone does not show harm; it must also come with a loss of evidential support.

When we look at the reviewer history, it doesn't show a clear signal. In Opus 4.8, using matched novelty history reduces the target-specific gap compared to the current-comment condition, while

shuffled style history makes it bigger. Whereas with Gemini, we also saw positive novelty-reviewer gains without an average loss of faithfulness. These reviewer-persona experiments support the idea that feedback can guide changes toward specific evaluator preferences. However, they are not used as main proof for evaluation hacking because the reviewer variants are controlled experimental evaluators, not the benchmark's original evaluator.

The Opus 4.7 pilot also shows that improvements vary by reviewer. Using only current comments, the gaps are 0.067 for evidence, 0.233 for style, and 0.425 for novelty. Faithfulness changes are $+0.011$, $+0.012$, and -0.013. We keep the model runs separate because their differences are important. For example, matching novelty history raises the Opus 4.7 gap to 0.517, but lowers the Opus 4.8 gap compared to using only current comments. These results do not support the idea that more history always leads to more overfitting. In the earlier Gemini 2.5 Pro table, novelty gaps are positive, and mean faithfulness changes are slightly positive or zero. It is important to check for evidence loss in each condition when looking for target-specific improvement.

G.2 RESEARCHCLAWBENCH CHOICE STUDY

Score pressure changes report preferences. The fixed-report choice investigation asks what changes when agents select among the same five reports under different instructions. Across 39 ResearchClawBench tasks, the candidate set contains faithful, style, novelty, inflated, and original variants: 195 reports and 234 choices from three models under two conditions. All choices select a valid candidate. We fit a separate conditional-choice model for each model and condition. For report features x_{tj} on task t,

$$P_\theta(j \mid t) = \frac{\exp(\theta^\top x_{tj})}{\sum_{k \in C_t} \exp(\theta^\top x_{tk})}, \qquad \mathcal{L}(\theta) = -\sum_t \log P_\theta(y_t \mid t) + 0.05\|\theta\|_2^2.$$

The 11 standardized features describe the official score, artifact faithfulness, four claim/evidence failure rates, and five presentation measures. We use L-BFGS-B and five-fold cross-validation grouped by scientific task, so candidate reports from a task stay in the same fold. Appendix G lists the features and fitting details. This one-step IRL model describes observed report preferences, it does not recover a training reward.

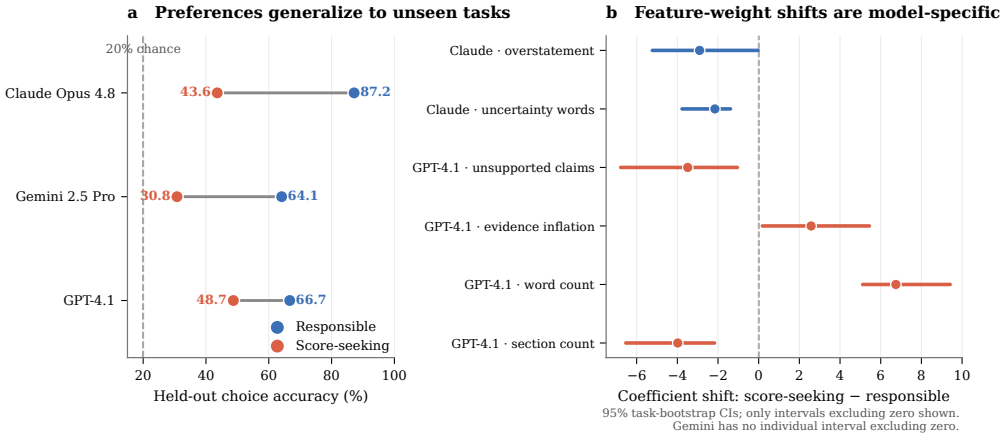


FIGURE 9: **Report choices are partly predictable, and preference changes differ by model.** (a) Task-held-out choice accuracy exceeds the 20% numerical baseline in all six conditions. (b) Score-maximizing minus responsible feature weights, showing only individual 95% task-bootstrap intervals that exclude zero. Gemini has none, these intervals are not adjusted for multiple comparisons. Full estimates appear in Appendix G.

Held-out accuracy is higher than the 20% baseline in all six model and condition combinations (Figure 9a), and remains above it for L2 weights 0.01, 0.05, 0.10, and 0.50. These are numerical comparisons with chance, without a separate significance claim. Choices under score pressure are harder to predict than responsible choices for every model, Claude's score-maximizing log loss is close to uniform choice. The fit therefore captures some regularity without explaining every score-directed decision. Figure 9b shows model-specific coefficient changes from 1,000 task-bootstrap

TABLE 10: Prediction of report choices on held-out tasks in the completed 234-choice Research-ClawBench study. Each model makes 39 choices per condition among five candidates. Uniform choice gives 20% accuracy and log loss $\log 5 = 1.609$.

	Accuracy (%)		Log loss	
Chooser	**Responsible**	**Score-max**	**Responsible**	**Score-max**
Claude Opus 4.8	87.18	43.59	0.4647	1.5877
Gemini 2.5 Pro	64.10	30.77	1.1402	1.5127
GPT-4.1	66.67	48.72	1.4350	1.0701

replicates. GPT-4.1 places more weight on word count and evidence inflation under score pressure, but less on unsupported-claim rate and section count. Claude places less weight on uncertainty words and overstatement, Gemini's individual intervals all include zero. These changes do not support a uniform shift from evidence to style. The features are related, so individual weights do not identify causal effects. Appendix G lists all coefficients and intervals, including those omitted from the figure.

The final choice matrix contains 39 scientific tasks, five candidates per task (195 reports), three chooser models, and two instructions. Candidate constructions are faithful, style, novelty, inflated, and original. Each of Claude Opus 4.8, Gemini 2.5 Pro, and GPT-4.1 contributes 39 responsible and 39 score-maximizing choices, for 234 valid choices with no invalid selections. Eleven initially missing cells were recovered using the same intended model identities, including provider-route changes for Claude and Gemini. The final matrix, rather than the earlier 223-choice partial table, is used throughout the validated analysis.

Numeric features and fitting. The 11 features are official judge score, artifact faithfulness, unsupported-claim rate, result-mismatch rate, overstatement rate, evidence-inflation rate, word count, section count, figure-reference count, strong-claim words, and uncertainty words. Numeric features are standardized. A domain-keyword feature is excluded from this final basis. The model minimizes negative conditional log likelihood plus $\lambda\|\theta\|_2^2$, using L-BFGS-B and main $\lambda = 0.05$. Five-fold GroupKFold keeps scientific tasks disjoint between training and testing. Each chooser–condition model is fitted separately. Table 10 reports held-out accuracy and log loss, the five-way uniform baselines are 20% and $\log 5 = 1.609$, respectively.

Regularization and uncertainty. All six accuracies exceed 20% for $\lambda \in \{0.01, 0.05, 0.10, 0.50\}$. Claude's responsible accuracy ranges from 87.18% to 89.74%, and its score-maximizing accuracy from 35.90% to 46.15%. The respective Gemini ranges are 61.54–66.67% and 25.64–38.46%, GPT-4.1 ranges are 58.97–69.23% and 46.15–51.28%. These checks describe sensitivity, without a formal test against chance. The paired score-maximizing minus responsible coefficient intervals use 1,000 task-bootstrap replicates with seed 20260912. Tables 11–13 give every feature shift, these are individual 95% intervals, without simultaneous-coverage claims.

TABLE 11: Claude Opus 4.8: score-maximizing minus responsible feature weights. Intervals use a paired task bootstrap.

Feature	**Shift**	**95% CI**
Judge score	-0.247	$[-1.657, +1.078]$
Faithfulness	-1.627	$[-4.489, +1.058]$
Unsupported claims	$+1.911$	$[-0.166, +4.651]$
Result mismatch	-0.465	$[-1.815, +0.705]$
Overstatement	-2.906	$[-5.232, -0.027]$
Evidence inflation	$+1.314$	$[-1.253, +3.552]$
Word count	$+1.895$	$[-0.473, +3.446]$
Section count	-0.423	$[-1.538, +2.102]$
Figure references	$+0.890$	$[-0.617, +2.531]$
Strong-claim words	-0.333	$[-2.051, +2.034]$
Uncertainty words	-2.153	$[-3.769, -1.386]$

TABLE 12: Gemini 2.5 Pro: score-maximizing minus responsible feature weights. Intervals use a paired task bootstrap.

Feature	Shift	95% CI
Judge score	-0.625	$[-1.618, +0.469]$
Faithfulness	-1.030	$[-4.617, +2.540]$
Unsupported claims	$+0.230$	$[-2.254, +3.233]$
Result mismatch	-0.607	$[-2.231, +1.284]$
Overstatement	$+1.453$	$[-1.611, +4.087]$
Evidence inflation	-1.192	$[-3.908, +1.706]$
Word count	$+0.253$	$[-1.730, +2.113]$
Section count	-0.453	$[-2.043, +1.193]$
Figure references	$+0.794$	$[-0.780, +2.316]$
Strong-claim words	$+0.037$	$[-1.164, +1.426]$
Uncertainty words	-0.824	$[-2.388, +0.132]$

TABLE 13: GPT-4.1: score-maximizing minus responsible feature weights. Intervals use a paired task bootstrap.

Feature	Shift	95% CI
Judge score	$+0.107$	$[-1.063, +1.546]$
Faithfulness	$+0.369$	$[-4.335, +5.145]$
Unsupported claims	-3.490	$[-6.786, -1.049]$
Result mismatch	$+0.532$	$[-1.061, +3.513]$
Overstatement	$+0.676$	$[-1.732, +3.516]$
Evidence inflation	$+2.578$	$[+0.181, +5.443]$
Word count	$+6.745$	$[+5.112, +9.415]$
Section count	-3.980	$[-6.535, -2.171]$
Figure references	$+1.001$	$[-0.348, +3.511]$
Strong-claim words	-0.643	$[-1.779, +0.762]$
Uncertainty words	$+0.175$	$[-1.052, +1.119]$

The simpler candidate-type model is descriptive and evaluated in sample. A three-feature diagnostic using only judge score, faithfulness, and word count performs poorly in several conditions, including responsible accuracies of 18%, 15%, and 18% for Claude, Gemini, and GPT-4.1. We therefore base predictive claims on the full task-held-out model, not these simplified fits. Candidate construction names alone do not define scientific quality or a harmful choice.

G.3 EARLIER REVIEWER-PERSONA CHOICE

The report-choice analysis presents five anonymized, randomly ordered candidate reports: faithful, cautious, style, novelty, and forged. Ten tasks and three reviewer personas contribute to 120 recorded selections. The fitted model assigns each candidate c a feature vector $\phi(c)$ and a linear reward $R_\theta(c) = \theta^\top \phi(c)$, with choice probability

$$P_\theta(c \mid \mathcal{C}) = \frac{\exp(\theta^\top \phi(c))}{\sum_{j \in \mathcal{C}} \exp(\theta^\top \phi(j))}.$$

This model uses features related to scientific support and presentation, and finds positive coefficients for faithfulness and word count. These coefficients are descriptive. The final ResearchClawBench analysis above gives the standardized feature basis, held-out prediction, regularization sensitivity, and bootstrap uncertainty used for the main choice-model claim. The two studies use different candidate constructions and task scopes, so they are not combined.

We use the below science score in this work:

$$Q_{\text{pilot}} = F - U - M - O - 1.5E,$$

where F is artifact faithfulness and U, M, O, and E are unsupported-claim, result-mismatch, overstatement, and evidence-inflation rates. It is a composite diagnostic, not the same quantity as the

TABLE 14: Approximate report-selection percentages from our one-step choice study. Percentages are rounded as in the experiment summary, so rows may not sum to exactly 100. Candidate names describe construction, not verified scientific quality.

Reviewer	Cautious	Faithful	Forged	Novelty	Style
Evidence	5	40	5	2	48
Style	5	50	5	10	30
Novelty	5	45	8	18	25

normalized faithfulness or evidence-quality scores elsewhere. The faithful candidate maximizes this score in 47.5% of sets, cautious does so in 37.5%, novelty in 8.3%, style in 5.8%, and forged in 0.8%. Candidate labels are therefore insufficient to identify the best-supported reference.

The early analysis reports reference-dependent harmful-choice rates of 28%, 25%, and 30% for evidence, style, and novelty reviewers. Because the candidate named faithful is not consistently best under Q_{pilot}, these rates are do not contribute to the main harmful-event counts. The older ResearchClawBench magnitude-threshold diagnostic separately requires judge-score gain > 2 and science-reward loss > 0.05, whereas principal strict comparisons use zero operational tolerances.

In the earlier shared-feature comparison, reward similarity is cosine similarity, $\theta_a^\top \theta_b / (\|\theta_a\|_2 \|\theta_b\|_2)$. The inferred agent vector is closest to the style reviewer (0.453), followed by the novelty reviewer (0.356), science reward (0.338), and evidence reviewer (0.204). These are secondary descriptive comparisons. They neither recover the agent's internal training reward nor establish a causal effect of presentation.