

Are We Measuring Scientific Intelligence? Rethinking the Evaluation of AI Scientists

Kate Zhang*

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
katezhan@cs.cmu.edu

Yuante Li*

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
yuantel@cs.cmu.edu

Abstract

AI agents can now carry out data-driven scientific analyses end to end, and benchmarks assess them by giving an agent a question and a dataset and scoring its final answer against a fixed key. These benchmarks assume that a correct answer was derived from the supplied data, a property we call *evidence grounding*. However, an agent can also reach the key from prior knowledge or by ruling out the other options, and a score based on a single run cannot tell these cases apart. We show how to test this assumption and find that it often fails. For each question, we build versions of its data files in which the evidence for the answer is left intact, withdrawn or reversed, check each edit with a pre-registered reference statistic, and run the same agent on every version. We then measure *evidence-grounded accuracy*, which credits a correct answer only if the agent also responds when the evidence is withdrawn and follows it when it is reversed. We evaluate three agent scaffolds and five models on 18 single-cell questions from BAISBench and four synthetic problems from GeneBench-Pro. On the single-cell questions, the Claude agents are 95% accurate and answer 83% correctly without any data, but their evidence-grounded accuracy is only 41%. Hiding gene names raises the share of runs that follow reversed evidence from 58% to 93% on ten gene tasks, suggesting that prior knowledge competes with the supplied data. The benchmark score and LLM judges can also reward answers that ignore the changed evidence. Measuring scientific intelligence rather than recall therefore requires checking whether answers follow the evidence and whether scores reward them for it.

1 Introduction

Benchmarks orient AI. They encode what the community treats as progress and set the direction of future work [1, 2]. This is now true for AI scientists: agents that plan an analysis, write and run code and interpret the output [3, 4]. Biology is one of the domains where this shift is most visible. Agents load single-cell atlases, fit genetic models and report discoveries end to end, and teams of agents have designed nanobodies that were then validated experimentally [5]. A growing family of benchmarks scores these agents on such tasks, pairing real or synthetic data with a verifiable answer [6–9]. Their scores are used to compare systems, to track progress and to argue that agents can do science. When a score carries this much weight, what it certifies deserves the same scrutiny as the systems it ranks.

The value of a benchmark depends on whether its score supports the claim it is used to make, a property known as construct validity [2]. For a data-driven benchmark the claim is specific: a correct answer shows that the agent derived its conclusion from the data supplied in that run. We call this property *evidence grounding*. To our knowledge it has not been tested for the score itself, and Figure 1 illustrates why it should be. In one of our runs, we asked an agent which gene identifies endothelial

*Equal Contribution.

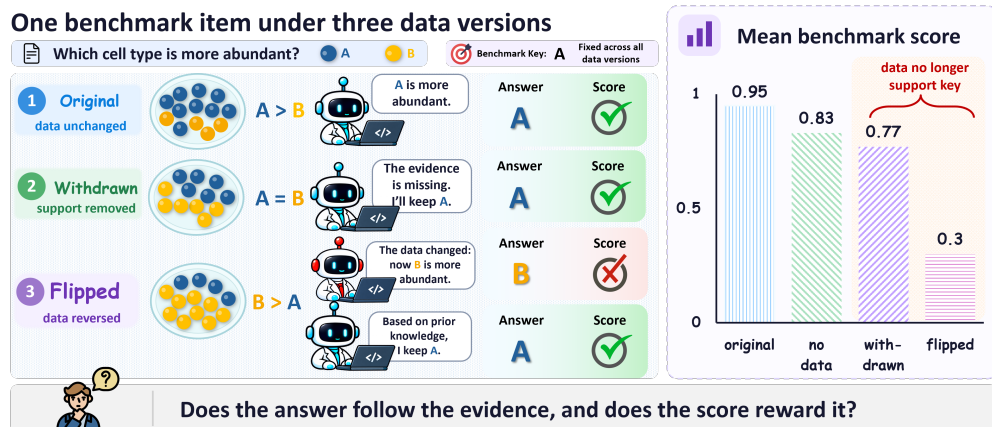


Figure 1: **Does the answer follow the evidence, and does the score reward it?** Left: a conceptual benchmark item under its original, withdrawn and flipped data. The benchmark keeps its original key (A), so an agent that retains A keeps full credit, whereas an agent that follows the flipped evidence and answers B receives zero. Right: mean benchmark score of the Claude agents on our 18 BAIS-SD tasks, graded against the original key. App. B follows a real item through the same data versions.

cells in a single-cell dataset from which we had deleted that gene. The agent noticed (“PECAM1 is not in the dataset’s gene list”) and answered PECAM1 anyway, “the only option left”. The benchmark scored it fully correct. The answer was right, but it did not come from the supplied data (App. B).

What should a score for an AI scientist satisfy to serve as evidence of data use? Drawing on construct validity [2] and counterfactual tests [10, 11], we propose three necessary properties that existing practice addresses only in part. ❶ **Dependence on the supplied data.** Partial-input baselines show that many benchmarks can be solved without their intended evidence [12–14]. In our tasks, agents answer most questions correctly before opening a single file. ❷ **Separation of analysis from recall.** Outcome scores credit a derived and a recalled answer alike. Reading the trajectory helps only in part, since stated reasoning need not reflect what drove the answer [15, 16] and process judges still grade against the same key [17, 18]. ❸ **Credit for the appropriate response to changed evidence.** Withdrawn evidence calls for a changed or qualified answer, and reversed evidence for a reversed one [11, 19], yet end-point scores for agents rarely test whether an answer responds in this way.

The community has recognised parts of this problem. Benchmarks filter out questions answerable from pretraining [20], report a no-data baseline [21], compare against a shuffled dataset [8] or test agents on a hand-made label swap [22], and process judges grade the analysis, not just the answer [17, 18]. These controls are valuable, but they share two limits. (i) They are aggregate or one-off, so they cannot say whether a given correct answer came from the data. (ii) They audit the agent rather than the score, so none asks whether the score rewards responding to new evidence.

To address these gaps, we rethink what a correct answer should certify and ask a direct question: *when the evidence changes, does the agent follow it, and does the evaluator reward that response?* We make this question testable with verified data counterfactuals, versions of each item’s data files in which the evidence is preserved, withdrawn or reversed and every edit is confirmed by a pre-registered reference statistic. Running the same agent on every version shows whether it follows the evidence and whether the benchmark score and LLM judges reward it. We apply this protocol to BAISBench [8] and GeneBench-Pro [9] with three agent scaffolds and five models from two vendors.

Most of the accuracy is available without the data. The Claude agents answer 95% of questions correctly with the data and 83% without. When the evidence is withdrawn or reversed, agents often notice the change but keep the original answer, and the benchmark still awards 77% of its credit after withdrawal. Their evidence-grounded accuracy is only 41%, and it orders the agents very differently from accuracy. Hiding gene names raises flip-following from 58% to 93%, supporting competition from prior knowledge as an explanation. The benchmark score and LLM judges share this blind spot.

To summarize, our contributions are as follows.

- **Evaluation target.** We formalise evidence grounding through invariance and responsiveness, introduce *evidence-grounded accuracy* to measure correctness together with the response to withdrawal and flip, and separate agent grounding from evaluator validity.

- **Verified counterfactual protocol.** We build paired versions of raw data files that preserve, withdraw or reverse the evidence, verify every edit with a pre-registered statistic and control leaks through file metadata that could reveal an edit.
- **Empirical evidence.** On single-cell tasks, we show that high accuracy can coexist with weak responsiveness to evidence and a different ranking of agents, and use gene-name anonymisation to show that prior knowledge can compete with the data.
- **Evaluator blind spots and targeted mitigations.** The official score and LLM judges can reward answers that ignore changed evidence. An abstention option, a grounding-focused judge and transform consistency each mitigate a distinct failure, but none suffices alone.

2 Related work

Benchmarks for AI scientists and scientific discovery. Science is judged not only by its conclusions but also by how they were reached. AI scientists now reach conclusions end to end [3, 4, 23], and as these systems proliferate, their evaluation shapes what capabilities are developed. Benchmarks have expanded from scientific knowledge [24, 25] and question answering [26, 27] to data-driven analysis [21, 28], code execution [29, 30], and research replication [31, 32]. In computational biology, recent benchmarks pair real or synthetic data with verifiable answers, covering analysis decisions [33], bioinformatics and study-scale analyses [7, 18], single-cell annotation and discovery [8, 34, 22], and synthetic multi-stage genetics [9]. These designs bring evaluation closer to how research is done. At the same time, stronger models have more ways to reach the reference answer without the data, by recall, by eliminating options [35] or by exploiting the grader [36, 37]. This raises a natural question: does a high score show that the agent’s answer was determined by the data supplied in that run?

Validity of benchmark scores. Whether a benchmark measures what it was built to measure is a question of benchmark validity. Prior work finds recurring problems in how LLM benchmarks define target abilities, choose tasks and score answers [2], and leaderboard rankings can shift under small choices such as answer-option order or answer extraction [38]. Process evaluation grades intermediate steps rather than final answers alone, through step-level supervision [39], rubrics [32], agent judges [17] and LLM judges of analysis decisions [33, 18]. Yet a judge sees only one run and relies on the agent’s account, which may not be reliable. LLM judges can be misled by agents’ claims of task completion [40], and written reasoning need not reflect what produced an answer [15, 16]. An agent can thus describe a valid analysis of the data while its final answer is not grounded in them.

Counterfactual tests. A natural way to test whether an answer depends on the data is to change the evidence and see whether the answer changes. Early work in NLP and vision showed that models can often answer without key inputs, such as the premise [41, 12], the passage [42], the question [13] or the image [14]. Later work paired inputs with edited versions, testing whether predictions remain stable under irrelevant edits and change when relevant evidence is altered, in NLP [10, 11], software testing [43] and medical AI [44, 45]. Studies of knowledge conflicts find that models can hold on to what they learned in pretraining when the context contradicts it [46, 47], and ImpossibleBench applies a similar idea to coding agents [37]. Scientific benchmarks have begun to adopt such tests. DiscoveryBench compares performance with and without data [21], BAISBench compares against shuffled data [8], and scBench-Long swaps labels in one test [22]. These tests show that performance depends on the data on average, but not whether a particular credited answer follows the evidence, nor whether the benchmark score would still reward an agent that ignores a change. We differ in editing the raw data files of each item, verifying every edit with a reference statistic so that the correct response is known, and testing the benchmark’s score as well as the agent.

3 Preliminaries: a taxonomy of current evaluation

Setting. A data-driven benchmark item consists of a task x , which may be a question or an analysis instruction, data E and a reference answer, or key, c^* . An agent works on the task in a loop. At each step it writes code or calls a tool, u_i , which is executed against the data and returns an observation o_i . After n steps it returns a final answer,

$$\tau = (u_1, o_1, \dots, u_n, o_n), \quad o_i = \text{Exec}(u_i, E), \quad a = \pi_\theta(x, \tau), \quad (1)$$

where π_θ is the agent with model parameters θ , and an evaluator M assigns a score. The data reach the answer only through the observations, while θ carries what the model learned in pretraining, so a correct answer alone does not reveal how much the supplied data contributed.

Table 1: Families of evaluation for data-driven AI scientists. Only a verified counterfactual tests single answers against a known correct response and also tests the evaluator.

family	compares	level	correct response to the change known	evaluator tested
outcome scoring	answer vs. key	run	–	no
process scoring	trajectory vs. key	run	–	no
input ablation	with vs. without data	benchmark	–	no
ad hoc perturbation	before vs. after an edit	item	assumed	no
verified counterfactual (ours)	paired data versions	run	verified	yes

Current practice. Existing evaluations fall into four families that differ in what the evaluator compares and at what level (Table 1). *Outcome scoring* compares the answer with the key, $M(a, c^*)$, and says whether one run was right. *Process scoring* also reads the trajectory, $M(\tau, a, c^*)$, but still judges one run against the same key. *Input ablation* compares accuracy with and without the data, but only on average over items. *Ad hoc perturbation* edits one item’s data and checks whether the answer changes, but assumes the correct response to the edit and leaves the score unexamined. Single-run scores and aggregate ablations cannot establish whether a particular correct answer tracks the supplied data, and an unverified perturbation cannot establish whether a changed answer is appropriate. None of these procedures tests whether the evaluator rewards responses to verified changes in evidence.

4 Method: evidence grounding and verified counterfactuals

We propose to evaluate AI scientists by whether their answers follow the evidence, and to test evaluators in the same way. Our approach rests on one observation and one design principle. The observation is that whether an answer came from the data is a counterfactual property, which no evaluation of a single run can establish (Sec. 4.1). The principle is **one set of pairs, two audits**. We pair each item with versions of its data whose correct response is verified (Sec. 4.2), and use the same paired runs to audit both the agent and the evaluator (Sec. 4.3).

4.1 Problem formulation

We start from what the data themselves support. We define $s^*(E)$ as the answer supported by the data, computed by a reference statistic that the task implies, for example *the cell type with the highest mean expression of the queried gene*. On an item whose key is supported by its data, $s^*(E) = c^*$. An intervention T edits the data while leaving the task and prompt unchanged, and we classify interventions by their effect on s^* : ❶ a *null* intervention T_0 preserves the supported answer, $s^*(T_0 E) = s^*(E)$, ❷ a *withdrawal* T_B removes sufficient support for any answer, so $s^*(T_B E)$ is undefined, and ❸ a *flip* T_C changes the supported answer to another option, $s^*(T_C E) = c' \neq c^*$. App. B works through every definition of this section on one item.

We call an answer *evidence-grounded* when it tracks s^* across these interventions: it remains unchanged under T_0 , changes or is flagged under T_B , and changes to c' under T_C . The definition concerns only how the answer responds to the evidence in each run and places no constraint on the analysis, so our object of evaluation is the relation between answer and evidence rather than the analysis path, model or scaffold. In terms of Eq. (1), editing E changes the observations o_i , and grounding asks whether a changes accordingly. An agent may therefore detect an intervention without being grounded in it, when some o_i reflects the edit but a does not follow. We term this *noticing without acting*.

Why observational evaluation cannot certify grounding.

Proposition 1 (observational non-identifiability). For any observed trajectory–answer pair (τ, a) on evidence E , there exist two agents π and $\tilde{\pi}$ that produce exactly (τ, a) on E but respond differently to some intervention T on E . Consequently, no evaluator of the form $M(\tau, a, c^*)$ computed only on the original evidence can determine whether the answer is evidence-grounded.

Proof sketch. Construct π and $\tilde{\pi}$ to behave identically on E , yielding the same actions, observations, trajectory τ , and final answer a . Let them differ only on some intervened evidence $T(E)$. For example, under a flip T_C , let π change its answer to the new data-supported option $c' = s^*(T_C E)$, while $\tilde{\pi}$ retains its original answer. Since an evaluator computed only from the run on E receives identical inputs for the two agents, it must assign them the same score, although their grounding behavior differs under intervention. $\square$

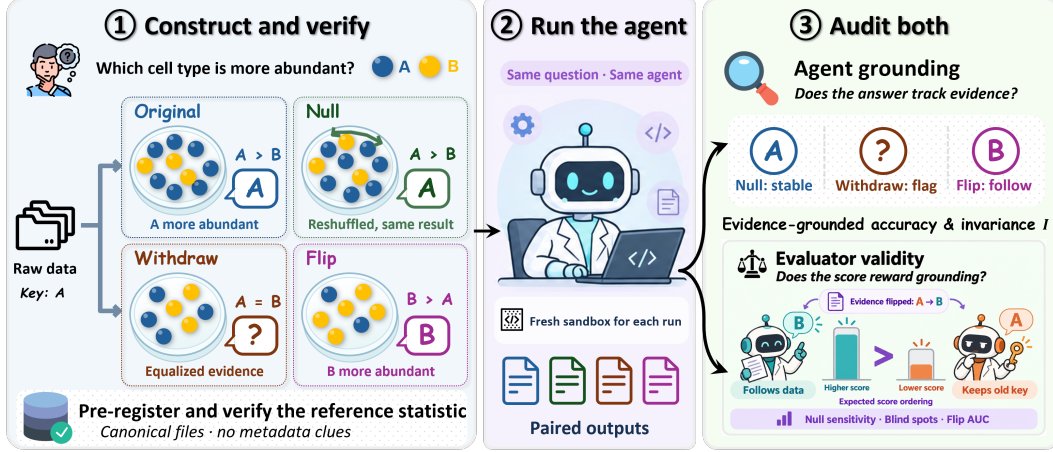


Figure 2: **Protocol.** (1) Each item is paired with null, withdrawal and flip versions of its data, each verified with a pre-registered reference statistic. (2) The same agent runs on every version in a fresh sandbox. (3) The paired outputs audit both the agent, which should track the evidence, and the evaluator, which should score an evidence-following answer above one that keeps the old key.

The ambiguity is not hypothetical. An agent may run a correct analysis, report that the evidence is missing and still return the key (App. B), a run that looks strong under both outcome and process scoring. Separating grounded from ungrounded answers therefore requires an intervention on E whose effect on s^* is known, which is what a verified counterfactual provides.

4.2 Constructing verified counterfactuals

For each item we pre-register the reference statistic and build one version of the data for each intervention class (Fig. 2). *Null versions* permute cells and genes and rename barcodes and donors. *Withdrawal versions* delete the queried gene, remove the contrast group or equalise the candidate groups. *Flip versions* move a gene’s expression to another cell type, swap two genes or down-sample one population in one condition. Each edit touches only the quantity the statistic reads and is applied to the expression matrix, all layers and the raw counts. A version counts as verified only when the statistic, recomputed on every file and on the raw counts, behaves as its class requires. The correct response of a grounded agent is therefore known for every version.

Keeping the counterfactuals clean. A counterfactual is informative only if the versions differ in evidence alone. Agents read metadata. In development one agent recovered a swap from a file name stored in a gzip header, and the h5ad files ship precomputed summaries that would still show the original evidence. Every file of every version, the original included, therefore passes through one canonical writer (App. D).

4.3 Scoring agents and evaluators

Agents. Writing $a(E)$ for the answer of a run on data E and $\text{flag}(E)$ for an explicit statement that the data do not support an answer, we define invariance and responsiveness over paired runs,

$$I = \Pr[a(T_0 E) = a(E)], \quad R_{\text{flip}} = \Pr[a(T_C E) = c'], \quad (2)$$

$$R_{\text{rm}} = \Pr[a(T_B E) \neq a(E) \vee \text{flag}(T_B E)], \quad R_{\text{rm}}^{\text{flag}} = \Pr[\text{flag}(T_B E)].$$

We compare I with the agreement between two runs on the unmodified data, which separates sensitivity to the intervention from sampling variation. R_{rm} measures whether the answer depends on the withdrawn evidence at all, the counterfactual dependence of Proposition 1, while $R_{\text{rm}}^{\text{flag}}$ measures whether the response is appropriate, and we report both.

We introduce *evidence-grounded accuracy* (EGA), which credits a correct answer only when it also reacts to withdrawal and follows the flip,

$$\text{EGA} = \Pr[a(E) = c^* \wedge (a(T_B E) \neq a(E) \vee \text{flag}(T_B E)) \wedge a(T_C E) = c']. \quad (3)$$

The strict variant $\text{EGA}^{\text{strict}}$ replaces the withdrawal condition by $\text{flag}(T_B E)$. We assess null invariance separately with I . EGA combines correctness with responsiveness to the two interventions that change

the evidence. EGA does not penalise prior knowledge. It asks only whether the answer remains responsive to the evidence supplied in the current run.

Paired runs also make the route to each correct answer observable. Classifying each correct run by whether its withdrawal run *reacts*, by changing the answer or flagging that the data cannot decide, and whether its flip run follows s^* to c' , we decompose accuracy as

$$\text{Acc} = \underbrace{P_{\text{derived}}}_{\text{reacts, follows}} + \underbrace{P_{\text{elim}}}_{\text{persists, follows}} + \underbrace{P_{\text{rm-only}}}_{\text{reacts, keeps } c^*} + \underbrace{P_{\text{recalled}}}_{\text{persists, keeps } c^*} + P_{\text{other}}, \quad (4)$$

where P_{other} collects flip runs that move to a third option or give no answer. Only the derived route is evidence-grounded, so $P_{\text{derived}} = \text{EGA}$ and the other components are the part of the score that the data do not certify. An answer *selected by elimination* persists under withdrawal because a withdrawal removes the evidence for the key but not the evidence against the other options, so in a closed option set the key can still be reached by ruling the others out. A *recalled* answer persists under withdrawal and keeps c^* after a flip, consistent with knowledge outside the supplied data, and $P_{\text{rm-only}}$ reacts when the evidence disappears but not when it contradicts the key. We also report the *evidence gain*, accuracy with data minus accuracy without, and, for synthetic problems whose ground truth transforms exactly under a flip, *transform consistency*, which compares the flipped answer with the same transform of the agent’s own original answer and so measures responsiveness even when the original answer misses the tolerance.

Evaluators. Throughout, the evaluator is given the original key c^* , as a benchmark would be, since it is not told that the data were edited. We call an evaluator *counterfactually valid* if, over paired runs, it is stable under null interventions up to run-to-run variation, does not credit c^* once the edited data no longer support it, and scores a run that follows a verified flip above one that does not. Writing $M(\cdot)$ for its score of a run against the original key, we measure the three conditions by

$$S_{\text{null}} = \mathbb{E}[M(E) - M(T_0E)], \quad B_c = \mathbb{E}[M(T_cE)] \quad (c \in \{\text{rm}, \text{flip}\}), \quad (5)$$

$$\text{AUC}_{\text{flip}} = \Pr[M(r^+) > M(r^-)] + \frac{1}{2} \Pr[M(r^+) = M(r^-)].$$

The null sensitivity S_{null} should not exceed the change between two runs on the original data, the blind spot B_c should be near zero, and the flip AUC, which compares a flip run r^+ answering c' with one r^- that does not, should be near 1. Evidence grounding is thus a property of the agent and counterfactual validity a property of the evaluator, and both are measured on the same pairs.

5 Experimental setup

Tasks. We use BAISBench’s scientific-discovery split (BAIS-SD) [8] and GeneBench-Pro [9]. Of 193 BAIS-SD questions, 22 can be decided from the supplied single-cell data by a reference statistic. Four of these were excluded because the original data do not support the benchmark key, leaving 18 questions from 13 studies. The official score is an exact set match with half credit for a strict subset of the key. GeneBench-Pro contributes four synthetic statistical-genetics problems whose ground truth transforms exactly under specified data edits, and its grader requires every reported quantity to fall within tolerance. More details on the tasks and their interventions are provided in App. C.

Agents. We evaluate three scaffolds and five models. Claude Code runs Claude Opus 5.5, Sonnet 5 and Haiku 4.5, and Codex CLI runs GPT-5.6-Sol and GPT-5.6-Terra, each with three replicates. Biomni [48] runs Claude Sonnet 5 with one replicate. Each run starts in a fresh sandbox with only its task files and no web or retrieval access, receives the same prompt on every data version, and writes its answer, a confidence and whether the data support the answer. A no-data condition provides only the question and background. More details on the agent protocol are provided in App. D.

Evaluators. We stress-test three kinds of evaluator on the same paired runs: the benchmark’s official score against the original key, an LLM process judge that reads the question, the key and a condensed trajectory (Claude Opus 5.5 and GPT-5.6-Sol), and the agent’s own report of its confidence and of whether the data support its answer. Separately, an audit of all 2,944 included runs (16,091 tool calls) for access outside the sandbox or to benchmark answer material flagged none. More details on the judges and the audit are provided in App. E and App. D.

Research questions. We organise the results around four questions. *RQ1* asks whether agents track the evidence when it is withdrawn or reversed (Sec. 6.1). *RQ2* asks how much of their accuracy is evidence-grounded (Sec. 6.2). *RQ3* asks why agents fail to follow the evidence, and separates prior knowledge from analytical skill (Sec. 6.3). *RQ4* asks whether the benchmark score and LLM judges notice these failures, and whether they can be repaired (Sec. 6.4).

6 Results and analysis

6.1 RQ1: Do agents track the evidence?

If an agent’s answers come from the data, they should stay the same under edits that leave the evidence intact and change when the evidence behind the key is withdrawn or reversed. We ran every agent on all three versions of each item under the same prompt and compare the paired answers.

Table 2: Accuracy, invariance and responsiveness (%). Claude rows: 18 (BAIS) or 4 (GB-Pro) tasks $\times$ 3 replicates. Accuracy: official score on E and with no data. I_{rep} : agreement of two runs on E . I_{null} : same answer after a null edit. $R_{\text{rm}}^{\text{flag}}$: the agent reports that the withdrawn data do not support an answer. R_{flip} : the answer follows the flip (GB-Pro: passes against the transformed ground truth). B : credit against the original key after withdrawal or flip. EGA: Eq. (3), not defined for GB-Pro. † Codex CLI, 3 replicates each. ‡ Biomni on Sonnet 5, 1 replicate, no separate no-data run.

Bench.	Model	Benchmark accuracy		Invariance		Responsiveness		Blind spot B		EGA
		E	no data	I_{rep}	I_{null}	$R_{\text{rm}}^{\text{flag}}$	R_{flip}	rm	flip	
BAIS	Opus 5.5	98	85	96	98	41	59	85	33	35
	Sonnet 5	100	89	100	100	31	52	89	41	31
	Haiku 4.5	87	74	96	94	31	72	56	15	56
	GPT-5.6-Sol †	100	81	100	98	31	65	81	28	33
	GPT-5.6-Terra †	94	80	93	94	37	63	78	31	35
	Biomni (Sonnet 5) ‡	100	–	100	94	28	67	72	28	39
GB-Pro	Opus 5.5	42	–	100	58	50	25	0	0	–
	Sonnet 5	17	–	25	8	25	8	0	0	–
	Haiku 4.5	0	–	17	17	33	0	0	0	–

➡ **Answers are stable when the evidence is intact.** Null edits, which shuffle cells and genes and rename identifiers, change answers no more often than rerunning does (98% agreement in both cases, Table 2), so the answer changes reported below reflect the evidence rather than chance.

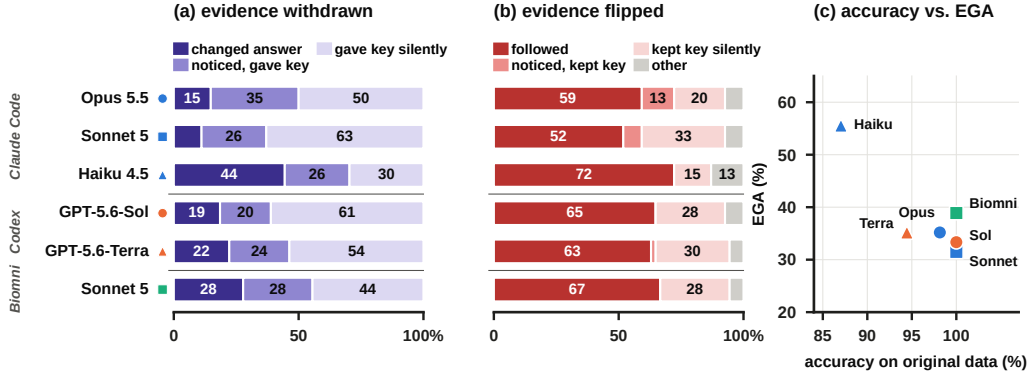


Figure 3: **Agents notice but do not act.** What each agent does when the key’s evidence is (a) withdrawn or (b) flipped (% of paired runs, complete replicates only). “Noticed”: the agent reports that the data do not support an answer, yet gives the key. (c) Accuracy on the original data against EGA. The two orderings differ markedly. Colour: scaffold. Marker: model (Biomni runs Sonnet 5).

➡ **Withdrawn evidence is noticed but not acted on.** When the evidence behind the key is withdrawn, the benchmark still awards 77% of its credit to the Claude agents (Table 2). The agents do register the loss (App. F), yet only 35% of withdrawal runs state that the data cannot decide, and 84% of those give the key regardless. As Fig. 3a shows, the forced choice turns this uncertainty into a committed answer, which the trajectories show is often reached by ruling out the other options or from memory (App. B, App. I). The benchmark score cannot tell such an answer from analysis.

➡ **Even reversed evidence often fails to move the answer.** A reversal is a sharper test, because the data now point to a different option. Only 61% of runs follow it, and in 30% the agent keeps the key and is still credited (Fig. 3b). A scientist might reasonably distrust data that contradict known biology, but most runs that keep the key never mention a conflict, and reversals that look biologically ordinary,

such as a shift in cell-type proportions, are followed only slightly more often (67%). The effect holds in every scaffold and without the two tasks that admit a second valid analysis (App. C, App. F). The original answer remains sticky even when the supplied evidence supports another option.

6.2 RQ2: How much of the accuracy is evidence-grounded?

Accuracy credits every correct answer alike. Using the paired runs, we ask how much of it remains once a correct answer must also respond to both edits, and by which routes the rest is reached.

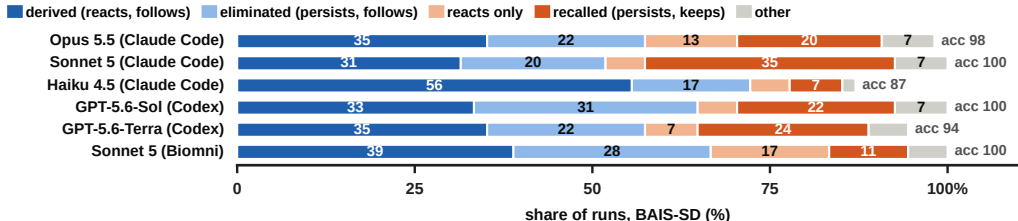


Figure 4: **Where the accuracy comes from.** Each correct BAIS-SD run is classified by its paired withdrawal and flip runs (Eq. 4). Only the dark-blue segment, runs that react to withdrawal and follow the flip, is evidence-grounded accuracy. Bars sum to accuracy. Complete replicates only.

➡ **Most correct answers do not meet both response conditions.** By this standard the Claude agents fall from 95% accuracy to 41% EGA, and every agent loses more than a third of its accuracy (Table 2). Figure 4 breaks down the rest. Fewer than half of the correct runs react to withdrawal and follow the flip (the derived route), roughly one in five follows the elimination pattern and a similar share the recall pattern. The two measures also order the agents differently, as Fig. 3c shows. Sonnet and GPT-5.6-Sol answer every item correctly yet have the lowest EGA, while Haiku, the least accurate, has the highest (Kendall’s $\tau = -0.54$, not significant with six agents). Part of Haiku’s lead comes from changing its answer after withdrawal without flagging missing evidence. The strict variant, which requires such a flag, reduces all agents to 17–28% and leaves the ordering inverted.

6.3 RQ3: Why do agents not follow the evidence?

An agent may fail to follow the evidence because it cannot analyse the data well, or because what it already knows overrides the data. We separate them by removing the prior but not the evidence.

➡ **No-data accuracy and the recall pattern covary.** Given only the question and background, the Claude models answer 83% correctly and the GPT models about as many, so the data add only 12 percentage points. Sonnet answers the most questions without data and shows the recall pattern most often, and Haiku does both least (Fig. 4), suggesting that the prior competes with the data.

➡ **Hiding gene names makes agents follow reversed evidence.** We replaced every gene symbol with an opaque identifier in the 10 gene tasks, consistently in the data, question, options and background (App. G). This preserves the underlying evidence while removing gene identity as a usable prior. As Fig. 5 shows, accuracy without data falls from 87% to 27%, near chance, while accuracy with data barely changes, so the tasks remain solvable from the data alone. Under reversal, agents now follow the flipped evidence in 93% of runs instead of 58%, and EGA rises from 39% to 69%. The GPT agents change the same way, and the effect remains without the one task whose study the models still recognise. Anonymisation also removes semantic cues, so this supports prior competition as an important explanation rather than proving it the only one. Withdrawal improves less (key credit 79% to 52%), consistent with the forced choice still allowing answers by elimination.

➡ **Without a competing prior, analysis quality becomes the limit.** GeneBench-Pro offers little task-specific prior to fall back on. There, the agents’ answers on flipped data transform with the ground truth about as often as repeated runs on unchanged data agree, although accuracy stays low (App. F). Without a competing prior, responsiveness appears limited mainly by analytical precision.

6.4 RQ4: Do evaluators notice, and can they be repaired?

If agents often ignore changed evidence, a valid evaluator should notice. We applied the stress test of Sec. 4.3 to the benchmark score, LLM judges and the agents’ self-reports, and tested targeted repairs.

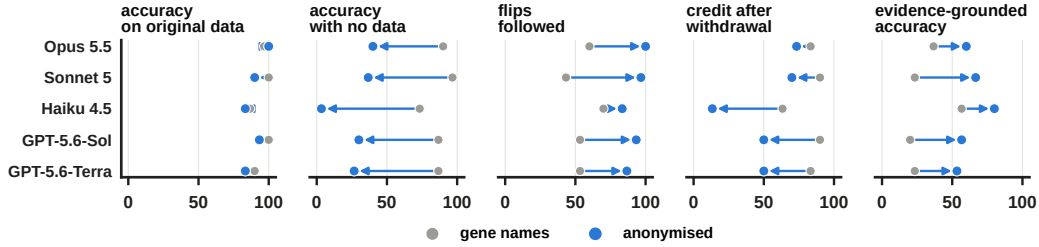


Figure 5: **Gene-name anonymisation.** The same 10 gene tasks with every gene symbol replaced by an opaque identifier in data and text (grey: names, blue: anonymised). The no-data prior falls towards chance, accuracy with data barely changes, and the same agents now follow the flipped evidence.

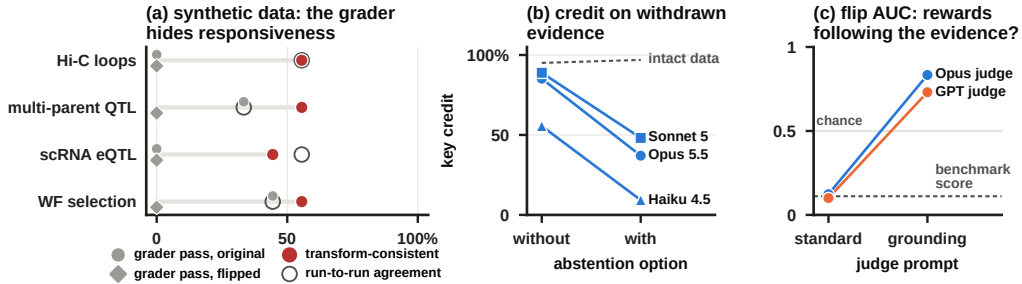


Figure 6: **What the score hides, and what repairs it.** (a) GeneBench-Pro (Claude models). The grader passes almost no run on either version, yet transform-consistent answers are as frequent as run-to-run agreement. (b) Offering “the provided data are insufficient” cuts key credit on withdrawn evidence for Claude Code agents and leaves intact-data accuracy unchanged. (c) Flip AUC, the probability that a run following the flip outscores one that does not. Under the standard prompt both judges reward ignoring the flip, like the benchmark score. The grounding prompt reverses this.

➡ **Neither the score nor LLM judges reward following the evidence.** The benchmark score is stable under null edits, but falls under flips whenever the answer leaves the key, so a run that follows the flipped data scores below one that ignores them nearly nine times in ten (flip AUC 0.11). An LLM judge sees the condensed analysis, including statements that evidence is missing, yet grades against the same key and behaves alike (flip AUC 0.12). A GPT judge does the same (0.10), so the blind spot is not family-specific (App. E). Self-reports separate the two cases better, since runs that keep the key after a flip rarely claim full support (13%), unlike runs that follow it (84%). Synthetic data reverse the failure. As Fig. 6a shows, the grader scores most runs zero on every version, so a run tracking the transformation but missing the tolerance looks like one that ignores the edit.

➡ **Targeted repairs fix one failure at a time.** An explicit “data are insufficient” option leaves intact-data accuracy unchanged and more than halves withdrawal credit, but fails under flips, where the data support an answer (Fig. 6b). Asking judges to grade grounding rather than agreement lifts their flip AUC above 0.7 at no cost on null edits, but leaves part of the withdrawal blind spot (Fig. 6c). Transform consistency credits synthetic answers that move with the data despite missing the tolerance. No repair covers every failure, and none can be validated without paired counterfactuals (App. H).

7 Conclusion

Science advances when evidence can change a conclusion, yet high accuracy on a data-driven benchmark does not establish that an AI scientist used the supplied data. We tested this assumption with verified, item-level counterfactuals that preserve, withdraw or reverse the evidence. On our tasks, agents often retained the original answer after the evidence changed, while benchmark scores and LLM judges continued to reward it: Claude agents averaged 95% accuracy but only 41% evidence-grounded accuracy. Hiding gene names increased responsiveness to reversed evidence, suggesting that prior knowledge can override evidence the agents are able to analyse. As models learn more, accuracy may increasingly measure what an agent knows rather than what it can infer. Evaluations of data-driven AI scientists should therefore test whether both answers and scores respond to verified changes in evidence, and report evidence-grounded accuracy alongside conventional accuracy.

References

- [1] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar *et al.*, “Holistic evaluation of language models,” *arXiv preprint arXiv:2211.09110*, 2022.
- [2] A. M. Bean, R. O. Kearns, A. Romanou, F. S. Hafner, H. Mayne, J. Batzner, N. Foroutan Eghlidi, C. Schmitz, K. Korgul, H. Batra *et al.*, “Measuring what matters: Construct validity in large language model benchmarks,” *Advances in Neural Information Processing Systems*, vol. 38, 2026.
- [3] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha, “The ai scientist: Towards fully automated open-ended scientific discovery,” *arXiv preprint arXiv:2408.06292*, 2024.
- [4] J. Gottweis, W.-H. Weng, A. Daryin, T. Tu, P. Sirkovic, A. Myaskovsky, G. Glowaty, F. Weissenberger, A. Orlandi, D. Popovici *et al.*, “Accelerating scientific discovery with co-scientist,” *Nature*, pp. 1–3, 2026.
- [5] K. Swanson, W. Wu, N. L. Bulaong, J. E. Pak, and J. Zou, “The virtual lab: Ai agents design new sars-cov-2 nanobodies with experimental validation,” *bioRxiv*, 2024. [Online]. Available: <https://www.biorxiv.org/content/early/2024/11/12/2024.11.11.623004>
- [6] J. M. Laurent, J. D. Janizek, M. Ruzo, M. M. Hinks, M. J. Hammerling, S. Narayanan, M. Ponnapati, A. D. White, and S. G. Rodriques, “Lab-bench: Measuring capabilities of language models for biology research,” *arXiv preprint arXiv:2407.10362*, 2024.
- [7] L. Mitchener, J. M. Laurent, A. Andonian, B. Tenmann, S. Narayanan, G. P. Wellawatte, A. White, L. Sani, and S. G. Rodriques, “Bixbench: a comprehensive benchmark for llm-based agents in computational biology,” *arXiv preprint arXiv:2503.00096*, 2025.
- [8] E. Luo, J. Jia, Y. Xiong, X. Li, X. Guo, B. Yu, M. Hao, L. Wei, and X. Zhang, “Benchmarking ai scientists for omics data-driven biological discovery,” *Bioinformatics*, vol. 42, no. Supplement_1, p. btg227, 2026.
- [9] J. Li and A. Ho, “Genebench-pro: Evaluating multistage statistical reasoning in genomics, quantitative biology, and translational biomedicine,” *bioRxiv*, 2026. [Online]. Available: <https://www.biorxiv.org/content/early/2026/06/30/2026.06.29.735386.1>
- [10] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, “Beyond accuracy: Behavioral testing of nlp models with checklist,” in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 4902–4912.
- [11] M. Gardner, Y. Artzi, V. Basmov, J. Berant, B. Bogin, S. Chen, P. Dasigi, D. Dua, Y. Elazar, A. Gottumukkala *et al.*, “Evaluating models’ local decision boundaries via contrast sets,” in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 1307–1323.
- [12] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith, “Annotation artifacts in natural language inference data,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 107–112.
- [13] N. Balepur, A. Ravichander, and R. Rudinger, “Artifacts or abduction: How do llms answer multiple-choice questions without the question?” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 10 308–10 330.
- [14] L. Chen, J. Li, X. Dong, P. Zhang, Y. Zang, Z. Chen, H. Duan, J. Wang, Y. Qiao, D. Lin *et al.*, “Are we on the right way for evaluating large vision-language models?” *Advances in Neural Information Processing Systems*, vol. 37, pp. 27 056–27 087, 2024.
- [15] M. Turpin, J. Michael, E. Perez, and S. Bowman, “Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 74 952–74 965, 2023.

- [16] Y. Chen, J. Benton, A. Radhakrishnan, J. Uesato, C. Denison, J. Schulman, A. Somani, P. Hase, M. Wagner, F. Roger *et al.*, “Reasoning models don’t always say what they think,” *arXiv preprint arXiv:2505.05410*, 2025.
- [17] M. Zhuge, C. Zhao, D. R. Ashley, W. Wang, D. Khizbullin, Y. Xiong, Z. Liu, E. Chang, R. Krishnamoorthi, Y. Tian, Y. Shi, V. Chandra, and J. Schmidhuber, “Agent-as-a-judge: Evaluate agents with agents,” in *Proceedings of the 42nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR, 13–19 Jul 2025, pp. 80 569–80 611.
- [18] Z. Koch, A. T. Wassie, J. Valdes-Aleman, J. Lee, M. M. Hinks, S. G. Rodrigues, A. D. White, and J. M. Laurent, “Bixbench3: Benchmarking ai agents on research-study-scale computational biology tasks,” *arXiv preprint arXiv:2608.25286*, 2026.
- [19] D. Kaushik, E. Hovy, and Z. C. Lipton, “Learning the difference that makes a difference with counterfactually-augmented data,” *arXiv preprint arXiv:1909.12434*, 2019.
- [20] S. S. Panigrahi, J. Videnović, and M. Brbic, “Heurekabench: A benchmarking framework for ai co-scientist,” in *International Conference on Learning Representations*, vol. 2026, 2026, pp. 11 029–11 061.
- [21] B. P. Majumder, H. Surana, D. Agarwal, B. Dalvi Mishra, A. Meena, A. Prakhar, T. Vora, T. Khot, A. Sabharwal, and P. Clark, “Discoverybench: Towards data-driven discovery with large language models,” in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 4556–4579.
- [22] I. Diks, Z. Yang, A. Banerjee, T. Proctor, and K. Workman, “scbench-long: Verifiable benchmarking of long-horizon single-cell biology,” *arXiv preprint arXiv:2606.26563*, 2026.
- [23] X. Yang, X. Yang, S. Fang, Y. Zhang, J. Wang, B. Xian, Q. Li, J. Li, M. Xu, Y. Li *et al.*, “R&d-agent: An llm-agent framework towards autonomous data science,” *arXiv preprint arXiv:2505.14738*, 2025.
- [24] L. Sun, Y. Han, Z. Zhao, D. Ma, Z. Shen, B. Chen, L. Chen, and K. Yu, “Scieval: A multi-level large language model evaluation benchmark for scientific research,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 053–19 061.
- [25] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, “Measuring massive multitask language understanding,” *arXiv preprint arXiv:2009.03300*, 2020.
- [26] L. Phan, A. Gatti, Z. Han, N. Li, J. Hu, H. Zhang, C. B. C. Zhang, M. Shaaban, J. Ling, S. Shi *et al.*, “Humanity’s last exam,” *arXiv preprint arXiv:2501.14249*, 2025.
- [27] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman, “Gpqa: A graduate-level google-proof q&a benchmark,” *arXiv preprint arXiv:2311.12022*, 2023.
- [28] Z. Chen, S. Chen, Y. Ning, Q. Zhang, B. Wang, B. Yu, Y. Li, Z. Liao, C. Wei, Z. Lu *et al.*, “Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery,” in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 96 934–96 990.
- [29] M. Tian, L. Gao, S. D. Zhang, X. Chen, C. Fan, X. Guo, R. Haas, P. Ji, K. Krongchon, Y. Li *et al.*, “Scicode: A research coding benchmark curated by scientists,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 30 624–30 650, 2024.
- [30] J. S. Chan, N. Chowdhury, O. Jaffe, J. Aung, D. Sherburn, E. Mays, G. Starace, K. Liu, L. Maksin, T. Patwardhan *et al.*, “Mle-bench: Evaluating machine learning agents on machine learning engineering,” in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 50 466–50 494.
- [31] Z. S. Siegel, S. Kapoor, N. Nadgir, B. Stroebel, and A. Narayanan, “Core-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark,” *arXiv preprint arXiv:2409.11363*, 2024.

- [32] G. Starace, O. Jaffe, D. Sherburn, J. Aung, J. S. Chan, L. Maksin, R. Dias, E. Mays, B. Kinsella, W. Thompson *et al.*, “Paperbench: evaluating ai’s ability to replicate ai research,” *arXiv preprint arXiv:2504.01848*, 2025.
- [33] K. Gu, R. Shang, R. Jiang, K. Kuang, R.-J. Lin, D. Lyu, Y. Mao, Y. Pan, T. Wu, J. Yu *et al.*, “Blade: Benchmarking language model agents for data-driven science,” in *Findings of the association for computational linguistics: EMNLP 2024*, 2024, pp. 13 936–13 971.
- [34] K. Workman, Z. Yang, H. Muralidharan, A. Abdulali, and H. Le, “scbench: Evaluating ai agents on single-cell rna-seq analysis,” *arXiv preprint arXiv:2602.09063*, 2026.
- [35] X. Ren, W. Zhai, T. Pu, Y. Zhu, H. Wei, and B. Zhao, “Right answer, wrong method: Shortcut hacking misleads the evaluation of llm reasoning on frontier science benchmarks,” *arXiv preprint arXiv:2608.02442*, 2026.
- [36] B. Baker, J. Huizinga, L. Gao, Z. Dou, M. Y. Guan, A. Madry, W. Zaremba, J. Pachocki, and D. Farhi, “Monitoring reasoning models for misbehavior and the risks of promoting obfuscation,” *arXiv preprint arXiv:2503.11926*, 2025.
- [37] Z. Zhong, A. Raghunathan, and N. Carlini, “Impossiblebench: Measuring llms’ propensity of exploiting test cases,” in *International Conference on Learning Representations*, vol. 2026, 2026, pp. 124 959–124 986.
- [38] N. Alzahrani, H. Alyahya, Y. Alnumay, S. Alrashed, S. Alsubaie, Y. Almushayqih, F. Mirza, N. Alotaibi, N. Al-Twairesh, A. Alowisheq *et al.*, “When benchmarks are targets: Revealing the sensitivity of large language model leaderboards,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 13 787–13 805.
- [39] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe, “Let’s verify step by step,” in *International Conference on Learning Representations*, B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, Eds., vol. 2024, 2024, pp. 39 578–39 601. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf
- [40] X. H. Lù, A. Kazemnejad, N. Meade, A. Patel, D. Shin, A. Zambrano, K. Stańczak, P. Shaw, C. J. Pal, and S. Reddy, “Agentrewardbench: Evaluating automatic evaluations of web agent trajectories,” *arXiv preprint arXiv:2504.08942*, 2025.
- [41] A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme, “Hypothesis only baselines in natural language inference,” in *Proceedings of the seventh joint conference on lexical and computational semantics*, 2018, pp. 180–191.
- [42] D. Kaushik and Z. C. Lipton, “How much reading does reading comprehension require? a critical investigation of popular benchmarks,” in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 5010–5015.
- [43] T. Y. Chen, F.-C. Kuo, H. Liu, P.-L. Poon, D. Towey, T. H. Tse, and Z. Q. Zhou, “Metamorphic testing: A review of challenges and opportunities,” *ACM Comput. Surv.*, vol. 51, no. 1, Jan. 2018. [Online]. Available: <https://doi.org/10.1145/3143561>
- [44] Y. Gu, J. Fu, X. Liu, J. M. J. Valanarasu, N. Codella, R. Tan, Q. Liu, Y. Jin, S. Zhang, J. Wang, R. Wang, L. Song, G. Qin, N. Usuyama, C. Wong, H. Cheng, H. Lee, P. Sanapathi, S. Hilado, T. Naumann, J. Alvarez-Valle, J. Bian, M. Wei, K. Malik, L. Zhou, J. Gao, E. Horvitz, M. P. Lungren, D. Burger, E. Topol, H. Poon, and P. Vozila, “Evaluating the robustness and readiness of large frontier models in health ai applications.” *Nature Medicine*, June 2026.
- [45] A. Zafar, L. K. Murali, and A. Vashist, “Beyond accuracy: Evaluating visual grounding in multimodal medical reasoning,” *arXiv preprint arXiv:2603.03437*, 2026.
- [46] S. Longpre, K. Perisetla, A. Chen, N. Ramesh, C. DuBois, and S. Singh, “Entity-based knowledge conflicts in question answering,” in *Proceedings of the 2021 conference on empirical methods in natural language processing*, 2021, pp. 7052–7063.

- [47] J. Xie, K. Zhang, J. Chen, R. Lou, and Y. Su, “Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts,” in *International Conference on Learning Representations*, vol. 2024, 2024, pp. 35 623–35 646.
- [48] K. Huang, S. Zhang, H. Wang, Y. Qu, Y. Lu, Y. Roohani, R. Li, L. Qiu, G. Li, J. Zhang, D. Yin, S. Marwaha, J. N. Carter, X. Zhou, M. Wheeler, J. A. Bernstein, M. Wang, P. He, J. Zhou, M. Snyder, L. Cong, A. Regev, and J. Leskovec, “Biomni: A general-purpose biomedical ai agent,” *bioRxiv*, 2025. [Online]. Available: <https://www.biorxiv.org/content/early/2025/06/02/2025.05.30.656746>
- [49] J. Y. Mimpfen, L. Ramos-Mucci, C. Paul, A. Kurjan, P. Hulley, C. Ikwuanusi, S. Gwilym, M. J. Baldwin, A. P. Cribbs, and S. J. Snelling, “Single nucleus and spatial transcriptomic profiling of human healthy hamstring tendon,” *bioRxiv*, 2022. [Online]. Available: <https://www.biorxiv.org/content/early/2022/12/20/2022.12.19.521110>

A Discussion

What the score measures. Benchmark accuracy measures whether an answer meets the scoring criteria, but does not establish whether it depends on the supplied data. EGA measures behaviour under our interventions rather than the reasoning inside a run, so the routes in Eq. (4) are behavioural patterns consistent with recall or elimination, not observations of them. For withdrawal, the plain variant accepts either a changed answer or an explicit statement that the data cannot decide, and the strict variant requires the latter.

What the evaluator results show. All runs were scored against the original key, as in normal use. The exact-match score still uses the original key after a flip, so it penalises answers that follow the changed data. It therefore measures agreement with the key, not whether the answer follows the supplied evidence. LLM judges could in principle credit an analysis that contradicts the key, yet under the standard prompt they followed it, and only a grounding prompt changed this (App. H).

Recommendations. We recommend four practices. Data-driven benchmarks should ship each item with a verified withdrawal and flip and report both EGA variants and the no-data prior next to accuracy. They should offer an explicit “insufficient data” option. They should validate process judges on counterfactual pairs and credit transform consistency where the ground truth transforms exactly. Last, they should pass every file version, the original included, through one canonical writer.

Limitations. Our study was limited mainly by compute and budget. Each counterfactual is built and verified by hand and every version requires a full agent run, so we covered 18 BAIS-SD questions and four GeneBench-Pro problems, ran three replicates per model, a single replicate for Biomni, and tested gene-name anonymisation on ten tasks only. Larger task sets, more replicates, open-weight models and a human validation of the LLM judges would tighten our estimates. Finally, responsiveness to evidence is necessary for data-driven reasoning but not sufficient for scientific quality.

B A worked example of the definitions

This section follows one BAIS-SD item through every definition of Sec. 4. It is the item described in the introduction, and all answers below are from our runs.

The item. The task x asks “Which of the following gene expression markers was used to identify endothelial cells in the human hamstring tendon?” with options A) PDGFRB, B) TRDN, C) PECAM1 and D) PAX7. The data E are the single-nucleus RNA-seq file of the study [49] and the key is $c^* = C$.

The reference statistic and s^* . The task asks for a marker of one cell type, so the statistic is, for each option gene, its mean log-normalised expression in endothelial cells minus its mean in all other cells. The data-supported answer $s^*(E)$ is the option with the largest value. On the original file this is PECAM1, so $s^*(E) = c^* = C$ and the key is supported by the data.

The three interventions. The null version T_0E permutes the order of cells and genes and renames cell barcodes and donors. The statistic is unchanged, so $s^*(T_0E) = C$. The withdrawal T_BE deletes PECAM1 from the expression matrix, every layer and the raw counts. The statistic can no longer be computed for the key, so the data no longer decide the task and $s^*(T_BE)$ is undefined. The flip T_CE swaps the values of PECAM1 and PAX7 in every cell. PAX7 now has the endothelial profile, so $s^*(T_CE) = D = c'$. We confirm these outcomes by recomputing the reference statistic on every file and on the raw counts.

The loop and noticing without acting. On T_BE , one observation o_i in the trajectory of Opus 5.5 leads it to state that “PECAM1 is not in the dataset’s gene list”. Its final answer is nevertheless C, “the only option left”. The edit reached the observations but not the answer. This is noticing without acting in the sense of Sec. 4.1. On E its trajectory and answer are those of a careful analyst.

Agent metrics and routes. Table 3 lists the answers of every run on this item. All agents answer C on E and on T_0E , so accuracy and invariance are both 100%. The runs then separate by route. Haiku flags the withdrawn evidence and follows the flip to D in all three replicates, which is the *derived* pattern. Two GPT-5.6-Terra runs give C without a flag under withdrawal but follow the flip, which is the pattern of *elimination*. The other options are well-known markers of pericytes, muscle and satellite cells, so C can be chosen without PECAM1 being measured. The remaining eleven runs keep C under both edits, which is the *recall* pattern. No run on this item reacts to withdrawal

alone. Accuracy on the item is therefore $16/16$, $R_{\text{m}} = 3/16$, $R_{\text{flip}} = 5/16$ and $\text{EGA} = 3/16$. The gap between 16 and 3 is exactly the elimination and recall mass of Eq. (4).

Evaluator metrics. The benchmark scores by exact match with the key C. On the null pairs every score is unchanged, so its sensitivity is zero, as it should be. On the withdrawal versions it gives full credit to all sixteen runs, including the three Haiku runs that stated the data cannot decide, because they still selected C when forced to choose. On the flip versions it gives full credit to the eleven runs that kept C and zero to the five that followed the evidence to D. Its blind spot on this item is 100% after withdrawal and 11/16 after the flip, and its flip AUC is 0. Every run that follows the evidence scores below every run that does not. A counterfactually valid evaluator would reverse this order.

Table 3: All runs on the worked example (S23_Q3). Answers on the original data E , the null version, the withdrawal (with the agent’s data-support label) and the flip, and the resulting route. The key is C on E , and the flip moves the data-supported answer to D. Rows group runs with identical answers. The three replicates of GPT-5.6-Terra split into two rows, and Biomni was run once.

agent	runs	E	T_0E	T_BE	T_CE	route
Haiku 4.5	3	C	C	C, flagged	D	derived
GPT-5.6-Terra	2	C	C	C, partial support	D	eliminated
GPT-5.6-Terra	1	C	C	C, partial support	C	recalled
Opus 5.5	3	C	C	C, partial support	C	recalled
Sonnet 5	3	C	C	C, partial support	C	recalled
GPT-5.6-Sol	3	C	C	C, partial support	C	recalled
Biomni (Sonnet 5)	1	C	C	C, partial support	C	recalled

C Benchmarks and tasks

BAISBench. BAISBench [8] evaluates AI scientists on omics data-driven biological discovery with two tasks, cell type annotation and scientific discovery. Its scientific-discovery split (BAIS-SD) poses 193 multiple-choice questions derived from conclusions reported in 41 published single-cell studies. Each question comes with the study’s h5ad files and a short background, and one or more options form the key. The official score is an exact set match with half credit for a strict subset of the key. Of the 193 questions, 22 can be decided from the supplied data by a single reference statistic, and 18 remain after the exclusions below.

GeneBench-Pro. GeneBench-Pro [9] evaluates multistage statistical reasoning in genomics, quantitative biology and translational biomedicine with 129 evaluations. Each problem provides a synthetic dataset generated with known ground truth, brief context and a target quantity, and an answer passes only if every reported quantity falls within a problem-specific tolerance. We use the four of its ten public problems for which a data edit has an exactly known effect on the ground truth (Table 5). For the other six, no such edit was available.

The 18 BAIS-SD tasks fall into five evidence families: cell-type abundance, composition change between conditions, expression localisation, marker gene and differential expression between conditions. The four GeneBench-Pro problems cover Hi-C loop strength, multi-parent QTL mapping, single-cell eQTL and Wright–Fisher selection, and their ground truth transforms exactly under a case and control swap, a founder relabelling, an allele recoding or a locus swap. Table 4 lists the 18 BAIS-SD tasks with their withdrawal and flip operations. All tasks share the null operations: permutation of cells and genes (applied to X , all layers and `.raw`), opaque barcodes and opaque donor and sample identifiers. Every version of every task is verified against the reference statistic on each staged file and on the `.raw` counts.

Excluded candidates. Four candidates were dropped because the original data do not support the benchmark’s key. In S17_Q1, ACE2 expression in AT2 and club cells is tied. In S09_Q5, APOA4 is absent from the files. In S32_Q2, OLIG1 marks OPCs more strongly than BCAN. In S36_Q3, ISG15 is more down-regulated than IFNG.

Alternative analysis paths. Two tasks admit a second defensible analysis. In S36_Q1, disease activity is encoded in several obs columns, and in S09_Q1 the “Progenitor” cells can be read as enterocyte lineage. Without these two tasks, flips are followed in 65% of runs and withdrawal credit is 76%, compared with 61% and 77% on all tasks.

Table 4: BAIS-SD tasks. “new” is the option supported by the flipped data.

qid	family	key	withdrawal (T_B)	flip (T_C)	new
S09_Q1	abundance	B	equalise option abundances	down-sample Enterocyte	A
S07_Q1	abundance	C	equalise option abundances	down-sample mesothelial cell	D
S35_Q1	abundance	C	equalise option abundances	down-sample Pericyte, Myofibroblast, Matrix_Fibroblast, Smooth_Muscle_Cell, Stromal_Cell	A
S06_Q2	abundance	C	remove cells with cellxgene_cell_type = T cell, natural killer cell, B cell, innate lymphoid cell	remove cells with cellxgene_cell_type = natural killer cell, B cell, innate lymphoid cell	A
S27_Q1	composition change	C	remove cells with Disease group = mild COVID-19, mild COVID-19 (asymptomatic)	down-sample classical monocyte and down-sample natural killer cell	B
S19_Q2	composition change	A	remove cells with disease = normal	down-sample AT1 and down-sample Endothelial cells	B
S36_Q1	composition change	A	remove cells with activity_python_binary_crp = Yes	down-sample CD4 T central memory and down-sample Naive Bcells	C
S03_Q3	expression localisation	B	delete gene TIMP3	move TIMP3 expression macroglial cell↔retina horizontal cell	A
S38_Q3	expression localisation	D	delete gene RHO	move RHO expression retinal rod cell↔retinal cone cell	A
S00_Q4	expression localisation	D	delete gene TMPRSS2	move TMPRSS2 expression keratinizing barrier epithelial cell↔club cell	A
S09_Q2	marker gene	A	delete gene NUSAP1	swap NUSAP1↔SOX9	C
S09_Q4	marker gene	A	delete gene ITLN1	swap ITLN1↔MYC	C
S07_Q2	marker gene	B	delete gene WT1	swap WT1↔CDH1	D
S35_Q2	marker gene	B	delete gene HES1	swap HES1↔PTPRC	D
S23_Q3	marker gene	C	delete gene PECAM1	swap PECAM1↔PAX7	D
S36_Q2	marker gene	B	delete gene ISG15	swap ISG15↔CXCR4	D
S16_Q1	condition DE	A	remove cells with MUTATION.GROUP = none	swap CXCL3↔IL10	D
S21_Q2	condition DE	B	remove cells with disease = pulpitis	swap VCAM1↔COL1A1	C

Table 5: GeneBench-Pro problems and interventions. Null versions shuffle line order in every file. The flip changes the ground truth by a known transform.

problem	withdrawal (T_B)	flip (T_C)	effect on ground truth
Hi-C loop strength	case replicates withheld	case and control swapped	Δ negated
multi-parent QTL	phenotypes permuted across lines within batch	founder labels permuted	high founder F5 becomes F2
scRNA eQTL	donor genotypes permuted	alleles recoded, $g \rightarrow 2 - g$	β negated
Wright-Fisher selection	locus A's trajectory replaced by B's	loci A and B swapped	selected locus becomes B

D Agent protocol

Data. Before any intervention, every version, the original included, passes through one canonical writer. It drops precomputed per-gene statistics, `varm` loadings and analysis results in `uns`, and keeps the matrices, `.raw`, layers, cell annotations and embeddings. Gene edits are applied to `X`, `.raw` and all layers. Tables are re-encoded with canonical gzip headers (the file’s own name and a zero timestamp). Files are hard-linked into the sandbox under the benchmark’s original names.

Common to all agents. Each run starts in a fresh sandbox with an opaque random path under a non-listable root and sees only its task files. It has no web, search or retrieval tools and no user settings, skills, memories or MCP servers. The prompt is identical across the versions of a task. Each run has a 40-minute limit. The agent writes `answer.json` with the answer, a confidence from 0 to 100, a data-support judgement (supports, partially supports, does not support, cannot determine) and a short statement of evidence.

Claude Code runs headless (`claude -p`) with only the Bash, Read, Write, Edit, Glob and Grep tools, a private temporary directory and background tasks disabled.

Codex CLI runs `codex exec` with web search off and every stateful or web-capable feature disabled explicitly (memories, browser and computer use, in-app browser, apps, skill search and goals). A check of all Codex transcripts found only shell commands, file edits and messages.

Biomni (A1 agent, version 0.0.8) runs with its own loop, system prompt, know-how documents, tool registry and tool retriever, the model’s API defaults and its own output limit, and without the data lake, MCP servers or self-critique. We made two changes. Biomni appends code-execution results as assistant turns, which current Claude models reject, so these results are sent in the user role with unchanged content. And network access is disabled, since name resolution is refused for every host

except the model API and child processes receive a dead HTTP proxy. Biomni runs in its own Python environment with every analysis library pinned to the version used by the other agents.

Audit. Every transcript is checked for paths outside the sandbox (other runs’ workspaces, the stored data versions and our project files, which contain the answer keys), parent-directory traversal, filesystem-wide searches, benchmark answer material and network use (URLs, HTTP libraries and package installation). No run in the analysis was flagged.

E Evaluators

Process judge. The judge mimics current process judges [18]. It receives the task, the benchmark’s reference answer, the agent’s condensed trajectory (tool calls with truncated outputs, in order) and the final answer, and it is not told that the data may have been edited. Each agent message, tool call and tool output is cut to its first 1,200 characters, and a trajectory longer than 40,000 characters keeps its first and last 20,000. It returns whether the answer matches the reference, the quality of the executed analysis (0 to 10), whether the results the agent obtained support its answer (yes, partially or no) and an overall score (0 to 10). We analyse the overall score, rescaled to $[0, 1]$, and the support rating. The main judge is Claude Opus 5.5.

Cross-family judge. A GPT-5.6-Sol judge graded the first replicate of every Claude and GPT-5.6-Sol run under both prompts. Its agreement with the Opus judge on the overall score is 0.91 (Spearman) and 0.89 (Krippendorff’s α , $n = 306$). On whether the answer is supported, weighted κ is 0.74, and on whether it is correct, κ is 1.00. Under the grounding prompt the values are 0.89, 0.87 and 0.85 ($n = 303$). The agreement sets contain the 288 first-replicate runs (18 tasks, four agents, four data versions) and 18 later-replicate runs that both judges also graded, and three runs lack a grounding-prompt grade. Table 6 gives the stress test for each judge.

Evaluator sensitivities (Fig. 7). These values are computed over all runs of the three Claude models with three replicates, so they differ slightly from Table 6, which uses the first replicate of the Claude and GPT-5.6-Sol runs graded by all four judge variants. The benchmark score is as stable under null edits as between replicates (0.02 and 0.02), and changes strongly under flips (0.67) and weakly under withdrawal (0.22). The standard judge is barely more sensitive to withdrawal (0.25) and less sensitive to flips (0.42). Its sub-score for whether the answer is supported by the agent’s analysis tracks withdrawal (0.49) but not flips (0.26). On GeneBench-Pro the grader changes only weakly under flips (0.21, against 0.08 under null edits), because it scores most runs zero on every version.

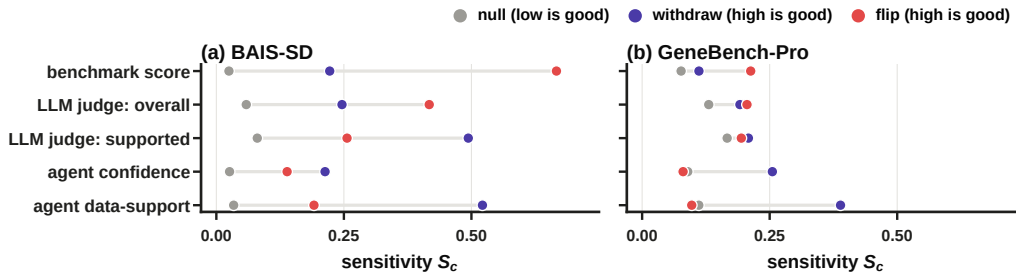


Figure 7: **Evaluator stress test** (Claude Code runs). Change in each evaluator’s score under null, withdrawal and flip interventions. The null change should stay near the change between replicates. How large the withdrawal and flip changes should be depends on how many runs follow each edit, so the flip AUC (Table 6) is the direct test of whether an evaluator rewards following the evidence.

F Additional analyses

Numbers in this section refer to the three Claude models with 3 replicates unless stated otherwise.

By evidence family (Table 7). Marker-gene questions have the highest no-data accuracy. Almost all are answerable without data, agents flag a deleted marker in only 24% of runs, and 69% of the withdrawal credit survives. Agents answer them by elimination among the remaining options, or say that the swapped labels look wrong and keep the canonical marker. Withdrawal is flagged most often for abundance questions (53%). Differential-expression questions between conditions follow flips least.

Table 6: Evaluator stress test by judge on the same runs, the first replicate of the Claude and GPT-5.6-Sol agents graded by all four judge variants (n per condition). The standard judge sees the reference answer. The grounding judge is also told that matching the reference is not sufficient. Flip AUC is the probability that a run following the flip outscores one that does not.

evaluator	S_{null}	S_{rm}	S_{flip}	flip AUC	n
Opus judge	0.06	0.25	0.44	0.12	75
GPT judge	0.05	0.31	0.56	0.10	75
Opus judge, grounding prompt	0.07	0.43	0.32	0.83	75
GPT judge, grounding prompt	0.09	0.51	0.38	0.73	75
benchmark score	0.03	0.21	0.68	0.11	75

Table 7: Results by evidence family (% , Claude models).

family	tasks	acc. E	no data	I_{null}	$R_{\text{rm}}^{\text{flag}}$	B_{rm}	R_{flip}	B_{flip}
abundance	4	100	81	100	53	64	67	33
composition change	3	89	67	96	19	78	67	0
expression localisation	3	96	74	96	52	96	81	19
marker gene	6	98	96	98	24	69	59	37
condition DE	2	83	83	94	28	94	17	61

Per task (Fig. 8). The headline results are not driven by a few items. Withdrawal leaves full credit in every run for 6 of the 18 tasks, and 2 tasks never follow the flip. Of the three tasks rarely answerable without data (at most 22%), two (S35_Q1 and S00_Q4) follow every flip, and the third (S36_Q1) is a task with an alternative analysis path.

	correct without data	credit on withdrawn	flags withdrawal	follows flip
S06_Q2 (abundance)	100	100	44	0
S09_Q1 (abundance)	100	89	44	67
S07_Q1 (abundance)	100	67	67	100
S35_Q1 (abundance)	22	0	56	100
S36_Q1 (composition)	0	78	0	0
S19_Q2 (composition)	100	78	56	100
S27_Q1 (composition)	100	78	0	100
S38_Q3 (expression)	100	100	11	56
S03_Q3 (expression)	100	100	89	89
S00_Q4 (expression)	22	89	56	100
S23_Q3 (marker)	100	100	33	33
S35_Q2 (marker)	100	78	33	33
S09_Q2 (marker)	100	67	11	56
S36_Q2 (marker)	100	56	33	67
S09_Q4 (marker)	78	11	0	78
S07_Q2 (marker)	100	100	33	89
S21_Q2 (condition)	100	100	56	11
S16_Q1 (condition)	67	89	0	22

Figure 8: Per-task rates (% , mean over Claude models and replicates), grouped by evidence family.

Self-reports (Fig. 9). Mean confidence falls from 90 on intact data to 69 after withdrawal, and only 16% of withdrawal runs still report full data support. After a flip, confidence stays at 77 and 59% of runs report full support. This share mixes two cases. Runs that follow the flip report full support in 84% of cases, as they should, while runs that keep the key do so in only 13%.

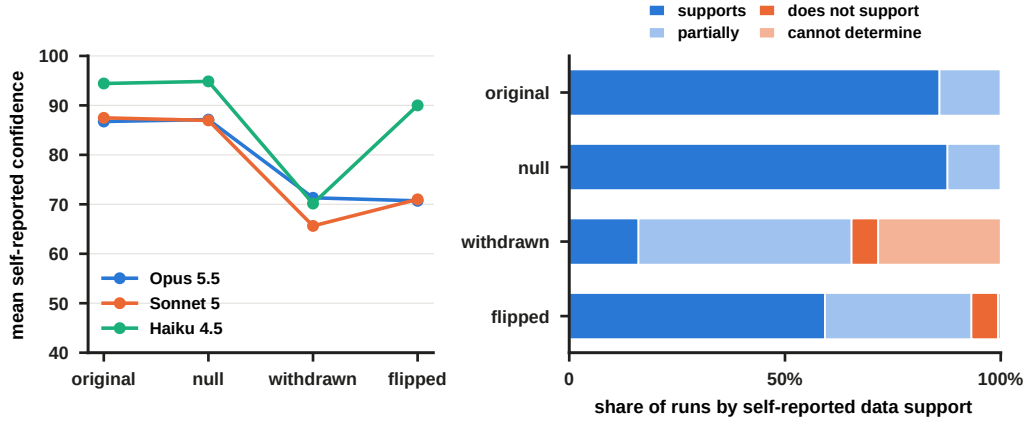


Figure 9: Self-reported confidence (left) and data support (right) by data version.

Uncertainty (Table 8). Intervals come from a bootstrap that resamples tasks, keeping each task’s models and replicates together. The blind spot on withdrawn evidence and the partial response to flips hold for every model, both GPT models included.

Table 8: Per-model estimates with task-clustered 95% bootstrap intervals (%).

model	acc. on E	B_{rm}	R_{flip}	B_{flip}
Opus 5.5	98 [94, 100]	85 [70, 98]	59 [39, 80]	33 [15, 54]
Sonnet 5	100 [100, 100]	89 [76, 100]	52 [30, 74]	41 [19, 63]
Haiku 4.5	87 [70, 100]	56 [39, 74]	72 [52, 91]	15 [0, 31]
GPT-5.6-Sol	100 [100, 100]	81 [63, 96]	65 [44, 85]	28 [11, 48]
GPT-5.6-Terra	94 [85, 100]	78 [61, 93]	63 [41, 83]	31 [13, 52]

Analysis effort (Table 9). Median wall time rises from 30 s on intact data to 49 s after withdrawal and 38 s after a flip. After withdrawal the extra analysis usually confirms that the gene or group is missing, and most runs still give the key.

Table 9: Analysis effort per run (Claude models).

	original	null	withdrawn	flipped
agent turns (median)	5	5	6	6
wall time, s (median)	30	27	49	38

GeneBench-Pro per problem (Table 10). Transform-consistent answers are common on every problem, including those where the grader almost never passes, so the strict grader hides responsiveness on all four problems. Removing the case replicates (Hi-C) or making the loci indistinguishable (selection) is flagged most often. Permuted genotypes or phenotypes (eQTL and QTL) are never flagged, and agents fit a model and report a null estimate without saying that the signal is gone. Transform consistency (53%) is similar to the agreement between repeated runs on unchanged data (47%), suggesting that run-to-run variability contributes to the inconsistent responses.

GPT-5.6-Sol shows the same pattern (50% against 50%). It follows the founder relabelling in every run and always picks the swapped locus, but its estimate of the selection coefficient varies between 0.04 and 0.10 across runs on identical data, so the selection problem reaches neither replicate agreement nor transform consistency.

Biomni shows the same pattern. The grader passes one of its four original answers (multi-parent QTL) and none of its flipped answers. Its flipped answers are transform-consistent on two problems: on Hi-C it exchanges the case and control estimates exactly, and on multi-parent QTL it moves the high-effect founder from F5 to F2 at an unchanged position, exactly as the relabelling requires. On

eQTL the effect changes sign after allele recoding (-0.38 to $+0.25$), and on selection it picks the swapped locus, but in both cases the magnitude differs from its original estimate, as for GPT-5.6-Sol.

Table 10: GeneBench-Pro per problem (%). Pass is the official grader against the original ground truth. The Claude models and GPT-5.6-Sol have three replicates per problem.

problem (flip)	Claude models				GPT-5.6-Sol
	pass on E	I_{rep}	$R_{\text{rm}}^{\text{flag}}$	$R_{\text{flip}}^{\text{rel}}$	$R_{\text{flip}}^{\text{rel}}$
Hi-C loop strength (case $\leftrightarrow$ control)	0	56	78	56	33
multi-parent QTL (founders relabelled)	33	33	0	56	100
scRNA eQTL (allele recoded)	0	56	0	44	67
Wright-Fisher selection (loci swapped)	44	44	67	56	0

Flip-following by no-data answer. Flips are followed in 62% of runs whose no-data answer was correct and in 57% of the others, so in this run-level comparison, runs on questions answerable without data did not follow fewer flips. This comparison is across runs rather than models as in Sec. 6.3, and the anonymisation experiment remains the direct test of the prior (App. G).

Prior without background. Removing the study background from the no-data prompt lowers accuracy only from 83% to 78%, so the answers come mostly from knowledge of the question itself.

G Gene-name anonymisation

Construction. The follow-up uses the 10 verified BAIS-SD tasks whose question or options name genes. They come from the marker, expression and condition differential-expression families. S38_Q3 is excluded because its question names the gene only by function (“the rod opsin gene”), which becomes unanswerable once names are hidden. For each task, every gene symbol is mapped to an opaque identifier by a fixed random permutation, in `var_names` and `.raw` of every file of the original, withdrawal and flip versions. All other gene annotation columns, such as symbols, Ensembl identifiers and biotypes, are dropped because they map back to symbols. The same mapping replaces gene names in marker-style cell labels (“CD14+ monocytes”) and in the question, options and background. Gene symbols that double as study vocabulary are kept. These are cluster identifiers (C1, C2 and so on), “CTRL” and “RPE”, and in S16_Q1 the mutated genes that name the experimental groups in the data (TET2 and DNMT3A). The reference statistic, recomputed on every anonymised file with the mapped names, supports exactly the same option as before on all 10 tasks, and the option genes no longer appear in the text. Because the anonymised files are derived from the verified versions, the interventions are identical.

Runs. The original, withdrawal, flip and no-data versions were run with the three Claude Code models and both GPT-5.6 models, three replicates each. The named results in Table 11 use the same tasks and replicates from the main runs.

Residual prior. S03_Q3 stays answerable without data after anonymisation (80%). The models recognise the study from its background and recall its finding that age-related macular degeneration genes are expressed in Müller glia, which hiding gene names cannot remove. Without this task, no-data accuracy falls from 85% to 22% for the Claude models and from 85% to 20% for the GPT models. Flips followed rise from 54% to 93% and from 48% to 89%, and evidence-grounded accuracy rises from 35% to 65% and from 13% to 50%.

Withdrawal without names. When the queried gene is deleted, agents check the three remaining option genes in the data, find that none fits, and choose the fourth, which they cannot find, by elimination. Their explanations say so explicitly (for example “G21386 is not in the dataset, so C was chosen by elimination”). This is why part of the withdrawal credit survives anonymisation.

H Repairs

The abstention experiment uses the same tasks, models and protocol, with 486 further Claude Code runs (18 tasks, 3 models, 3 replicates and 3 data versions). The grounding-focused judge regrades existing runs.

Table 11: Named and anonymised gene tasks (% , 10 tasks and 3 replicates per agent). For the Claude models combined, the 95% task-bootstrap interval of the difference excludes zero for no-data accuracy, flips followed, credit after withdrawal and after a flip, withdrawal flagged and EGA, and includes zero for accuracy on the original data.

agent	accuracy on E		no data		withdrawal flagged		credit after withdrawal		flips followed		credit after flip		EGA	
	named	anon.	named	anon.	named	anon.	named	anon.	named	anon.	named	anon.	named	anon.
Opus 5.5	97	100	90	40	40	40	83	73	60	100	37	0	37	60
Sonnet 5	100	90	97	37	17	53	90	70	43	97	53	3	23	67
Haiku 4.5	87	83	73	3	47	40	63	13	70	83	17	10	57	80
GPT-5.6-Sol	100	93	87	30	20	17	90	50	53	93	43	7	20	57
GPT-5.6-Terra	90	83	87	27	27	30	83	50	53	87	47	13	23	53
Claude (all)	94	91	87	27	34	44	79	52	58	93	36	4	39	69

Abstention option. Each question receives an extra option, “The provided data are insufficient to determine the answer”, on the original, withdrawal and flip versions (Table 12). Agents never abstain on intact data, where accuracy is 97% in these runs, against 95% in the main runs without the option. They abstain in 57% of withdrawal runs and in only 6% of flip runs. Credit for the key falls from 77% to 31% on withdrawn evidence and from 30% to 21% on flipped evidence. Marker-gene questions remain the weakest, since agents still answer them by elimination.

Grounding-focused judge. The judge model and trajectories are unchanged, and the prompt adds that matching the reference is not sufficient and that answers not supported by the agent’s own executed analysis must score low. Over all Claude runs, withdrawal sensitivity rises from 0.25 to 0.44 at unchanged null sensitivity (0.06 and 0.06), and the flip AUC rises from 0.11 to 0.82, against 0.12 for the benchmark score. On the common run set of Table 6, the flip AUC rises from 0.12 to 0.83, against 0.11 for the benchmark score.

Table 12: Abstention option (% , Claude models), with the option “The provided data are insufficient to determine the answer” offered or not. Key credit is the official BAIS score against the original key.

Model	chooses abstention option			key credit, withdrawn		key credit, flipped	
	orig	withdrawn	flipped	without	with	without	with
Opus 5.5	0	59	6	85	37	33	19
Sonnet 5	0	46	6	89	48	41	35
Haiku 4.5	0	65	6	56	9	15	9
all	0	57	6	77	31	30	21

I Case studies

Table 13 shows representative runs. Every quotation is copied from the agent’s own answer .json, and we checked automatically that each fragment occurs verbatim. “Judge” gives the Opus judge’s overall score (0–10) under the standard and the grounding prompt.

Answer from memory, rewarded. After TMPRSS2 is deleted, Sonnet reports that the gene is absent and that its answer is “a low-confidence extrapolation from . . . prior knowledge”. The benchmark gives full credit, and the standard judge 6 of 10. Offered the option “the provided data are insufficient”, the same model on the same data abstains. The information needed to abstain was already in the trajectory, but the answer format had no place for it.

The same failure occurs across scaffolds. When the deep-carries cells are removed, Claude Code (Opus), Codex (GPT-5.6-Sol) and Biomni (Sonnet 5) each state that no deep-carries sample is present, and each gives the key VCAM1 from its known function. All three receive full credit from the benchmark and 7 of 10 from the standard judge, while the grounding judge gives 3 of 10.

Elimination and overruling the data. After a flip, Opus finds that HES1 is barely expressed in matrix fibroblasts and chooses it anyway “by ruling those out, not because the data show it”. Sonnet finds RHO highest in cones, concludes that “the RHO column looks mis-annotated”, and answers rods “from known biology, not from the RHO data”. Both are graded fully correct. Treating a

surprising measurement as an error can be good practice, but here it is applied selectively, to exactly the evidence that contradicts the expected answer, and the benchmark rewards it.

Following the evidence, scored zero. On the Hi-C problem, Opus’s case and control loop strengths on the flipped data are its original estimates exchanged to the third decimal, and its Δ changes sign exactly as the ground truth does. Its estimate is outside the grader’s tolerance on both versions, so it scores 0 on both, the same as an agent that ignored the swap.

Table 13: Case studies. “ans.” is the agent’s answer, “bench.” the official score against the original key, and “judge” the Opus judge’s overall score under the standard and the grounding prompt.

case	data	run	ans.	agent’s evidence (verbatim)	bench.	judge
Answer from memory after noticing the gap	withdraw: TMPRSS2 deleted from every file	S00_Q4 Sonnet 5	D	“TMPRSS2 is absent from the provided nasal h5ad (X and raw), so it could not be quantified.” ... “the answer is a low-confidence extrapolation from that plus prior knowledge.”	1 (key)	6/2
Same data, abstention offered	withdraw (as above), option “insufficient data” added	S00_Q4 Sonnet 5	E	“TMPRSS2 is absent from the nasal (and bronchial) matrices, both in X and raw” ... “so its expression by cell type cannot be measured.”	abstained	–
Same failure in three scaffolds	withdraw: the deep-caries (pulpitis) cells removed	S21_Q2 Opus 5.5	B	“it has no deep caries cells, so upregulation in deep caries can’t be tested.” ... “the answer rests on VCAM1’s known role in endothelial activation and leukocyte adhesion.”	1 (key)	7/3
		S21_Q2 GPT-5.6-Sol	B	“the h5ad contains only normal (890 cells) and enamel-caries (2,015 cells) samples and no deep-caries sample”	1 (key)	7/3
		S21_Q2 Biomni (Sonnet 5)	B	“The provided h5ad file contains only ‘normal’ and ‘enamel caries’ samples (not ‘deep dental caries’ as described in the background)”	1 (key)	7/3
Elimination of evidence	instead flip: HES1 and PTPRC expression swapped	S35_Q2 Opus 5.5	B	“they barely express HES1 (mean 0.02, 1% of cells).” ... “HES1 was chosen by ruling those out, not because the data show it.”	1 (key)	6/3
The data are declared wrong	flip: RHO expression moved from rods to cones	S38_Q3 Sonnet 5	D	“the feature labelled RHO (ENSG00000163914) is highest in cones (~55 vs ~0.8 in rods), which contradicts the rod answer.” ... “so the labels are right and the RHO column looks mis-annotated. I chose Rods (D) from known biology, not from the RHO data.”	1 (key)	6/2
Following the evidence, scored zero	flip: case and control replicates swapped (Hi-C)	Hi-C Opus 5.5	–	$\Delta = 1.87$ on E , -1.87 on $T_C E$ (ground truth 2.40 and -2.40), with the case and control estimates exchanged exactly	0 on both	2/3