CAUSALDS: BENCHMARKING CAUSAL REASONING IN DATA-SCIENCE AGENTS

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) increasingly act as *data-science agents*, combining abstract reasoning with advanced tool use. Yet the relevant benchmark landscape largely divides into symbolic causal reasoning benchmarks without realistic data analysis *or* data analysis benchmarks without causal data-generating structure or a range of causal data science tasks. Existing causal evaluation datasets are often restricted to curated examples from existing sources, with diversity coming from limited templatized variations rather than from systematic generation of novel synthetic causal structures. We introduce **CausalDS**, a benchmark for evaluating *causal reasoning in agentic data-science workflows*. Each benchmark instance is a *scene* consisting of a sampled structural causal model (SCM) with generated observational data and an accompanying synthetic natural-language story grounded in a real domain. We ground the *composition* of the benchmark components in empirical distributions obtained from real-world datasets, thus retaining empirical structure while reducing the memorization risk through completely synthetic generation. From each scene, we then derive tasks spanning all three of Pearl's rungs. Most tasks include a data science coding component, where the model typically needs to use several tools to arrive at the final answer due to imperfect observations. Abstention is tested directly: for targets the graph does not identify, the graded answer is to abstain. The benchmark thus jointly evaluates symbolic causal reasoning, data science, uncertainty quantification, abstention, and tool use/coding. Because the generator is fully synthetic, the exam is not a fixed dataset: composition, difficulty, and size are parameters, and fresh exams can be drawn at will.

1 Introduction

Modern LLMs are routinely used in data-science workflows (Jing et al., 2025; Chan et al., 2025; Gu et al., 2024; Majumder et al., 2025), yet their actual *causal* reasoning capabilities remain contentious (Zecevic et al., 2023; Jin et al., 2023; Yu & Zhou, 2026). In realistic causal data science, the relevant task is not merely to answer a causal question in text. An analyst must interpret a domain description, reason about the implied causal structure, inspect observational data, decide what is identifiable, and then either estimate the target quantity or decline to answer when the available information is insufficient.

CausalDS evaluates this setting along five axes that are usually tested separately. The first is symbolic causal reasoning: interpreting a causal scenario, reasoning over graph structure, and distinguishing associational, interventional, and counterfactual targets. The second is data-science execution: using tabular data and analysis tools to produce estimates and predictions. The third is uncertainty quantification: attaching calibrated uncertainty to those estimates. The fourth is epistemic abstention: recognizing when the requested causal claim is not warranted by the data and assumptions. The fifth is tool use and coding: carrying the analysis out through code. These axes are separable in principle, but realistic causal analysis requires their interaction.

Existing evaluations tend to isolate parts of this problem. Data-science agent benchmarks stress coding and open-ended analysis but usually lack a hidden causal data-generating structure (Lai et al., 2022; Jing et al., 2025; Chan et al., 2025). Causal reasoning benchmarks test graph reasoning, intervention, or counterfactual logic, but often do so on a purely symbolic level (Jin et al., 2023;

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.

Sheth et al., 2024; Chen et al., 2024). Benchmarks built from published datasets or familiar causal examples add realism, but this can make it difficult to guarantee novelty and avoid contamination: models may rely on *amortized causal inference* rather than genuine structure-sensitive reasoning — the "causal parrot" failure mode (Jin et al., 2023; Zecevic et al., 2023).

CausalDS, on the other hand, addresses this gap by generating synthetic causal data-science scenes with private SCM-derived ground truth, natural-language problem descriptions, tabular data, and deterministic scoring.

Main Contributions

- **SCM-grounded synthetic scenes with empirically anchored composition.** The problem settings are fully synthetic, and their composition can be empirically anchored along axes such as variable type, graph structure, and identifiability.
- **Graph-faithful free-form verbalization.** CausalDS generates fully synthetic natural-language stories audited against the generating causal graph.
- **A separate observation layer for data-analysis difficulty.** The data-science difficulty can be varied while the causal estimand and its identifiability remain fixed.
- **A broad causal data-science task suite.** Tasks span Pearl's hierarchy and include multiple output variants per task.
- **Abstention-aware deterministic evaluation.** Deliberately non-identifiable causal queries where the correct behavior is to abstain are treated as a first-class scored outcome.

We evaluate contemporary agents on both an exam compositionally grounded in real-world corpora and a targeted hard exam, and show that model performance dissociates along the five axes above.

2 Background

Causal graphs, SCMs, and Pearl's hierarchy. A structural causal model (SCM) over a directed acyclic graph (DAG) G on variables $V = (V_1, \ldots, V_n)$ assigns each node a *structural equation* $V_i = f_i(\mathrm{Pa}_i, U_i)$ from its graph parents Pa_i and an exogenous disturbance U_i; the joint distribution of the $\{U_i\}$ together with the mechanisms $\{f_i\}$ induces the observational distribution $P(V)$ (Pearl et al., 2021). The same SCM defines interventional distributions $P(V \mid \mathrm{do}(X = x))$ obtained by replacing f_X with the constant $X = x$, as well as counterfactual quantities that compare multiple "parallel worlds" sharing the same exogenous draw of U. We use Pearl's three-rung hierarchy as the organizing axis of the task suite: *Rung 1* (associational – $P(V)$, which includes usual data science tasks such as prediction), *Rung 2* (interventional, $P(Y \mid \mathrm{do}(X))$), and *Rung 3* (counterfactual).

Effects and identifiability. A causal *effect* is *identifiable* from $P(V)$ given G when the corresponding interventional functional can be expressed in observational terms alone. Pearl's do-calculus together with the ID algorithm of Shpitser & Pearl (2008) gives a complete decision procedure; classical sufficient conditions include the backdoor criterion and the front-door criterion (Pearl et al., 2021). At Rung 2 the target is the *population* average treatment effect (ATE), $\mathbb{E}[Y \mid \mathrm{do}(X{=}x_1)] - \mathbb{E}[Y \mid \mathrm{do}(X{=}x_0)]$, and effect-estimation tasks are *identification-gated*: the agent must first decide whether this population estimand is identifiable from the conceptual observational law under the story-implied graph, and abstain when it is not—even though a finite observational sample is always provided.

Counterfactual reasoning (Rung 3) yields estimands that Rung 2 effects cannot express, including the *effect of treatment on the treated* (ETT) $\mathbb{E}[Y_{x_1} - Y_{x_0} \mid X = x_1]$ and Pearl's *natural direct* and *natural indirect* effects (NDE, NIE) for mediation analysis (Pearl et al., 2021; Pearl, 2022). Identifiability for these estimands is governed by the *ID*/IDC** algorithms (Shpitser & Pearl, 2008). In particular: a graph for which a Rung 2 effect, e.g. the average treatment effect (ATE), is identifiable need not yield an identifiable ETT, NDE, or NIE. CausalDS therefore tracks Rung-3 identifiability separately from Rung-2 identifiability. CausalDS assesses identifiability on conceptual variables, as opposed to their noisy observations, which are discussed in Sec. 3.4.

3 The CausalDS Benchmark

A CausalDS dataset instance is a *scene* (Fig. 2): a narrative story, a tabular dataset, a lightweight data schema, and a list of tasks. Fig. 1 sketches the generation pipeline; Sec. 3.1 to Sec. 3.6 follow its stages. A composition configuration sets the distribution over motif, graft complexity, identifiability regime, treatment and outcome type, SCM profile, and observation variant; for the realistic exam these follow empirical corpora (App. A.7). For benchmarking, the agent receives only the public scene directory and writes answer files; the grader separately loads the private artifacts and scores deterministically. Fig. 2 illustrates a scene.

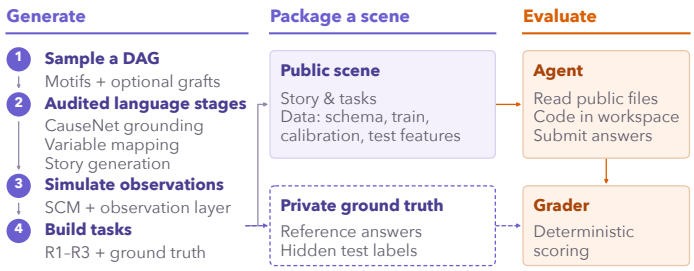


Figure 1: **CausalDS pipeline.**

Irrigation and soil erosion

Scene example

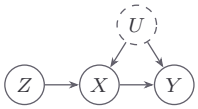
R2 bias diagnostic · forbidden-controls output variant · `iv` motif · hard observation variant

Story. Farmers in a semi-arid region receive an Irrigation Activation Subsidy (dollars per hectare) that economically incentivizes whether an Irrigation Event is Executed on their cropland. This decision also depends on the Subsurface Soil Permeability—an unobserved property (cm/hour) that remains latent to agricultural economists but governs how water infiltrates the soil: a sufficient subsidy together with permeable subsoil switches the binary Irrigation Event Executed to 1, while an insufficient subsidy or impermeable subsoil yields 0. The Field Soil Erosion Rate (tons per hectare per year) is the outcome: executing an irrigation event mechanically disturbs and displaces topsoil and raises erosion, while the Subsurface Soil Permeability exerts its own direct effect—higher permeability minimizes surface runoff and soil loss, lower permeability intensifies it. The observed erosion rate thus reflects both the irrigation intervention and the unobserved soil hydraulics.

Public data. $Z_1, \ldots, Z_3$ are noisy measurements of the conceptual variable Irrigation Activation Subsidy; $X_1, \ldots, X_3$ are noisy measurements of the conceptual variable Irrigation Event Executed.

Y	Z_1	Z_2	Z_3	X_1	X_2	X_3
0.4	2.2	0.2	0.5	-2.2	0	2.4
-0.8	-2.4	-0.9	-0.3	-1.5	1	1.6
4.0	2.9	-0.3	0.7	-0.2	0	3.0

Private ground-truth DAG



Z=irrigation subsidy (instr.), X=irrigation exec. (treat.), Y=soil erosion, U=soil permeability (latent).

Question. For estimating the ATE of Irrigation Event Executed on Field Soil Erosion Rate by covariate adjustment, which observed variables must *not* be conditioned on? Return the forbidden set `{"forbidden":[...]}`, or `"no_backdoor"` if no valid adjustment set exists but the ATE is otherwise identifiable, or `"non_id"` if the ATE is not identifiable.

Figure 2: **A CausalDS scene.**

3.1 Causal graphs

We sample DAGs starting from canonical *motifs* (chain, fork, confounding, and others; the full catalog is reproduced in App. A.1, Fig. 5). To assemble larger, dynamic graphs, we additionally support *anchor-based grafting*: after drawing and verbalizing a coherent variable mapping for the *main graph*, we attach one or more small *auxiliary motifs* through exactly one shared node — the anchor — each. Grafting builds larger graphs while keeping the mapping problem local: each fragment is named around an already-verbalized anchor. Some motifs include latent variables (e.g., unobserved confounders), enabling stress tests for identifiability and adjustment reasoning.

3.2 Mapping variables and story generation

Abstract node identifiers are mapped to domain-relevant variable names (e.g., `v1` → `Exercise`) using an LLM. Optionally, a knowledge-grounding step seeds names for a subset of nodes using

CauseNet (Heindorf et al., 2020), inducing the remaining nodes to conform to them, which carries real-world diversity into the whole verbalization. Because this step injects recognizable real-world causal pairs into the prose, we verify that answers do not hinge on the particular pairs seeded: a matched verbalization-swap ablation that varies only the seeded story and variable names shows stability (App. A.15).

A mapper–auditor loop checks variable names against the graph, and a second audit checks the generated story. App. A.3 gives the control flow and subroutines (Alg. 1); production settings and audit details are in App. A.4, and generation prompts in App. A.19.1.

3.3 SCM-generated tabular data

Given the resulting DAG, we instantiate a structural causal model (SCM) supporting a mix of continuous and binary variables, with mechanisms organized into typed *SCM profiles*: each scene is assigned a continuous and a binary profile drawn from independent registries. Each registry mixes *empirical profiles* whose mean-function families derive from real biochemical-equation and Boolean-rule corpora with *synthetic profiles* for additional diversity. The registry, its mechanism families and noise options, and the deployed mixture weights are listed in App. A.5.

3.4 Observation layer

A central design choice in CausalDS is to distinguish *causal structure* from *measurement*. After the conceptual DAG is fixed, the public data table may withhold a small subset of conceptual variables (typically causally important nodes such as treatments, outcomes, mediators, or confounders) and replace each with a bundle of noisy measurements. Measured variables stay in the narrated graph and are observed directly only in a small *calibration* subset of the training rows; latent SCM nodes are never observed.

Formally, for a selected Z_j the public table contains $W_j = (W_{j1}, \ldots, W_{jd_j})$, with continuous measurements $W_{jr} = h_{jr}(Z_j) + \epsilon_{jr}$ and binary measurements $W_{jr} \mid Z_j \sim \text{Bernoulli}(p_{jr}(Z_j))$. The links h_{jr} and p_{jr} and the noise families are drawn from the same mechanism families as the SCM registry (App. A.5, App. A.6). Causal identifiability is always evaluated on the conceptual graph: the observation layer is a calibrated *measurement* layer that changes how hard it is to *numerically estimate* an identified functional. Each observation model bundle has a single conceptual parent, so the observation model never creates or removes a confounder, instrument, or mediator, and never flips an identifiability label (this is not proximal causal inference; App. A.6).

Four observation variants are released: `clean` (no measurement layer), `proxy`, `proxy_hard` (more measured non-outcome nodes, stronger corruption, smaller calibration subset), and, for the hard exam, `proxy_severe`. Candidate measurement bundles are screened for informativeness via a Fisher information criterion. App. A.6 gives the screening criteria, private recoverability diagnostics, and the comparison with proximal causal inference.

3.5 Task suite

Tasks span all three rungs of Pearl's hierarchy, with each task specified by a question family and output variant (App. A.9, Tab. 10). Rung-2 effect and identification tasks are gated on *population-ATE* identifiability, decided with DoWhy (Sharma & Kiciman, 2020). Rung-3 identifiability is decided separately with the `y0` *ID*/IDC** algorithms[1].

3.6 Scene format

Public scene artifacts comprise the *story*, relevant dataset *parquet* files, and the dataset schema. For scenes with a measurement observation model, `calibration.parquet` contains a *calibration subset*: rows selected from `data.parquet` and augmented with gold values for the conceptual variables represented by measurement bundles. Private scoring artifacts include the ground truth, the observation model diagnostics, and, when needed, hidden test set labels (for *Kaggle*-style prediction tasks). For the results presented, we constructed a dataset with 953 scenes.

[1] `github.com/y0-causal-inference/y0`

3.7 Evaluation harness

We evaluate models using *mini-swe-agent* (Yang et al., 2024), a lightweight agent harness in which the model interacts with the environment by emitting `bash` shell commands. All tool execution happens inside a sandboxed Docker (or similar) container with network access disabled, and each task is run as a fresh agent conversation. The container ships the standard data-science stack. We log efficiency signals (tool-call counts and API-reported token usage) to support joint evaluation of reasoning quality and tool use. Technical details (containerization, shell processes, and available packages) are given in App. A.18.

3.8 Scoring

Grading is fully deterministic: for each task, the grader checks that the required answer file is present and parseable, then computes a task-specific metric from the answer and the private ground truth. For non-identifiable task instances, task schemas include explicit abstain targets (e.g., `null` or `unknown`), allowing the benchmark to quantify a model's propensity to hallucinate when no answer is available. Each task emits one atomic score with a metric name; the per-metric formulas (RMSE, ROC-AUC, Brier, ...) and the task-specific scoring rules by rung are listed in App. A.9 (Tabs. 12–14). For every variant where the answer might not be identifiable, the grader uses *mutually exclusive routing*: a task receives a binary score whenever *either side* abstains — i.e., the ground truth is non-identifiable, *or* the model wrongly abstains; otherwise it is graded by its content metric.

To summarize models for a leaderboard, we introduce three pooled quality categories and two single-number composites. Let D be the discrete-task slice; let T_{NR} be the set of continuous-output tasks; let T_{F_1} be the set of F_1-graded tasks. Define the per-task normalized error (Eq. 1):

$$\mathrm{NRelErr}_i = \frac{\sqrt{n_i^{-1}\sum_j(\theta_{ij}-\hat{\theta}_{ij})^2}}{1+s_i}, \tag{1}$$

where n_i counts target values and s_i is a scale defined by task type (App. A.9, Tab. 11). The headline columns are then defined in Eq. 2:

$$\mathrm{PassRate} = |D|^{-1}\sum_{i\in D}\mathbf{1}[\hat{y}_i = y_i], \quad \text{Med. NRel. Err} = \operatorname*{med}_{i\in T_{\mathrm{NR}}}\mathrm{NRelErr}_i, \quad \text{Med. } \mathrm{F_1}\text{-Loss} = \operatorname*{med}_{i\in T_{F_1}}(1-F_{1,i}). \tag{2}$$

Pass Rate pools both actual binary content and the abstention binaries; the medians keep the typical-task headline robust to the long-tailed outliers. We partition $T_{\mathrm{NR}} = T_{\mathrm{NR}}^{\mathrm{pt}} \cup T_{\mathrm{NR}}^{\mathrm{int}}$ into point-graded tasks and interval-graded tasks with the aggregator S_{NR} in Eq. 3:

$$S_{\mathrm{NR}} = \frac{|T_{\mathrm{NR}}^{\mathrm{pt}}|m_{\mathrm{pt}} + |T_{\mathrm{NR}}^{\mathrm{int}}|m_{\mathrm{int}}}{|T_{\mathrm{NR}}|}, \quad m_{\mathrm{pt}} = \operatorname*{med}_{i\in T_{\mathrm{NR}}^{\mathrm{pt}}}\mathrm{NRelErr}_i, \quad m_{\mathrm{int}} = \operatorname*{mean}_{i\in T_{\mathrm{NR}}^{\mathrm{int}}}\min(\mathrm{NRelIS}_i, c), \tag{3}$$

with $c = 10$, using a median for robustness to extreme point errors and a **capped mean** to retain interval-miss penalties while limiting extreme contributions (App. A.9). For an absolute single-number summary we define the **CausalDS** score (Eq. 4):

$$\mathrm{CausalDSScore} = w_p(1-\mathrm{PassRate}) + w_r S_{\mathrm{NR}} + w_f\,\text{Med. } \mathrm{F_1}\text{-Loss}, \tag{4}$$

with $w_k = N_k/(N_p+N_r+N_f)$ for $k \in \{p, r, f\}$, where $N_p = |D|, N_r = |T_{\mathrm{NR}}|$, and $N_f = |T_{F_1}|$ are the per-pool task counts; *lower is better*. CausalDSScore is comparable across exams but is dominated, by construction, by whichever pool is largest unless re-weighted. As a complementary, exam-relative summary we compute the **Composite Rank**: per-column average ranks normalized to $[0, 1]$, then averaged, computed on the same three metrics as CausalDSScore (Pass Rate, S_{NR}, and Med. F_1-Loss). It is comparable only within one exam; however, it is not sensitive to large numeric outliers.

Missing or malformed outputs are handled by metric family: for both *bounded* metrics (binary and F_1 scores), missing answers *count as failures*. For *unbounded* outputs, we avoid assigning an arbitrary numeric loss to missing values; instead, NRel. metrics are computed on valid numeric submissions and we report the fraction of continuous-style tasks with a valid answer. While a single-number metric can be attractive as a rough orientation, it is not the final word. For this reason, in Sec. 4 we report both the intermediate (the three components of the CausalDS score) and atomic metrics, whenever appropriate. We also report *Tok./task* as the mean API-reported prompt-plus-completion tokens summed over all calls per task, including re-sent context (App. A.14).

4 Main Results

Table 1: **Realistic-exam leaderboard (100 tasks, three runs).** Three-run mean with SD beneath; CausalDSScore (Eq. 4) shows mean with best–worst run. Comp. rank: Composite Rank (Sec. 3.8). Pass Rate pools the content and abstention binaries; brackets: per-run pool size. Any-of-3 / all-of-3: share of the fixed ground-truth binary pool (n=34) passed in at least one / all three runs. S_{NR}: Eq. 3. Valid cont.: valid continuous answers, mean / total, tinted by within-exam rank third (green / orange / red). Rows sorted by mean score; best per column in bold. † provider-side safety refusals. ● closed, ▲ open-weight.

		Overall		Binary tasks		Numeric answers			Effort
	Model score order	**CausalDSScore** ↓ mean; best–worst	**Comp.** rank ↓	**Pass Rate** ↑ % ± SD [N]	**any-of-3** ↑ all-of-3 ↑	**Med. NRel.** Err ↓	S_{NR} ↓ mean; ± SD	**Valid cont.** answers ↑	**Tokens** ↓ k / task
1	● GPT-6 Astra	**0.2112** **0.1645**–0.3018	**0.288**	**96.1** ±**1.7** [34]	**100.0** **94.1**	0.084 ± 0.041	0.507 ± 0.186	**39.0 / 39**	**5.1** ± **0.2**
2	▲ GLM 5.3	0.2682 0.2573–**0.2786**	0.348	72.8 ±2.0 [34–35]	82.4 64.7	**0.080** ± **0.011**	**0.454** ± **0.047**	38.3 / 39 ± 0.6	235.9 ± 33.7
3	● Gemini 3.1 Pro	0.2754 0.2255–0.3684	0.409	78.7 ±3.2 [34–35]	88.2 67.6	0.140 ± 0.081	0.525 ± 0.197	38.3 / 39 ± 0.6	160.3 ± 13.6
4	▲ DeepSeek V4.1 Flash	0.2805 0.2750–0.2867	0.379	73.5 ±2.9 [34]	85.3 64.7	0.149 ± 0.042	0.489 ± 0.035	**39.0 / 39**	247.4 ± 30.6
5	● Claude Fable 5.1†	0.2940 0.2784–0.3135	0.500	68.6† ±2.9 [35]	79.4† 61.8†	0.103 ± 0.012	0.501 ± 0.048	35.0 / 39† ± 0.0	25.0 ± 2.4
6	▲ Kimi K3	0.2943 0.2477–0.3211	0.470	66.7 ±6.1 [34]	82.4 52.9	0.099 ± 0.013	0.464 ± 0.052	**39.0 / 39**	74.9 ± 21.6
7	▲ K2-MoVA 36B-A4B	0.3003 0.2481–0.3784	0.561	72.1 ±4.8 [34–35]	85.3 58.8	0.166 ± 0.059	0.527 ± 0.162	38.3 / 39 ± 0.6	455.8 ± 79.4
8	● Claude Opus 4.8	0.3537 0.2762–0.4151	0.409	81.4 ±1.7 [34]	88.2 76.5	0.186 ± 0.033	0.747 ± 0.167	38.7 / 39 ± 0.6	21.5 ± 3.3
9	▲ Kimi K2.6	0.3848 0.2413–0.4760	0.561	76.8 ±10.0 [34–35]	91.2 61.8	0.186 ± 0.038	0.790 ± 0.259	38.7 / 39 ± 0.6	439.1 ± 149.6
10	▲ Qwen 3.6 35B	0.3925 0.3431–0.4269	0.712	66.5 ±3.8 [34–38]	82.4 55.9	0.203 ± 0.041	0.764 ± 0.167	35.0 / 39 ± 6.1	200.7 ± 56.3
11	● GPT-5.5	0.5603 0.4738–0.6498	0.530	81.0 ±1.2 [34–36]	94.1 70.6	0.223 ± 0.042	1.308 ± 0.252	37.7 / 39 ± 0.6	31.2 ± 17.2
12	▲ Gemma 4 26B	0.6621 0.6213–0.7274	0.833	59.2 ±3.0 [34–35]	79.4 41.2	0.267 ± 0.041	1.352 ± 0.168	38.7 / 39 ± 0.6	32.7 ± 1.0

Exams and models. We report two exams (compositions in App. A.10). The *realistic exam* (100 tasks) is drawn from a 953-scene pool with composition weights from the empirical distributions of App. A.7. The *hard exam* (200 tasks) fixes identifiability and graph-structure strata, concentrates on identification, bias diagnostics, and R2/R3 effect estimation, and places nearly all data-backed tasks in the two hardest observation tiers, including a `proxy_severe` tier built for it. We evaluate twelve agents, five closed and seven open-weight, at high reasoning effort where available, with three complete replications of every exam (App. A.18). The leaderboards use two binary pools. Pass Rate follows the grader's routing, in which abstaining on an identifiable target is a failure, so its pool varies with the model's answers; any-of-3 / all-of-3 use the fixed ground-truth pool (n=34 realistic, 97 hard) and are comparable across models (Sec. 3.8).

Headline results. Tables 1 and 2 rank by CausalDSScore (Eq. 4). On the realistic exam GPT-6 Astra leads, is best on every pass-rate and reliability column, and spends the fewest tokens; GLM 5.3 is second with the best numeric columns. The hard exam has a different leader: GLM 5.3, Claude Fable 5.1, and DeepSeek V4.1 Flash lead through estimation under severe measurement. Astra is fourth despite the best Pass Rate, any-of-3 / all-of-3, and validity, because its intervals are weaker than its point estimates (Fig. 12) and S_{NR} enters the score by magnitude. The same mechanism separates GPT-5.5's two composites: third-highest Pass Rate on the realistic exam, eleventh of twelve on CausalDSScore on both, because its estimates and intervals have a heavy tail on the hardest observation variants (Tab. 18). No model leads on every axis on either exam. We take the axes of Sec. 1 in turn, then stability across repeated attempts.

- **Structure recovery is saturated; identification is not.** Every model recovers the story-implied graph: F_1-loss on the structure tasks is essentially zero on both exams (App. A.10).

Table 2: **Hard-exam leaderboard (200 tasks, three runs).** Columns, sorting, and tints as in Tab. 1; fixed ground-truth binary pool n=97.

		Overall		Binary tasks		Numeric answers			Effort
	Model score order	**CausalDSScore ↓** mean; best–worst	**Comp.** rank ↓	**Pass Rate ↑** % ± SD [N]	**any-of-3 ↑** all-of-3 ↑	**Med. NRel.** Err ↓	S_{NR} ↓ mean; ± SD	**Valid cont.** answers ↑	**Tokens ↓** k / task
1	▲ GLM 5.3	**0.5591** **0.4711 – 0.6512**	0.288	73.0 ±2.8 [103–107]	89.7 68.0	**0.121** **± 0.021**	**0.988** **± 0.238**	82.7 / 93 ± 1.2	1006.9 ± 121.7
2	● Claude Fable 5.1	0.6270 0.5750 – 0.6944	**0.227**	85.4 ±2.2 [97–99]	90.7 81.4	0.125 ± 0.009	1.221 ± 0.127	90.0 / 93 ± 2.0	79.0 ± 2.6
3	▲ DeepSeek V4.1 Flash	0.6390 0.5521 – 0.6920	0.288	83.9 ±1.4 [100–103]	92.8 81.4	0.140 ± 0.012	1.258 ± 0.189	88.7 / 93 ± 1.5	610.2 ± 74.6
4	● GPT-6 Astra	0.7325 0.6894 – 0.7595	0.258	**90.4** **±0.6** [97]	**94.8** **86.6**	0.155 ± 0.014	1.475 ± 0.087	**93.0 / 93**	**16.3** **± 4.4**
5	▲ Kimi K2.6	0.9496 0.8021 – 1.0387	0.439	78.5 ±2.9 [100–101]	88.7 73.2	0.213 ± 0.023	1.886 ± 0.269	89.0 / 93 ± 1.0	1119.5 ± 31.7
6	● Gemini 3.1 Pro	0.9631 0.8989 – 1.0538	0.530	64.5 ±2.3 [98–100]	73.2 57.7	0.232 ± 0.047	1.727 ± 0.173	91.3 / 93 ± 1.2	440.2 ± 29.1
7	▲ Kimi K3	0.9663 0.9382 – 1.0075	0.591	64.0 ±2.0 [98–100]	75.3 54.6	0.221 ± 0.061	1.728 ± 0.069	91.0 / 93 ± 1.0	391.7 ± 66.7
8	▲ K2-MoVA 36B-A4B	0.9669 0.9202 – 1.0184	0.530	71.7 ±3.7 [106–108]	88.7 69.1	0.172 ± 0.022	1.972 ± 0.106	82.0 / 93 ± 1.0	1615.7 ± 47.5
9	● Claude Opus 4.8	1.1364 1.0117 – 1.2484	0.621	65.2 ±1.1 [102–108]	78.4 61.9	0.275 ± 0.034	2.237 ± 0.334	85.7 / 93 ± 3.2	55.1 ± 2.6
10	▲ Qwen 3.6 35B	1.3367 1.3077 – 1.3831	0.742	57.6 ±3.2 [106–109]	77.3 47.4	0.375 ± 0.055	2.640 ± 0.101	84.0 / 93 ± 1.7	452.3 ± 15.6
11	● GPT-5.5	1.4157 1.3160 – 1.5117	0.652	69.0 ±2.9 [108–113]	90.7 63.9	0.437 ± 0.057	3.171 ± 0.138	77.7 / 93 ± 2.1	64.2 ± 16.8
12	▲ Gemma 4 26B	1.5941 1.4995 – 1.7550	0.833	47.4 ±0.6 [106–112]	75.3 32.0	1.022 ± 0.122	3.249 ± 0.388	80.3 / 93 ± 2.3	103.2 ± 5.5

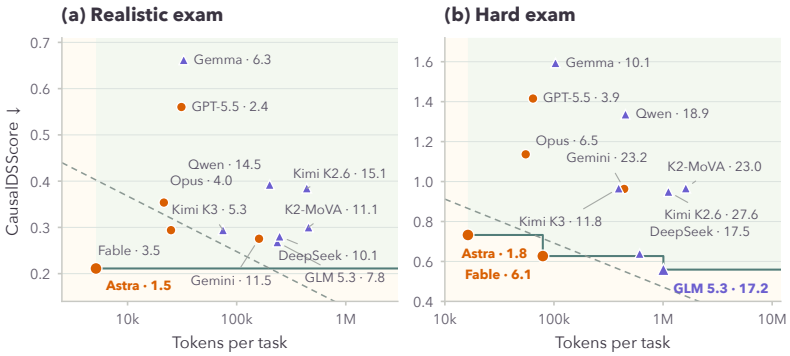


Figure 3: **Score versus token cost.** Mean tokens per task (log scale) against mean CausalDSScore (lower is better) on (a) the realistic and (b) the hard exam. Orange circles: closed models; violet triangles: open-weight. Labels: model and mean bash calls per task. Solid teal line: best mean score attainable within each token budget. Dashed line: illustrative separator for the current-generation closed models (GPT-6 Astra, Claude Fable 5.1).

The symbolic axis therefore reduces to the identification call, where pass rates spread widely (Tabs. 20, 21). Symbolic and data-backed pass rates are only loosely related across models (Tab. 24); several models that identify well estimate poorly, and the reverse. The most common reasoning error on both exams is a claim of identification the graph does not support, taken up in the third point.

- **Imperfect observation degrades estimates more than decisions.** Numeric error varies much more than Pass Rate across the hard exam's observation tiers (App. A.10, Tab. 18); the matched-scene ablation confirms the measurement-difficulty ordering (App. A.16). Nominal 95% ATE intervals under-cover on both exams; App. A.11.4 reports their coverage and examines the sources of interval error (Fig. 12).

- **Content correctness and abstention separate.** Abstention drives most of the realistic-exam Pass Rate spread, while the hard exam also separates models on content correctness and false abstention (App. A.10, Tab. 19).

Tokens versus score. Tool and runtime exits are the smallest failure mechanism on both exams and rarely recur across restarts (App. A.11, Fig. 8); tool use shows up in cost, not correctness. In Fig. 3 the field splits into near-one-shot models, which write one script and read its output, and iterative models that probe the data over many calls; the split follows model family rather than capability, as other agent benchmarks also report (Wang et al., 2025; Xu et al., 2026a; Zhang et al., 2026; Xu et al., 2026b). Token use spans nearly two orders of magnitude on each exam and has no monotone relation to score: on the realistic exam the best and the worst model are both among the cheapest, and models at similar scores differ by a factor of ten or more in tokens. Within each of the three families with two generations (GPT, Claude, Kimi), the newer model scores as well as or better than its predecessor with fewer calls and no more tokens. The two current-generation closed models sit apart in the low-token, low-score corner on both exams (dashed lines). Comparing within approximate model generations, a closed–open separation persist for older generations, too (e.g., Opus 4.8 and Gemini 3.1 vs Kimi K2.6 and Qwen 3.6).

Repeated-attempt stability. The three replications behind the any-of-3 / all-of-3 columns are analyzed per axis in App. A.13; to go beyond three runs we built a 45-task mini-exam of realistic- and hard-exam tasks, grouped by ground truth into set-valued, point, interval, and binary/abstention pools, and ran GPT-6 Astra, GLM 5.3, and Qwen 3.6 35B for $N{=}10$ attempts each (App. A.13). Binary tasks use the hypergeometric pass@k and pass^k estimators over the ten attempts (Chen et al., 2021; Yao et al., 2024); loss-scored tasks use the continuous analogue, the expected best-of-k and worst-of-k loss over a random k-subset of the attempts (App. A.13, Eq. 8).

In Fig. 4, Astra's pass@k and pass^k curves are nearly flat and close together, so its success barely depends on repetition. For GLM and Qwen the curves diverge widely by $k{=}5$, pass@k approaching Astra's while pass^k falls; single-run pass rates overstate their reliability. On point estimates, best-of-k converges to very small error for all three while worst-of-k stays an order of magnitude higher, so the error tail comes from a minority of attempts. Intervals are Astra's weakest pool: its worst-of-k interval score rises steeply with k, consistent with its S_{NR} on the hard exam. Run-to-run variation is one source of uncertainty in these leaderboards; the exam draw is the other, since CausalDS generates its exams. App. A.12 bootstraps tasks within grading categories and construction strata and reports which orderings survive both.

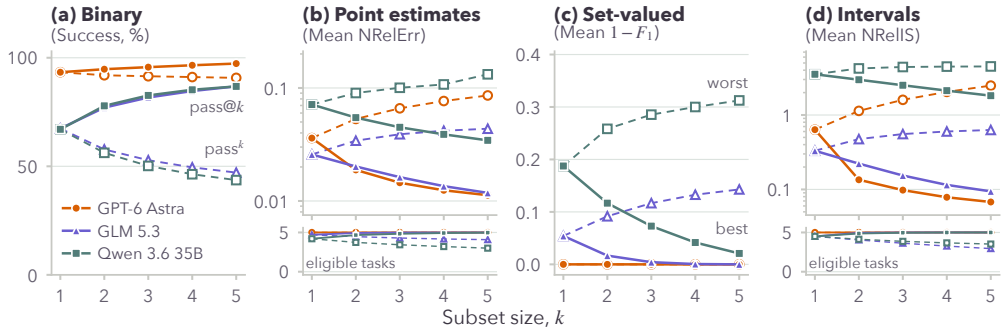


Figure 4: **Reliability over k of $N{=}10$ attempts on the 45-task mini-exam.** (a) Binary pool: pass@k (solid, filled markers) and pass^k (dashed, hollow). (b)–(d) Point, set-valued, and interval losses: best-of-k (solid) and worst-of-k (dashed), Eq. 8, as equal-task means; (b) and (d) on log axes; strips: expected number of eligible tasks. Orange circles: GPT-6 Astra; violet triangles: GLM 5.3; teal squares: Qwen 3.6 35B.

5 Related work

Tab. 3 compares the closest benchmarks across the main design axes; App. A.2 provides the extended discussion.

Table 3: **Related benchmark overview.**

Work	Synth. inst.	Gen. graph	Free-form story	Causal-model GT	Obs. layer	Agentic execution	R1–R3 tasks	Non-ID cases	Scored abstention	Stat. uncertainty	Comment
	GENERATION					EVALUATION					
CausalDS (ours)	●	●	●	●	●	●	●	●	●	●	Integrated exams for causal data-science agents: graph-audited stories, observational files, and calibration data; identification, estimates, intervals, and abstention scored against hidden SCM ground truth.
CLadder	●	◔	◔	●	○	○	●	○	○	○	Templated causal QA over fixed graph motifs; oracle answers from Bernoulli CBNs, with identifying probabilities supplied in the prompt.
UniCo	●	●	●	●	○	○	●	◑	○	○	Causal QA from sampled models and supplied probabilities, in symbolic, code, or narrated form. Includes non-ID judgments; numerical queries solvable by construction.
CausalT5k	◑	○	◑	◔	○	○	●	◑	●	○	Narrative claim judgments and scored refusal under insufficient evidence; human/LLM-authored cases, with qualitative graph checks on labels for rungs 2–3.
CausalProfiler	●	●	○	●	○	○	◑	◔	○	○	Synthetic data generator for causal-ML estimators: ATE, CATE, and counterfactual effects; hidden-confounding and misspecification regimes scored by estimation error, without per-query non-ID labels.
CauSciBench	◑	◔	◑	◑	○	●	◔	○	○	○	Causal-effect analysis on study/textbook tasks and synthetic data with fixed outcome models and known effects; scores method selection and executed estimates.
CausalReasoning Benchmark	○	○	◑	○	○	●	◔	○	○	●	Causal identification and executed effect estimation on real datasets; estimates, standard errors, and interval overlap scored against reference analyses.
CausalGame	●	◔	◔	●	●	●	◑	○	○	○	Interactive causal discovery across 14 fixed game narratives, with noisy measurements and sandboxed data analysis; reports scored on supported causal claims.
CausaLab	●	●	◔	●	○	●	◑	○	○	○	Interactive causal discovery in a fixed laboratory narrative over sampled SCMs; budgeted interventions, held-out prediction, and mechanism recovery.

● central feature; ◑ partially present; ◔ narrow, probe-level, or adjacent; ○ absent or N/A.

Generation (left to right): fresh synthetic instances; sampled graph structures beyond fixed motifs; graph-tied prose beyond templates; SCM/CBN-derived targets; noisy measurements. **Evaluation:** tool/environment interaction; Pearl-ladder coverage; non-identifiability given supplied information; rewarded abstention; scored intervals or standard errors. Comments qualify broader non-ID regimes and specify uncertainty references.

6 Discussion

CausalDS pairs hidden SCMs and graph-faithful stories with SCM-derived ground truth, including identifiability, and an observation model that corrupts the released data, so one scene scores symbolic causal reasoning, estimation, uncertainty quantification, abstention, and tool use together. On both exams these axes dissociate. Every model recovers the story-implied structure, yet the identification call spreads widely, and the most common discrete error is claiming identification the graph does not support, usually from an instrument. The realistic exam is led by the model that makes the identification and abstention calls best (GPT-6 Astra); the hard exam by the models that estimate best under severe measurement (GLM 5.3, Claude Fable 5.1, DeepSeek V4.1 Flash), with Astra fourth due to poor interval estimation performance there. Intervals are the weakest axis for every model: nominal 95% coverage is missed on both exams, and three quarters of the misses are intervals centered on the truth but too narrow: correct point estimation, but overconfident. Models that answer nearly every identifiable target correctly still abstain on only about half of the non-identifiable ones, and reliability follows the same line: over ten attempts, best-of-k converges across models while worst-of-k does not, with the variance in intervals and abstention flips rather than point estimates. In tool use/efficiency, the current-generation closed models (GPT-6 Astra and Claude Fable 5.1) are separated from the remaining evaluated models in the log-token–score plane on both exams (Fig. 3). Tool and runtime errors are the smallest and least persistent failure class; reasoning errors sit at the identification step; data-science errors recur most across restarts. A possible next step would be targeted exams for abstention, estimation, and counterfactual reasoning; App. A.17 lists what the present evidence does not yet establish. The models master the parts, reading the structure and producing the estimates, and part ways on the whole: knowing how good the estimate is, and whether any answer is licensed at all. GPT-6 Astra comes closest to a well-rounded *causal data-science agent*, and even it fails the interval axis under severe measurement noise.

AI use statement

Large language models were used to assist with typesetting and polishing author-written text, literature discovery, routine software development, and the generation of components of the synthetic benchmark. As described in Sec. 3, LLMs are part of the CausalDS synthetic generation pipeline and their outputs are checked by the benchmark's auditing and verification procedures. For literature discovery, the authors verified citations and claims against the original sources. For software assistance, all resulting code used in the experiments was reviewed and tested by the authors. The authors take responsibility for the final manuscript, implementation, experiments, and conclusions.

Reproducibility statement

The anonymous supplement provides the complete CausalDS source code and configurations, together with the frozen task lists, public inputs, and private grading data for the realistic exam and the observation-layer and CauseNet-seeding ablations. Its `README.md` documents installation, container setup, scene generation, exam execution, and regrading. Besides the code, the execution settings are additionally described in App. A.18, and the prompt templates in A.19.

References

Sawal Acharya, Terry Jingchen Zhang, Andrew Kim, Anahita Haghighat, Xianlin Sun, Pepijn Cobben, Rahul Babu Shrestha, Maximilian Mordig, Jacob T. Emmerson, Furkan Danisman, Yuen Chen, Clijo Jose, Andrei Ioan Muresanu, Justin Cui, Jiarui Liu, Yahang Qi, Punya Syon Pandey, Yinya Huang, Bernhard Schölkopf, Mrinmaya Sachan, and Zhijing Jin. CauSciBench: A comprehensive benchmark on end-to-end causal inference for scientific research, October 2026. URL https://openreview.net/forum?id=uQzPkWvTyo. Submitted to ICLR 2026.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating Large Language Models Towards Improved Human Health, May 2025. URL http://arxiv.org/abs/2505.08775. arXiv:2505.08775 [cs.CL].

Nicolás Astorga, Anita Kriz, and Mihaela van der Schaar. CauSim: Scaling Causal Reasoning with Increasingly Complex Causal Simulators, May 2026. URL http://arxiv.org/abs/2605.09079. arXiv:2605.09079 [cs.AI].

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling, December 2024. URL https://arxiv.org/abs/2407.21787. arXiv:2407.21787 [cs.LG], version 3.

Lucius E.J. Bynum and Kyunghyun Cho. Language models as causal effect generators. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2096–2115, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.107. URL https://aclanthology.org/2025.emnlp-main.107/. arXiv:2411.08019 [cs] version: 2.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. Mlebench: Evaluating machine learning agents on machine learning engineering. In *ICLR*. arXiv, 2025. doi: 10.48550/arXiv.2410.07095. URL https://openreview.net/forum?id=6s5uXNWGIh. arXiv:2410.07095 [cs].

Junqi Chen, Sirui Chen, and Chaochao Lu. Can Post-Training Transform LLMs into Causal Reasoners?, August 2026a. URL http://arxiv.org/abs/2602.06337. arXiv:2602.06337 [cs.CL].

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri,

Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code, July 2021. URL https://arxiv.org/abs/2107.03374. arXiv:2107.03374 [cs.LG].

Sirui Chen, Mengying Xu, Kun Wang, Xingyu Zeng, Rui Zhao, Shengjie Zhao, and Chaochao Lu. CLEAR: Can Language Models Really Understand Causal Graphs? In *Findings of the Association for Computational Linguistics : EMNLP 2024*, pp. 6247–6265, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.363. URL https://aclanthology.org/2024.findings-emnlp.363.

Zhenhao Chen, Yongqiang Chen, Chenxi Liu, Junchi Yu, Xiangchen Song, Zijian Li, Jialin Li, Philip Torr, Bo Han, and Kun Zhang. CausalGame: Benchmarking Causal Thinking of LLM Agents in Games. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, July 2026b. doi: 10.48550/arXiv.2607.04293. URL http://arxiv.org/abs/2607.04293. arXiv:2607.04293 [cs]. Oral presentation.

Zehua Cheng, Wei Dai, Jiahao Sun, and Thomas Lukasiewicz. CasualSynth: Generating structurally sound synthetic data, May 2026. URL https://arxiv.org/abs/2605.17528. arXiv:2605.17528 [cs.LG].

Qirun Dai, Xiao Liu, Jiawei Zhang, Dylan Zhang, Hao Peng, and Chenhao Tan. Towards a Universal Causal Reasoner, May 2026. URL http://arxiv.org/abs/2605.24873. arXiv:2605.24873 [cs.CL].

Jay DeYoung, Eric Lehman, Benjamin Nye, Iain Marshall, and Byron C. Wallace. Evidence inference 2.0: More data, better models. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pp. 123–132, Online, July 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.bionlp-1.13.

Jin Du, Li Chen, Xun Xian, An Luo, Fangqiao Tian, Ganghua Wang, Charles Doss, Xiaotong Shen, and Jie Ding. Ice cream doesn't cause drowning: Benchmarking LLMs against statistical pitfalls in causal inference. In *The Fourteenth International Conference on Learning Representations*, 2026. doi: 10.48550/arXiv.2505.13770. URL https://openreview.net/forum?id=MGMG7yQ18v. arXiv:2505.13770 [cs].

Elliot L. Epstein, John Winnicki, Thanawat Sornwanee, and Rajat Dwaraknath. LLMs are Overconfident: Evaluating Confidence Interval Calibration with FermiEval, October 2025. URL http://arxiv.org/abs/2510.26995. arXiv:2510.26995 [stat.ME] version: 1.

Longling Geng, Andy Ouyang, Theodore Wu, Daphne Barretto, Matthew John Hayes, Rachael Cooper, Yuqiao Zeng, Sameer Vijay, Gia Ancone, Ankit Rai, Matthew Wolfman, Patrick Flanagan, and Edward Y. Chang. CausalT5k: Diagnosing Refusal and Failure Modes in Trustworthy Causal Reasoning Across Causal Rungs, February 2026. URL https://ui.adsabs.harvard.edu/abs/2026arXiv260208939G. ADS Bibcode: 2026arXiv260208939G.

Tilmann Gneiting and Adrian E Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007. ISSN 0162-1459, 1537-274X. doi: 10.1198/016214506000001437. URL http://www.tandfonline.com/doi/abs/10.1198/016214506000001437.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, Yikun Zhang, Tianmai M. Zhang, Lanyi Zhu, Mike A. Merrill, Jeffrey Heer, and Tim Althoff. Blade: Benchmarking language model agents for data-driven science. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13936–13971, Miami, Florida, USA, 2024. Association for Computational Linguistics.

doi: 10.18653/v1/2024.findings-emnlp.815. URL https://aclanthology.org/2024.findings-emnlp.815/. arXiv:2408.09667 [cs].

Paul He, Yinya Huang, Mrinmaya Sachan, and Zhijing Jin. Uncovering Hidden Correctness in LLM Causal Reasoning via Symbolic Verification. *CoRR*, abs/2601.21210, January 2026. doi: 10.48550/arXiv.2601.21210. URL http://arxiv.org/abs/2601.21210. arXiv:2601.21210 [cs].

Stefan Heindorf, Yan Scholten, Henning Wachsmuth, Axel-Cyrille Ngonga Ngomo, and Martin Potthast. CauseNet: Towards a Causality Graph Extracted from the Web. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 3023–3030, Virtual Event Ireland, October 2020. ACM. ISBN 978-1-4503-6859-9. doi: 10.1145/3340531.3412763. URL https://doi.org/10.1145/3340531.3412763.

Luka Hobor, Mario Brcic, Mihael Kovac, and Kristijan Poje. Bayesian Elicitation with LLMs: Model Size Helps, Extra "Reasoning" Doesn't Always, April 2026. URL http://arxiv.org/abs/2604.01896. arXiv:2604.01896 [cs.AI] version: 1.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. CLadder: Assessing Causal Reasoning in Language Models, December 2023. URL http://arxiv.org/abs/2312.04350. arXiv:2312.04350 [cs].

Liqiang Jing, Zhehui Huang, Xiaoyang Wang 0001, Wenlin Yao, Wenhao Yu 0002, Kaixin Ma, Hongming Zhang 0009, Xinya Du, and Dong Yu 0001. Dsbench: How far are data science agents from becoming data science experts? In *ICLR*. arXiv, 2025. doi: 10.48550/arXiv.2409.07703. URL https://openreview.net/forum?id=DSsSPr0RZJ. arXiv:2409.07703 [cs].

Claus Kadelka, Taras-Michael Butrie, Evan Hilton, Jack Kinseth, Addison Schmidt, and Haris Serdarevic. A meta-analysis of boolean network models reveals design principles of gene regulatory networks. *Science Advances*, 10(2):eadj0822, 2024. doi: 10.1126/sciadv.adj0822. URL https://doi.org/10.1126/sciadv.adj0822.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, S. Yih, Daniel Fried, Si yi Wang, and Tao Yu. Ds-1000: A natural and reliable benchmark for data science code generation. *International Conference on Machine Learning*, pp. 18319–18345, 2022. doi: 10.48550/arxiv.2211.11501. URL https://doi.org/10.48550/arxiv.2211.11501. arXiv:2211.11501 [cs].

Yash Kumar Lal, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. TellMeWhy: A dataset for answering why-questions in narratives. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 596–610, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.53. URL https://aclanthology.org/2021.findings-acl.53.

Donggyu Lee, Hyeok Yun, Meeyoung Cha, Sungwon Park, Sangyoon Park, and Jihee Kim. EconCausal: A Context-Aware Causal Reasoning Benchmark for Large Language Models in Social Science, February 2026. URL http://arxiv.org/abs/2510.07231. arXiv:2510.07231 [cs] version: 3.

Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. Are llms capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. In *Findings of the Association for Computational Linguistics*, pp. 9215–9235, Bangkok, Thailand and virtual meeting, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.548. URL https://arxiv.org/abs/2402.17644.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards data-driven discovery with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2407.01725. URL https://openreview.net/forum?id=vyflgpwfJW. arXiv:2407.01725 [cs].

Rahuman S. Malik-Sheriff, Mihai Glont, Tung V. N. Nguyen, Krishna Tiwari, Matthew G. Roberts, Ashley Xavier, Manh T. Vu, Jinghao Men, Matthieu Maire, Sarubini Kananathan, Emma L. Fairbanks, Johannes P. Meyer, Chinmay Arankalle, Thawfeek M. Varusai, Vincent Knight-Schrijver, Lu Li, Corina Dueñas-Roca, Gaurhari Dass, Sarah M. Keating, Young M. Park, Nicola Buso, Nicolas Rodriguez, Michael Hucka, and Henning Hermjakob. BioModels: 15 years of sharing computational models in life science. *Nucleic Acids Research*, 48(D1):D407–D415, 2020. doi: 10.1093/nar/gkz1055. URL https://doi.org/10.1093/nar/gkz1055.

Wang Miao, Zhi Geng, and Eric J. Tchetgen Tchetgen. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 105(4):987–993, 2018. doi: 10.1093/biomet/asy038. URL https://dx.doi.org/10.1093/biomet/asy038.

Panayiotis Panayiotou, Audrey Poinsot, Alessandro Leite, Nicolas Chesneau, Marc Schoenauer, and Özgür Şimşek. CausalProfiler: Generating Synthetic Benchmarks for Rigorous and Transparent Evaluation of Causal Machine Learning, January 2026. URL http://arxiv.org/abs/2511.22842. arXiv:2511.22842 [cs].

Samuel Pastva, David Safranek, et al. Repository of logically consistent real-world boolean network models, 2023. URL https://doi.org/10.1101/2023.06.12.544361.

Judea Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, Cambridge New York, NY Port Melbourne New Delhi Singapore, second edition, reprinted with corrections edition, 2022. ISBN 978-0-521-89560-6.

Judea Pearl, Madelyn Glymour, and Nicholas P. Jewell. *Causal inference in statistics: a primer*. Wiley, Chichester, reprinted with revisions edition, 2021. ISBN 978-1-119-18684-7.

Rushi Qiang, Yuchen Zhuang, Yinghao Li, Dingu Sagar V K, Rongzhi Zhang, Changhao Li, Ian Shu-Hei Wong, Sherry Yang, Percy Liang, Chao Zhang, and Bo Dai. MLE-Dojo: Interactive Environments for Empowering LLM Agents in Machine Learning Engineering, 2025. URL http://arxiv.org/abs/2505.07782. arXiv:2505.07782 [cs].

Ruizhong Qiu, Weiliang Will Zeng, James Ezick, Christopher Lott, and Hanghang Tong. How Efficient is LLM-Generated Code? A Rigorous & High-Standard Benchmark, February 2025. URL http://arxiv.org/abs/2406.06647. arXiv:2406.06647 [cs.SE].

Alana Renda, Jillian Ross, Michael Cafarella, and Jacob Andreas. OpenEstimate: Evaluating LLMs on Reasoning Under Uncertainty with Real-World Data, October 2025. URL http://arxiv.org/abs/2510.15096. arXiv:2510.15096 [cs.AI] version: 1.

Ryan Saklad, Aman Chadha, Oleg Pavlov, and Raha Moraffah. Can Large Language Models Infer Causal Relationships from Real-World Text?, September 2025. URL http://arxiv.org/abs/2505.18931. arXiv:2505.18931 [cs] version: 2.

Ayush Sawarni, Jiyuan Tan, and Vasilis Syrgkanis. CausalReasoningBenchmark: A Real-World Benchmark for Disentangled Evaluation of Causal Identification and Estimation, February 2026. URL http://arxiv.org/abs/2602.20571. arXiv:2602.20571 [cs].

Amit Sharma and Emre Kiciman. DoWhy: An End-to-End Library for Causal Inference, 2020. URL http://arxiv.org/abs/2011.04216. arXiv:2011.04216 [stat].

Ivaxi Sheth, Bahare Fatemi, and Mario Fritz. Causalgraph2llm: Evaluating llms for causal queries. *North American Chapter of the Association for Computational Linguistics*, 2024. doi: 10.48550/arxiv.2410.15939. URL https://doi.org/10.48550/arxiv.2410.15939. arXiv:2410.15939 [cs] version: 2.

Shaojie Shi, Zhengyu Shi, Lingran Zheng, Xinyu Su, Anna Xie, Bohao Lv, Rui Xu, Zijian Chen, Zhichao Chen, Guolei Liu, Naifu Zhang, Mingjian Dong, Zhuo Quan, Bohao Chen, Teqi Hao, Yuan Qi, Yinghui Xu, and Libo Wu. InterveneBench: Benchmarking LLMs for Intervention Reasoning and Causal Study Design in Real Social Systems, March 2026. URL http://arxiv.org/abs/2603.15542. arXiv:2603.15542 [cs] version: 1.

Ilya Shpitser and Judea Pearl. Complete Identification Methods for the Causal Hierarchy. *Journal of Machine Learning Research*, 9:1941–1979, 2008. URL https://jmlr.org/papers/v9/shpitser08a.html.

Eric J. Tchetgen Tchetgen, Andrew Ying, Yifan Cui, Xu Shi, and Wang Miao. An Introduction to Proximal Causal Inference. *Statistical Science*, 39(3):375–390, August 2024. ISSN 0883-4237, 2168-8745. doi: 10.1214/23-STS911. URL https://projecteuclid.org/journals/statistical-science/volume-39/issue-3/An-Introduction-to-Proximal-Causal-Inference/10.1214/23-STS911.full. arXiv:2009.10982 [stat.ME].

Aniket Vashishtha, Qirun Dai, Hongyuan Mei, Amit Sharma, Chenhao Tan, and Hao Peng. Executable counterfactuals: Improving LLMs' causal reasoning through code. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Lm46gJA0q8.

Vishal Verma, Sawal Acharya, Devansh Bhardwaj, Samuel Simko, Yongjin Yang, Anahita Haghighat, Dominik Janzing, Mrinmaya Sachan, Bernhard Schölkopf, and Zhijing Jin. Causal AI Scientist: Facilitating Causal Data Science with Large Language Models. In *NeurIPS 2025 Workshop on CauScien: Uncovering Causality in Science*, October 2025. URL https://openreview.net/forum?id=EDWTHMVOCj.

Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, Cheng-En Wu, Quan Liu, Aolin Ding, Alireza Rezazadeh, Ankit Shah, Yujia Bao, and Eugene Siow. MCP-Bench: Benchmarking Tool-Using LLM Agents with Complex Real-World Tasks via MCP Servers, August 2025. URL http://arxiv.org/abs/2508.20453. arXiv:2508.20453 [cs.CL].

Ulrike Wittig, Maja Rey, Andreas Weidemann, Renate Kania, and Wolfgang Muller. SABIO-RK: An updated resource for manually curated biochemical reaction kinetics. *Nucleic Acids Research*, 2018. doi: 10.1093/nar/gkx1065. URL https://doi.org/10.1093/nar/gkx1065.

Xinbo Xu, Ruihan Yang, Haiyang Shen, Wendong Xu, Bofei Gao, Ruoyu Wu, Kean Shi, Weichu Xie, Xuanzhong Chen, Ming Wu, Jason Zeng, Michael Heinrich, Elvis Zhang, Liang Chen, Kuan Li, and Baobao Chang. RoadmapBench: Evaluating Long-Horizon Agentic Software Development Across Version Upgrades, May 2026a. URL http://arxiv.org/abs/2605.15846. arXiv:2605.15846 [cs.SE] version: 1.

Zhangchen Xu, Junda Chen, Yue Huang, Dongfu Jiang, Jiefeng Chen, Hang Hua, Zijian Wu, Zheyuan Liu, Zexue He, Lichi Li, Shizhe Diao, Jiaxin Pei, Jinsung Yoon, Hao Zhang, Mengdi Wang, Radha Poovendran, Misha Sra, Alex Pentland, and Zichen Chen. AutoLab: Can Frontier Models Solve Long-Horizon Auto Research and Engineering Tasks?, June 2026b. URL http://arxiv.org/abs/2606.05080. arXiv:2606.05080 [cs.AI].

Zhi Xu and Yun Fu. NoisyCausal: A Benchmark for Evaluating Causal Reasoning Under Structured Noise, May 2026. URL http://arxiv.org/abs/2605.04313. arXiv:2605.04313 [cs.CL].

John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. Intercode: Standardizing and benchmarking interactive coding with execution feedback. *Neural Information Processing Systems*, October 2023. doi: 10.48550/arxiv.2306.14898. URL https://doi.org/10.48550/arxiv.2306.14898. arXiv:2306.14898 [cs].

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Adriano Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Neural Information Processing Systems*, 2024. doi: 10.48550/arxiv.2405.15793. URL https://doi.org/10.48550/arxiv.2405.15793. arXiv:2405.15793 [cs].

Junlin Yang, Dylan Zhang, Xiangchen Song, Qirun Dai, Xiao Liu, Yuen Chen, Aniket Vashishtha, Jing Shi, Chenhao Tan, and Hao Peng. CausaLab: A Scalable Environment for Interactive Causal Discovery Toward AI Scientists, May 2026. URL http://arxiv.org/abs/2605.26029. arXiv:2605.26029 [cs.AI].

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains, 2024. URL http://arxiv.org/abs/2406.12045. arXiv:2406.12045 [cs].

Zhenyu Yu and Shuigeng Zhou. Caliper: Probing Lexical Anchors versus Causal Structure in LLMs, June 2026. URL http://arxiv.org/abs/2606.04915. arXiv:2606.04915 [cs.CL].

M. Zecevic, Moritz Willig, D. Dhami, and K. Kersting. Causal parrots: Large language models may talk causality but are not causal. *Trans. Mach. Learn. Res.*, 2023, 2023. doi: 10.48550/arxiv.2308.13067. URL https://doi.org/10.48550/arxiv.2308.13067. arXiv:2308.13067 [cs].

Yinger Zhang, Shutong Jiang, Renhao Li, Jianhong Tu, Yang Su, Lianghao Deng, Xudong Guo, Chenxu Lv, and Junyang Lin. DeepPlanning: Benchmarking Long-Horizon Agentic Planning with Verifiable Constraints, January 2026. URL http://arxiv.org/abs/2601.18137. arXiv:2601.18137 [cs.AI] version: 1.

Yu Zhou, Xingyu Wu, Jibin Wu, Liang Feng, and Kay Chen Tan. CausalBN-Bench: A comprehensive benchmark for causal learning capability of LLMs. *IEEE Transactions on Artificial Intelligence*, pp. 1–15, 2026. doi: 10.1109/tai.2026.3703427. URL https://doi.org/10.1109/tai.2026.3703427. arXiv:2404.06349 [cs].

A Appendix

A.1 Motif catalog

Fig. 5 shows the base motif templates used in this work. When grafting, these are the graph skeletons used by the sampler. Gray dashed nodes denote latent variables.

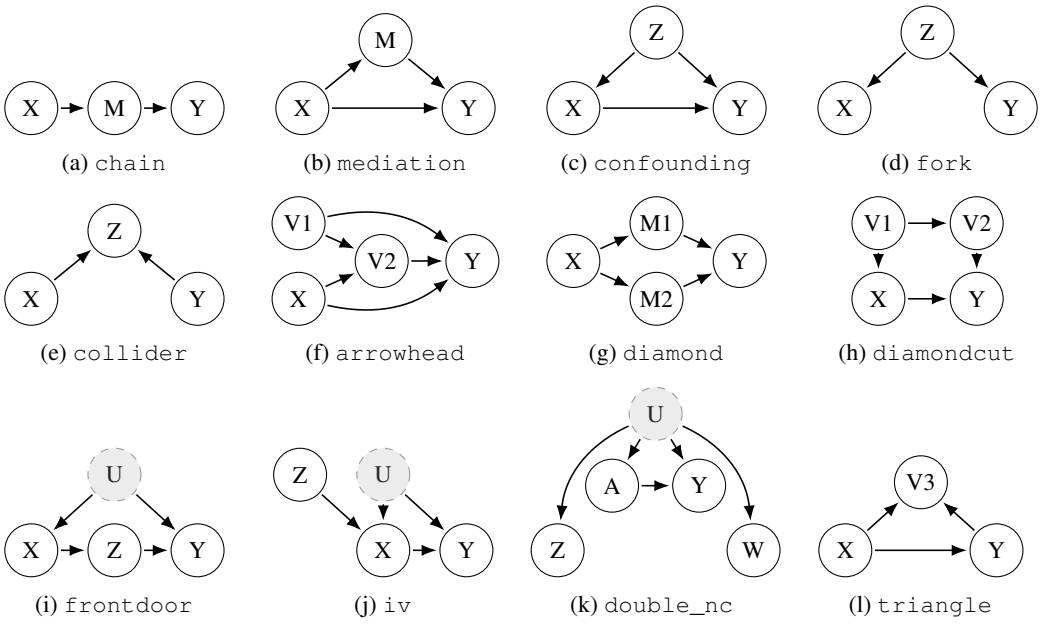


Figure 5: **Base motif templates.** Latent variables are shown as gray dashed nodes.

A.2 Extended related work

Tab. 3 summarizes the closest comparisons; the following discussion places them within four broader lines of work.

Real-study causal analysis benchmarks. The closest real-data benchmarks evaluate causal analysis over scientific or policy studies. *CauSciBench* (Acharya et al., 2026) combines research-paper and textbook tasks with synthetic scenarios whose true effects are sampled under fixed outcome models. Agents select a causal method and implement it in Python; the reported metrics score method-selection accuracy and relative effect error, with requested standard errors excluded from scoring. The real-paper subset uses reference estimates, while the synthetic subset supplies model-derived targets. *CausalReasoningBenchmark* (Sawarni et al., 2026) evaluates structured identification specifications and executed effect estimation on curated real datasets. Its scoring includes point estimates, standard errors, and confidence-interval overlap against reference analyses. Thus, statistical-uncertainty evaluation has a direct precedent, with a different reference: CausalDS scores estimates and intervals against hidden SCM-derived targets.

InterveneBench emphasizes intervention-centered study-design reasoning in real social systems without predefined graphs or structural equations (Shi et al., 2026). The study-derived instances provide strong external realism, but their dependence on existing papers, datasets, or study designs makes them potentially vulnerable to data contamination (the "causal parrot" issue).

Synthetic causal benchmarks and causal reasoning probes. A second line evaluates formal causal reasoning in controlled settings. *CLadder* (Jin et al., 2023) constructs templated natural-language questions from fixed graph motifs and Bernoulli causal Bayesian networks, with oracle-computed answers spanning all three rungs of Pearl's ladder. Its queries are identifiable by construction, and the required probabilities are supplied in the prompt.

UniCo (Dai et al., 2026) samples graphs and conditional probability tables, obtains exact-inference answers for 18 query types across the ladder, and renders questions in symbolic, code, and LLM-narrated forms. CausalDS addresses a different statistical problem: UniCo supplies probabilities or mechanisms needed to answer a query, while CausalDS withholds generating mechanisms and requires estimation from finite observational samples. UniCo's generated benchmark does not evaluate tool-executed data analysis or statistical intervals, and has no noisy-measurement layer; its code form is an input representation of the model. Its identification questions do include cases with no identifying variable set, but its numerical queries are answerable by construction and do not score abstention on non-identifiable estimates. CausalDS combines tool-executed estimation, interval scoring, and such abstention, including scenes with noisy measurements and calibration data. The benchmarks therefore measure different capabilities: reasoning from supplied causal information in UniCo, and causal data-science analysis of narrated observational files in CausalDS.

CauSim (Astorga et al., 2026) builds executable SCMs of increasing complexity and renders them in natural language, checking each translation by mapping it back to an executable SCM. This supplies supervision for reasoning with a given causal simulator or its verbalization. CausalDS instead keeps the generating mechanisms private and asks an agent to estimate identified quantities from observational data. The distinction is therefore between answering queries about a supplied causal model and performing statistical analysis of data generated by a hidden one.

Related graph-centric or format-dependent tests of causal understanding include *CausalGraph2LLM*, *CLEAR*, and *CausalBench* (Sheth et al., 2024; Chen et al., 2024; Zhou et al., 2026). *Executable Counterfactuals* focuses on code-mediated counterfactual reasoning (Vashishtha et al., 2026).

CausalProfiler (Panayiotou et al., 2026) is a synthetic data generator for causal-ML estimators. It samples causal models, observational data, queries, and ground truth for finite-sample estimation of ATE, CATE, and counterfactual total effects. Its estimation-error comparisons include identification-consistent, misspecified-graph, and hidden-confounding regimes, without assigning query-level non-identifiability labels or scoring abstention. It therefore provides a precedent for finite-sample effect evaluation against known causal-model targets. CausalDS combines that statistical setting with narrated scenes and tool-using agents that perform the analysis.

Language Models as Causal Effect Generators introduces sequence-driven SCMs for treatment-effect benchmarking (Bynum & Cho, 2025); *CausalSynth* realizes SCM-sampled records as unstructured text under iterative consistency checks (Cheng et al., 2026). *NoisyCausal* evaluates natural-language causal QA under structured noise (distractors, perturbed values, latent confounders, partial masking) (Xu & Fu, 2026).

CausalDS differs by making the benchmark unit a narrated fully-fledged scene: a hidden SCM is realistically grounded into domain variables, rendered as a graph-audited free-form story, packaged with synthetic SCM-driven data files, and the agent's tool-backed answers are then scored deterministically.

Answerability and abstention. A third line asks whether models recognize when a causal answer is unwarranted. *CausalT5k* (Geng et al., 2026) evaluates narrative causal-claim judgments across all three rungs of Pearl's ladder and scores informed refusal under insufficient evidence. Human-authored case types are expanded by an LLM and reviewed by humans; solver checks validate labels for rungs 2 and 3 against qualitative causal structures. Its utility/safety split penalizes both unsupported answers and over-refusal. *CausalPitfalls* stresses classical statistical traps such as confounding, Simpson's paradox, and selection bias (Du et al., 2026), and *CauGym* generates probability-style questions from random DAGs and, in its insufficient-information split, requires the answer `LACK_CONDITION` under exact-match scoring (Chen et al., 2026a). CausalDS's abstention behaves differently: a formally specified estimand is judged non-identifiable from the observational law under the story-implied graph, and the same scene couples the outcome of that judgment to numerical estimation. This abstention criterion is complementary to verifier-style scoring approaches proposed for causal expressions (He et al., 2026): a verifier checks whether an emitted causal expression is formally correct, whereas abstention scoring asks the orthogonal question of whether the agent recognizes that no formally correct answer exists.

Agentic data-science benchmarks and causal-agent systems. Outside causality, *DS-1000*, *DS-Bench*, *MLE-bench*, *MLE-Dojo*, *BLADE*, and *DiscoveryBench* evaluate code generation, ML engineer-

ing, or open-ended data-analysis workflows (Lai et al., 2022; Jing et al., 2025; Chan et al., 2025; Qiang et al., 2025; Gu et al., 2024; Majumder et al., 2025), often using interactive execution frameworks such as *InterCode* and *SWE-agent/mini-swe-agent* (Yang et al., 2023; 2024). These benchmarks motivate our file-backed, tool-using evaluation protocol, but they do not provide hidden causal structure, interventional or counterfactual task families, or abstention targets for non-identifiable estimands. Within causal inference, *CAIS* automates causal method selection and execution from a dataset-plus-query input (Verma et al., 2025).

CausalGame (Chen et al., 2026b) evaluates active experimental discovery across 14 fixed game narratives driven by SCMs, with hidden confounders, injected measurement noise, and selection bias. Agents collect data and can run sandboxed Python analyses; their explanation reports are scored on supported causal claims. *CausaLab* (Yang et al., 2026) uses a fixed laboratory narrative with a freshly sampled hidden SCM in each episode. Agents choose budgeted interventions through an experimental interface, receive measurements, and are scored on held-out-target prediction and DSL-based mechanism recovery. Its hidden disturbances enter the structural dynamics. Both evaluate an experimental causal discoverer, including its choice of experiments; CausalDS asks whether an agent can act as a causal data scientist on a common exam: every model receives the same evidence and is evaluated against the same ground truth, allowing controlled comparison on the several axes CausalDS examines.

Prior work establishes precedents for individual components of CausalDS. To the best of our knowledge, no other benchmark integrates sampled causal graphs, graph-audited free-form stories, SCM-generated tabular data with clean and noisy-measurement variants, and tool-using agent tasks spanning all three rungs of Pearl's ladder with deterministic scoring of identification, numerical estimates, statistical intervals, and abstention on non-identifiable estimands against hidden causal-model ground truth.

A.3 Scene synthesis algorithms

Naming the abstract nodes of a sampled graph can introduce semantic drift and "accidental edges": a choice of variable names that implies a direct cause the graph does not contain, or hides one it does. Scene synthesis therefore audits both the variable mapping and the story against the graph and repairs them iteratively. Alg. 1 gives the top-level control flow; the audit rules and yields are in App. A.4, and the prompt templates in App. A.19.1.

Algorithm 1: Scene synthesis: variable mapping and story verbalization

Input: DAG G; optional stage plan P; budgets R, T, S for seeding, mapping audit/repair, and story verification.
Output: domain D, mapping M, verified story σ, or failure.

If P is absent, set $P \leftarrow [G]$; let H_0 be the first stage and initialize its fixed-name assignment $F_0 \leftarrow \emptyset$.
if CauseNet seeding is enabled **then**
    Try up to R CauseNet matches on H_0; take the first passing PREAUDIT (Alg. 2), or fail if seeding is mandatory.
end if
$(D, M) \leftarrow$ MAPSTAGE(H_0, F_0, T) (Alg. 3); **return** failure if it fails.
for each auxiliary stage H_i with shared anchor a_i **do**
    Extend M with MAPSTAGE(H_i, F_i, T), where F_i fixes a_i to its current name and forbids names from other stages; fail on mapping failure, anchor drift, or name duplication across stages.
end for
if there are auxiliary stages and the merged-graph audit is enabled **then**
    Run the audit/repair loop on M over the full G for up to T rounds, preserving anchor meanings; **return** failure if it fails.
end if
$\sigma \leftarrow$ VERBALIZE(D, M, G, S) (Alg. 4); **return** failure if it fails.
return (D, M, σ).

Alg. 1 calls three subroutines. PREAUDIT (Alg. 2) is the LLM feasibility check on a CauseNet seeding: whether the fixed real-world concepts can be completed into a plausible graph before any mapping is attempted. MAPSTAGE (Alg. 3) names the variables of one (sub)graph with a *mapper–auditor* loop: the mapper proposes a domain and names, the auditor checks every edge and non-edge for plausibility and flags names that imply links the graph lacks, and the mapper regenerates from the

auditor's feedback. Grafted graphs are mapped stage-wise through the same loop: the host graph first, then each auxiliary fragment with the shared anchor fixed and names from other stages forbidden, and finally a global audit of the merged graph. The mapper may use web search to pin down realistic variable names, units, and story details. VERBALIZE (Alg. 4) drafts the narrative from the mapped graph and revises it against a story-to-DAG verifier until no hard violation (missing variable, missing edge, reversed direction) remains.

Algorithm 2 LLM pre-audit for CauseNet seeding

Input: stage graph H with CauseNet-fixed nodes F.
Output: feasibility verdict (`feasible`, `confidence`, `reason`).

1: Extract directed edges E, all unordered non-edges $\mathcal{N}$, and the fixed–fixed subset $\mathcal{N}_F \subseteq \mathcal{N}$.
2: Prompt with fixed nodes, remaining placeholders, E, $\mathcal{N}_F$, and $\mathcal{N}$.
3: Ask whether a skilled mapper could interpret the fixed concepts and name placeholders so the graph constraints remain plausible.
4: Reject iff a fixed–fixed non-edge has an unavoidable mainstream direct-cause reading, or a fixed–fixed edge is impossible.
5: Otherwise accept, using `low` confidence when only narrow mainstream interpretations make the constraints work; uncertain cases default to feasible (as the following stages can still reject them)
6: **return** the feasibility verdict.

Algorithm 3 MAPSTAGE: name the (sub)graph variables with audit/repair

Input: (sub)graph H; fixed-name assignment F (possibly empty); audit retry budget T.
Output: mapping M for H, or failure.

1: The mapper proposes M under fixed-name, completeness, and uniqueness constraints.
2: **for** $t = 1$ to T **do**
3: The auditor checks edge plausibility, absence of plausible direct links for non-edges, and conditional independence/type-consistency signals.
4: **if** no violations remain **then**
5: **return** M
6: **end if**
7: **if** a violation cannot be fixed without changing fixed names **then**
8: **return** failure
9: **end if**
10: The mapper regenerates M from a feedback prompt provided by the auditor, listing the violations.
11: **end for**
12: **return** failure.

Algorithm 4 VERBALIZE: story generation with story-to-DAG verifier loop

Input: domain D, mapping M, DAG G, story retry budget S.
Output: verified story σ, or failure.

1: Build yellow-flag context Φ by scanning M for derived-name markers and restrictive qualifiers (cf. App. A.4)
2: The story writer drafts σ from (D, M, Φ).
3: **for** $s = 1$ to S **do**
4: The verifier checks σ against (G, M): missing variables, missing edges, and direction contradictions are hard violations; extra direct effects, domain drift, graph jargon, and plausibility/coherence issues are warnings.
5: **if** there are no hard violations **then**
6: **return** σ
7: **end if**
8: The story writer revises σ from the verifier's feedback listing the hard violations, warnings, and Φ.
9: **end for**
10: **return** failure.

A.4 Mapping and narration audit details

Production configuration and yield. The initial production batch used Kimi K2.5 for variable mapping, narration, and their audits. A regeneration run was done with Kimi K2.6 to retry all 151 failed instances, complete five previously unattempted instances, and replace one scene removed after a quality check. Tab. 4 reports the generation yield, showing that mapping was the main failure stage in both batches. In the results presented in this work, we adopted a policy of seeding the outcome node and one of its neighbors (on the realistic exam, 83 scenes seeded with two nodes, 17 with one). Seeding is not fully deterministic: the mapper may still rename a seeded node within its topic.

Table 4: **Generation yield.** Failures are classified by stopping stage.

Batch (model)	Attempted	Mapping failures	Story failures	Completed	Yield
Initial (Kimi K2.5)	895	146	5	744	83.1%
Regeneration (Kimi K2.6)	157	43	0	114	72.6%

Both the mapping and story auditors are augmented with a deterministic preprocessor that scans variable meanings for derived-name markers (e.g. “corrected”, “residual”, “count of”, “index based on”) and qualifiers (e.g. “based only on”, “not influenced by other factors”), and appends a yellow-flag for the auditor. This targets a recurring failure mode where a construct is narrated as directly causing a record/ID meant to merely keep track of it.

Story-to-graph fidelity check. As both our generation and auditing are LLM-based, we use the exam results as an independent check that the released stories reliably encode their hidden graphs. The `causal_sketch` family asks an agent to determine the graph implied by the story alone. It is scored by edge-set F_1 against the hidden graph, so a narrated but unrecoverable edge lowers recall and an edge the prose over-implies lowers precision. Over twelve models and three replications of the realistic exam, the 17 sketch tasks (12 directed, 5 “skeleton”) are recovered with $F_1{=}1$ in 606 of 612 task-runs (Tab. 20). Three of the misses are provider-side safety refusals by Claude Fable 5.1 on one scene; the remaining three are single-run slips (F_1 of 0.5 twice on one skeleton task, 0.83 once on a grafted directed graph) by models that recovered the same scenes perfectly in their other runs. The four sketch tasks of the verbalization-swap ablation (App. A.15) are also recovered perfectly by both models. Twelve models from independent providers thus agree with the hidden graph on every scene they are asked about, which indicates more-than-sufficient story quality for the purposes of this benchmark; the stories are also plausible enough to trigger Anthropic’s refusal mechanism, as mentioned previously.

A.5 SCM profile registry

CausalDS supports a rich family of functional relations in the generated SCMs: each scene draws one continuous and one binary profile from the two profile registries in Tab. 5, so that the node types can be either continuous or binary. A profile determines the functional mechanism family and the parameter ranges for the node. Root nodes are noise-only (continuous) or Bernoulli with a profile-specific base-rate range (binary). Continuous noise is additive and drawn from Gaussian, Laplace, Student-t, or Gaussian-mixture families. Binary mechanisms produce a Bernoulli output through a logistic, threshold, or noisy-gate link with no separate noise term, the Bernoulli draw being the source of stochasticity. These noise specifications support Monte Carlo evaluation of treatment effects, mediation, and counterfactuals using the same SCM. The empirical profiles take their families from the equation and rule corpora described in App. A.7. The profile mixture weights in the last column of Tab. 5 were likewise derived from the same corpora, and the synthetic profiles serve as a uniform coverage prior in the mixture. Synthetic regimes include neural-network mechanisms, mixture noise, and sharp thresholds. The SCM profiles are not a targeted difficulty axis for the realistic exam, as their weights and compositions are determined by the empirical grounding (the “realistic” part); for the hard exam, we tilt them towards more complex mechanisms in the observation model (cf. App. A.6, App. A.10).

Table 5: **SCM profile registry.**

Profile	Source	Mechanism	Weight
Continuous profiles			
`interaction_response`	empirical	product, saturating, or cooperative (Hill) response of the parents	0.234
`symbolic_transform`	empirical	rational, power, exp/log, or piecewise transform of a linear parent index	0.053
`linear_additive`	empirical	affine mean function $b + \sum_i w_i x_i$	0.013
`generic_additive_baseline`	synthetic	affine mean function with random-sign Gaussian weights and Gaussian bias	0.140
`generic_smooth_handcrafted`	synthetic	linear term, shifted $\tanh$ nonlinearity, and pairwise parent interactions	0.140
`generic_neural_blackbox`	synthetic	random $\tanh$ network (2–4 layers, 8–64 units, spectral-normalized weights), output calibrated to $[-1, 1]$	0.140
`generic_mixture_noise`	synthetic	handcrafted family; each node draws Gaussian-mixture noise (2–6 components) with probability 0.65	0.140
`generic_heteroscedastic`	synthetic	handcrafted family; each node draws a parent-dependent noise scale with probability 0.7	0.140
Binary profiles			
`multi_input_logic_gate`	empirical	each node draws a random Boolean gate (prob. 0.64; each input negated with prob. 0.52), noisy-AND (0.19), or noisy-OR (0.16)	0.204
`single_regulator_logistic`	empirical	logistic link on a linear-plus-softsign index	0.096
`generic_shifted_logistic`	synthetic	logistic link with a shifted softsign nonlinearity	0.140
`generic_sharp_threshold`	synthetic	sharp logistic threshold on the weighted parent sum	0.140
`generic_noisy_or`	synthetic	noisy-OR gate over binarized parents (flip probability 2–10%)	0.140
`generic_noisy_and`	synthetic	noisy-AND gate over binarized parents (flip probability 2–10%)	0.140
`generic_signed_logic_high_negation`	synthetic	random Boolean gate; each input negated with probability 0.55	0.140

A.6 Observation-model implementation details

Mechanism families and noise. As noted, every observation bundle has a single conceptual parent Z ("measured node"). Continuous *observed* columns use either the handcrafted form $h_r(z) = wz+b+w_{\mathrm{nl}} \tanh(z-c)$ or, with probability p_{nn}, a small spectral-normalized $\tanh$ network calibrated to the realized range of Z. Binary columns use a logistic link on $wz + b + w_{\mathrm{nl}} \operatorname{softsign}(z - c)$. The shift c is drawn per measurement so that the flat region of the nonlinearity, where $h_r'(z) \approx 0$ and the Fisher information vanishes, does not systematically coincide with the realized treatment-contrast region (cf. "*Admissibility checks*" below). Continuous noise is additive with a base scale σ_0 drawn per measurement; its family is a Gaussian mixture with probability p_{mix} and otherwise Gaussian, Laplace, or Student-t with equal probability. With probability p_{het} the scale is heteroscedastic, $\sigma(z) = \operatorname{softplus}(w_\sigma z+b_\sigma)$ clamped to $[\sigma_{\min}, \sigma_{\max}]$ and multiplying σ_0, so the local Gaussian approximation to the measurement's Fisher information is location-dependent: $I_r(z) \approx (h_r'(z))^2/(\sigma_0\,\sigma(z))^2$. A released variant fixes these probabilities together with the number of measured nodes, the number of measurements per node, the range of σ_0, and the fraction of training rows used for calibration. `proxy` measures 1–3 non-outcome nodes with 2–10 columns each, $(p_{\mathrm{nn}}, p_{\mathrm{mix}}, p_{\mathrm{het}}) = (0.3, 0.3, 0.3)$, $\sigma_0 \in [1, 3]$, and a calibration fraction of 15%; `proxy_hard` measures 2–5 nodes with $(0.45, 0.45, 0.6)$, $\sigma_0 \in [1.5, 4]$, and 10%; the hard exam's `proxy_severe` measures 3–6 nodes, allows the outcome node to be among the candidates, with 2–4 columns each (fewer columns make the problem harder), $(0.5, 0.5, 0.7)$, $\sigma_0 \in [2.5, 5]$, and 4%.

Admissibility checks on candidate bundles. Candidate bundles for a continuous conceptual variable Z are screened by estimated *Fisher information* to reject nearly uninformative measurements.

Let $I_r(z)$ denote the estimated local Fisher information about Z carried by measurement W_r at $Z = z$, and let $I(z) = \sum_r I_r(z)$ denote the bundle's total information. These estimates use local fits to the generated (Z, W_r) pairs: a linear-Gaussian approximation for continuous measurements and a Bernoulli logistic model for binary measurements. We evaluate $I(z)$ on a grid over the realized support of Z and accept the bundle only if its Fisher-information estimates pass all three checks: (i) $\min_z I(z) \geq \tau_{\min}$ (no dead zones), (ii) $\bar{I} \geq \tau_{\text{avg}}$ (sufficient average information), and (iii) the lower-tail quantiles $I_{0.10}$ and $I_{0.25}$ clear their respective thresholds. Here, $\bar{I}$ is the mean over grid points and I_q is the q-quantile of the grid values. For binary Z, we sum class-separation scores across measurements: squared standardized mean differences for continuous measurements, and mutual information for binary measurements. Scenes that also place a measurement bundle on the outcome apply the same checks with stricter thresholds. The sampler retries the mechanism draw up to a configured budget and fails the bundle afterward. The setting used for the results presented is $\tau_{\min} = 0.03$.

Calibration rows. The calibration dataset is a coverage-oriented selected subset of the public training rows. We first select rows from each quantile bin of *every* measured conceptual variable. We then fill the remaining slots within each bin by selecting rows whose variable ranks are farthest (in the joint space of all selected conceptual variables to be "proxied") from those already selected.

This preserves each measured variable's marginal support, which fitting the measurement model $W_{jr} \mid Z_j$ across the range of Z_j requires, and since every measurement depends on a single conceptual parent with independent noise, the conditional law of W_{jr} given Z_j is the same in the calibration rows as in the population. The *joint dependence* among measured variables is attenuated by the same construction: spreading points in rank space favors regions of low joint density — in the exam scenes, pairwise rank correlations between measured variables drop to roughly a quarter to a third of their population values. Population relationships among conceptual variables are thus hard to estimate from the calibration set; in particular, when the outcome is measured, an outcome regression directly fitted on the calibration rows is biased toward no effect.

Recoverability diagnostics in the private bundle. Beyond the Fisher-information screen at generation time, each measured conceptual variable Z_j is privately scored by how well it can be recovered from its d_j measurements. Every score is evaluated on the scene's held-out test split and expressed as R^2 for continuous Z_j and AUC for binary Z_j, so scores are comparable across scenes. For diagnostics, we use three reference fits: the *lower bound* S_{low} is the best single-measurement *linear* model (OLS or logistic regression) fitted on the public calibration rows, the score a naive agent would obtain. S_1 is the best single-measurement *nonlinear* model. The *upper bound* S_{upp} is a gradient-boosting model on all d_j measurements jointly, fitted on the full training rows with their gold conceptual values (i.e., labels no agent ever sees), defining an oracle-like ceiling that caps even an oracle's downstream performance. The gap between the bounds:

$$\Delta \;=\; S_{\text{upp}} - S_{\text{low}} \;=\; \underbrace{(S_1 - S_{\text{low}})}_{\text{misspecification gap}} \;+\; \underbrace{(S_{\text{upp}} - S_1)}_{\text{multi-measurement/data information gain}}, \tag{5}$$

measures how much a skilled data-science agent can gain over the naive fit: the first part is what a flexible nonlinear model recovers from the single best column, the second what having all d_j columns *and* all rows would add on top. A large Δ means the scene rewards a careful measurement model, and a small Δ means one column already carries most of the recoverable signal. A further fit, S_{cal}, applies the same gradient-boosting model to all d_j measurements using the public calibration rows alone. It is recorded next to the two bounds as the score an agent can reach from the released files with a reasonably flexible model.[2] Fig. 6 shows S_{low}, S_{cal}, and S_{upp} on a representative scene.

Realized measurement mechanisms and difficulty. Within one observation variant the bundles still vary from scene to scene: the number of measured nodes, the share of neural mechanisms, and the noise family and scale are all random draws. Unlike the SCM profile (App. A.5), whose mechanism shape reaches an agent only through the joint distribution it estimates from, a measurement mechanism has to be *inverted*: the agent recovers the measured variable from its bundle through the calibration rows. The functional form of the link thus bears more directly on the task, and, e.g., a non-monotone neural network mechanism is what widens the misspecification gap of Eq. 5 for an agent that fits

[2] In practice, the agents are expected to exceed this since we don't optimize the model iteratively.

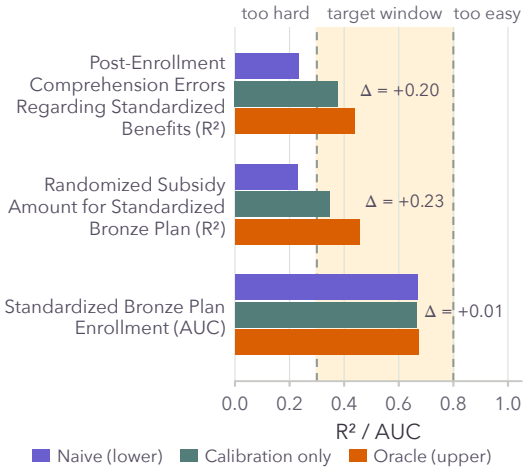


Figure 6: **Per-scene recoverability diagnostic.** The `proxy_severe` variant of one hard-exam-pool scene with three measured conceptual variables (two continuous, one binary). Violet bars show the naive lower bound, teal bars show the calibration-only fit, orange bars show the "oracle" upper bound. The shaded band marks the target window (0.3–0.8) used during scene generation: below it is "too hard," above it "too easy." We display the gap (Eq. 5) adjacent to the bars.

linear calibration models. This is the axis along which the variants are constructed to be hard, and it shows within variants as well. Regressing a per-task unit loss (clipping and scaling continuous-task losses to $[0, 1]$) on model identity and task composition (output variant, structural label, treatment and outcome type) within one variant slice, with the realized share of neural-network mechanisms added as a covariate, a larger share of neural-network mechanisms goes with a higher loss in the realistic exam's `proxy_hard` slice and the hard exam's `proxy_severe` slice (68 scenes), by 0.22 and 0.08 loss units per standard deviation of the share (p-values $p = 0.02$ and 0.015 from reshuffling the share across scenes).

Relation to proximal causal inference. The calibrated observation layer is distinct from proximal causal inference (Tchetgen et al., 2024; Miao et al., 2018). Proximal causal inference addresses causal-effect identification when relevant confounding mechanisms are not directly observed, using auxiliary measurements or negative controls under additional assumptions about their relation to the unobserved confounding structure. In *CausalDS*, the observation model has no such role: the conceptual DAG and SCM are fixed before the observation layer is applied, and each measurement bundle is generated as a noisy observation of *one* conceptual variable. These released columns are not additional causal variables in the conceptual graph and cannot provide a new route to identification as they cannot create or remove confounding, instruments, or mediators. Thus, the observation layer changes estimation difficulty from the released files, not the conceptual estimand or its identifiability.

A.7 Optional empirical grounding of composition axes

CausalDS is fully synthetic at the scene level, but a number of *composition axes* can be grounded from external corpora when we want a specific benchmark mixture to better resemble real causal-analysis workloads, as is the case for our *realistic exam* (Sec. 4).

The key design choice is to *ground composition axes*: we derive axis-specific empirical distributions from several benchmark corpora and companion data, weight them by how directly they expose the quantity of interest (as our concepts such as question type might not map one-to-one), and then allow benchmark-design tilts when needed for difficulty, coverage, or stress-testing (e.g. the exam-side difficulty knob of App. A.8). The composition axes are split into a *generation* side and an *exam* side. Generation axes (variable types; main-motif weights, auxiliary-motif weights, and graft probability; identifiability proportion; typed SCM mechanism profiles; observation-model profile composition) determine the type of scenes that get sampled and built, while exam axes shape which task, required

output type, and measurement view are selected from a scene in a given exam. This split lets the same scene support multiple released exams without re-running expensive verbalization.

The exam side would require estimating the full joint $P(\text{motif}, \text{question type}, \text{output variant}, \text{observation variant})$ from available data, which is not possible given the available empirical sources. Thus, we factor the *exam side* through the question type:

$$P(S, Q, \text{output}, \text{obs}) = P(S)\,P(Q \mid S)\,P(\text{output} \mid Q)\,P(\text{obs} \mid Q), \qquad \text{output} \perp\!\!\!\perp \text{obs} \mid Q,$$

where Q denotes the question type, and S a structural label: motif-based or grafted DAG. This factorization is illustrated in Fig. 7.

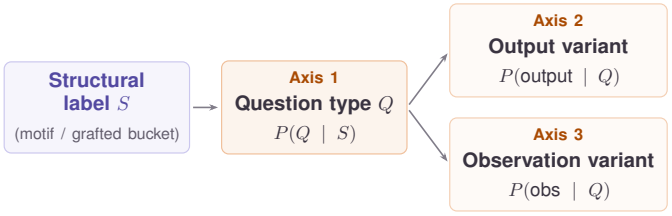


Figure 7: **Exam-composition factorization.** Conditional on the structural label S (motif or the synthetic `grafted` bucket), the exam builder draws the question type Q, then independently draws the output variant and the observation variant given Q. Each conditional corresponds to one exam-side axis: $P(Q \mid S)$ is Exam Axis 1, $P(\text{output} \mid Q)$ is Exam Axis 2, and $P(\text{obs} \mid Q)$ is Exam Axis 3.

Tab. 6 lists the external sources used, Tab. 7 shows which source served which axis and in what role, and Tab. 8 gives, per axis, the estimated quantity, how the sources are combined, and the benchmark-design tilt applied on top of the empirical estimate for the realistic exam. Further details follow in *Notes on individual axes.* When numbers are quoted, we refer to the specific composition chosen for the presented *realistic* exam.

Table 6: **External grounding sources.** *Abbr.*: abbreviations used in Tab. 7.

Abbr.	Source	Kind
EC	*EconCausal* (Lee et al., 2026)	real-study causal benchmark
CSB	*CauSciBench*, real-study subset (Acharya et al., 2026)	real-study causal benchmark
CSB-QR	*CauSciBench*, QR subset (Liu et al., 2024)	textbook benchmark
CRB	*CausalReasoningBenchmark* (Sawarni et al., 2026)	real-study causal benchmark
IB	*InterveneBench* (Shi et al., 2026)	real-study causal benchmark
ReCITE	*ReCITE* (Saklad et al., 2025)	real-paper causal graphs
CLadder	*CLadder* (Jin et al., 2023)	synthetic motif benchmark
EI	*EvidenceInference* (DeYoung et al., 2020)	biomedical RCT reports
TMW	*TellMeWhy* (Lal et al., 2021)	narrative why-questions
CG	*CausalGame* (Chen et al., 2026b)	challenge suite
CP	*CausalPitfalls* (Du et al., 2026)	challenge suite
BM	*BioModels* (Malik-Sheriff et al., 2020)	biochemical equations
SABIO	*SABIO-RK* (Wittig et al., 2018)	biochemical equations
BBM	*BiodivineBooleanModels* (Pastva et al., 2023)	Boolean rules
KDP	*Kadelka DesignPrinciples* (Kadelka et al., 2024)	Boolean rules

Notes on individual axes. *Variable types.* Some high-cardinality discrete variables are better modeled as effectively continuous in downstream estimation, and a modest tilt toward continuous variables raises data-analysis difficulty without changing the causal semantics.

Auxiliary motifs and grafts. The graft-count prior is hand-authored rather than taken from corpus frequencies because the available real-paper graph complexity source (*ReCITE*) violates $P(2 \text{ grafts}) < P(1 \text{ graft})$, a constraint we hold to be structurally justified.

Identifiability. Real-world data on non-identifiable cases is scarce as "solvability" is typically desirable for published problems. This generation axis uses the population ATE as a one-item proxy for

Table 7: **Source roles by composition axis.** •: the source contributes frequency mass to the axis's empirical prior; ○: the source is a companion that calibrates structure or coverage. Source abbreviations as in Tab. 6.

Axis	EC	CSB	CSB-QR	CRB	IB	ReCITE	CLadder	EI	TMW	CG	CP	BM	SABIO	BBM	KDP
Generation axes															
Variable types	•	•	•	•											
Main motif	•	•		•	•	○	○								
Aux. motifs / grafts						•	○								
Identifiability	•	•								○	○				
SCM profiles												•	•	•	•
Exam axes															
Question type	•	•		•			○	•							
Output variant		•	•	•											
Observation variant				○		•		•	○						

identifiability, while on the exam, identifiability is checked for each requested estimand type. Under the benchmark's assumptions, the ATE is non-identifiable in `double_nc` and `iv`. The generation configuration requests a quota of non-identifiable cases (10% for the presented realistic exam). If the two motifs above cannot satisfy it, we inject latent confounding $U \to \{X, Y\}$ into other motifs and accept only graphs whose population-ATE identifiability check returns `False` (i.e., we use rejection sampling).

SCM profiles. The empirical evidence comes from a biology-heavy source pool and is concentrated on multi-parent interaction shapes and multi-input gates. Because of this narrow domain and limited mechanism diversity, the empirical part is demoted to $\alpha = 0.3$ in favor of the uniform coverage prior, and the compact profile names keep the registry domain-neutral.

Question type. Let $\hat{p}_{\text{emp}}(Q)$ be the credibility-weighted family marginal of the sources, which has mass on identification and effect estimation only, and $p_0(Q) = 0.15\,\hat{p}_{\text{emp}}(Q) + 0.85/|\mathcal{Q}|$ its mixture with a uniform floor over the $|\mathcal{Q}| = 10$ deployable families. The structural lift is

$$P(Q \mid S) \;\propto\; p_0(Q) \left(\frac{P_{\textit{CLadder}}(Q \mid S)}{P_{\textit{CLadder}}(Q)} \right)^{\lambda}, \qquad \lambda = 0.15, \tag{6}$$

CLadder being the only source with joint graph × question-type rows. It is generated with equal counts across supported combinations per graph-family and question-type cell, so its question marginal carries no frequency information; only the ratio does, albeit weakly, which is why we pick a small λ.

Observation variant. We use information *accessibility* as a proxy for observation difficulty and order it as: `clean` (the needed information is stated in the given passage) < `partly_explicit` (only partly stated there and the rest must be inferred) < `document_context` (it is stated elsewhere in the document, outside the given passage) < `external_implicit` (it is not in the document at all).

A.8 Exam-side difficulty knob

After the distributions of App. A.7 are built, a separate **difficulty knob** $d \in [0, 1]$ tilts the three exam-side distributions: the question-type conditional $P(Q \mid S)$, the within-family output-variant distribution $P(\text{OutputVariant} \mid \text{QuestionType})$, and the released observation-variant distribution $P(\text{obs} \mid Q)$. We partition question families by Pearl rung $r(Q) \in \{1, 2, 3\}$, index output variants by a within-family difficulty rank $r_q(v)$ with midpoint c_q, and index observation variants by $o(\text{obs}) \in \{0, 1, 2\}$ for $\{\texttt{clean}, \texttt{proxy}, \texttt{proxy_hard}\}$.[3] We multiply the initial weights by the above

[3] `proxy_severe` was constructed for the hard exam specifically.

Table 8: **Per-axis estimation and design tilts.** The quantities estimated, how the sources of Tab. 7 are combined, and the manual tilt applied on top of the empirical estimate.

Axis	Estimated quantity	Combination of sources	Design tilt
Generation axes			
Variable types	Continuous vs. non-continuous share, separately for treatment and outcome	Types coded from source columns, metadata, and variable descriptions; sources credibility-weighted by how well their constructs map to ours	Slight tilt toward continuous variables
Main motif	Weights for the motif inventory	Design-family labels (adjusted observational, DiD/event study, RD, IV, matching) mapped to motifs explicitly; per-source distributions credibility-weighted; *ReCITE* supplies the relative weights when a label is mapped to multiple motifs	*CLadder*'s distribution used as coverage for motifs absent from design-family labels
Auxiliary motifs and grafts	Auxiliary-pool weights; graft-count distribution	*ReCITE* structural catalog over explicit real-paper graphs; *CLadder* for coverage	Graft-count prior hand-authored with a small adjustment towards *ReCITE*'s distribution
Identifiability	Share of scenes with a non-identifiable population ATE (as a stand-in)	Empirical sources are 99.6% identifiable; challenge suites give the *kinds* of obstacles, not their frequency	Manually requested non-identifiable quota.
SCM profiles	Profile weights per node type	We exclude rare equation and rule types and group the remainder into three continuous and two binary profiles. Profile weights are estimated separately for continuous and binary nodes, independently of graph motif.	Convex mix with a uniform synthetic coverage prior (weight 0.3 on the empirical part)(App. A.5)
Exam axes			
Question type	$P(Q \mid S)$	Credibility-weighted empirical marginal over task-family labels, mixed with a uniform coverage prior, then reshaped by a *CLadder* structural lift (Eq. 6)	Floor keeps every rung-1 and rung-3 family off zero under the R2-heavy empirical marginal; small lift exponent
Output variant	$P(\text{output} \mid Q)$	We manually map source answer formats to our output variants. For `identification` and `effect_estimate`, we mix source frequencies with uniform weights over compatible variants; other families start with uniform weights	Tilt toward harder answer contracts in prediction, association, and causal sketch which are among the easier tasks.
Observation variant (Exam Axis 3)	$P(\text{obs} \mid Q)$ over `clean`, `proxy`, `proxy_hard`	Source items are grouped into the four accessibility levels. We estimate distributions directly for graph-recovery and effect-estimation questions, reuse them for related question families, and use the overall source distribution for the remaining families.	The two hardest source levels both map onto `proxy_hard`

difficulty factors, then normalize them to sum to one:

$$\begin{aligned}
\hat{p}_d(Q \mid S) &\propto \hat{p}(Q \mid S)\exp\big(\beta\,(2d{-}1)\,(r(Q){-}2)\big),\\
\hat{p}_d(v \mid q) &\propto \hat{p}(v \mid q)\exp\big(\delta\,(2d{-}1)\,(r_q(v) - c_q)\big),\\
\hat{p}_d(\text{obs} \mid Q) &\propto \hat{p}(\text{obs} \mid Q)\exp\big(\gamma\,(2d{-}1)\,(o(\text{obs}){-}1)\big),
\end{aligned} \tag{7}$$

In Eq. 7, $d = 0.5$ leaves the initial weights unchanged; $d = 0$ favors R1, easier output variants, and `clean`, while $d = 1$ favors R3, harder output variants, and `proxy_hard`. The constants control the strength of each adjustment: β for the Pearl-rung mix, δ for output-variant difficulty within each question family, and γ for observation-variant difficulty. All default to 1 and can be set independently. These difficulty rankings reflect benchmark-design choices: Pearl rung is a coarse indicator, and output variants are ranked within each question family, for example with CIs ranked as more difficult than point estimates. The observation ordering `clean` < `proxy` < `proxy_hard` follows increasing measurement corruption. The realistic exam uses $d = 0.5$, leaving the empirical weights from App. A.7 unchanged.

Post hoc check of the three factors. We examine how model performance varies with observation variant, Pearl rung, and output variant on both exams, using the twelve models and three replications of the leaderboards (Tab. 9). *Score* is the CausalDSScore of a level's tasks pooled over all model runs; *Band* is the 5–95% range over 500 resamples of the level's tasks; *Model-centered* Δ subtracts each model's full-exam score from its score on the level and averages over models; *Adjusted* Δ compares the mean per-task loss ($0/1$ for discrete tasks, normalized error for point tasks, capped normalized interval score for intervals, F_1-loss for graph and set tasks) with a reference level after an additive regression adjustment for model and the other factors. Positive differences indicate worse performance.

The observation ranking holds. On the realistic exam `clean` and `proxy` are indistinguishable while `proxy_hard` is harder on every summary, with an adjusted Δ of $+0.19$ that survives adjustment for model, rung, and output variant; on the hard exam the three proxy tiers order as intended, with adjusted Δ of $+0.48$ (`proxy_hard`) and $+0.73$ (`proxy_severe`) relative to `proxy`. The rung ranking does not hold: performance does not worsen from R1 to R3 on either exam, and on the realistic exam R3 is the easiest rung once observation variant is adjusted for (-0.27 against R1), largely because most of its tasks are counterfactual-identification binaries with high pass rates and its numeric errors are small, whereas R1 carries the prediction tasks with the largest normalized errors (Tab. 20). The output-variant ranking is supported in the aggregate but not family by family: a regression with fixed effects for model, task family, and observation variant estimates a loss increase of 0.25 ± 0.03 per rank step on the realistic exam and 0.96 ± 0.03 on the hard exam, and the interval outputs are the hardest variants of their families on both, yet the direction of the within-family relationship agrees with the chosen ranking in only 5 of 9 realistic-exam families and 3 of 4 hard-exam families. Given this mixed evidence, the hard exam specifies its sampling weights directly and sets $\beta = \delta = \gamma = 0$ (App. A.10).

Table 9: **Model performance by observation variant and Pearl rung.** Twelve models, three replications, corrected grades. *Score*: CausalDSScore of the level's tasks pooled over all model runs; *Band*: 5–95% range over 500 task resamples within the level; *Model-centered* Δ: mean over models of the level score minus that model's full-exam score; *Adjusted* Δ: mean per-task loss relative to the reference level (ref.) after additive adjustment for model and, for observation variants, rung and output variant; rung is adjusted for model and observation variant only, since output variants are nested in rungs. Positive is harder. The hard exam has no R1 tasks, and its `clean` slice holds only symbolic output variants, so its adjusted contrast is not separable (n/s) and `proxy` is the reference there.

Exam	Axis	Level	Tasks	Score	Band	Model-centered Δ	Adjusted Δ
Realistic	Observation	`clean`	48	0.280	$[0.159, 0.439]$	-0.076	ref.
		`proxy`	32	0.155	$[0.087, 0.229]$	-0.200	$+0.031$
		`proxy_hard`	20	0.901	$[0.463, 1.340]$	$+0.540$	$+0.193$
Realistic	Rung	R1	28	0.486	$[0.260, 0.798]$	$+0.137$	ref.
		R2	51	0.376	$[0.208, 0.562]$	$+0.020$	-0.092
		R3	21	0.136	$[0.075, 0.206]$	-0.218	-0.267
Hard	Observation	`clean`	33	0.164	$[0.105, 0.231]$	-0.827	n/s
		`proxy`	24	0.552	$[0.296, 0.904]$	-0.431	ref.
		`proxy_hard`	55	1.063	$[0.709, 1.493]$	$+0.073$	$+0.483$
		`proxy_severe`	88	1.376	$[1.030, 1.778]$	$+0.384$	$+0.734$
Hard	Rung	R2	104	0.978	$[0.726, 1.261]$	-0.013	ref.
		R3	96	1.002	$[0.679, 1.299]$	$+0.014$	-0.336

A.9 Task families and scoring rules

Task families. A task is denoted `Rung::family::variant`. Tab. 10 lists the ten question families by rung; the families expand on CLadder's collection. Within a family, the **output variant** fixes what the agent must return and therefore how the task is graded: a sign, a boolean, a set, a point estimate, an interval, or an abstention-eligible value. Effect families carry several variants of the same estimand (point, 95% interval, sign, agreement with the observed association), so one scene can test the same quantity at different levels of demand. Tabs. 12–14 enumerate every variant with its metric.

Table 10: **Question families per rung.**

Rung	Family	Description
$R1$	Prediction	Fit a predictive model for outcome Y from observed columns; write held-out point predictions or, for continuous Y, central prediction intervals.
$R1$	Association	Quantify the observational relation between treatment X and outcome Y, marginally or after conditioning on a third variable.
$R1$	Collider phenomenon	Test whether conditioning on a named collider opens a spurious association between two of its parents (explaining-away).
$R2$	Causal sketch	Recover the story-implied graph as a directed edge set or as an undirected skeleton.
$R2$	Identification	Decide whether the population ATE of X on Y is identifiable from the observed conceptual variables and by what strategy; includes adjustment-set queries.
$R2$	Effect estimation	Estimate the population average treatment effect (ATE); output point, uncertainty, sign, or agreement with the observed association.
$R2$	Bias diagnostics	Diagnose whether an adjustment strategy is biased (collider conditioning, forbidden controls).
$R3$	Counterfactual identification	Decide whether the target counterfactual is identifiable.
$R3$	Counterfactual effects	Estimate the effect of treatment on the treated (ETT), or abstain if non-identifiable.
$R3$	Mediation	Decompose the effect into natural direct (NDE) and natural indirect (NIE) components; report point, sign, or which dominates.

Scoring rules. We list the scoring rules used in the benchmark in this appendix. Tab. 11 lists the base metrics, the normalized quantity each task contributes when pooled, and the pooled category it feeds. Tabs. 12–14 then assign them to task variants. The abstain convention from Sec. 3.8 applies whenever the target estimand is non-identifiable or the model abstains (routing details below), superseding the task-specific metrics.

Interval scores and the aggregator S_{NR}. The interval score is strictly proper (Gneiting & Raftery, 2007): its expectation is minimized by the true central (equal-tailed) $(1-\alpha)$ interval, so the agent picks its own estimator and the submitted $(\hat{\ell}, \hat{u})$ are scored directly against the scalar target τ, row-wise against y_j for prediction intervals, with no reference interval computed or stored. Propriety is preserved by averaging, so interval scores are aggregated by means (as opposed to e.g., medians): over test rows within a prediction-interval task, and across tasks inside the aggregate numeric metric S_{NR} (Eq. 3). For this reason, S_{NR} splits the numeric pool T_{NR} into a point pool $T_{\mathrm{NR}}^{\mathrm{pt}}$ and an interval pool $T_{\mathrm{NR}}^{\mathrm{int}}$. As point-graded tasks emit normalized errors, a median over the point pool is appropriate as it tracks typical-scene performance without being tail-driven. Interval-graded tasks, however, carry a miss penalty (after the above averaging) $2/\alpha$ where one overconfident miss can reach NRelIS (Tab. 11) values an order of magnitude above the typical level. While a median would discard such

Table 11: **Base metrics and their pooled forms.** $\mathrm{sd}(Y_{\mathrm{test}})$ is the outcome standard deviation on the private test set, θ and τ the GT target, $(x)_+ = \max(x, 0)$, and c the cap of the interval pool. N counts task–output variant pairs; † marks the same pair in both prediction rows. Abstention reroutes existing variants.

Used for	Metric	Definition	Normalized contribution	Pool	N
continuous prediction	RMSE	$\sqrt{n^{-1}\sum_j (y_j - \hat{y}_j)^2}$ over the test rows	$\mathrm{RMSE}/(1 + \mathrm{sd}(Y_{\mathrm{test}}))$	T_{NR}	$1^\dagger$
binary prediction	ROC-AUC, Brier, log-loss	$\mathrm{Brier} = n^{-1}\sum_j (\hat{p}_j - y_j)^2$; ROC-AUC and log-loss reported alongside	$\sqrt{\mathrm{Brier}}/(1 + \mathrm{sd}(Y_{\mathrm{test}}))$	T_{NR}	$1^\dagger$
point effects, association strengths	Absolute error	$\lvert\hat{\theta} - \theta\rvert$	$\lvert\hat{\theta} - \theta\rvert/(1 + \lvert\theta\rvert)$	T_{NR}	6
prediction intervals ($\alpha = 0.1$, mean over test rows); effect intervals ($\alpha = 0.05$)	Interval score IS_α	$(\hat{u} - \hat{\ell}) + \frac{2}{\alpha}(\hat{\ell} - \tau)_+ + \frac{2}{\alpha}(\tau - \hat{u})_+$	$\mathrm{NRelIS} = \mathrm{IS}_\alpha/(1 + s)$ with $s = \mathrm{sd}(Y_{\mathrm{test}})$ or $\lvert\tau\rvert$; capped mean across tasks, $c = 10$	T_{NR}	4
edge sets, adjustment sets, forbidden controls	F_1	$2\lvert\hat{S} \cap S\rvert/(\lvert\hat{S}\rvert + \lvert S\rvert)$; 1 when both sets are empty	$1 - F_1$	Med. F_1-Loss	4
discrete answers	Exact match	$\mathbf{1}[\text{answer} = \text{GT}]$	as is	Pass Rate	18
non-identifiable target or null answer	Abstention binary	$\mathrm{AbstMatch} = 1$ if the model's abstention decision matches target non-identifiability, and 0 otherwise	as is	Pass Rate	—

signals (and abandon propriety), a raw mean lets one catastrophic miss dominate the metric. Hence we settle on the *capped mean* $\mathrm{mean}\ \min(\mathrm{NRelIS}_i, c)$ with $c = 10$ as a robustness compromise. With $\alpha = 0.05$, a miss by $0.1\,|\tau|$ incurs $\approx 4|\tau|/(1 + |\tau|)$ (well below the cap), while a miss by $|\tau|$ reaches the cap once $|\tau| \geq 1/3$. The cap bounds any single task's contribution to c, so the aggregates cannot be dominated by a single catastrophic miss.

Permissive band for sign tasks. For the `sign_only` variants we adopt a "permissive" criterion: the GT label is `0` when $|\tau| < 3\,\mathrm{sd}(Y)/\sqrt{n}$, i.e. when the effect is below three standard errors of an outcome mean estimated from n directly observed outcome values. Inside this band both `0` and the true sign are accepted as correct answers. On views where the outcome is observed directly, n is the training-row count. If the outcome is not directly measured (its values are released only through a measurement bundle), n is the number of calibration rows, the only rows carrying gold outcome values.

Abstention pool routing. Some variants exist only in compatible scenes (interval prediction requires a continuous outcome, collider tasks a collider), and the exam builder samples among the available ones. Independently of that, the grader routes a task into the abstention pool whenever *either* the GT estimand is non-identifiable *or* the model returned a null-equivalent answer. In these cases, the per-variant content metric is replaced with a single uniform abstention binary $\mathrm{AbstMatch}_i$ (Sec. 3.8). This covers every variant whose estimand can be non-identifiable: the rung-2 `identification` family, the rung-2 `effect_estimate` family, `bias_diagnostic.forbidden_controls_list`, and the rung-3 `counterfactual_identification`, `counterfactual_effect`, and `mediation_effect` variants. For families that ask for identifiability this routing changes no score as the identifiability judgment is the content there. The task, however, is assigned to the abstention pool so the same binary is counted once. For the other families it replaces the content metric, which would be meaningless on a non-identifiable estimand. Each answered task therefore belongs to exactly one pool: a wrong abstention call receives a 0 on the

Table 12: **Rung 1 scoring rules.**

Task	Output variant	Metric	Score definition
prediction	point_predictor	RMSE / AUC	RMSE on the private test set for continuous Y; ROC-AUC for binary $Y \in \{0, 1\}$.
prediction	prediction_interval	Int. score	Proper interval score on the private test set; report coverage and mean width.
association	sign_only	Exact	1 iff answer sign matches the GT sign in $\{+, -, \texttt{unknown}\}$.
association	effect_size_point	Abs. err.	$\lvert\widehat{r} - r\rvert$ where r is the GT Pearson correlation between treatment and outcome on the ground-truth data.
association	sign_before_after	Exact	1 iff both sign_before and sign_after match GT.
association	delta_point	Abs. err.	Absolute error on $\widehat{\Delta}$ where Δ is the GT change in association after conditioning.
association	delta_sign_only	Exact	1 iff answer sign matches the sign of the GT conditional-association change.
association	argmax_change	Exact	1 iff the selected conditioning variable matches the GT variable with the largest absolute association change.
collider_phenomenon	induced_association_boolean	Exact	1 iff answer association_present matches GT.
collider_phenomenon	induced_association_sign_only	Exact	1 iff answer sign matches the GT induced-association sign after conditioning on the collider.
collider_phenomenon	induced_association_strength_point	Abs. err.	Absolute error on the GT conditional association induced by conditioning on the collider.

abstention binary but does not receive a substitute content score. The R3 abstention check uses the per-estimand R3 identifiability (counterfactual_identification.⟨ETT,NDE,NIE⟩); for the dominance variant we require both NDE and NIE to be identifiable. Null-equivalent labels (none, null, unknown, n/a, na, missing field, empty string) are accepted interchangeably.

Table 13: **Rung 2 scoring rules.**

Task	Output variant	Metric	Score definition
causal_sketch	edges_only	Edge F_1	Directed edge-set F_1 against GT edges in story-name space.
causal_sketch	skeleton_edges	Skel. F_1	Undirected edge-set F_1 against the GT skeleton, ignoring direction.
identification	one_valid_adjustment_set	Exact	1 iff answer adjust equals any GT valid backdoor set; if the ATE is identifiable but no valid backdoor set exists (e.g., front-door), 1 iff the answer is the no_backdoor sentinel.
identification	method_label	Exact	1 iff answer method matches GT, with none denoting not identifiable.
identification	identifiable_boolean	Exact	1 iff answer identifiable matches the GT population-ATE identifiability.
identification	minimal_adjustment_set_size	Exact	1 iff answer k equals the size of a GT minimal valid backdoor set; if the ATE is identifiable but no valid backdoor set exists, 1 iff the answer is the no_backdoor sentinel.
identification	n_valid_adjustment_sets	Exact	1 iff answer n equals the number of GT valid backdoor sets.
identification	all_minimal_adjustment_sets	Set F_1	Set-level F_1 over predicted versus GT minimal valid backdoor sets.
effect_estimate	ate_point	Abs. err. / Exact	If identifiable, $\lvert\widehat{\tau} - \tau\rvert$ where τ is the SCM Monte Carlo GT ATE; also report relative error. If non-identifiable, 1 iff the answer abstains with ate=null.
effect_estimate	ate_uq_95	Int. score / Exact	If identifiable, the proper interval score for τ; report coverage and width. If non-identifiable, 1 iff the answer abstains with ate=null.
effect_estimate	ate_sign_only	Exact	1 iff answer sign matches the GT ATE sign in $\{+, -, 0, \texttt{unknown}\}$.
effect_estimate	ate_vs_assoc_sign_match	Exact	1 iff answer matches agrees with whether the GT ATE sign matches the GT observational-association sign.
bias_diagnostic	collider_bias_boolean	Exact	1 iff answer bias_present matches GT.
bias_diagnostic	forbidden_controls_list	Set F_1	Set-level F_1 over predicted versus GT forbidden observed controls.

A.10 Further exam results

Tables in this section further break down the results reported in the main paper (Sec. 4). Note that all slices (i.e., per-motif) are marginal views of the same exams: the axes are not varied independently, so a difference along one axis can also reflect the values of the other axes that happen to fall into that slice.

Exam composition. Tables 15, 16, and 17 compare the two exams. The realistic exam covers all ten task families, while the hard exam focuses on identification, bias diagnostics, and R2/R3 effect tasks. The hard exam has larger shares of front-door, double_nc, and grafted scenes, and more of its tasks require data; the continuous/binary outcome mix is the same in both exams. Some realistic exam motif-by-family cells contain only one or two tasks, limiting comparisons at that level. The hard exam was constructed by targeting fixed identifiability and graph structure strata. Then, the task-family weights are specified per structure and output- and observation-variant weights are specified within each task family. The observation mix is tilted toward harder measurements: 143 of the 167 data-backed hard-exam tasks use proxy_hard or the "new" proxy_severe tier. The latter combines more complex measurement links, stronger noise, and a smaller calibration subset (App. A.6).

Table 14: **Rung 3 scoring rules.**

Task	Output variant	Metric	Score definition
`counterfactual_identification`	`identifiable_boolean`	Exact	1 iff answer `identifiable` matches the GT identifiability of the target estimand (ETT, NDE, or NIE).
`counterfactual_effect`	`effect_point`	Abs. err. / Exact	If identifiable, $\lvert\widehat{\mathrm{ETT}} - \mathrm{ETT}\rvert$ using SCM GT; also report relative error. If non-identifiable, 1 iff the answer abstains with `value=null`.
`counterfactual_effect`	`effect_uq_95`	Int. score / Exact	If identifiable, the proper interval score for the ETT; report coverage and width. If non-identifiable, 1 iff the answer abstains with `value=null`.
`counterfactual_effect`	`sign_only`	Exact	If identifiable, 1 iff answer `sign` matches the GT ETT sign in $\{+, -, 0, \texttt{unknown}\}$. If non-identifiable, 1 iff `sign=unknown`.
`mediation_effect`	`effect_point`	Abs. err. / Exact	For NDE or NIE, if identifiable, absolute error on the requested SCM GT effect; also report relative error. If non-identifiable, 1 iff the answer abstains with `value=null`.
`mediation_effect`	`effect_uq_95`	Int. score / Exact	For NDE or NIE, if identifiable, the proper interval score for the requested effect; report coverage and width. If non-identifiable, 1 iff the answer abstains with `value=null`.
`mediation_effect`	`sign_only`	Exact	For NDE or NIE, if identifiable, 1 iff answer `sign` matches the GT sign in $\{+, -, 0, \texttt{unknown}\}$. If non-identifiable, 1 iff `sign=unknown`.
`mediation_effect`	`direct_vs_indirect_dominance`	Exact	If identifiable, 1 iff answer `dominant` matches whether $\lvert\mathrm{NDE}\rvert$ exceeds $\lvert\mathrm{NIE}\rvert$, whether $\lvert\mathrm{NIE}\rvert$ exceeds $\lvert\mathrm{NDE}\rvert$, or whether they are tied within tolerance. If non-identifiable, 1 iff `dominant=null`.

Table 15: **Realistic-exam composition.** Scenes by structural motif and task family. Pred: prediction; Assoc: association; Ident: identification; CF-ID: counterfactual identification; CFE: counterfactual effect. `grafted_`k denotes scenes with k auxiliary motifs anchor-grafted onto a host main graph.

Motif	Pred	Assoc	Collider	Sketch	Ident	Effect	Bias	CF-ID	CFE	Mediation	Total
arrowhead	0	0	0	1	0	0	0	1	0	0	2
chain	1	0	0	1	1	1	0	1	0	0	5
collider	0	0	0	1	0	0	1	1	0	0	3
confounding	6	1	0	3	2	3	3	4	3	0	25
diamond	0	0	0	0	0	0	0	0	0	1	1
diamondcut	0	0	0	0	2	0	0	0	0	0	2
double_nc	0	0	0	0	0	0	0	0	1	0	1
fork	1	2	0	0	1	0	0	0	0	0	4
frontdoor	1	0	0	0	0	1	0	0	0	1	3
iv	2	3	0	8	4	3	3	3	1	0	27
mediation	0	1	0	2	0	1	0	0	0	0	4
triangle	1	0	0	0	0	1	0	0	0	0	2
grafted_1	2	3	2	1	3	3	0	0	2	0	16
grafted_2	1	1	0	0	1	0	0	2	0	0	5
Total	**15**	**11**	**2**	**17**	**14**	**13**	**7**	**12**	**7**	**2**	**100**

Performance by observation variant. Table 18 breaks Pass Rate and median normalized error down by observation variant (`clean`: every conceptual variable measured directly). Across the hard exam's three proxy tiers, pass rates shift by under twenty points and in no consistent direction, whereas median error grows five- to fifteen-fold from `proxy` to `proxy_severe`, and under the severe tier the weakest models have medians many times those of the leaders. The hard-exam `clean` slice holds only symbolic binaries with a larger non-identifiable share, so its Pass Rate is not comparable to the

Table 16: **Hard-exam composition.** Scenes by structural motif and task family; columns as in Tab. 15.

Motif	Pred	Assoc	Collider	Sketch	Ident	Effect	Bias	CF-ID	CFE	Mediation	Total
arrowhead	0	0	0	0	0	1	0	0	1	4	6
chain	0	0	0	0	2	0	0	0	1	10	13
confounding	0	0	0	0	2	8	1	0	5	0	16
diamond	0	0	0	0	1	1	0	0	3	7	12
diamondcut	0	0	0	0	0	1	0	0	1	0	2
double_nc	0	0	0	0	4	5	1	0	3	0	13
frontdoor	0	0	0	0	5	13	0	0	6	6	30
iv	0	0	0	0	3	10	1	0	8	0	22
mediation	0	0	0	0	3	2	0	0	3	2	10
grafted_1	0	0	0	0	0	15	6	0	17	1	39
grafted_2	0	0	0	0	0	15	4	0	16	2	37
Total	**0**	**0**	**0**	**0**	**20**	**71**	**13**	**0**	**64**	**32**	**200**

Table 17: **Exam composition by observation variant, outcome type, and input mode.** Counts (percent). Observation-variant percentages refer to data-backed tasks alone. *Outcome type* refers to the conceptual outcome variable.

Axis	Level	Realistic ($n = 100$)	Hard ($n = 200$)
Observation variant (data-backed only)	`clean`	22 (43.1%)	0 (0%)
	`proxy`	10 (19.6%)	24 (14.4%)
	`proxy_hard`	19 (37.3%)	55 (32.9%)
	`proxy_severe`	0 (0%)	88 (52.7%)
Outcome type	Continuous	59 (59%)	118 (59%)
	Binary	41 (41%)	82 (41%)
Input mode	Symbolic	49 (49%)	33 (16.5%)
	Data-backed	51 (51%)	167 (83.5%)

proxy tiers'. On the realistic exam the small `proxy_hard` slice (note its run-to-run SDs) is where pass rates are low for several closed and open models alike. The `proxy_hard`-versus-`clean` gap survives adjustment for rung and output variant (App. A.8), and the matched-scene ablation of App. A.16 confirms the ordering within identical scenes.

Content correctness and abstention. Table 19 splits Pass Rate into the content pool and the abstention pool; both can be graded because identifiability is determined by the hidden SCM (Sec. 3.8). On the realistic exam content PR is close to saturation for most models, so the spread in Pass Rate is almost entirely abstention: GPT-6 Astra abstains almost exactly when the ground truth requires it, whereas several models that answer nearly every identifiable target correctly abstain on only about half of the non-identifiable ones. On the hard exam content PR spreads as well, and Astra's lead there comes from abstention; its content PR is below that of several open models. The two abstention errors differ across exams: on the realistic exam missed abstention, committing to a number the graph does not identify, dominates for every model; on the hard exam false abstention, refusing an identifiable target, becomes substantial for weaker and for more cautious models, and for GPT-5.5 exceeds its missed abstention. App. A.11 traces both errors to the identification step and to the motifs on which each occurs.

Per-rung and per-family aggregates (Tabs. 20, 21). On the realistic exam, R1 binary performance is near ceiling: ten of twelve models score 100%, with Qwen 3.6 35B and GPT-5.5 at 91.7%. Association shows essentially no model separation, while prediction Med. NRel. Err spans 0.491–0.671. The 17 causal-sketch tasks are also recovered almost perfectly (cf. App. A.4); set-valued losses are omitted from these tables for this reason. They are reported separately in Tab. 37.

On both exams, most R2 binary separation comes from abstention. Content PR is 100% for ten models on the realistic exam and eleven on the hard exam, whereas abstention PR ranges from 20.5–

Table 18: **Pass Rate and median NRel. Err by observation variant.** Three-run mean ± SD per model and exam. Hard / Severe: `proxy_hard` / `proxy_severe`; NRel. Err over valid numeric answers. — : empty pool (no severe variant on the realistic exam; hard-exam `clean` tasks are all binary). Rows in realistic-exam leaderboard order; best per column and exam in bold. ● closed, ▲ open-weight.

		Pass Rate (%) ↑				Med. NRel. Err ↓			
Model	Exam	Clean	Proxy	Hard	Severe	Clean	Proxy	Hard	Severe
● GPT-6 Astra	Realistic	**92.6 ± 3.2**	**100.0 ± 0.0**	**100.0 ± 0.0**	—	0.053 ± 0.006	0.101 ± 0.063	0.147 ± 0.004	—
	Hard	**100.0 ± 0.0**	**100.0 ± 0.0**	**91.3 ± 4.3**	82.9 ± 2.7	—	0.060 ± 0.017	0.089 ± 0.011	0.331 ± 0.063
▲ GLM 5.3	Realistic	74.1 ± 3.2	76.9 ± 0.0	50.0 ± 16.7	—	0.041 ± 0.006	0.302 ± 0.001	**0.092 ± 0.026**	—
	Hard	77.5 ± 2.2	62.6 ± 14.1	72.4 ± 4.0	73.2 ± 2.7	—	**0.025 ± 0.000**	**0.060 ± 0.008**	0.297 ± 0.085
● Gemini 3.1 Pro	Realistic	83.3 ± 5.6	82.1 ± 4.4	38.9 ± 9.6	—	0.068 ± 0.047	0.217 ± 0.146	0.168 ± 0.067	—
	Hard	82.6 ± 4.3	50.0 ± 0.0	55.1 ± 5.0	62.8 ± 5.4	—	0.059 ± 0.027	0.183 ± 0.003	0.429 ± 0.160
▲ DeepSeek V4.1 Flash	Realistic	75.9 ± 3.2	69.2 ± 7.7	77.8 ± 19.2	—	0.061 ± 0.038	0.288 ± 0.022	0.175 ± 0.027	—
	Hard	84.1 ± 9.1	80.1 ± 7.4	83.1 ± 0.4	85.2 ± 2.6	—	0.042 ± 0.013	0.093 ± 0.028	0.282 ± 0.041
● Claude Fable 5.1	Realistic	77.8 ± 0.0	59.0 ± 4.4	58.3 ± 14.4	—	**0.027 ± 0.007**	0.380 ± 0.001	0.114 ± 0.010	—
	Hard	82.6 ± 4.3	87.5 ± 0.0	85.9 ± 2.3	**86.2 ± 2.1**	—	0.032 ± 0.001	0.066 ± 0.011	**0.212 ± 0.055**
▲ Kimi K3	Realistic	64.8 ± 6.4	69.2 ± 0.0	66.7 ± 33.3	—	0.042 ± 0.010	0.301 ± 0.000	0.146 ± 0.017	—
	Hard	62.8 ± 3.3	62.5 ± 0.0	63.8 ± 5.0	65.0 ± 6.8	—	0.048 ± 0.007	0.147 ± 0.044	0.334 ± 0.024
▲ K2-MoVA 36B-A4B	Realistic	74.7 ± 7.7	66.7 ± 4.4	80.6 ± 17.3	—	0.122 ± 0.043	0.301 ± 0.000	0.180 ± 0.070	—
	Hard	68.6 ± 4.7	70.0 ± 0.0	68.9 ± 4.4	75.2 ± 5.4	—	0.037 ± 0.006	0.112 ± 0.012	0.307 ± 0.005
● Claude Opus 4.8	Realistic	88.9 ± 0.0	79.5 ± 4.4	44.4 ± 19.2	—	0.135 ± 0.014	**0.099 ± 0.000**	0.216 ± 0.040	—
	Hard	82.6 ± 4.3	68.1 ± 9.0	61.1 ± 5.0	58.5 ± 3.0	—	0.128 ± 0.082	0.196 ± 0.036	0.709 ± 0.312
▲ Kimi K2.6	Realistic	78.5 ± 13.5	76.9 ± 0.0	66.7 ± 33.3	—	0.108 ± 0.100	0.258 ± 0.075	0.180 ± 0.018	—
	Hard	76.8 ± 2.5	83.8 ± 7.6	72.4 ± 8.8	81.6 ± 2.5	—	0.069 ± 0.048	0.215 ± 0.052	0.479 ± 0.142
▲ Qwen 3.6 35B	Realistic	69.2 ± 5.2	79.5 ± 4.4	16.7 ± 16.7	—	0.220 ± 0.023	0.164 ± 0.120	0.186 ± 0.080	—
	Hard	67.7 ± 7.5	69.2 ± 15.9	53.1 ± 2.8	52.9 ± 4.4	—	0.233 ± 0.025	0.248 ± 0.054	1.253 ± 0.300
● GPT-5.5	Realistic	85.5 ± 3.0	77.3 ± 8.2	72.2 ± 25.5	—	0.182 ± 0.040	0.215 ± 0.100	0.216 ± 0.051	—
	Hard	87.2 ± 11.4	80.0 ± 0.0	62.7 ± 1.8	62.0 ± 2.4	—	0.117 ± 0.064	1.226 ± 0.846	1.606 ± 0.598
▲ Gemma 4 26B	Realistic	51.9 ± 8.5	76.9 ± 7.7	27.8 ± 25.5	—	0.213 ± 0.014	0.113 ± 0.023	0.333 ± 0.021	—
	Hard	50.0 ± 2.2	57.4 ± 8.5	46.4 ± 6.2	45.0 ± 3.5	—	0.215 ± 0.058	0.947 ± 0.475	3.344 ± 1.268

Table 19: **Pass Rate decomposition.** Three-run mean ± SD (PR: mean only); parentheses: per-run pool size or range. The abstention pool holds non-identifiable targets and model abstentions (App. A.9); its PR, missed abstention (answered, or answered invalidly, where abstention was required), and false abstention (abstained on an identifiable target) sum to 100%. Rows sorted by mean PR over the two exams; best per column and exam in bold. ● closed, ▲ open-weight.

		Pass rates (%) ↑			Abstention errors (%) ↓	
Model	Exam	PR	Content PR (N)	Abstention PR (N)	Missed abstention	False abstention
● GPT-6 Astra	Realistic	**96.1**	96.3 ± 3.2 (18)	**95.8 ± 3.6** (16)	**4.2 ± 3.6**	**0.0 ± 0.0**
	Hard	**90.4**	77.9 ± 0.3 (40–41)	**99.4 ± 1.0** (56–57)	**0.0 ± 0.0**	**0.6 ± 1.0**
▲ DeepSeek V4.1 Flash	Realistic	73.5	98.0 ± 3.4 (17–18)	48.0 ± 1.7 (16–17)	48.0 ± 1.7	3.9 ± 3.4
	Hard	83.9	85.1 ± 2.5 (40–41)	83.0 ± 2.8 (60–62)	8.8 ± 4.3	8.2 ± 1.5
▲ Kimi K2.6	Realistic	76.8	98.0 ± 3.4 (17–18)	54.6 ± 15.0 (16–18)	41.7 ± 9.5	3.7 ± 6.4
	Hard	78.5	85.0 ± 4.3 (40)	74.2 ± 3.4 (60–61)	18.1 ± 3.3	7.7 ± 0.9
● Claude Fable 5.1	Realistic	68.6	82.5 ± 3.0 (19)	52.1 ± 3.6 (16)	47.9 ± 3.6	**0.0 ± 0.0**
	Hard	85.4	87.1 ± 3.5 (41–42)	84.1 ± 1.8 (56–57)	14.7 ± 2.1	1.2 ± 1.0
● GPT-5.5	Realistic	81.0	88.9 ± 5.6 (18)	72.5 ± 6.9 (16–18)	21.8 ± 9.6	5.7 ± 5.6
	Hard	69.0	63.5 ± 4.9 (42–47)	72.7 ± 2.6 (66–70)	10.5 ± 4.7	16.8 ± 2.8
● Claude Opus 4.8	Realistic	81.4	**100.0 ± 0.0** (18)	60.4 ± 3.6 (16)	39.6 ± 3.6	**0.0 ± 0.0**
	Hard	65.2	78.5 ± 1.3 (40–41)	56.8 ± 1.8 (62–67)	30.8 ± 2.8	12.4 ± 3.6
▲ GLM 5.3	Realistic	72.8	96.4 ± 3.1 (18–19)	45.8 ± 3.6 (16)	54.2 ± 3.6	**0.0 ± 0.0**
	Hard	73.0	81.8 ± 5.9 (40–44)	67.2 ± 1.9 (62–64)	21.7 ± 0.6	11.1 ± 1.4
▲ K2-MoVA 36B-A4B	Realistic	72.1	98.0 ± 3.4 (16–17)	47.9 ± 12.9 (17–19)	41.1 ± 16.0	10.9 ± 5.0
	Hard	71.7	85.7 ± 4.0 (39–40)	63.4 ± 3.5 (66–69)	19.8 ± 1.9	16.8 ± 1.9
● Gemini 3.1 Pro	Realistic	78.7	96.3 ± 3.2 (18)	59.2 ± 3.1 (16–17)	38.8 ± 4.4	2.0 ± 3.4
	Hard	64.5	**87.7 ± 0.2** (40–41)	48.3 ± 3.7 (57–59)	48.3 ± 2.3	3.4 ± 1.7
▲ Kimi K3	Realistic	66.7	96.3 ± 6.4 (17–18)	34.4 ± 11.4 (16–17)	63.6 ± 14.7	2.0 ± 3.4
	Hard	64.0	86.1 ± 1.5 (40–41)	48.6 ± 2.2 (57–60)	47.5 ± 2.5	4.0 ± 2.5
▲ Qwen 3.6 35B	Realistic	66.5	89.8 ± 7.7 (17–21)	41.2 ± 5.9 (17)	52.9 ± 5.9	5.9 ± 0.0
	Hard	57.6	76.6 ± 3.5 (37–40)	46.9 ± 5.1 (67–71)	34.3 ± 4.2	18.8 ± 2.4
▲ Gemma 4 26B	Realistic	59.2	88.5 ± 0.4 (17–18)	29.1 ± 9.0 (16–18)	65.2 ± 14.0	5.7 ± 5.6
	Hard	47.4	69.5 ± 6.5 (39–40)	35.2 ± 3.3 (66–73)	44.6 ± 2.7	20.2 ± 4.4

100% and 34.2–100%, respectively. GPT-6 Astra is the only model perfect across all R2 abstention families on both exams. Other models vary substantially by family: on the realistic exam, Claude Opus 4.8 reaches 86.7% abstention PR for identification but 11.1% for both effect estimation and bias

Table 20: **Per-rung and per-family aggregates on the realistic exam.** Three-run means; row blocks are `rung:family` slices with the task count n; each rung block opens with the pooled per-rung slice. Family abbreviations as in Tab. 15. — marks an empty pool, bold marks the best model per row. Model markers: ● closed, ▲ open-weight.

Family (n)	Metric	● GPT-6	▲ GLM	● Gemini	▲ DeepSeek	● Fable	▲ K3	▲ K2-MoVA	● Opus	▲ K2.6	▲ Qwen	● GPT-5.5	▲ Gemma
R1:all (28)	PR (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	91.7	91.7	**100.0**
	Cont. PR (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	91.7	91.7	**100.0**
	Med. NRel	0.272	0.281	0.283	**0.271**	0.272	0.291	0.281	0.318	0.273	0.283	0.291	0.348
R1:Pred (15)	Med. NRel	0.530	0.530	0.530	0.530	**0.491**	0.530	0.531	0.531	0.532	0.546	0.671	0.538
R1:Assoc (11)	PR (%)	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	Cont. PR (%)	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	Med. NRel	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
R1:Collider (2)	PR (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	83.3	83.3	**100.0**
	Cont. PR (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	83.3	83.3	**100.0**
	Med. NRel	0.013	0.009	0.010	0.009	**0.004**	0.014	0.013	0.161	0.112	—	—	0.164
R2:all (51)	PR (%)	**100.0**	45.2	62.7	47.6	54.8	35.7	56.7	57.1	54.8	47.8	88.1	37.1
	Cont. PR (%)	**100.0**	**100.0**	**100.0**	88.9	**100.0**	88.9	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
	Abst. PR (%)	**100.0**	30.3	52.8	36.4	42.4	21.2	45.5	45.5	42.4	34.3	84.8	20.5
	Med. NRel	**0.059**	0.071	0.127	0.092	0.094	0.084	0.181	0.161	0.146	0.380	1.069	2.101
R2:Ident (14)	PR (%)	**100.0**	47.6	90.5	47.6	57.1	38.1	54.8	90.5	61.9	71.4	90.5	38.1
	Cont. PR (%)	**100.0**	**100.0**	**100.0**	83.3	**100.0**	83.3	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
	Abst. PR (%)	**100.0**	26.7	86.7	33.3	40.0	20.0	37.8	86.7	46.7	60.0	86.7	13.3
R2:Effect (13)	PR (%)	**100.0**	55.6	8.3	77.8	44.4	44.4	80.6	11.1	55.6	8.3	88.9	30.6
	Abst. PR (%)	**100.0**	55.6	8.3	77.8	44.4	44.4	80.6	11.1	55.6	8.3	88.9	30.6
	Med. NRel	**0.059**	0.071	0.127	0.092	0.094	0.084	0.181	0.161	0.146	0.380	1.069	2.101
R2:Bias (7)	PR (%)	**100.0**	33.3	58.3	25.0	58.3	25.0	41.7	33.3	41.7	41.7	83.3	41.7
	Cont. PR (%)	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	Abst. PR (%)	**100.0**	11.1	44.4	0.0	44.4	0.0	22.2	11.1	22.2	22.2	77.8	22.2
R3:all (21)	PR (%)	92.2	90.4	88.2	90.2	74.1	86.3	80.4	**98.0**	90.6	77.5	73.5	70.6
	Cont. PR (%)	94.4	94.7	94.4	**100.0**	74.4	97.2	97.0	**100.0**	97.0	87.0	85.6	82.3
	Abst. PR (%)	86.7	80.0	73.3	71.1	73.3	63.3	52.4	**93.3**	79.0	55.6	49.0	46.7
	Med. NRel	0.033	**0.024**	0.033	0.042	0.027	0.047	0.094	0.026	0.049	0.085	0.037	0.091
R3:CF-ID (12)	PR (%)	94.4	94.4	94.4	91.7	72.2	88.9	83.3	**97.2**	94.4	91.7	88.9	77.8
	Cont. PR (%)	**100.0**	**100.0**	96.3	**100.0**	74.1	**100.0**	96.3	**100.0**	**100.0**	**100.0**	**100.0**	88.4
	Abst. PR (%)	77.8	77.8	**88.9**	66.7	66.7	61.1	50.0	**88.9**	77.8	66.7	61.1	50.0
R3:CFE (7)	PR (%)	83.3	83.3	66.7	**100.0**	91.7	75.0	91.7	**100.0**	86.7	46.7	41.7	50.0
	Cont. PR (%)	66.7	83.3	83.3	**100.0**	**100.0**	83.3	**100.0**	**100.0**	83.3	44.4	44.4	50.0
	Abst. PR (%)	**100.0**	83.3	50.0	**100.0**	83.3	66.7	83.3	**100.0**	88.9	50.0	44.4	50.0
	Med. NRel	0.046	**0.015**	0.025	0.055	0.027	0.023	0.097	0.040	0.058	0.101	0.069	0.115
R3:Mediation (2)	PR (%)	**100.0**	83.3	**100.0**	33.3	50.0	**100.0**	0.0	**100.0**	66.7	50.0	33.3	66.7
	Cont. PR (%)	**100.0**	83.3	**100.0**	—	50.0	**100.0**	—	**100.0**	—	—	—	—
	Abst. PR (%)	—	—	—	—	—	—	0.0	—	—	—	—	—
	Med. NRel	**0.004**	—	0.040	0.019	—	0.089	0.090	0.011	0.014	—	0.015	0.012

diagnostics. Numeric effect-estimation accuracy also separates models: GPT-6 Astra has the lowest Med. NRel. Err on the realistic exam (0.059), and GLM 5.3 on the hard exam (0.105); GPT-5.5 and Gemma 4 26B have the largest errors on both exams. Thus GPT-5.5's relatively strong R2 abstention performance does not extend to estimation accuracy, which is related to its near-one-shot strategy as discussed in Sec. 4.

R3 content performance varies more than R2 on both exams: content PR starts to become a genuine challenge, spanning 74.4–100% on the realistic exam and 54.9–84.2% on the hard exam. Claude Opus 4.8 leads R3 abstention on the realistic exam (93.3%), while GPT-6 Astra again leads on the hard exam (98.4%). Every model has higher R3 median numeric error on the hard exam, with Med. NRel. Err ranging significantly from 0.121–0.968, compared with 0.024–0.094 on the realistic exam. Within Hard, every model has lower content PR for counterfactual effects than for mediation effects. The latter task family grows from just two Realistic tasks to 32 Hard tasks.

Per-motif aggregates (Tab. 22). The two motifs that are non-identifiable by construction, `iv` and `double_nc` (App. A.7), rank models mainly on the abstention call (solely for the hard exam). The

Table 21: **Per-rung and per-family aggregates on the hard exam.** Three-run means. Notation same as in Tab. 20.

Family (n)	Metric	▲ GLM	● Fable	▲ DeepSeek	● GPT-6	▲ K2.6	● Gemini	▲ K3	▲ K2-MoVA	● Opus	▲ Qwen	● GPT-5.5	▲ Gemma
R2:all (104)	PR (%)	71.9	84.4	85.8	**100.0**	74.8	59.6	51.8	68.6	62.9	49.7	83.0	41.3
	Cont. PR (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	80.1
	Abst. PR (%)	65.5	80.6	82.4	**100.0**	69.0	49.6	40.0	61.8	54.4	40.2	79.5	34.2
	Med. NRel	0.121	**0.105**	0.144	0.183	0.267	0.271	0.294	0.157	0.256	0.414	0.762	1.322
R2:Ident (20)	PR (%)	86.7	88.2	92.2	**100.0**	88.2	90.2	78.8	84.6	94.1	76.8	92.5	57.6
	Cont. PR (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	80.1
	Abst. PR (%)	74.1	75.0	83.3	**100.0**	75.0	79.2	55.6	69.4	87.5	59.1	86.3	37.4
R2:Effect (71)	PR (%)	66.1	86.4	87.4	**100.0**	72.9	35.9	40.3	68.9	45.2	33.0	79.3	34.7
	Abst. PR (%)	66.1	86.4	87.4	**100.0**	72.9	35.9	40.3	68.9	45.2	33.0	79.3	34.7
	Med. NRel	0.121	**0.105**	0.144	0.183	0.267	0.271	0.294	0.157	0.256	0.414	0.762	1.322
R2:Bias (13)	PR (%)	50.0	66.7	61.1	**100.0**	44.4	61.1	16.7	22.2	50.0	38.9	72.2	27.8
	Abst. PR (%)	50.0	66.7	61.1	**100.0**	44.4	61.1	16.7	22.2	50.0	38.9	72.2	27.8
R3:all (96)	PR (%)	73.8	**86.2**	82.6	82.1	81.8	68.8	74.4	74.3	67.2	64.8	58.0	53.2
	Cont. PR (%)	77.0	83.6	80.8	71.6	80.6	**84.2**	82.1	81.7	72.3	70.6	54.9	66.7
	Abst. PR (%)	69.9	90.3	85.1	**98.4**	83.3	46.2	63.2	66.0	60.7	58.0	62.9	36.9
	Med. NRel	**0.121**	0.151	0.151	0.141	0.200	0.193	0.173	0.175	0.300	0.525	0.809	0.968
R3:CFE (64)	PR (%)	75.5	**85.5**	85.1	81.3	81.0	63.5	70.6	77.2	67.2	62.5	56.3	46.6
	Cont. PR (%)	70.6	78.2	78.7	64.4	78.3	**80.6**	77.4	79.6	67.2	64.3	43.5	56.7
	Abst. PR (%)	80.4	93.3	91.0	**98.4**	83.3	46.9	64.1	75.3	67.0	61.1	68.9	38.3
	Med. NRel	**0.125**	0.139	0.135	0.151	0.182	0.170	0.189	0.206	0.404	0.378	0.897	0.999
R3:Mediation (32)	PR (%)	68.9	**88.9**	74.8	84.8	84.8	88.4	88.4	66.4	67.6	73.1	63.7	77.9
	Cont. PR (%)	88.9	**93.9**	84.8	84.8	84.8	90.9	90.9	85.6	81.8	81.8	76.5	84.8
	Abst. PR (%)	0.0	—	—	—	—	—	—	0.0	0.0	0.0	0.0	—
	Med. NRel	**0.109**	0.170	0.161	0.119	0.220	0.284	0.171	0.180	0.215	1.233	0.939	1.073

Table 22: **Per-motif aggregates.** Three-run means. Count in brackets. Grafted scenes split by graft count. Exam slices with fewer than 3 tasks not presented. Avg is the mean over the models with a value; bold marks the best model per row. Model markers: ● closed, ▲ open-weight.

Motif	Exam (n)	Metric	● GPT-6	▲ GLM	● Gemini	▲ DeepSeek	● Fable	▲ K3	▲ K2-MoVA	● Opus	▲ K2.6	▲ Qwen	● GPT-5.5	▲ Gemma	Avg
grafted_1	Realistic (16)	PR (%)	86.7	93.3	93.3	93.3	93.3	80.0	86.7	**100.0**	93.3	75.6	75.6	60.0	85.9
		Med. NRel	0.169	0.179	**0.139**	0.227	0.333	0.186	0.177	0.206	0.364	0.299	0.217	0.252	0.229
	Hard (39)	PR (%)	79.2	56.2	53.2	**81.2**	65.3	36.8	64.6	43.8	72.2	52.1	46.9	31.2	56.9
		Med. NRel	0.171	**0.108**	0.168	0.135	0.144	0.187	0.226	0.497	0.252	0.559	1.400	1.399	0.437
iv	Realistic (27)	PR (%)	**95.2**	40.5	54.8	45.2	47.6	31.0	52.4	54.8	50.0	40.5	76.2	33.3	51.8
		Med. NRel	0.035	**0.026**	0.043	0.036	0.028	0.044	0.052	0.052	0.052	0.043	0.052	0.052	0.043
	Hard (22)	PR (%)	**100.0**	66.7	30.3	83.3	77.3	39.4	71.2	56.1	74.2	57.6	81.8	37.9	64.6
grafted_2	Realistic (5)	PR (%)	**100.0**	**100.0**	83.3	**100.0**	50.0	**100.0**	72.2	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	92.1
		Med. NRel	0.246	0.247	0.247	0.246	0.246	0.247	0.246	0.247	0.246	**0.179**	0.247	0.250	0.241
	Hard (37)	PR (%)	84.4	68.1	51.1	**86.7**	**86.7**	51.1	73.3	57.8	65.1	55.6	75.0	42.2	66.4
		Med. NRel	0.142	0.109	0.138	0.083	**0.082**	0.155	0.087	0.180	0.105	0.224	0.337	0.433	0.173
confounding	Realistic (25)	PR (%)	**100.0**	**100.0**	**100.0**	**100.0**	83.3	94.4	94.4	**100.0**	95.2	95.2	85.7	88.9	94.8
		Med. NRel	0.230	**0.221**	0.234	0.230	0.241	0.237	0.245	0.260	0.254	0.257	0.301	0.343	0.254
	Hard (16)	PR (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	94.4	73.3	88.9	86.7	95.3
		Med. NRel	0.148	0.154	0.219	0.125	**0.108**	0.223	0.152	0.171	0.854	0.409	0.576	1.647	0.399
frontdoor	Realistic (3)	PR (%)	**100.0**	**100.0**	83.3	33.3	**100.0**	**100.0**	0.0	**100.0**	66.7	33.3	33.3	66.7	68.1
		Med. NRel	**0.224**	0.225	0.288	0.225	0.225	0.225	0.288	0.292	0.225	0.361	0.232	0.348	0.263
	Hard (30)	PR (%)	**86.7**	31.2	61.1	35.7	83.3	55.6	20.5	27.4	55.6	12.7	16.1	4.2	40.8
		Med. NRel	0.163	**0.121**	0.544	0.254	**0.121**	0.284	0.327	0.311	0.346	1.472	0.548	2.025	0.543
chain	Realistic (5)	PR (%)	**100.0**	**100.0**	**100.0**	**100.0**	66.7	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	97.2
		Med. NRel	0.293	0.322	0.384	0.312	**0.094**	0.337	0.304	0.332	0.322	0.390	0.302	0.296	0.307
	Hard (13)	PR (%)	91.7	92.6	95.8	91.7	**100.0**	**100.0**	92.6	87.5	91.7	87.5	92.6	95.8	93.3
		Med. NRel	**1.042**	1.490	2.777	1.809	4.347	4.455	6.179	7.182	5.714	6.964	6.765	7.260	4.665
mediation	Realistic (4)	Med. NRel	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.008
	Hard (10)	PR (%)	**100.0**	**100.0**	91.7	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	91.7	88.9	97.7
		Med. NRel	0.217	0.337	1.171	0.275	**0.149**	0.445	1.128	0.670	**0.149**	3.589	0.983	1.637	0.896
double_nc	Hard (13)	PR (%)	**100.0**	**100.0**	97.4	**100.0**	**100.0**	94.9	92.3	97.4	97.4	61.5	94.9	64.1	91.7
diamond	Hard (12)	PR (%)	77.8	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	90.5	**100.0**	90.5	**100.0**	96.6
		Med. NRel	0.101	**0.049**	0.177	0.159	0.083	0.113	0.214	0.238	0.172	0.183	0.218	0.160	0.156
arrowhead	Hard (6)	PR (%)	66.7	**77.8**	**77.8**	66.7	**77.8**	66.7	66.7	66.7	66.7	66.7	50.0	52.8	66.9
		Med. NRel	0.696	**0.166**	1.057	0.239	0.529	1.888	1.324	0.640	0.183	0.766	2.802	0.828	0.927
fork	Realistic (4)	PR (%)	**100.0**	**100.0**	**100.0**	66.7	**100.0**	66.7	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	94.4
		Med. NRel	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
collider	Realistic (3)	PR (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	66.7	97.2

two structures are not equally hard: on `double_nc` ten of the twelve models abstain correctly on more than 92% of the tasks (the smaller models — Qwen 3.6 35B and Gemma 4 26B — only achieve 62–64%), whereas the `iv` pass rates spread from 31.0% to 95.2% on the realistic exam and from 30.3% to 100.0% on the hard exam, with GPT-6 Astra alone at the top both times. Often, the instrument is what tempts a model into an identification it cannot support, the over-claim traced on the running example in App. A.11.2. `confounding` is the mirror image: its queries are identifiable, pass rates are saturated on both exams (averages 94.8% and 95.3%), and the numeric error column separates the models instead, tightly on the realistic exam (0.221–0.343) and widely on the hard exam, where five (more recent) models stay at or below 0.154 while Qwen 3.6 35B, GPT-5.5, Kimi K2.6, and Gemma 4 26B sit at 0.41–1.65. `frontdoor`, whose effect is identifiable by construction, is the hard exam's discriminator: over its 30 tasks the average pass rate is only 40.8%, from 86.7% (GPT-6 Astra) and 83.3% (Claude Fable 5.1) down to 4.2% (Gemma 4 26B), the over-refusal on identifiable front-door targets that the error taxonomy (App. A.11) and the matched-scene ablation (App. A.16) sections both isolate. The models that do answer also estimate well (0.121 for GLM 5.3 and Claude Fable 5.1).

Grafting is benign on the realistic exam (average pass rates 85.9% and 92.1% for one and two grafts, errors within 0.14–0.36), but on the hard exam the grafted scenes (76 of the 200 tasks) are where the field spreads: 31.2–81.2% on `grafted_1` and 42.2–86.7% on `grafted_2`, with the `grafted_1` errors of Qwen 3.6 35B, GPT-5.5, and Gemma 4 26B at 0.56–1.40 against 0.11–0.50 for the rest. The reason for this is that graft count, task family, and observation variant co-vary across these slices; for the hard exam, grafted motifs cover about half of counterfactual-effect and most of the bias-diagnostic tasks (Tab. 16). The simple identifiable structures form a floor on both exams: `chain`, `fork`, and `collider` on the realistic exam and `mediation` and `diamond` on the hard exam are passed by all but one or two models per row, and `diamond` is estimated tightly (0.049–0.238). The exception is the hard-exam `chain` block, where pass rates stay above 87% but the median error is the largest in the table (1.04–7.26) as ten of its thirteen tasks are mediation effects (R3:Mediation in Tab. 21 spans 0.109–1.233). Slices with fewer than three tasks are not included in the table: `arrowhead`, `diamond`, `diamondcut`, `double_nc`, and `triangle` on the realistic exam and `diamondcut` on the hard exam.

Per-axis Composite Rank (Tab. 23). Each cell of Tab. 23 is the leaderboards' Composite Rank recomputed on the tasks of that slice only. The three CausalDSScore grading pools (Pass Rate, S_{NR}, Med. F_1-Loss) are first averaged over the three runs; then the twelve models are ranked on each pool, the ranks are scaled to $[0, 1]$, and the cell is the mean over the grading pools the slice contains. Each column is therefore a ranking within its slice. On the realistic exam the columns split between models: GPT-6 Astra takes R2, `clean`, and `proxy` along with the overall rank, Claude Fable 5.1 takes R1, Claude Opus 4.8 R3 (0.045), and K2-MoVA 36B-A4B `proxy_hard`, the one column an open-weight model wins there. The hard exam concentrates the lead: Claude Fable 5.1 is best overall (0.227) and takes R3 and `proxy_severe` (both 0.045), GPT-6 Astra keeps the three lighter observation columns, and DeepSeek V4.1 Flash takes R2 (0.227), with GLM 5.3 and DeepSeek V4.1 Flash tied for third overall (0.288). The rank inversions between the exams are large: Claude Fable 5.1's move is explained by its refusals on the realistic exam; Claude Opus 4.8 moves from 0.409 to 0.621 and from the best R3 rank to 0.727, GPT-5.5 from 0.530 to 0.652 with an R3 rank of 0.955, and Gemini 3.1 Pro from 0.409 to 0.530, while the two open-weight leaders rise from 0.348 and 0.379 to 0.288. GPT-6 Astra's `clean` lead on both exams matches its perfect identification abstention (Tab. 21). With F_1-Loss saturated for every model, the pools that move a slice rank are Pass Rate and the interval-sensitive S_{NR}. GPT-5.5's realistic R1 rank of 0.932 combines the slice's lowest pass rate, shared with Qwen 3.6 35B, with the largest prediction error (Tab. 20); its hard-exam ranks of 0.818 on both proxy columns and 0.955 on R3, and Gemma 4 26B's 0.955–1.000 on the same columns, follow from poorly-covering intervals in the field (Fig. 12; App. A.11.4), while Tab. 18 shows the pooled errors behind the observation columns.

What the breakdowns add. Detailed analyses further elucidate the models' performance along the five main axes CausalDS tests. Symbolic causal reasoning is often saturated: graph recovery, association, and the R1 binaries on the realistic exam are passed by nearly every model. Data-science execution separates the field progressively under hard measurement: R2 content is perfect for ten or more models on each exam, while the hard exam's errors spread by an order of magnitude on, e.g., `confounding` motifs, with the five most recent models (GPT-6 Astra, GLM 5.3, DeepSeek

Table 23: **Composite Rank by rung and observation variant.** Realistic and hard exam; 0 is best. *Overall* is the whole exam, the Comp. Rank of Tabs. 1 and 2; — marks a slice absent from the exam. Rows follow realistic-exam Overall (ties by CausalDSScore); best per column and exam in bold. Model markers: ● closed, ▲ open-weight.

		Rung			Observation variant				
Model	Exam	R1	R2	R3	`clean`	`proxy`	`proxy_hard`	`proxy_severe`	Overall
● GPT-6 Astra	Realistic	0.295	**0.288**	0.227	**0.258**	**0.167**	0.348	—	**0.288**
	Hard	—	0.258	0.182	**0.250**	**0.091**	**0.045**	0.273	0.258
▲ GLM 5.3	Realistic	0.250	0.439	0.136	0.439	0.621	0.379	—	0.348
	Hard	—	0.318	0.273	0.523	0.500	0.182	0.227	0.288
▲ DeepSeek V4.1 Flash	Realistic	0.386	0.470	0.455	0.409	0.576	0.318	—	0.379
	Hard	—	**0.227**	0.182	0.341	0.136	0.227	0.136	0.288
● Gemini 3.1 Pro	Realistic	0.341	0.409	0.364	0.409	0.379	0.561	—	0.409
	Hard	—	0.561	0.500	0.432	0.727	0.591	0.545	0.530
● Claude Opus 4.8	Realistic	0.614	0.500	**0.045**	0.439	0.273	0.621	—	0.409
	Hard	—	0.621	0.727	0.432	0.682	0.682	0.773	0.621
▲ Kimi K3	Realistic	0.432	0.530	0.591	0.470	0.667	0.364	—	0.470
	Hard	—	0.561	0.409	0.705	0.682	0.545	0.409	0.591
● Claude Fable 5.1	Realistic	**0.205**	0.485	0.500	0.439	0.803	0.500	—	0.500
	Hard	—	0.288	**0.045**	0.432	0.227	0.136	**0.045**	**0.227**
● GPT-5.5	Realistic	0.932	0.500	0.682	0.500	0.379	0.561	—	0.530
	Hard	—	0.561	0.955	0.295	0.636	0.818	0.818	0.652
▲ K2-MoVA 36B-A4B	Realistic	0.477	0.379	0.818	0.591	0.803	**0.227**	—	0.561
	Hard	—	0.561	0.545	0.614	0.545	0.455	0.500	0.530
▲ Kimi K2.6	Realistic	0.705	0.545	0.455	0.470	0.561	0.576	—	0.561
	Hard	—	0.470	0.409	0.568	0.136	0.500	0.409	0.439
▲ Qwen 3.6 35B	Realistic	0.795	0.652	0.773	0.773	0.364	0.742	—	0.712
	Hard	—	0.742	0.818	0.659	0.773	0.864	0.864	0.742
▲ Gemma 4 26B	Realistic	0.568	0.803	0.955	0.803	0.409	0.803	—	0.833
	Hard	—	0.833	0.955	0.750	0.864	0.955	1.000	0.833

V4.1 Flash, Claude Fable 5.1, K2-MoVA 36B-A4B) staying accurate where GPT-5.5, Qwen 3.6 35B, Kimi K2.6, and Gemma 4 26B blow up. Uncertainty quantification enters these tables only through S_{NR}, and it is what sinks the noisy observation and R3 ranks of GPT-5.5 and Gemma 4 26B (Fig. 12; App. A.11.4). Epistemic abstention is decisive for ranking the field on both exams: abstention pass rates span 20–100% whereas binary content can be saturated. Only GPT-6 Astra makes the call reliably across families and exams. Tool use and coding leave no column here, but the near-one-shot closed models, GPT-5.5 and Claude Opus 4.8, sit with Qwen 3.6 35B and Gemma 4 26B at the bottom of the `proxy_severe` error column (Tab. 18; App. A.14), consistent with the observation model scaling data-science difficulty independently of the other axes. A caveat nonetheless stands: as every slice is a marginal view of exams whose axes co-vary, the causal reading of any single axis rests on the matched-scene ablation (App. A.16) and the error taxonomy (App. A.11), and cells with a handful of tasks support no reliable conclusion.

A.11 Detailed error taxonomy

The leaderboard (Sec. 4) records whether an agent passed and how far its numbers were off. It does not say which part of an integrated item failed. Because every scene is generated from a known SCM, each task has an exact answer — a graph, an identifiability verdict, an effect value — and abstention is graded as a first-class answer (Sec. 3.8); a miss can therefore be attributed to the decision that produced it rather than to a disagreement about the reference. This section does that per failed task, for all twelve models and all three replications of both exams.

Input mode. The coarsest cut is whether a task needs the data at all. `causal_sketch`, `identification`, `counterfactual_identification`, and the forbidden-controls `bias_diagnostic` variant are solvable from the story-implied graph (*symbolic*); every other family needs the released files (*data-backed*). The two Pass Rate columns of Tab. 24 do not move together on either exam. GPT-6 Astra leads both. On the realistic exam DeepSeek V4.1 Flash and

K2-MoVA rank in the bottom half on symbolic tasks and in the top half on data-backed ones, while GPT-5.5, Gemini 3.1 Pro, and Qwen 3.6 35B are the reverse; on the hard exam GPT-5.5, Gemini, and Claude Opus 4.8 keep symbolic Pass Rates near the top with data-backed Pass Rates among the lowest. Structure recovery is saturated (F_1-Loss 0 for every model; the per-family view is in App. A.10), so the symbolic column measures identification alone.

Table 24: **Performance by task input mode.** *Symbolic* tasks (`causal_sketch`, `identification`, `counterfactual_identification`, forbidden-controls `bias_diagnostic`) are solvable from the story-implied graph; *data-backed* tasks need the released files. Pass Rate pools the content and abstention binaries of the slice; NRel. Err is over valid numeric answers. Three-run mean ± SD, with the per-run pool size in parentheses. Median F_1-loss on the symbolic set/graph tasks is 0.000 for every model on both exams and is omitted. Rows follow the realistic-exam leaderboard; bold marks each exam's best per column. Model markers: ● closed, ▲ open-weight.

		Symbolic	Data-backed	
Model	Exam	Pass Rate (%) ↑	Pass Rate (%) ↑	Med. NRel. Err ↓
● GPT-6 Astra	Realistic	**97.0 ± 2.6** (22)	**94.4 ± 4.8** (12)	0.084 ± 0.041 (39)
	Hard	**100.0 ± 0.0** (23)	**87.4 ± 0.8** (74)	0.155 ± 0.014 (93)
▲ GLM 5.3	Realistic	68.2 ± 0.0 (22)	81.0 ± 5.2 (12–13)	**0.080 ± 0.011** (38–39)
	Hard	77.5 ± 2.2 (23–24)	71.6 ± 3.2 (80–83)	**0.121 ± 0.021** (82–84)
● Gemini 3.1 Pro	Realistic	86.4 ± 4.5 (22)	65.0 ± 3.0 (12–13)	0.140 ± 0.081 (38–39)
	Hard	82.6 ± 4.3 (23)	59.0 ± 1.6 (75–77)	0.232 ± 0.047 (90–92)
▲ DeepSeek V4.1 Flash	Realistic	65.2 ± 2.6 (22)	88.9 ± 4.8 (12)	0.149 ± 0.042 (39)
	Hard	84.1 ± 9.1 (23)	83.8 ± 1.8 (77–80)	0.140 ± 0.012 (87–90)
● Claude Fable 5.1	Realistic	63.6 ± 4.5 (22)	76.9 ± 0.0 (13)	0.103 ± 0.012 (35)
	Hard	82.6 ± 4.3 (23)	86.2 ± 1.9 (74–76)	0.125 ± 0.009 (88–92)
▲ Kimi K3	Realistic	60.6 ± 2.6 (22)	77.8 ± 12.7 (12)	0.099 ± 0.013 (39)
	Hard	62.8 ± 3.3 (23–24)	64.3 ± 3.2 (75–76)	0.221 ± 0.061 (90–92)
▲ K2-MoVA 36B-A4B	Realistic	65.7 ± 6.2 (22–23)	83.8 ± 8.4 (12–13)	0.166 ± 0.059 (38–39)
	Hard	68.6 ± 4.7 (23–24)	72.5 ± 4.4 (83–85)	0.172 ± 0.022 (81–83)
● Claude Opus 4.8	Realistic	83.3 ± 2.6 (22)	77.8 ± 4.8 (12)	0.186 ± 0.033 (38–39)
	Hard	82.6 ± 4.3 (23)	60.3 ± 2.3 (79–85)	0.275 ± 0.034 (82–88)
▲ Kimi K2.6	Realistic	74.2 ± 2.6 (22)	81.8 ± 24.6 (12–13)	0.186 ± 0.038 (38–39)
	Hard	76.8 ± 2.5 (23)	79.0 ± 3.1 (77–78)	0.213 ± 0.023 (88–90)
▲ Qwen 3.6 35B	Realistic	75.8 ± 2.6 (22)	51.5 ± 6.2 (12–16)	0.203 ± 0.041 (28–39)
	Hard	67.7 ± 7.5 (24–25)	54.7 ± 4.6 (81–84)	0.375 ± 0.055 (83–86)
● GPT-5.5	Realistic	87.9 ± 5.2 (22)	69.3 ± 6.7 (12–14)	0.223 ± 0.042 (37–38)
	Hard	87.2 ± 11.4 (23–24)	64.2 ± 0.8 (85–89)	0.437 ± 0.057 (76–80)
▲ Gemma 4 26B	Realistic	57.6 ± 2.6 (22)	62.2 ± 4.2 (12–13)	0.267 ± 0.041 (38–39)
	Hard	50.0 ± 2.2 (23–24)	46.7 ± 1.4 (83–88)	1.022 ± 0.122 (79–83)

Mechanisms. Every graded task (10,800 task-grades) gets one label. A task is a *combined failure* if it is an official discrete miss (`correct=False`, abstention included), a non-submitted or unparseable exit, or a material continuous miss: point relative error above 0.25, an interval that misses the SCM value or has NRelIS > 2 (Sec. 3.8), or prediction n-relative RMSE above 0.50. The mechanism is *tool/runtime* if the agent loop produced no usable submission, *reasoning* if a valid discrete answer was wrong, and *data science* if an identifiable numeric answer was materially wrong. Missing files count as tool/runtime and schema errors after a `Submitted` exit are ≈0; the one persistent tool case is Claude Fable 5.1 exiting `RepeatedFormatError` on every run of realistic `scene_000018` without a bash call (App. A.18). The official Pass Rate sees only the first two mechanisms; a bad interval is never a binary fail. Trajectories of the eight leading models (7,200 traces) annotate these grade-level labels — a recovered bash error does not relabel a submitted discrete miss — and supply the first-commitment audit of App. A.11.1.

Combined failures are 28.9±0.3% of task-grades (15.8% realistic, 35.4% hard), and their mix is stable across replications (Tab. 25): half data science, somewhat less reasoning, one in twenty tool/runtime. Restricted to the official discrete pool, misses are 88% reasoning and 12% tool/runtime.

The realistic exam is the more tool-exposed (15.9% of its combined fails in run 1, then 8.2% and 7.8%); on the hard exam tool/runtime is 3.5% and data science 55.3%.

Table 25: **Combined-failure mix by mechanism.** Share of combined failures (%), mean ± sample SD, over three replications; N is the per-run combined-fail count. Combined failure as defined in the text. Violet marks the Realistic exam, orange the Hard exam.

Slice	N / run	Reasoning (%)	Data science (%)	Tool / runtime (%)
Both exams	1039 ± 12	43.0 ± 0.8	52.1 ± 0.9	4.8 ± 0.9
Realistic	190 ± 16	51.4 ± 3.3	37.8 ± 2.6	10.6 ± 4.5
Hard	849 ± 15	41.2 ± 0.6	55.3 ± 0.3	3.5 ± 0.3

Persistence. Over the 3,600 complete (exam, model, scene, task) triples, a data-science fail that occurs once recurs on all three runs 51.9% of the time, a reasoning fail 35.6%, a tool/runtime fail 8.3% (10 of 121; Fig. 8). The sandbox is not the exam. The same separation appears in the metrics Sec. 4 summarizes: the gap between any-of-3 and all-of-3 in Tabs. 1 and 2 is the share of the binary pool that passes on some runs and not others; point estimates reproduce almost exactly across restarts (median per-task NRelErr SD below 0.001 for every model on the realistic exam and below 0.03 on the hard exam, over tasks answered in all three runs), whereas the per-task SD of NRelIS is one to two orders of magnitude larger, driven by a few `ate_uq_95` items that swing between ≈0 and the cap (the point/interval split by k is Tab. 37). The worst/best CausalDSScore in the leaderboards scores each restart as a complete exam, which keeps the covariance between score components.

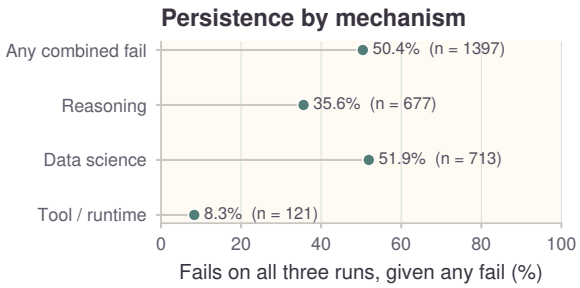


Figure 8: **Persistence by mechanism.** Share of complete (exam, model, scene, task) triples that fail on all three replications, among those that fail on at least one (n in parentheses; 3,600 triples, twelve models, both exams). Teal dots are the measured shares.

`scene_000700` (realistic exam; ATE with 95% CI; grafted confounding host; `proxy_hard`) is the extreme case. Treatment and confounder are latent behind noisy bundles, with gold values on the 1,600 calibration rows only, and the treatment is a sharp threshold of the confounder (treated share by confounder decile $1, 1, 1, 1, 1, 0.95, 0.68, 0.42, 0.09, 0$), so the ATE must be extrapolated where there is no overlap and the answer is decided by the adjustment form: linear or quadratic adjustment on the gold rows gives 0.34–0.41 (SCM value 0.42), an additive spline 0.11, a reconstructed-confounder regression 0.20 or 0.82. Agents choose anew on each restart. Claude Opus 4.8 scores NRelIS 0.18, 5.0, 9.9 on its three runs, GPT-6 Astra 0.22, 10, 0.23, and six of twelve models flip between below 0.6 and above 3.5. These are shifted-point misses in the sense of App. A.11.4 (Opus's point moves $0.41 \to 0.20 \to 0.82$), the mirror image of hard `scene_000070` in App. A.11.2, where every model lands the same attenuated point on every run. Each flip moves CausalDSScore by 0.04–0.10; with `scene_000126`, this item accounts for the whole per-run spread of Opus (0.276, 0.370, 0.415) and of Gemini 3.1 Pro (0.368, 0.232, 0.225).

By model. The field mix is not a model property (Fig. 9, Tab. 26). GPT-6 Astra's fails are 80% data science with no tool/runtime, and its rare discrete misses are wrong effect signs (26 of 32), not false identifiability (2). Gemma 4 26B, GLM 5.3, Kimi K3, and Qwen 3.6 35B fail at least as often on reasoning as on data science. GPT-5.5 inverts the field's abstention error: 41% of its reasoning fails are false abstentions against 19% overall — the same signature as its column in Tab. 19 — and it is the only leading model with a double-digit tool share (15%). Claude Opus 4.8 sits on the field

mean, which here means two large piles rather than balance. Within data science, UQ misses with an acceptable point are 54–66% for every model: models differ in which class they enter, not in how that class breaks.

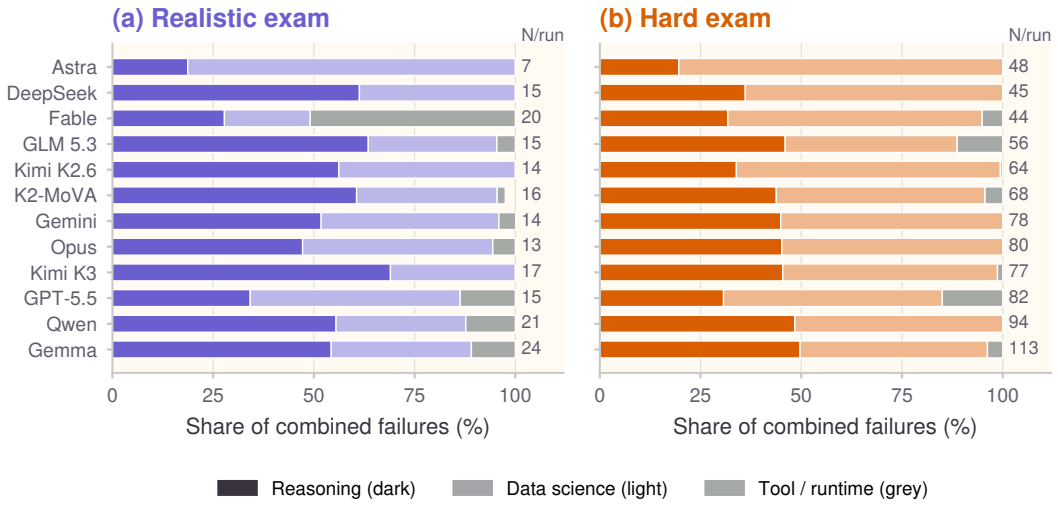


Figure 9: **Combined-failure mix by model and exam.** One bar per model: reasoning (full exam color), data science (light), tool / runtime (gray); shares of that model's combined failures, mean over three replications, with the per-run count N at the bar end. (a) Realistic exam in violet, (b) Hard exam in orange. Models are ordered by pooled combined fails per run.

Table 26: **Subtype shares by model.** Both exams, mean over three replications; N is combined fails per run, mean $\pm$ SD. False ID / false abstention are shares of that model's *reasoning* fails; UQ-ok / uncovered+bad are shares of its *data-science* fails. The mechanism split itself is Fig. 9. Model markers: ● closed, ▲ open-weight.

Model by N	N / run	Of reasoning fails (%)		Of data-science fails (%)	
		False ID	False abs.	UQ-ok	Uncov.+bad
● GPT-6 Astra	55 ± 5	6	3	62	20
▲ DeepSeek V4.1 Flash	60 ± 2	53	22	62	19
● Claude Fable 5.1	64 ± 4	72	3	66	21
▲ GLM 5.3	71 ± 3	64	20	66	16
▲ Kimi K2.6	78 ± 4	61	18	61	24
▲ K2-MoVA 36B-A4B	84 ± 3	51	34	59	25
● Gemini 3.1 Pro	92 ± 4	82	5	64	22
● Claude Opus 4.8	93 ± 5	61	19	56	24
▲ Kimi K3	94 ± 3	80	6	66	20
● GPT-5.5	96 ± 4	22	41	54	30
▲ Qwen 3.6 35B	115 ± 7	58	25	60	22
▲ Gemma 4 26B	137 ± 6	57	22	56	27

A.11.1 Reasoning and data-science failure tracks

Reasoning. A reasoning fail is a valid discrete answer that is wrong: the grader could score the JSON and the binary, set, or sign target was missed. The pool is the discrete pool, not "binary tasks": an effect item that was estimated when it should have been abstained on, or abstained on when identifiable, is graded here; a wrong identification strategy behind a submitted number is not, and the census below exists to catch it. Both abstention errors are gradable only because identifiability is fixed by the hidden graph: `y0.id` and `dowhy` decide it from the SCM, not from a textbook default, so "estimated when it should have abstained" is a fact about the scene, not a judgment call (App. A.9). Three subtypes account for 96% of the pool (Fig. 10a): false identifiability (59%, 796

grades), false abstention (19%), and a wrong effect sign (17%). The first two are the abstention errors of Tab. 19 under other names — a missed abstention there is a false-identifiability grade here — with a different denominator (that table divides by the abstention pool per exam, this section by reasoning fails over both exams); the model ordering agrees, with GPT-6 Astra lowest on both and GPT-5.5 the false-abstention outlier on the hard exam. Wrong adjustment sets are 14 grades in all, and graph reconstruction fails 3 times in 10,800.

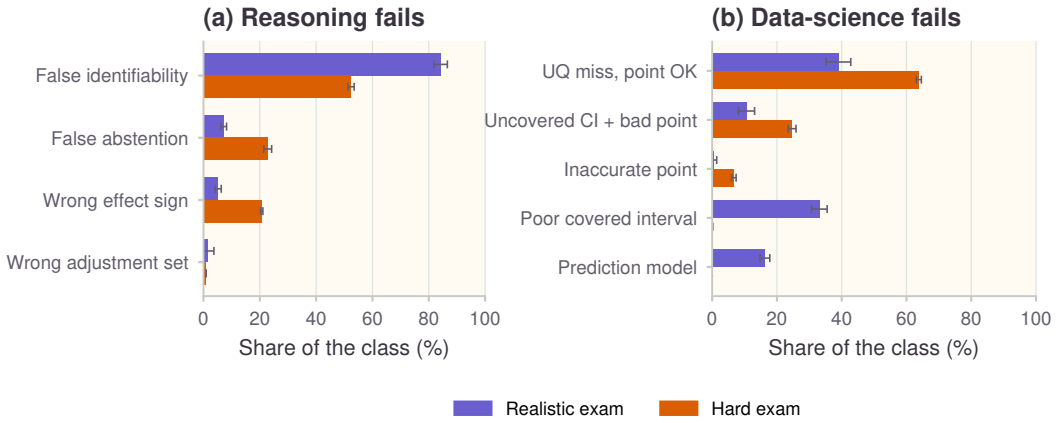


Figure 10: **Failure subtypes by exam.** (a) Reasoning-fail subtypes as a share of reasoning fails; (b) data-science-fail subtypes as a share of data-science fails. Bars are means over three replications, whiskers the sample SD; violet Realistic exam, orange Hard exam. "UQ miss, point OK": the 95% interval misses the SCM value while point relative error is ≤ 0.25. Pooled counts are given in the text.

The two abstention errors are not one skill on different motifs. Of the 796 false-identifiability grades, 502 sit on an IV host and 243 on a grafted graph that still carries an instrument-like unused cause; front-door hosts contribute none. False abstention concentrates on front-door mediation and counterfactual-effect items (Tab. 27): the agent sees an unobserved common cause, calls it non-identification, and never applies the front-door functional. The first is the over-claim diagnosed trace by trace on the running example (App. A.11.2); the per-motif view already ranks models on the `iv` row purely by that call (Tab. 22). The second is the exit the matched-scene ablation isolates under `proxy_hard` (App. A.16). Wrong signs are a hard-exam phenomenon (217 `sign_only` grades on the hard exam against 15 on the realistic exam); the realistic reasoning pile is 84% false identifiability.

Table 27: **Reasoning fails by task family.** Pooled over three replications, twelve models, both exams. Counts, not rates: families have different exam weights.

Family	False ID	False abs.	Wrong sign	Wrong adj. set
Effect estimate	304	89	0	0
Identification	146	21	0	14
Bias diagnostic	172	0	0	0
CF effect	146	88	216	0
CF identification	28	4	0	0
Mediation effect	0	59	16	0

First-commitment census. The grade cannot see a wrong meta-approach behind a submitted number: backdoor-only on a front-door host is data science if a number came out. We therefore coded the first causal commitment — the first assistant turn that names identifiability or a method, including the reasoning summary and the first bash, or the first planning turn after an inspect-only opening — on every leading-model fail with a trajectory plus six effect-item passes per exam and model: 1,971 traces (Tab. 28). Most traces inspect the files before committing; GPT-5.5 and Astra more often commit in the first message, before any tool result.

Table 28: **First-commitment census: strata and first-turn regimes.** Leading-model traces with files (eight models, both exams, three replications; corrected grades). Fail strata are a census; effect-item passes are 6 per exam×model. Inspect-first = the first assistant turn has no identifiability or method claim; pre-tool commit = the causal plan is stated before any tool result. Hidden-reasoning rates per stratum and per model are in Fig. 11.

Stratum	n traces	Inspect-first	Pre-tool commit
Discrete reasoning fail	756	531	225
DS fail, effect / CF / mediation	941	781	160
DS fail, prediction	69	69	0
Effect-item pass (sample)	96	75	21
Tool / runtime	109	60	20

On discrete reasoning fails the first commitment already contains the error 61% of the time (56% on the hard exam, 80% on the realistic), typically IV on a non-identifiable ATE. On data-science effect fails it is motif-consistent in 94% of traces; the 54 exceptions are almost all one thing, a front-door host whose first sentence treats unobserved U as fatal — and that code is the first sentence, not the script that is submitted, as hard `scene_000070` in App. A.11.2 shows. Prediction fails have no causal plan at all. So the continuous pile is not a relabeled reasoning pile; its one leak is front-door graphs read as non-identified backdoors. The flags are not uniform (Fig. 11b): GPT-6 Astra almost never states a false first plan, GPT-5.5 rarely, while Gemini 3.1 Pro, Kimi K3, and Claude Opus 4.8 do so in more than a third of coded traces. That is a process difference; at the grade level (Tab. 26) Astra's fails are still mostly data science.

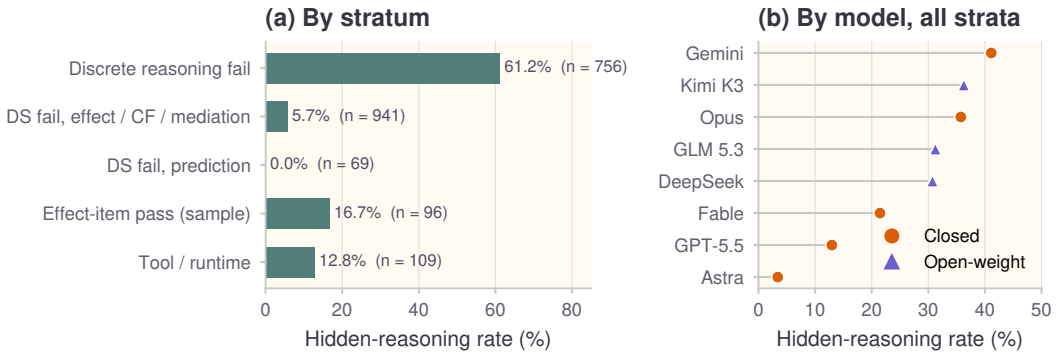


Figure 11: **Hidden reasoning in the first commitment.** Share of traces whose first causal commitment is a false-identifiability plan, false-abstention plan, wrong identification strategy, linearity rescue, or wrong estimand. (a) By census stratum, with n; (b) by model over all strata: orange circles closed models, violet triangles open-weight.

Data science. A data-science fail is an identifiable numeric, interval, or prediction answer that is materially wrong, and the dominant subtype is not a blown point estimate. Across both exams 60% of these fails are *UQ misses with an acceptable point*: the 95% interval misses the SCM value while point relative error is ≤ 0.25 (Fig. 10b; 982 of 1,623). Another 23% miss the interval and the point; the rest are isolated inaccurate points, poor covered intervals, and prediction-model misses, the last two almost entirely on the realistic exam, which still has a prediction family and few severe-proxy interval items. The observation layer moves the pile from covered-but-poor intervals on `clean` to uncovered intervals on `proxy_hard` and `proxy_severe`; App. A.11.4 gives the geometry by view. Among the eight leading models with traces, data-science fails run the longest tool loops (mean 14.5 bash calls versus 7.8 on passes and 10.1 on reasoning fails; per-task budgets are in App. A.14), and 56% show a recovered bash or format error. The typical data-science miss is not "never opened the data". It is a submitted, calibration-aware estimate whose interval is too tight, too shifted, or both; App. A.11.3 shows the script and App. A.11.4 the interval.

A.11.2 Failure-mode examples

One scene per large subtype makes the tracks concrete. Table 29 gives every example in one format — ground truth, the outcome of all twelve models on each run, the stable split, a one-line reading — and Tab. 30 adds the per-agent diagnoses for the running example. They are illustrations, not a sample. What the tables do not carry is the traces.

Table 29: **Scene-level forks.** One row per example scene, twelve models, three replications. Outcome strips show runs 1–3 per model: ● pass, ● reasoning miss (false identifiability or false abstention), ○ data-science miss (uncovered interval), × tool / runtime exit. Scene IDs are colored by exam (Realistic, Hard); K3, K2.6, MoVA, GLM abbreviate Kimi K3, Kimi K2.6, K2-MoVA 36B-A4B, GLM 5.3. ● Closed; ▲ open-weight. Quotes are in the text.

Scene task; host; view; class	Ground truth	Outcomes, runs 1–3	Stable split	Reading
scene_000154 Forbidden-controls bias diagnostic; IV, proxy_hard False identifiability	Not point-identified; correct: non_id	● Astra ●●● ● Gemini ●●● ▲ K2.6 ●●● ● GPT-5.5 ●●● ▲ DeepSeek ●●● ▲ MoVA ●●● ● Opus ●●● ▲ GLM ●●● ▲ Qwen ●●● ● Fable ●●● ▲ K3 ●●● ▲ Gemma ●●●	Astra, Gemini 3/3 non_id; Opus, DeepSeek, GLM, K2-MoVA, both Kimis, Qwen 3/3 no_backdoor; GPT-5.5 2/3, Fable 1/3 correct	IV over-claim: an instrument is read as identifying the population ATE
scene_000470 ATE with 95% CI; IV, proxy_severe False identifiability	Not point-identified; correct: null	● Astra ●●● ● Gemini ●●● ▲ K2.6 ●●● ● GPT-5.5 ●●● ▲ DeepSeek ●●● ▲ MoVA ●●● ● Opus ●●● ▲ GLM ●●● ▲ Qwen ●●● ● Fable ●●● ▲ K3 ●×● ▲ Gemma ●●●	Astra, Fable, DeepSeek, GPT-5.5, Kimi K2.6, Qwen 3/3 null; Opus, Gemini 3/3 number	The same fork when the requested object is an interval
scene_000236 NIE point; front-door, proxy False abstention	NIE identifiable (front-door); correct: a number	● Astra ●●● ● Gemini ●●● ▲ K2.6 ●●● ● GPT-5.5 ●●● ▲ DeepSeek ●●● ▲ MoVA ●●● ● Opus ●●● ▲ GLM ●●● ▲ Qwen ●●● ● Fable ●●● ▲ K3 ●●● ▲ Gemma ●●●	GLM, GPT-5.5, K2-MoVA 3/3 null; Astra, Fable, Gemini, both Kimis, Gemma 3/3 number	Front-door false abstention: latent U read as fatal
scene_000070 ATE with 95% CI; front-door, proxy_hard UQ miss, point OK	SCM ATE -0.121; correct: a covering front-door CI	● Astra ○○○ ● Gemini ○○○ ▲ K2.6 ○○○ ● GPT-5.5 ○○○ ▲ DeepSeek ○○○ ▲ MoVA ●●○ ● Opus ○○○ ▲ GLM ○○○ ▲ Qwen ○○○ ● Fable ○○○ ▲ K3 ○○○ ▲ Gemma ○○●	All eight leading models, all runs: uncovered CI, point relative error ≈ 0.09	Identification held; the interval did not
scene_000700 ATE with 95% CI; grafted confounding, proxy_hard Shifted-point interval miss	SCM ATE 0.42; no overlap in the confounder tails	● Astra ●○○ ● Gemini ○○○ ▲ K2.6 ●○○ ● GPT-5.5 ●○● ▲ DeepSeek ○○○ ▲ MoVA ●●× ● Opus ●○○ ▲ GLM ○○● ▲ Qwen ×●● ● Fable ○○○ ▲ K3 ●○○ ▲ Gemma ○○○	Six of twelve models flip between covered and far off across runs	Shifted-point misses; the adjustment form decides
scene_000018 Prediction interval; —, clean Tool / runtime	Any valid 90% prediction interval	● Astra ●●● ● Gemini ●●● ▲ K2.6 ●●● ● GPT-5.5 ●●● ▲ DeepSeek ●●● ▲ MoVA ●●● ● Opus ●●● ▲ GLM ●●● ▲ Qwen ●●● ● Fable ××× ▲ K3 ●●● ▲ Gemma ●●●	Fable 3/3 RepeatedFormatError; eleven models pass every run	The tool pile in miniature

On the running example (scene_000154, Fig. 2) the population ATE is not point-identified — the subsidy instrument identifies at most a compliance-local effect (y0.id and dowhy agree) — and most models answer no_backdoor: no adjustment set exists, therefore the instrument identifies the ATE. It is the modal discrete miss of the field (502 of 796 false-identifiability grades sit on an IV host), and Tab. 30 shows the routes to it: Kimi K2.6 derives the LATE-versus-ATE gap and then bets against it on guessed grader convention, Claude Opus 4.8 bridges it with a linearity assumption the story never states, Qwen and Gemma never surface the estimand question, Gemini 3.1 Pro starts from

the same over-claim and reasons its way back to `non_id`, and GPT-5.5 abstains in its first and only step.

Table 30: **Per-agent responses on the running example's bias diagnostic.** Forbidden-controls bias diagnostic for `scene_000154` (Fig. 2); quotes are verbatim from the agents' own reasoning traces. All six agents reconstruct the same graph and correctly conclude that no observed backdoor adjustment set exists; the answers diverge only on the final step—whether the subsidy instrument point-identifies the *population* ATE. Failing agents ordered by how much of the correct argument appears in their trace. `non_id` is the correct answer; `no_backdoor` over-claims. ● Closed; ▲ open-weight.

Model	Answer	Diagnosis verbatim quotes from the reasoning trace
● Gemini 3.1 Pro	`non_id`	Starts where the failures end ("S is a valid instrument"), then prosecutes its own claim at length: recalls that nonparametric IV with a binary treatment yields only Balke–Pearl bounds, notices that the story's threshold mechanism (subsidy sufficient *and* permeable soil) "definitively breaks the linear SEM assumptions" that would rescue point identification, and reverses to `non_id`.
● GPT-5.5	`non_id`	Answers in its first and only step, without opening anything in the workspace: the subsidy is "an instrument-like cause of treatment, not a variable that blocks the backdoor path", so the ATE "is not nonparametrically identifiable from this graph by adjustment or another graph-based strategy". The nonparametric standard is applied by default, with no deliberation about what the grader might accept.
▲ Kimi K2.6	`no_backdoor`	Its thinking trace contains the correct answer in full: it derives that IV identifies "the local average treatment effect (LATE)... but the population ATE may not be", and even works out the bias-amplification algebra for conditioning on an instrument—then bets against its own analysis on guessed grader convention: "the question likely comes from a course or dataset where they expect `no_backdoor`" and "the problem might not be that nuanced".[4]
● Claude Opus 4.8	`no_backdoor`	Reaches the correct fork, then bridges it with parametric assumptions it reads into the narration: "the population ATE may be identifiable via IV under linearity/additivity assumptions. The story describes [the outcome] as linear-additive combination... So ATE is identifiable by IV". The story states no functional form (the outcome "reflects the combined action" of treatment and confounder), and the task is symbolic—there is no released data against which linearity could be checked, and its sole exploration is reading the story. The "rescue" rests on an assumption that is neither stated nor checkable.
▲ Qwen 3.6 35B	`no_backdoor`	Reverses its verdict four times ("Let me go with `non_id` as the safer answer... OK let me just commit to `no_backdoor`"), in between spending most of a 60-message trajectory re-printing the 1.6 kB story it already had verbatim in the prompt. The final commitment verifies the three instrument conditions—relevance, exclusion, independence, all of which *do* hold—and treats that checklist as the finish line: "So IV identification is valid! The answer should be `no_backdoor`." But those conditions (plus the monotonicity it *never* invokes) deliver only the complier-local effect, not the population ATE the task asks for. Unlike the two agents above, it never registers that a bridging step is missing: where Kimi K2.6 derived the LATE-vs-ATE gap and gambled against it, and Claude Opus 4.8 filled it with an invented linearity assumption, Qwen never surfaces the estimand question at all—LATE, monotonicity, and effect homogeneity appear nowhere in its trace.
▲ Gemma 4 26B	`no_backdoor`	Never questions the step at all: "Is the ATE identifiable? Yes, via IV" is asserted flatly, and the entire deliberation—written into a 22 kB draft of the answer file itself, whose first version answered `[]`—concerns which answer label fits the definition of a forbidden control, not whether the instrument identifies anything.

On hard `scene_000470` the same fork appears on an interval item. Astra's first turn is the whole argument ("Fossil reserves alone do not identify the population ATE without additional assumptions"); Opus writes the same nonparametric sentence and then runs a Wald ratio off a first stage with R^2=0.006, rescued by a homogeneity assumption read out of an OLS fit — the model that states the principle is not the one that acts on it. On `scene_000236` the front-door NIE is identifiable and GLM abstains with the mediation-formula condition applied one graph too early ("with U latent, $E[Y_{x0,m}]$ cannot be identified via the mediation formula"); GPT-5.5 makes the same exit on `scene_000188` from a single bundled script that decides identifiability and writes the null together. On `scene_000070` identification held and the interval did not. Fable names the front-door criterion and calls the effect "essentially null" (≈ -0.017, CI $\approx[-0.048, 0.013]$, SCM -0.121); Opus reaches the same number from the other side — first sentence an open backdoor,

[4] Kimi K2.6's meta-reasoning is *precisely* the failure mode *CausalDS*'s combination of fully synthetic generation and first-class abstention grading is designed to expose: fitting the answer to an imagined exam sourced from a textbook rather than to the problem at hand.

submitted script a front-door functional, the near-zero located in the reconstruction ("the coefficient of $E \mid G$ is nearly zero ... measurement noise attenuation"). The field agrees on a tight interval around a mediator regression that the noisy bundles have flattened.

A.11.3 What the agent actually ran

The census above codes the named plan; this section codes the bash that produced the number, on every leading-model data-science fail and interval pass with a trajectory (1,598 traces: 941 fails, 657 passes). The label is the identification functional the last substantial Python script implements — front-door, g-formula, AIPW / double ML, IPW, 2SLS, OLS with covariates, an unadjusted mean difference, or a mediation product; a front-door mention in that script counts as a front-door attempt, since parametric front-door is often `sm.OLS` of Y on M and X, and `LinearRegression` imports are calibration helpers, not the estimator.

99.5% of data-science fails read `calibration.parquet`, 40% mention attenuation, reliability, or measurement error, and scikit-learn and statsmodels are used as calibration and OLS helpers rather than as a causal API. The functional does not separate a pass from a fail (Tab. 31). Pooled, fails look more front-door-heavy only because front-door hosts are over-represented among them (226 fails vs. 109 interval passes); within a host the mixes line up, and about one in six last scripts match no pattern (Astra 40%) — custom one-shots, not a hidden second functional. GPT-5.5 is the only model that leans on AIPW as a named object (21% of its data-science fails). The interval recipe is even more uniform: 95% of data-science fails compute a bootstrap percentile interval, an analytic Wald interval, or both, and passes use the same recipes in similar proportions. So the continuous pile is not "OLS on a front-door graph". The agents reconstructed the concepts and ran a motif-typical script about as often as the passers; what differs is the interval they then submitted.

Table 31: **Last-script functional on leading-model traces, within structural motif.** Front-door = the last Python script names or implements a front-door formula; OLS = `sm.OLS`/`smf.ols` in that script and no front-door mention.

Host	n fail / pass	Leading functional, fails	Leading functional, passes
Front-door	226 / 109	front-door 67.3%; OLS 20.4%	front-door 55.0%; OLS 23.9%
Confounding	115 / 67	OLS 44.3%; g-formula 20.0%; AIPW 15.7%	OLS 50.7%; g-formula 19.4%; AIPW 14.9%

A.11.4 Interval coverage and error geometry

Interval calibration is the weakest axis for every model. Two pools are used. Fig. 12 covers the identifiable `ate_uq_95` tasks with a valid interval (7 per run on the realistic exam, 34 on the hard): coverage of the nominal 95% interval is below nominal for every model on both exams, well below for most, and only loosely related to the leaderboard (the best-covered model on the realistic exam is not the score leader). The geometry below uses the wider pool of every graded effect, counterfactual, and mediation interval on both exams, all twelve models and three replications (2,231 usable triples $(\hat{\tau}, [L, U], \tau)$ recovered from the grade report, 15 unparseable, observation variants joined from `exam.json` where the report omits them); its empirical coverage is 39% (37% hard, 56% realistic). The two coverage numbers are therefore not the same statistic; the matched-scene ablation (App. A.16) sees the same under-coverage on a fixed set of scenes. The pattern is consistent with recent uncertainty-elicitation benchmarks, in which nominal intervals from frontier models cover far below their nominal level (Hobor et al., 2026; Epstein et al., 2025; Renda et al., 2025).

Error geometry. An uncovered interval almost always contains the agent's own point and is almost never off-center relative to it. The miss therefore splits on whether the point is acceptable, the same cut as Fig. 10b: 72% of uncovered intervals are *centered too narrow* — point and midpoint within the point-OK band of τ, interval too short — and 27% are a *shifted point* (Tab. 32). Among the centered misses the median half-width would have to grow by 2.7×, at the same center, to cover.

The observation layer changes how often the point breaks, not the shape of the typical miss (Fig. 13). Coverage falls from 79% on `clean` to 29% on `proxy_severe`, and the shifted-point share of the

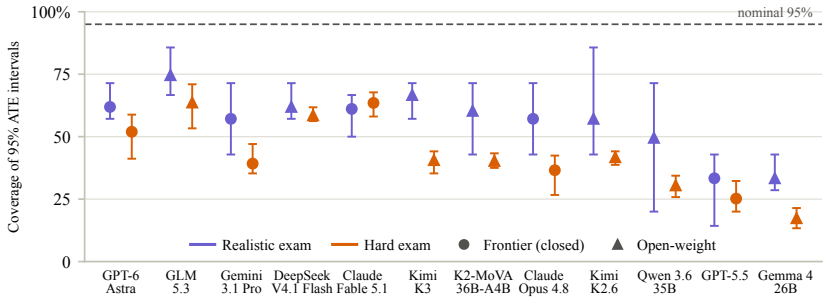


Figure 12: **Coverage of nominal** 95% **ATE intervals.** Per model on the realistic (violet) and hard (orange) exam, over identifiable `ate_uq_95` tasks with a valid interval (7 and 34 per run; fewer where a run returned no interval). Marker: mean over three runs; whisker: worst to best run. Circles: closed models; triangles: open-weight. Models in realistic-exam leaderboard order.

Table 32: **Geometry of uncovered effect-family** 95% **intervals.** Twelve models, both exams, three replications. Centered too narrow = point and midpoint both within relative error 0.25 of the SCM value; shifted point = the point itself is outside that band. Width is the median normalized width; expansion is the factor on half-width, at the submitted center, that would have covered.

Class	n	Share of uncovered %	Med. norm. width	Med. expansion
Centered too narrow	984	72.5	0.054	2.70
Shifted point	371	27.3	0.100	7.10
Off-center interval	3	0.2	0.260	1.95
All uncovered	1358	100	0.065	3.59
All covered	873	—	0.074	0.44

misses rises from zero to 39%; even under the worst measurement the majority of misses stay centered too narrow, and every one of the 15 `clean` misses does. A pooled covered-versus-uncovered width comparison is the wrong statistic, for the compositional reason of App. A.11.3: uncovered rows sit on the noisier views, which have wider intervals either way. Within a view, the models that cover on `proxy_hard` and `proxy_severe` did widen (median normalized width 0.062 vs. 0.048 and 0.165 vs. 0.098).

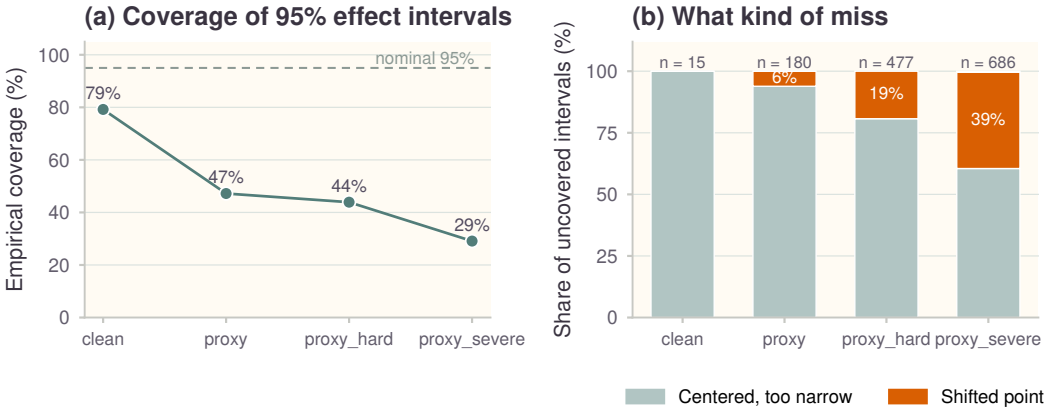


Figure 13: **Interval coverage by observation variant.** (a) Empirical coverage of the nominal 95% effect-family intervals per observation variant (teal, measured; dashed gray line is the nominal level). (b) Share of uncovered intervals that are centered too narrow (light teal) versus a shifted point (orange), with the number of uncovered intervals above each bar.

Models differ in how tight they are willing to be, not in the shape of the error (Fig. 14). GLM 5.3 covers most often and would have needed a $1.8\times$ expansion on its centered misses; GPT-5.5 and Gemma 4 26B cover least and would have needed 6–$8\times$. Claude Opus 4.8 is not the calibration leader once the target is a 95% interval rather than a point, and GPT-6 Astra, the most data-science-heavy model, covers 46% with intervals that are merely somewhat too tight. The data-science class is therefore a UQ class in a precise sense: overconfidence after a correct identification decision, not a second helping of false identifiability.

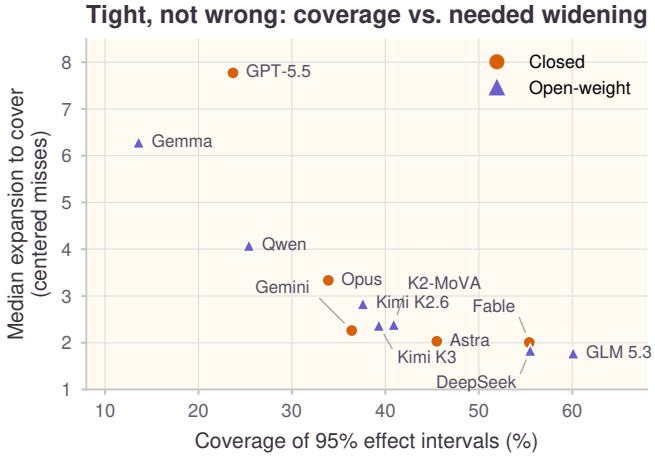


Figure 14: **Coverage versus the widening that would have covered.** Per model, empirical coverage of effect-family 95% intervals against the median expansion factor on the centered-too-narrow misses (both exams, three replications). Orange circles closed models, violet triangles open-weight.

A.12 Stability analysis of exam generation

Because CausalDS is a generator rather than a fixed dataset, an exam is one draw of tasks from the scene collection, and a leaderboard is only as meaningful as it is stable under redrawing that exam. Sec. 4 reports run-to-run variation as an SD over three replications, and App. A.11 shows it concentrates in a few interval items and abstention flips; this section holds the models fixed and varies the exam instead. Tables 33 and 34 repeat the per-run means of Tabs. 1 and 2 with 95% finite-exam intervals. Pass Rate uses a Wilson interval on the run-averaged rate, with the mean per-run denominator as sample size (34–35.7 on the realistic exam, 97–111 on the hard). Every other column uses percentile intervals of a task bootstrap stratified by grading category: $B{=}10{,}000$ pseudo-exams redraw tasks with replacement within each of the five ground-truth scoring pools, so that a pseudo-exam has the same number of exact-match, interval, abstention, point, and graph/set tasks as the real one (realistic: exact-match content 18, interval 14, abstention 16, point 25, graph/set 27; hard: exact-match content 41, interval 59, abstention 56, point 34, graph/set 10), each replication is rescored as a full exam under the leaderboard routing, and the three values are averaged. The Rank column is the CausalDSScore rank over the same pseudo-exams; all-of-3 and any-of-3 are computed on the ground-truth binary pool of App. A.13 with the same intervals; Med. F_1-Loss is omitted because point estimate and interval are 0.000 for all but two hard-exam models. Table 35 uses the same pseudo-exams, in which every model is scored on the same drawn tasks, to give the probability that one model beats another on CausalDSScore (ties count one half). Table 36 recomputes the ranks under four scoring-rule variants: interval cap $c{=}5$ or 20 instead of 10 in S_{NR} (Eq. 3), the median instead of the capped mean for the interval term, and equal pool weights instead of count-proportional weights in Eq. 4.

Resampling within grading categories describes a redrawn exam of exactly the same grading composition. A redrawn exam would not have that: the generator fixes scene strata, not scoring pools, so the closer analogue resamples within the groups the exam was assembled from — on the hard exam its seven construction strata (identifiable front-door scenes (30), identifiable grafted scenes (46), identifiable mediator scenes (54), identifiable simple scenes (14), non-identifiable double-negative-control scenes (13), non-identifiable IV scenes (22), other non-identifiable scenes (21)), on the realistic exam

Table 33: **Realistic-exam leaderboard with finite-exam intervals (100 tasks).** Per-run means over three replications with 95% finite-exam intervals in brackets. Rank is the CausalDSScore rank with its interval; all-of-3 and any-of-3 are the fractions of the ground-truth binary pool (n=34) passed in all three and in at least one replication. Rows ordered by CausalDSScore; model names abbreviated as in Tab. 20. Model markers: ● closed, ▲ open-weight.

#	Model	CausalDSScore ↓	Rank	Pass Rate ↑	Med. NRel. Err ↓	S_{NR} ↓	all-of-3 ↑	any-of-3 ↑
1	● GPT-6	0.211 [0.137, 0.293]	1 [1, 2]	96.1% [83.6, 99.2]	0.084 [0.038, 0.281]	0.507 [0.324, 0.713]	94.1% [85.3, 100.0]	100.0% [100.0, 100.0]
2	▲ GLM	0.268 [0.193, 0.353]	2 [2, 6]	72.8% [56.2, 84.8]	0.080 [0.030, 0.294]	0.454 [0.279, 0.657]	64.7% [52.9, 76.5]	82.4% [70.6, 94.1]
3	● Gemini	0.275 [0.197, 0.368]	3 [2, 7]	78.7% [62.5, 89.1]	0.140 [0.040, 0.301]	0.525 [0.341, 0.753]	67.6% [52.9, 79.4]	88.2% [76.5, 97.1]
4	▲ DeepSeek	0.281 [0.202, 0.371]	4 [2, 8]	73.5% [56.9, 85.4]	0.149 [0.041, 0.291]	0.489 [0.306, 0.704]	64.7% [50.0, 79.4]	85.3% [73.5, 94.1]
5	● Fable	0.294 [0.204, 0.396]	5 [2, 9]	68.6% [52.0, 81.4]	0.103 [0.028, 0.318]	0.501 [0.293, 0.745]	61.8% [47.1, 76.5]	79.4% [64.7, 91.2]
6	▲ K3	0.294 [0.225, 0.374]	6 [4, 9]	66.7% [49.9, 80.1]	0.099 [0.050, 0.307]	0.464 [0.302, 0.653]	52.9% [41.2, 64.7]	82.4% [70.6, 94.1]
7	▲ K2-MoVA	0.300 [0.231, 0.369]	7 [1, 9]	72.1% [55.5, 84.2]	0.166 [0.065, 0.307]	0.527 [0.376, 0.678]	58.8% [44.1, 73.5]	85.3% [73.5, 94.1]
8	● Opus	0.354 [0.238, 0.492]	8 [2, 10]	81.4% [65.4, 91.0]	0.186 [0.052, 0.343]	0.747 [0.465, 1.087]	76.5% [64.7, 88.2]	88.2% [76.5, 97.1]
9	▲ K2.6	0.385 [0.264, 0.536]	9 [6, 11]	76.8% [60.5, 87.8]	0.186 [0.067, 0.304]	0.790 [0.489, 1.177]	61.8% [50.0, 73.5]	91.2% [82.4, 100.0]
10	▲ Qwen	0.392 [0.288, 0.518]	10 [3, 11]	66.5% [50.1, 79.7]	0.203 [0.068, 0.324]	0.764 [0.489, 1.109]	55.9% [44.1, 67.6]	82.4% [70.6, 94.1]
11	● GPT-5.5	0.560 [0.341, 0.804]	11 [9, 12]	81.0% [65.2, 90.6]	0.223 [0.069, 0.353]	1.308 [0.739, 1.946]	70.6% [55.9, 85.3]	94.1% [85.3, 100.0]
12	▲ Gemma	0.662 [0.434, 0.929]	12 [11, 12]	59.2% [42.7, 73.9]	0.267 [0.081, 0.404]	1.352 [0.768, 2.038]	41.2% [29.4, 52.9]	79.4% [67.6, 91.2]

Table 34: **Hard-exam leaderboard with finite-exam intervals (200 tasks).** Per-run means over three replications with 95% finite-exam intervals in brackets. Rank is the CausalDSScore rank with its interval; all-of-3 and any-of-3 are the fractions of the ground-truth binary pool (n=97) passed in all three and in at least one replication. Rows ordered by CausalDSScore; model names abbreviated as in Tab. 20. Model markers: ● closed, ▲ open-weight.

#	Model	CausalDSScore ↓	Rank	Pass Rate ↑	Med. NRel. Err ↓	S_{NR} ↓	all-of-3 ↑	any-of-3 ↑
1	▲ GLM	0.559 [0.422, 0.708]	1 [1, 3]	73.0% [63.8, 80.5]	0.121 [0.082, 0.196]	0.988 [0.677, 1.336]	68.0% [58.8, 77.3]	89.7% [83.5, 95.9]
2	● Fable	0.627 [0.426, 0.852]	2 [1, 4]	85.4% [77.1, 91.0]	0.125 [0.086, 0.161]	1.221 [0.786, 1.714]	81.4% [73.2, 88.7]	90.7% [84.5, 95.9]
3	▲ DeepSeek	0.639 [0.447, 0.855]	3 [1, 4]	83.9% [75.5, 89.8]	0.140 [0.111, 0.204]	1.258 [0.832, 1.737]	81.4% [73.2, 88.7]	92.8% [87.6, 96.9]
4	● GPT-6	0.733 [0.524, 0.965]	4 [2, 4]	90.4% [82.9, 94.8]	0.155 [0.112, 0.249]	1.475 [1.029, 1.974]	86.6% [80.4, 92.8]	94.8% [90.7, 99.0]
5	▲ K2.6	0.950 [0.715, 1.205]	5 [5, 8]	78.5% [69.5, 85.4]	0.213 [0.147, 0.364]	1.886 [1.361, 2.452]	73.2% [63.9, 82.5]	88.7% [81.4, 94.8]
6	● Gemini	0.963 [0.744, 1.207]	6 [5, 9]	64.5% [54.7, 73.3]	0.232 [0.136, 0.372]	1.727 [1.253, 2.253]	57.7% [49.5, 67.0]	73.2% [64.9, 81.4]
7	▲ K3	0.966 [0.752, 1.195]	7 [5, 8]	64.0% [54.2, 72.8]	0.221 [0.148, 0.333]	1.728 [1.264, 2.228]	54.6% [45.4, 63.9]	75.3% [67.0, 83.5]
8	▲ K2-MoVA	0.967 [0.744, 1.207]	8 [5, 8]	71.7% [62.5, 79.3]	0.172 [0.120, 0.288]	1.972 [1.436, 2.548]	69.1% [59.8, 78.4]	88.7% [82.5, 94.8]
9	● Opus	1.136 [0.888, 1.398]	9 [8, 10]	65.2% [55.7, 73.7]	0.275 [0.190, 0.444]	2.237 [1.663, 2.841]	61.9% [52.6, 71.1]	78.4% [70.1, 86.6]
10	▲ Qwen	1.337 [1.101, 1.579]	10 [10, 11]	57.6% [48.2, 66.5]	0.375 [0.244, 1.084]	2.640 [2.083, 3.214]	47.4% [38.1, 56.7]	77.3% [69.1, 85.6]
11	● GPT-5.5	1.416 [1.175, 1.658]	11 [10, 11]	69.0% [59.9, 76.9]	0.437 [0.287, 1.641]	3.171 [2.563, 3.787]	63.9% [54.6, 73.2]	90.7% [85.6, 95.9]
12	▲ Gemma	1.594 [1.334, 1.867]	12 [11, 12]	47.4% [38.3, 56.7]	1.022 [0.502, 2.177]	3.249 [2.610, 3.905]	32.0% [23.7, 40.2]	75.3% [67.0, 83.5]

a plain resample of its 100 tasks — letting the pool mix and the CausalDSScore weights vary as they would from one generated exam to the next. Under that scheme the CausalDSScore intervals widen by 1.3–1.6× on the realistic exam and 1.1–1.4× on the hard, and no rank interval moves by more than one place at either end. The ranking is therefore stable to how the exam was composed, not only to which tasks happened to fill a fixed composition.

Table 35: **Paired rank stability.** Probability that the row model has the lower (better) CausalDSScore than the column model over the pseudo-exams of Tabs. 33 and 34. Realistic exam above the diagonal, hard exam below the diagonal. Rows and columns follow the realistic-exam rank; the hard-exam rank is in parentheses. Model markers: ● closed, ▲ open-weight.

	1	2	3	4	5	6	7	8	9	10	11	12
1 ● GPT-6 Astra (4)	–	0.98	0.98	0.99	0.99	1.00	0.96	1.00	1.00	1.00	1.00	1.00
2 ▲ GLM 5.3 (1)	0.96	–	0.59	0.75	0.78	0.97	0.76	0.92	0.99	0.95	0.99	1.00
3 ● Gemini 3.1 Pro (6)	0.00	0.00	–	0.56	0.69	0.78	0.70	0.94	0.98	0.94	0.99	1.00
4 ▲ DeepSeek V4.1 Flash (3)	0.89	0.19	1.00	–	0.66	0.82	0.68	0.89	0.99	0.92	0.99	1.00
5 ● Claude Fable 5.1 (2)	0.88	0.23	1.00	0.56	–	0.52	0.56	0.81	0.96	0.89	0.98	1.00
6 ▲ Kimi K3 (7)	0.00	0.00	0.48	0.00	0.00	–	0.54	0.81	0.95	0.90	0.98	1.00
7 ▲ K2-MoVA 36B-A4B (8)	0.00	0.00	0.48	0.00	0.00	0.51	–	0.78	0.92	0.97	0.99	1.00
8 ● Claude Opus 4.8 (9)	0.00	0.00	0.03	0.00	0.00	0.02	0.01	–	0.64	0.74	0.97	1.00
9 ▲ Kimi K2.6 (5)	0.01	0.00	0.56	0.00	0.00	0.58	0.57	0.98	–	0.54	0.94	1.00
10 ▲ Qwen 3.6 35B (10)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	–	0.96	1.00
11 ● GPT-5.5 (11)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.25	–	0.92
12 ▲ Gemma 4 26B (12)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	–

Table 36: **Rank under scoring-rule variants.** CausalDSScore rank (1 = best) under the scoring-rule variants defined in the text, both exams; base is the published rule. Rows ordered by the realistic-exam CausalDSScore. Model markers: ● closed, ▲ open-weight.

	Realistic exam					Hard exam				
Model	base	c=5	c=20	median	equal w	base	c=5	c=20	median	equal w
● GPT-6 Astra	1	1	1	1	1	4	4	4	3	4
▲ GLM 5.3	2	3	2	3	2	1	1	1	4	1
● Gemini 3.1 Pro	3	2	4	2	3	6	7	8	8	5
▲ DeepSeek V4.1 Flash	4	4	3	7	4	3	3	3	2	3
● Claude Fable 5.1	5	5	5	6	7	2	2	2	1	2
▲ Kimi K3	6	6	6	8	5	7	8	5	7	6
▲ K2-MoVA 36B-A4B	7	7	7	5	6	8	6	6	6	8
● Claude Opus 4.8	8	9	8	4	8	9	9	9	9	9
▲ Kimi K2.6	9	8	9	9	9	5	5	7	5	7
▲ Qwen 3.6 35B	10	10	10	11	10	10	11	10	10	10
● GPT-5.5	11	11	11	10	11	11	10	11	11	11
▲ Gemma 4 26B	12	12	12	12	12	12	12	12	12	12

On the realistic exam the intervals support three orderings: GPT-6 Astra first (rank interval $[1, 2]$, beating every other model in at least 96% of pseudo-exams), Gemma 4 26B last ($[11, 12]$), GPT-5.5 eleventh ($[9, 12]$). Ranks 2–7 (GLM 5.3, Gemini 3.1 Pro, DeepSeek V4.1 Flash, Claude Fable 5.1, Kimi K3, K2-MoVA) are one band — every rank interval contains 2 and reaches 6–9, and neighbors beat each other with probability 0.52–0.78 — and Claude Opus 4.8, Kimi K2.6, and Qwen 3.6 35B form a second band at 8–10. A 100-task exam cannot order either band, and its Pass Rate intervals are ± 15 points wide on a binary pool of 34. The hard exam is a different instrument with larger pools, and its intervals are correspondingly narrower: GLM 5.3 is first with probability 0.77–0.96 against the next three, Fable and DeepSeek are tied for second (0.56), Astra is fourth ($[2, 4]$), Kimi K2.6, Gemini, Kimi K3, and K2-MoVA are a band at 5–8 with pairwise probabilities near 0.5, and the bottom four are ordered except Qwen and GPT-5.5 ($[10, 11]$ each). The scoring rule adds little uncertainty of its own on either exam. Under the interval-cap and equal-weight variants no rank moves more than two places and the extremes are fixed; only the median variant reshuffles the middle (Opus $8 \to 4$ on the realistic exam, GLM $1 \to 4$ on the hard), because it discards the capped interval tail that the mean deliberately keeps (Sec. 3.8). What survives both the task draw and the rule is, on the realistic exam, Astra first and Gemma last with GPT-5.5 just above it; on the hard exam, GLM, Fable, DeepSeek, and Astra as the top four and Gemma, GPT-5.5, and Qwen as the bottom three.

A.13 Per-axis reliability across three runs

Three replications already separate the models on reliability. In Tabs. 1 and 2 all-of-3 spreads much more than any-of-3, widest for the weaker open models, and the best–worst CausalDSScore range runs from a few percent of the mean to a factor of two on the realistic exam. The variance is not in the

point estimates, which reproduce almost exactly across restarts (App. A.13, Tab. 37); it comes from a few interval tasks that swing between near-correct and the cap, and from abstention flips re-scored in a small pass/fail pool. Persistence follows the same pattern: a data-science miss recurs on all three runs about half the time, a reasoning miss about a third, a tool exit rarely (Fig. 8).

Setup. The finite-exam intervals of those leaderboard columns are in App. A.12; here we give their definitions, the continuous analogue, and the per-axis breakdowns. Each task has $N{=}3$ complete independent restarts, each a fresh single-shot trajectory. Because the ground truth is generated, a task can be assigned to a scoring pool by what the SCM says rather than by what the model answered: non-identifiable targets form the binary abstention pool, where a committed answer is wrong, not incomplete; identifiable targets keep their native metric, with failed attempts handled according to that metric. The pools are therefore fixed and common to all twelve models — binary 34 / 97, point 25 / 34, interval 14 / 59, graph/set 27 / 10 on the realistic / hard exam. Point and interval losses are pooled separately here while the leaderboard's Med. NRel. Err mixes them, so the continuous numbers below are not comparable to Tab. 1. Any-of-N is the fraction of tasks passed in at least one of the N observed runs; all-of-N is the fraction passed in every run. The Mean rows average each task's success fraction over the fixed ground-truth binary pool. These summaries use all $N{=}3$ runs. The main-text mini-exam instead uses $N{=}10$ attempts to estimate pass@k and pass$\hat{}k$ over size-k subsets for $k < N$ (Sec. 4); at $k = N$, those estimators reduce to any-of-N and all-of-N. For losses, Mean averages each task's valid losses, best-of-3 takes its smallest valid loss, and worst-of-3 its largest when all three attempts are valid; these are the $k = 1$ and $k = N = 3$ cases of Eq. 8. Best-of-3 is an oracle selection using the graded loss (Brown et al., 2024). The reported values are equal-task means over eligible tasks, whose pools can differ between summaries. Set/graph failures receive loss 1; point and interval failures have no valid loss.

Subset estimators. The mini-exam figure of Sec. 4 (Fig. 4) uses the following estimators with $N{=}10$ attempts; the Mean, best-of-3, and worst-of-3 columns above are their $k = 1$ and $k = N = 3$ cases. On the numeric pools an invalid attempt has no loss and is handled by the estimators' conditioning. For the binary pool we use the unbiased hypergeometric pass@k and pass$\hat{}k$ estimators over the $N{=}10$ attempts (Chen et al., 2021; Yao et al., 2024). For the loss-scored tasks we use the continuous analogue, as in eff@k (Qiu et al., 2025) and worst-at-k (Arora et al., 2025): the expected best (smallest) and worst (largest) loss over a uniformly drawn k-subset of the attempts. With a task's v valid losses sorted as $\ell_{(1)} \le \cdots \le \ell_{(v)}$,

$$b(k) = \frac{\sum_{r=1}^{v} \ell_{(r)} \binom{N-r}{k-1}}{\binom{N}{k} - \binom{N-v}{k}}, \qquad w(k) = \frac{\sum_{r=1}^{v} \ell_{(r)} \binom{r-1}{k-1}}{\binom{v}{k}}, \tag{8}$$

where $b(k)$ (*best-of-k*) conditions on the subset containing at least one valid loss and $w(k)$ (*worst-of-k*) on all k losses being valid; both reduce to the mean loss at $k = 1$, so no attempt is privileged. Curves are equal-task means over the tasks for which the statistic is defined, so that a single failing task stays visible. The strips under the loss panels show the expected number of such tasks, the sum over tasks of the conditioning probability, which is the denominator in Eq. 8 divided by $\binom{N}{k}$. Failures on bounded losses (set-valued $1 - F_1$) score 1 and count as valid; failures on the unbounded point and interval losses stay excluded.

Table 37: **Continuous-task reliability.** Equal-task means of the per-task mean, best-of-3, and worst-of-3 losses over three full-exam runs. Parentheses give mean valid tasks/run for Mean, tasks with at least one valid loss for best-of-3, and tasks valid in all three runs for worst-of-3, each over the pool total. Set/graph failures have loss 1; point and interval losses are uncapped. Models follow realistic-exam CausalDSScore; lowest per exam and pool in bold. Model markers: ● closed, ▲ open-weight.

	Realistic			Hard		
Model	Mean	best-of-3	worst-of-3	Mean	best-of-3	worst-of-3
Point (NRelErr)						
● GPT-6 Astra	0.125 (25.0/25)	0.120 (25/25)	0.132 (25/25)	0.131 (34.0/34)	0.069 (34/34)	0.221 (34/34)
▲ GLM 5.3	0.126 (24.7/25)	0.121 (25/25)	0.136 (24/25)	0.088 (30.0/34)	0.063 (33/34)	**0.104** (26/34)
● Gemini 3.1 Pro	0.127 (24.3/25)	0.122 (25/25)	0.139 (23/25)	0.111 (32.3/34)	0.083 (33/34)	0.143 (31/34)
▲ DeepSeek V4.1 Flash	0.129 (25.0/25)	0.121 (25/25)	0.144 (25/25)	0.105 (32.0/34)	0.073 (34/34)	0.150 (30/34)
● Claude Fable 5.1	**0.120** (23.0/25)	**0.117** (23/25)	**0.125** (23/25)	**0.081** (34.0/34)	**0.057** (34/34)	0.108 (34/34)
▲ Kimi K3	0.134 (25.0/25)	0.125 (25/25)	0.144 (25/25)	0.110 (32.7/34)	0.082 (33/34)	0.146 (32/34)
▲ K2-MoVA 36B-A4B	0.137 (24.7/25)	0.128 (25/25)	0.152 (24/25)	0.109 (30.0/34)	0.077 (32/34)	0.135 (26/34)
● Claude Opus 4.8	0.146 (24.7/25)	0.141 (25/25)	0.154 (24/25)	0.148 (31.7/34)	0.127 (34/34)	0.180 (29/34)
▲ Kimi K2.6	0.137 (24.7/25)	0.125 (25/25)	0.153 (24/25)	0.123 (31.3/34)	0.089 (34/34)	0.156 (29/34)
▲ Qwen 3.6 35B	0.148 (22.3/25)	0.133 (25/25)	0.141 (18/25)	0.181 (30.0/34)	0.119 (32/34)	0.237 (28/34)
● GPT-5.5	0.141 (23.7/25)	0.134 (25/25)	0.154 (21/25)	0.170 (27.0/34)	0.137 (31/34)	0.183 (21/34)
▲ Gemma 4 26B	0.164 (24.7/25)	0.148 (25/25)	0.174 (24/25)	0.157 (28.3/34)	0.123 (30/34)	0.207 (27/34)
Interval (NRelIS)						
● GPT-6 Astra	1.526 (14.0/14)	1.038 (14/14)	2.475 (14/14)	41.787 (59.0/59)	1.525 (59/59)	121.394 (59/59)
▲ GLM 5.3	**1.206** (13.7/14)	1.038 (14/14)	1.406 (13/14)	**1.552** (52.7/59)	**0.671** (58/59)	**2.440** (45/59)
● Gemini 3.1 Pro	1.459 (14.0/14)	**0.958** (14/14)	2.250 (14/14)	3.177 (59.0/59)	1.945 (59/59)	4.348 (59/59)
▲ DeepSeek V4.1 Flash	1.290 (14.0/14)	1.087 (14/14)	1.409 (14/14)	2.245 (56.7/59)	1.511 (59/59)	2.647 (54/59)
● Claude Fable 5.1	1.407 (12.0/14)	1.268 (12/14)	1.647 (12/14)	2.100 (56.0/59)	1.465 (59/59)	2.793 (52/59)
▲ Kimi K3	1.220 (14.0/14)	1.027 (14/14)	**1.363** (14/14)	3.035 (58.3/59)	1.900 (59/59)	4.561 (57/59)
▲ K2-MoVA 36B-A4B	1.349 (13.7/14)	1.088 (14/14)	1.959 (13/14)	3.828 (52.0/59)	2.923 (57/59)	5.038 (45/59)
● Claude Opus 4.8	1.955 (14.0/14)	1.344 (14/14)	2.503 (14/14)	4.150 (54.0/59)	3.143 (58/59)	5.293 (50/59)
▲ Kimi K2.6	2.098 (14.0/14)	0.963 (14/14)	3.223 (14/14)	3.493 (57.7/59)	2.538 (59/59)	4.287 (56/59)
▲ Qwen 3.6 35B	2.097 (12.7/14)	1.334 (14/14)	2.484 (10/14)	5.145 (54.0/59)	3.585 (57/59)	6.999 (51/59)
● GPT-5.5	4.410 (14.0/14)	3.098 (14/14)	6.159 (14/14)	5.887 (50.7/59)	3.791 (59/59)	7.335 (38/59)
▲ Gemma 4 26B	6.495 (14.0/14)	5.932 (14/14)	7.205 (14/14)	6.183 (52.0/59)	4.854 (56/59)	7.038 (47/59)
Set/graph $(1 - F_1)$						
● GPT-6 Astra	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
▲ GLM 5.3	0.006	**0.000**	0.019	0.067	**0.000**	0.100
● Gemini 3.1 Pro	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
▲ DeepSeek V4.1 Flash	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
● Claude Fable 5.1	0.074	0.074	0.074	**0.000**	**0.000**	**0.000**
▲ Kimi K3	0.010	**0.000**	0.031	0.033	**0.000**	0.100
▲ K2-MoVA 36B-A4B	0.027	**0.000**	0.080	0.067	**0.000**	0.200
● Claude Opus 4.8	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
▲ Kimi K2.6	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
▲ Qwen 3.6 35B	**0.000**	**0.000**	**0.000**	0.167	0.100	0.200
● GPT-5.5	**0.000**	**0.000**	**0.000**	0.133	**0.000**	0.300
▲ Gemma 4 26B	0.062	**0.000**	0.185	0.267	0.100	0.600

Continuous reliability. Restarts barely move point estimates. On the realistic exam the twelve Mean values span 0.120–0.164 and best-of-3 0.117–0.148, with Claude Fable 5.1 lowest on both exams; the agents refit the same data to the same number, consistent with the run-to-run SDs in App. A.11. What restarts do change is eligibility and intervals. Qwen 3.6 35B's missing submissions are transient — 22.3 of 25 realistic point tasks valid per run on average, all 25 in at least one of the three runs — so the tool-output exits behind its low valid-answer rate (App. A.18) are recoverable by restarting; GLM 5.3 and GPT-5.5 show the same pattern on the hard exam. Intervals are the unstable pool: best-of-3 and worst-of-3 differ by up to 3× (Kimi K2.6 0.96 vs. 3.22, GPT-6 Astra 1.04 vs. 2.48 on the realistic exam), and because these losses are uncapped a few blown intervals dominate the means — Astra's hard-exam interval loss is 41.8 for Mean and 121 for worst-of-3 against a best-of-3 of 1.5. That is the interval swing of App. A.11 seen from the retry side, and the reason the

leaderboard caps the interval term (Sec. 3.8). Set/graph losses are zero or near it for every model but Gemma 4 26B (worst-of-3 0.19 realistic, 0.60 hard).

Table 38: **Binary reliability by rung (%).** Mean is the three-run mean success rate on the fixed ground-truth pool; any-of-3/all-of-3 are the shares passed in at least one/all three runs. n is the ground-truth pool size; model columns follow realistic-exam CausalDSScore, with best per row in bold. Model markers: ● closed, ▲ open-weight.

Rung (n)	Metric	● GPT-6	▲ GLM	● Gemini	▲ DeepSeek	● Fable	▲ K3	▲ K2-MoVA	● Opus	▲ K2.6	▲ Qwen	● GPT-5.5	▲ Gemma
Realistic													
R1 (3)	Mean	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
	any-of-3	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
	all-of-3	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
R2 (14)	Mean	**100.0**	45.2	64.3	47.6	54.8	35.7	59.5	57.1	54.8	50.0	88.1	38.1
	any-of-3	**100.0**	57.1	78.6	71.4	71.4	64.3	78.6	71.4	78.6	71.4	**100.0**	64.3
	all-of-3	**100.0**	35.7	50.0	28.6	42.9	14.3	42.9	50.0	28.6	28.6	71.4	21.4
R3 (17)	Mean	92.2	92.2	88.2	90.2	78.4	86.3	80.4	**98.0**	92.2	80.4	76.5	70.6
	any-of-3	**100.0**	**100.0**	94.1	94.1	82.4	94.1	88.2	**100.0**	**100.0**	88.2	88.2	88.2
	all-of-3	88.2	82.4	76.5	88.2	70.6	76.5	64.7	**94.1**	82.4	70.6	64.7	47.1
Hard													
R2 (45)	Mean	**100.0**	74.1	60.0	88.1	84.4	52.6	74.1	67.4	79.3	56.3	90.4	48.9
	any-of-3	**100.0**	86.7	66.7	95.6	88.9	62.2	86.7	77.8	86.7	75.6	**100.0**	75.6
	all-of-3	**100.0**	64.4	53.3	77.8	80.0	44.4	62.2	57.8	71.1	33.3	77.8	24.4
R3 (52)	Mean	82.1	82.7	70.5	87.2	**87.8**	76.3	83.3	72.4	83.3	70.5	69.2	57.7
	any-of-3	90.4	**92.3**	78.8	90.4	**92.3**	86.5	90.4	78.8	90.4	78.8	82.7	75.0
	all-of-3	75.0	71.2	61.5	**84.6**	82.7	63.5	75.0	65.4	75.0	59.6	51.9	38.5

Table 39: **Binary reliability by observation condition (%).** Definitions and model order follow Tab. 38. Best per row in bold. Model markers: ● closed, ▲ open-weight.

Observation (n)	Metric	● GPT-6	▲ GLM	● Gemini	▲ DeepSeek	● Fable	▲ K3	▲ K2-MoVA	● Opus	▲ K2.6	▲ Qwen	● GPT-5.5	▲ Gemma
Realistic													
`clean` (18)	Mean	**92.6**	74.1	83.3	75.9	77.8	64.8	75.9	88.9	79.6	70.4	87.0	51.9
	any-of-3	**100.0**	88.9	94.4	88.9	83.3	83.3	88.9	94.4	94.4	77.8	88.9	72.2
	all-of-3	**88.9**	61.1	66.7	66.7	72.2	50.0	61.1	83.3	61.1	61.1	83.3	33.3
`proxy` (13)	Mean	**100.0**	76.9	82.1	69.2	59.0	69.2	66.7	79.5	76.9	79.5	79.5	76.9
	any-of-3	**100.0**	76.9	84.6	76.9	69.2	76.9	76.9	84.6	84.6	92.3	**100.0**	92.3
	all-of-3	**100.0**	76.9	76.9	61.5	46.2	61.5	53.8	76.9	69.2	61.5	61.5	61.5
`proxy_hard` (3)	Mean	**100.0**	55.6	44.4	77.8	77.8	66.7	88.9	44.4	66.7	22.2	77.8	33.3
	any-of-3	**100.0**	66.7	66.7	**100.0**	**100.0**	**100.0**	**100.0**	66.7	**100.0**	66.7	**100.0**	66.7
	all-of-3	**100.0**	33.3	33.3	66.7	66.7	33.3	66.7	33.3	33.3	0.0	33.3	0.0
Hard													
`clean` (23)	Mean	**100.0**	79.7	82.6	84.1	82.6	63.8	69.6	82.6	76.8	72.5	88.4	50.7
	any-of-3	**100.0**	91.3	91.3	95.7	87.0	69.6	78.3	91.3	82.6	91.3	**100.0**	78.3
	all-of-3	**100.0**	69.6	73.9	69.6	78.3	56.5	60.9	73.9	69.6	47.8	69.6	30.4
`proxy` (8)	Mean	**100.0**	75.0	50.0	95.8	87.5	62.5	87.5	79.2	87.5	79.2	**100.0**	62.5
	any-of-3	**100.0**	87.5	50.0	**100.0**	87.5	75.0	87.5	87.5	**100.0**	87.5	**100.0**	75.0
	all-of-3	**100.0**	62.5	50.0	87.5	87.5	50.0	87.5	62.5	75.0	75.0	**100.0**	50.0
`proxy_hard` (23)	Mean	**91.3**	79.7	55.1	85.5	88.4	63.8	79.7	65.2	75.4	60.9	75.4	55.1
	any-of-3	**95.7**	87.0	69.6	91.3	91.3	73.9	91.3	69.6	87.0	73.9	87.0	69.6
	all-of-3	**87.0**	69.6	39.1	82.6	82.6	52.2	69.6	60.9	65.2	47.8	56.5	39.1
`proxy_severe` (43)	Mean	82.9	78.3	65.1	**89.1**	86.8	67.4	82.2	64.3	86.0	58.1	72.1	52.7
	any-of-3	90.7	90.7	69.8	90.7	**93.0**	79.1	**93.0**	74.4	90.7	69.8	86.0	76.7
	all-of-3	76.7	67.4	60.5	**86.0**	81.4	55.8	69.8	55.8	79.1	41.9	58.1	25.6

Binary reliability by slice. Rung 1 is saturated: its three realistic binary tasks are solved by every model on every run. Rung 2 is where restarts add coverage — realistic Mean → any-of-3 takes DeepSeek from 47.6 to 71.4%, Kimi K2.6 from 54.8 to 78.6%, Kimi K3 from 35.7 to 64.3% — while all-of-3 stays at 14–43% for the open models against 100% for GPT-6 Astra; on the hard exam (n=45) all-of-3 runs from 24.4% (Gemma) to 100% (Astra). Rung 3 has high coverage everywhere (any-of-3 $\geq$ 75%) over a wide reliability floor (all-of-3 38.5–94.1%). Across observation conditions the any-of-3 minus all-of-3 gap is a model property more than a view property: on the hard exam's `proxy_severe` slice (n=43) it is under 12 points for DeepSeek, Fable, Gemini, and Kimi K2.6 and 51 points for Gemma, whose gap is about the same on `clean`.

Abstention is the clearest pattern (Tab. 40), and it is only measurable because non-identifiability is a property of the generated scene rather than a matter of opinion. Astra abstains correctly on every run of every non-identifiable task but one. The open models recover much of what they miss on a

Table 40: **Abstention reliability by task family (%).** Restricted to non-identifiable targets; definitions and model order follow Tab. 38. Family abbreviations follow Tab. 15; best per row in bold. Model markers: ● closed, ▲ open-weight.

Family (n)	Metric	● GPT-6	▲ GLM	● Gemini	▲ DeepSeek	● Fable	▲ K3	▲ K2-MoVA	● Opus	▲ K2.6	▲ Qwen	● GPT-5.5	▲ Gemma
Realistic													
	Mean	**100.0**	26.7	86.7	33.3	40.0	20.0	40.0	86.7	46.7	60.0	86.7	13.3
Ident (5)	any-of-3	**100.0**	40.0	**100.0**	80.0	60.0	60.0	60.0	**100.0**	80.0	**100.0**	**100.0**	40.0
	all-of-3	**100.0**	20.0	60.0	0.0	20.0	0.0	20.0	80.0	20.0	20.0	80.0	0.0
	Mean	**100.0**	55.6	11.1	77.8	44.4	44.4	88.9	11.1	55.6	11.1	88.9	33.3
Effect (3)	any-of-3	**100.0**	66.7	33.3	**100.0**	66.7	**100.0**	**100.0**	33.3	**100.0**	33.3	**100.0**	66.7
	all-of-3	**100.0**	33.3	0.0	66.7	33.3	0.0	66.7	0.0	0.0	0.0	66.7	0.0
	Mean	**100.0**	11.1	44.4	0.0	44.4	0.0	22.2	11.1	22.2	22.2	77.8	22.2
Bias (3)	any-of-3	**100.0**	33.3	66.7	0.0	66.7	0.0	66.7	33.3	33.3	33.3	**100.0**	66.7
	all-of-3	**100.0**	0.0	33.3	0.0	33.3	0.0	0.0	0.0	0.0	0.0	33.3	0.0
	Mean	77.8	77.8	**88.9**	66.7	66.7	66.7	55.6	**88.9**	77.8	66.7	66.7	55.6
CF-ID (3)	any-of-3	**100.0**	**100.0**	**100.0**	66.7	66.7	66.7	66.7	**100.0**	**100.0**	66.7	66.7	66.7
	all-of-3	**66.7**	**66.7**	**66.7**	**66.7**	**66.7**	**66.7**	33.3	**66.7**	**66.7**	**66.7**	**66.7**	33.3
	Mean	**100.0**	83.3	50.0	**100.0**	83.3	66.7	83.3	**100.0**	**100.0**	50.0	50.0	50.0
CFE (2)	any-of-3	**100.0**	**100.0**	50.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
	all-of-3	**100.0**	50.0	50.0	**100.0**	50.0	50.0	50.0	**100.0**	**100.0**	0.0	0.0	0.0
Hard													
	Mean	**100.0**	83.3	79.2	83.3	75.0	58.3	75.0	87.5	75.0	79.2	95.8	41.7
Ident (8)	any-of-3	**100.0**	87.5	87.5	**100.0**	87.5	75.0	87.5	**100.0**	87.5	87.5	**100.0**	62.5
	all-of-3	**100.0**	75.0	62.5	75.0	62.5	37.5	62.5	75.0	62.5	62.5	87.5	12.5
	Mean	**100.0**	68.2	36.4	92.4	86.4	40.9	78.8	51.5	81.8	39.4	92.4	47.0
Effect (22)	any-of-3	**100.0**	81.8	40.9	95.5	90.9	54.5	95.5	63.6	90.9	59.1	**100.0**	72.7
	all-of-3	**100.0**	59.1	31.8	86.4	81.8	31.8	63.6	40.9	72.7	18.2	86.4	18.2
	Mean	**100.0**	50.0	61.1	61.1	66.7	16.7	22.2	50.0	44.4	38.9	72.2	27.8
Bias (6)	any-of-3	**100.0**	83.3	83.3	83.3	66.7	16.7	33.3	66.7	50.0	83.3	**100.0**	83.3
	all-of-3	**100.0**	33.3	50.0	16.7	66.7	16.7	16.7	33.3	33.3	0.0	33.3	0.0
	Mean	**100.0**	88.3	50.0	**100.0**	93.3	68.3	90.0	75.0	91.7	75.0	83.3	46.7
CFE (20)	any-of-3	**100.0**	**100.0**	65.0	**100.0**	95.0	90.0	**100.0**	85.0	**100.0**	90.0	**100.0**	75.0
	all-of-3	**100.0**	75.0	35.0	**100.0**	90.0	45.0	80.0	60.0	85.0	55.0	60.0	20.0

typical run — realistic `identification` abstention has any-of-3 of 100% for Qwen and 80% for DeepSeek and Kimi K2.6 — yet their all-of-3 on the same pool is 0–20%: they can recognize a non-identifiable scene, not reliably. The forbidden-controls `bias_diagnostic` family is the hardest abstention call under all three summaries on both exams. The realistic per-family pools are small (n=2–5), so those rows are indicative; the hard-exam pools (n=6–22) carry the claim.

A.14 Cost and efficiency

Fig. 15 and Tab. 41 unpack the mean costs in the leaderboards (Tabs. 1 and 2), showing per-attempt calls and total tokens through termination, regardless of outcome. Each model contributes 300 realistic-exam and 600 hard-exam attempts across three replications. Token accounting follows Sec. 3.8. Cumulative token usage can exceed a model's context-window size even when every individual request fits within it. Quantiles use the empirical inverse CDF. We wish to shed light on how the benchmark's fifth axis, tool use and coding, is reflected in these metrics: how each model spends, where the spending concentrates, and whether it buys anything.

Interaction style. The call number distributions sort the field into three styles. GPT-6 Astra and GPT-5.5 answer with a single call at the median on both exams: one script, one submission. Claude Fable 5.1 and Claude Opus 4.8 run short loops of bounded length, with 95th-percentile call counts of 8 on the realistic exam and 12–18 on the hard exam. The remaining eight models iterate open-endedly: their 95th percentiles run from 14 to 57 calls on the realistic exam and from 19 to 69 on the hard exam, and Gemini 3.1 Pro, Kimi K2.6, and Qwen 3.6 35B need 19–24 calls for a typical hard-exam attempt. The mean-call thresholds of Tab. 41 draw the same line, near one-shot (N) for the first two styles and iterative (I) for the third. Because the harness re-sends the growing context on every call, cumulative tokens follow calls: the median across models of the within-model Spearman correlation between an attempt's calls and its total tokens is 0.95 on the realistic exam and 0.98 on the hard exam, so the call count is the primary quantity and the tokens are a secondary, tail-dominated one. For the near one-shot models the mean token cost lies well above the median attempt, twelvefold for GPT-5.5 on the realistic exam, so their Tok./task entries on the leaderboards are set by a minority of long attempts (Fig. 15).

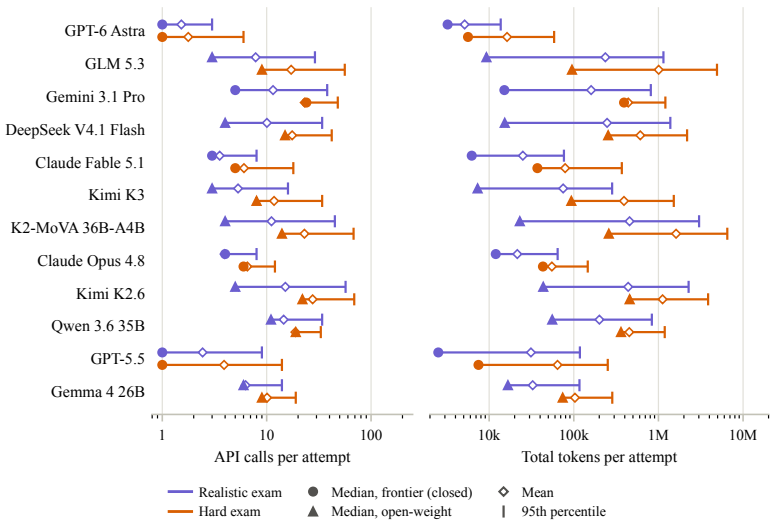


Figure 15: **Per-attempt cost by model.** Realistic exam (violet) and hard exam (orange): API calls (left) and total prompt+completion tokens (right), log scale. Each line runs from the median attempt (circle — closed model, triangle — open-weight model) to the 95th percentile (tick); the hollow diamond is the mean, reported on the leaderboards. Rows follow the realistic-exam leaderboard (mean CausalDSScore), as in Tab. 41.

Table 41: **Per-attempt cost quantiles.** Mean, median (med.), and p95 of API calls and total prompt+completion tokens (thousands) across three replications; rows follow the realistic-exam leaderboard (mean CausalDSScore), with all attempts pooled in the last row. N/I denote near one-shot/iterative (< 5 mean calls on the realistic exam and < 10 on the hard exam for N). Column minima in bold. Model markers: ● closed, ▲ open-weight.

		Realistic exam (100 tasks, 300 attempts per model)						Hard exam (200 tasks, 600 attempts per model)					
		Calls			Tokens (k)			Calls			Tokens (k)		
Model	Strategy	mean	med.	p95	mean	med.	p95	mean	med.	p95	mean	med.	p95
● GPT-6 Astra	N	**1.5**	**1**	**3**	**5.1**	3.2	**14**	**1.8**	**1**	**6**	**16**	**5.6**	**59**
▲ GLM 5.3	I	7.8	3	29	236	9.3	1,146	17.2	9	56	1,007	95	4,919
● Gemini 3.1 Pro	I	11.5	5	38	160	15	813	23.2	24	48	440	392	1,208
▲ DeepSeek V4.1 Flash	I	10.1	4	34	247	15	1,382	17.5	15	42	610	255	2,180
● Claude Fable 5.1	N	3.5	3	8	25	6.2	76	6.1	5	18	79	37	369
▲ Kimi K3	I	5.3	3	16	75	7.3	284	11.8	8	34	392	93	1,519
▲ K2-MoVA 36B-A4B	I	11.1	4	45	456	23	3,021	23.0	14	68	1,616	259	6,510
● Claude Opus 4.8	N	4.0	4	8	22	12	64	6.5	6	12	55	43	146
▲ Kimi K2.6	I	15.1	5	57	439	44	2,272	27.6	22	69	1,119	457	3,862
▲ Qwen 3.6 35B	I	14.5	11	34	201	56	836	18.9	19	33	452	361	1,187
● GPT-5.5	N	2.4	**1**	9	31	**2.5**	118	3.9	**1**	14	64	7.5	252
▲ Gemma 4 26B	I	6.3	6	14	33	17	117	10.1	9	19	103	74	285
All models	—	7.8	4	31	161	12	798	14.0	8	45	496	71	2,304

Where the cost goes. The hard exam raises typical cost more than extreme cost: pooled over models, the median attempt grows sixfold, from 12k to 71k tokens, and the 95th percentile threefold (Tab. 41). The styles absorb this differently. The iterative models lengthen their *typical* attempt, Gemini 3.1 Pro's median moving from 5 to 24 calls and Kimi K2.6's from 5 to 22, whereas GPT-6 Astra and GPT-5.5 keep the one-call median and grow only in the tail, their 95th-percentile calls moving from 3 to 6 and from 9 to 14. Tab. 42 locates the costly attempts, those above their model's own 95th percentile. On both exams they are interval and numerical-point targets under the harder observation variants. The absence of costly `clean` attempts on the hard exam coincides with that slice holding only identification and bias-diagnostic tasks, i.e., symbolic tasks. Cost therefore concentrates where the data-science and uncertainty-quantification axes meet, on numeric estimation under measurement error.

What spending buys. Between models, tokens buy score in the Pareto sense of Fig. 3: GLM 5.3 attains the best hard-exam CausalDSScore at 1,007k tokens per attempt, over sixty times GPT-6 Astra's. Within a model they do not. The trajectory-level taxonomy shows that *the failing attempts are the long ones*: among the leading models, data-science fails run the longest tool loops and reasoning fails the next longest (App. A.11.1), and the first-commitment census finds GPT-6 Astra and GPT-5.5 committing to a plan before any tool result more often than the other models, i.e. in a near one-shot fashion. The abstention decision shows the same asymmetry in cost. On the hard exam an over-claim, a committed answer on a non-identifiable target, costs a median 116k tokens against 26k for a correct abstention and is costly sixteen times as often (6.5% against 0.4% of attempts; Tab. 42). The hard exam's non-identifiable structures, `iv` and `double_nc`, carry mostly effect and counterfactual-effect tasks (Tab. 16), so the over-claim is an estimation pipeline run to completion on a quantity the graph does not identify; on the realistic exam, where most non-identifiable items are symbolic (Tab. 15), the two outcomes have nearly the same median cost, though over-claims are still costly five times as often. The routes into the over-claim are traced in App. A.11.2, where GPT-5.5 reaches the abstention in its first and only step. Read together with the abstention axis, the iterative open models pay for a missed abstention twice, in the answer and in the tokens, whereas the near one-shot models miss cheaply, and their misses fall mostly on the uncertainty-quantification axis (Sec. 4; App. A.11).

Comparability across benchmarks. Call counts are not commensurate across harnesses: a single bash call here can run a complete analysis script, so the quantity that transfers is the interaction style, single-pass script submission versus multi-round iteration. The split is one instance of the model- and benchmark-specific tool-use patterns reported elsewhere: RoadmapBench and AutoLab report opposite exploration failures for GPT-5.4 (Xu et al., 2026a;b), GPT-5 uses tools heavily on MCP-Bench (Wang et al., 2025), and more tool use improves long-horizon planning on DeepPlanning (Zhang et al., 2026). AutoLab also finds that the GPT family's one-pass style persists under a prompt that discourages it (App. A.18).

Table 42: **Task composition of costly attempts.** Pooled over models and replications. Costly is the percentage of the category's attempts whose total tokens exceed their model's own within-exam 95th percentile. — marks a category absent from the exam.

	Share (%)		Median (k)		Costly (%)	
Category	Real.	Hard	Real.	Hard	Real.	Hard
Interval	14.0	29.5	70	243	13.5	9.8
Numerical point	25.0	17.0	47	256	9.0	7.7
Non-identifiable	16.0	28.0	14	34	3.5	2.1
abstained	8.8	20.1	13	26	1.3	0.4
over-claimed	6.8	7.6	16	116	6.5	6.5
Graph/set	27.0	5.0	3.7	8.7	0.0	0.0
Binary content	18.0	20.5	8.8	42	1.7	1.0
`clean`	48.0	16.5	9.5	8.3	1.7	0.0
`proxy`	32.0	12.0	7.5	100	2.3	4.7
`proxy_hard`	20.0	27.5	93	100	17.2	5.8
`proxy_severe`	—	44.0	—	124	—	6.5

A.15 Robustness to the CauseNet-seeded verbalization

CauseNet seeding (Sec. 3.2) injects real-world causal pairs into the story and variable names. One concern is whether this semantic layer leaks the answer, letting a model solve a scene from a familiar causal template in the prose rather than from the story-implied graph and data — the "causal parrot" failure mode. The more general concern is stability: scores should be a function of the formal problem, not of the particular story drawn for it. Because every story is audited to be faithful to its graph (cf. Sec. A.4), a model whose answers move under a story swap is failing on its own terms — exactly the behavior the benchmark is meant to expose — but a persistent per-scene movement across models could indicate dependence on the story draw. We probe both with a *matched verbalization-swap* ablation that holds the formal problem fixed and varies only the seeded verbal grounding.

Construction. From the main scene pool we form ten *bundles*. Within a bundle the motif, the raw node order, and the per-node variable-type signature are identical, and one representative member's clean data and private ground truth are mapped back onto every member's variable names, so all members share the same numbers, the same scoring target, and the same private truth. The members thus differ only in their CauseNet-seeded story and variable names, giving four verbalizations of a single fixed formal problem. Each bundle contributes one task, for 40 exam items spanning nine distinct motifs and both symbolic and data-backed families. We evaluate two open-weight agents, Kimi K2.6 and Qwen 3.6 35B with one attempt per item each.

Diagnostic loss. To place heterogeneous task outputs on one scale we use a capped "diagnostic loss" (higher is worse): $1 -$ score for boolean, F_1, and abstention tasks; normalized relative error for scalar estimates, normalized RMSE for prediction, and normalized interval score for intervals, each capped at 1; and 1 for a missing or unparseable answer or a runtime failure, including context overflow. Binary prediction uses $\sqrt{\text{Brier}}/(1 + \text{sd}(Y_{\text{test}}))$ (Eq. 1). The quantity of interest is the *within-bundle range* of this loss: how much the outcome moves when only the verbalization changes while the formal problem is held fixed.

Table 43: **Matched verbalization-swap ablation.** Capped diagnostic loss of each model on the four verbalizations s_1–s_4 of a bundle's fixed formal problem (ordered by source scene ID) and the within-bundle range (max−min). Slice is the bundle task's `rung:family`, abbreviated as in Tab. 15. *abst* marks an abstention on an identifiable target, *miss* a missing or unparseable answer or a runtime failure, and *id* an identifiable call on a non-identifiable target; each scores loss 1. The lower block counts bundles per model by range.

			▲ Kimi K2.6					▲ Qwen 3.6 35B				
Bundle	Slice	Motif	s_1	s_2	s_3	s_4	Range	s_1	s_2	s_3	s_4	Range
1	R2:Effect	arrowhead	0.009	0.009	0.009	0.009	0.000	0.009	0.009	0.006	0.006	0.003
2	R2:Sketch	chain	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	R1:Assoc	collider	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	R2:Ident	confounding	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	R1:Pred	diamond	0.167	0.168	0.167	0.173	0.006	0.171	0.279	0.172	0.174	0.107
6	R2:Bias	diamondcut	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	R3:CFE	frontdoor	0.109	*abst*	*abst*	0.109	0.891	0.386	*abst*	*abst*	*miss*	0.614
8	R3:CF-ID	double_nc	0.000	0.000	0.000	0.000	0.000	0.000	*id*	0.000	0.000	1.000
9	R1:Collider	collider	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
10	R2:Effect	mediation	0.002	0.002	0.002	0.002	0.000	0.001	0.002	0.002	0.002	0.002
Bundles with nontrivial spread (> 0.001)							2/10					5/10
Bundles with large spread (> 0.05)							1/10					3/10

The stronger agent is more stable. Table 43 shows that most bundles are flat: every discrete-output bundle outside Rung 3 is invariant for both models, and the ordinary Rung-1/2 numeric bundles move only marginally, determined by the particular numeric approach the model picks on the scene. Kimi (the stronger agent, cf. Tab. 1) is largely invariant, with nontrivial spread on $2/10$ bundles and a single large (> 0.05) swing. Qwen is materially more sensitive: $5/10$ bundles move and three swing widely. The instability concentrates in the two Rung-3 bundles (counterfactual effect and identification), where the ranges reflect discrete flips (commit vs. abstain, identifiable vs. not) rather than numeric dispersion, and, for Qwen, one binary prediction bundle where the ranking of test cases is preserved across stories (AUC 0.73–0.75, with the discrepant story in fact the highest) while the emitted probabilities become poorly calibrated (Brier 0.128 against 0.048–0.050 on the other three verbalizations): the story changed how confident the predictions are, not which cases the model scores higher.

Shared false abstentions. The movement that does exist is attributable to the story layer, not to sampling variation. On the stable numeric bundles, Kimi's estimates are identical across all four verbalizations to ten or more significant digits and Qwen's vary by at most 3×10^{-3} in relative error: given the same numbers, the agents run essentially deterministic pipelines. In bundle 7, by contrast, both models abstain on the same two stories and attempt the estimate on the other two (Kimi with numerically identical estimates; Qwen with one estimate and one context-overflow failure) — two agents agreeing on *which* stories trigger an unwarranted abstention is not single-run stochasticity.

This is precisely the persistent cross-model movement flagged above as the possible signature of story-draw dependence, and the only place the ablation finds it. We nevertheless argue that the coordinated behavior is mostly model-driven: it is a residue of the "causal parrot" failure mode (also cf. the next paragraph), and a competent agent would score identically on all four members. That this happens less often overall for the model otherwise benchmarked as the stronger one speaks in favor of this interpretation and against the "null hypothesis" that the verbalization seeding makes the benchmark unstable. What remains on the benchmark side is limited: for susceptible models, the story a scene is told with can make it harder or easier. This is thus a latent factor in the *difficulty*, not an indictment of the *stability* of the benchmark.

A concrete semantic-lure failure. The clearest case is bundle 8 (`double_nc` motif, symbolic counterfactual identification of the ETT), where the estimand is *non-identifiable*. Kimi answers non-identifiable throughout. Qwen also answers non-identifiable on the three stories seeded with the hypothermia–unconsciousness, illness–disability, and events–arrest pairs, but flips to *identifiable* on the fourth (`bundle008__scene_000409`; source scene `scene_000409`; the other ablation members are `bundle008__scene_000104`, `bundle008__scene_000527`, and `bundle008__scene_000684`), whose prose describes rainfall-gauge and satellite-precipitation measurements of a latent precipitation confounder:

> "Because researchers cannot directly observe the true annual precipitation totals across the entire heterogeneous mountainscape, they rely on two complementary assessments: the Rainfall Gauge Measurement collected at fixed field stations, and the Satellite Precipitation Estimate derived from orbital remote sensing algorithms. [...] The true precipitation levels driving these fire dynamics remain latent variables inferred through the gauge and satellite observations rather than measured continuously across all terrain."

From those measurement descendants Qwen constructs a proxy/backdoor identification argument. The lure is a sophisticated one: the two-proxy prose mirrors the double-negative-control setting of proximal causal inference (Miao et al., 2018; Tchetgen et al., 2024), and the conditions the model asserts have exactly that shape. Even in that framework, however, point identification rests on completeness assumptions that are not encoded in the graph and that the model neither states nor checks. Under the benchmark's graph-level identifiability policy, observed descendants of a latent confounder do not license identification, so this is a template imported from the story without the assumptions it needs, rather than a property of the graph — exactly the behavior the fully synthetic construction is meant to expose.

Pair commonness does not explain the variation. We recovered the CauseNet support count and salience percentile of each scene's seeded pair and regressed the diagnostic loss on commonness with bundle fixed effects. The within-bundle slope is negative for both models (Kimi -0.05, bootstrap 95% CI $[-0.18, 0.00]$; Qwen -0.45, CI $[-1.08, 0.00]$), and the within-bundle Spearman correlation is $+0.05$ (Kimi) and -0.27 (Qwen). The sign is consistent with more common pairs being marginally easier, but the intervals include zero, and two bundles carry no within-bundle commonness variation. We therefore report commonness as an inconclusive moderator: the seeded verbal grounding affects the weaker agent, but that effect is not sufficiently explained by CauseNet pair commonness in this diagnostic.

Takeaway. The two results read as the two halves of the motivating concern. The stronger agent's near-invariance —exact agreement of its answers across seeded pairs with different real-world scenarios—indicates that the score measures the formal problem rather than the seeded verbalization: the benchmark is robust to CauseNet seeding for an agent that works from the released structure and data. The weaker agent's movement is shown likely to be causal-parrot susceptibility itself: the example above indicates that the model inferred semantics that the graph and data did not support.

A.16 Observation-layer hardness under matched scenes

The observation layer (Sec. 3.4) is designed to change how hard an identified functional is to estimate from the released files while preserving the conceptual SCM, the target estimand, and the identifiability label. Here, we present a matched ablation in which the same conceptual scene/task is evaluated under all three released observation views.

Construction. From the main scene pool we select ten matched scene/task pairs whose source scene releases all three observation variants, spanning the data-backed continuous families. Only identifiable targets enter, so abstention is never the correct response. The $10 \times 3 = 30$ instances form a single randomized exam. The observation model label used is hidden from the agent, as are previously submitted answers. We evaluate the two open-weight agents, Kimi K2.6 and Qwen 3.6 35B (the pair also used in App. A.15) with one run each. The paired deltas are descriptive contrasts on these ten selected pairs, not estimates of a population-level effect of the observation layer.

Diagnostic loss. For evaluation purposes, we define a (capped) *diagnostic loss* (higher is worse): the normalized relative error (point/scalar outputs) or normalized interval score (intervals), capped at 1; a missing, unparseable, or runtime-failed answer scores 1; and — because every selected target is identifiable — an abstention also scores 1, as it's always incorrect. This convention is specific to this ablation and differs from the main leaderboard grading, where abstention is scored as its own metric and the continuous pools contain only valid numeric answers. The paired contrasts are: $\Delta_{\text{proxy}} = L_{\text{proxy}} - L_{\text{clean}}$ and $\Delta_{\text{hard}} = L_{\text{proxy_hard}} - L_{\text{clean}}$ per pair, with paired-bootstrap 95% intervals over the ten pairs. The survivor-only view — deltas over pairs with *valid* answers in both arms — is reported alongside.

Table 44: **Matched observation-layer ablation.** (a) Mean paired change in capped diagnostic loss relative to `clean`, with paired-bootstrap 95% CI; Worse counts pairs with $\Delta > 0$. Survivor-only means condition on valid numeric answers in both arms, with their contributing count n. (b) Valid numeric answers, false abstentions on the six abstention-eligible tasks, context-overflow exits, mean API calls per task, and median capped loss among valid numeric answers.

(a) Paired contrasts against `clean`.

Model	View	Mean Δ	95% CI	Worse	Survivor-only (n)
▲ Kimi K2.6	`proxy`	+0.003	[−0.275, +0.255]	7/10	+0.003 (10)
	`proxy_hard`	+0.200	[+0.054, +0.380]	8/10	+0.147 (8)
▲ Qwen 3.6 35B	`proxy`	+0.304	[+0.014, +0.602]	6/10	+0.019 (3)
	`proxy_hard`	+0.332	[+0.070, +0.629]	6/10	+0.111 (3)

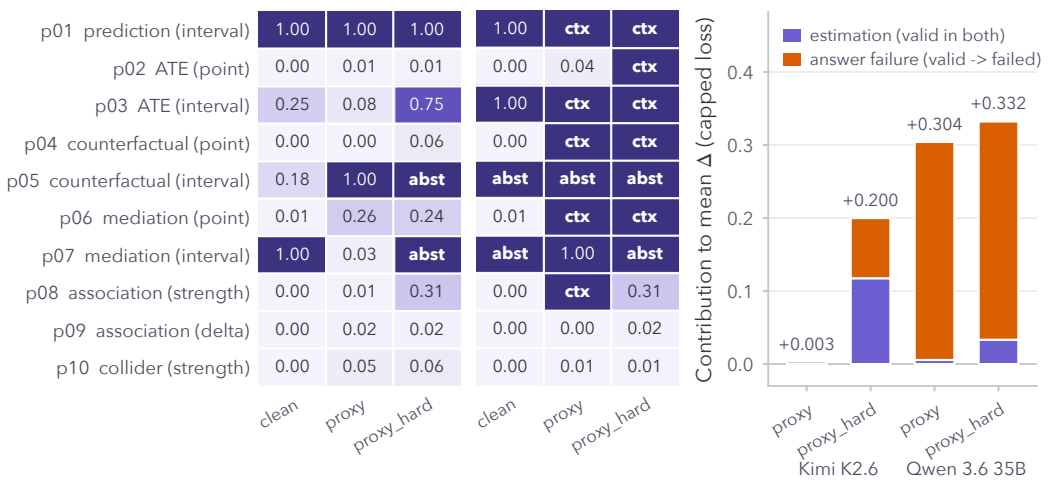
(b) Per-view behavior.

Model	View	Valid/10	Abst./6	Ctx./10	Calls/task	Med. valid loss
▲ Kimi K2.6	`clean`	10	0	0	16.5	0.005
	`proxy`	10	0	0	26.7	0.039
	`proxy_hard`	8	2	0	26.0	0.149
▲ Qwen 3.6 35B	`clean`	8	2	0	13.3	0.003
	`proxy`	4	1	5	19.6	0.028
	`proxy_hard`	3	2	5	19.7	0.016

The matched contrast confirms the penalty under harder observation. Table 44a shows both models scoring worse under `proxy_hard` (per-instance results in Fig. 16), with paired-bootstrap intervals excluding zero (Kimi +0.20, 8/10 pairs worse; Qwen +0.33). The middle tier separates the models: Qwen already degrades at `proxy` (+0.304, CI [+0.014, +0.602]), while Kimi's `proxy` contrast is flat on average (+0.003 against a +0.01 median, the mean pulled down by one interval pair whose `clean` answer was already at the loss cap while its `proxy` answer happened to score well). Median deltas are moderate throughout (≤ 0.06).

Hardness acts through two separate effects. The first effect is statistical: among instances that produce a valid answer, estimation degrades. The median survivor loss rises $0.005 \rightarrow 0.149$ (Kimi) and $0.003 \rightarrow 0.016$ (Qwen) from `clean` to `proxy_hard` (Tab. 44b), and Kimi's survivor-only Δ_{hard} is +0.15 with 7/8 pairs worse.

The second effect is agentic: the harder views increasingly prevent a valid answer at all. Over `clean`/`proxy`/`proxy_hard`, valid answers fall 10/10/8 (Kimi) and 8/4/3 (Qwen). The underlying driver is workload: both models work harder as the view hardens (Kimi 16.5 $\rightarrow \sim$26 calls



(a) Kimi K2.6. (b) Qwen 3.6 35B. (c) Contributions to the mean paired contrast.

Figure 16: **Matched observation-layer ablation.** (a, b) Capped diagnostic loss for all 60 instances. Shading and printed values give the loss; *abst* marks false abstention and *ctx* context-window failure. (c) Exact decomposition of the mean paired loss change relative to `clean`.

per task, Qwen 13.3 $\rightarrow \sim$20), and the extra work ends differently for the two agents. For Qwen, whose deployment serves a 32k-token window, the longer exploration is precisely what exhausts the context: ten of its fifteen failures are context-window overflows (five in each proxy tier), and the remaining five are false abstentions. Kimi never exhausts its context; instead it starts abstaining under `proxy_hard` — plausibly the added measurement uncertainty tipping it into judging an identifiable target unanswerable (its abstention cases are examined below). Failing to finish is not incidental noise but a second, separate signal that the observation layer stresses the whole agentic workflow, not only estimator accuracy. The decomposition in Fig. 16c separates the two contributions to the capped mean — for Qwen the failing pairs account for $+0.30$ of the $+0.33$, while Kimi's splits into $+0.12$ from surviving estimates and $+0.08$ from abstention onset — though these shares reflect the failure-scores-1 convention as much as the data and should not be read as a ranking of the two effects.

Abstention as a hardness-triggered exit. All seven abstentions in the ablation fall on the same two pairs (twelve instances: two models under three variants each) — an ETT interval and an NIE interval, both on frontdoor-motif scenes whose targets the *ID*/IDC**-based algorithm certifies as identifiable. The submitted explanations argue from textbook sequential-ignorability and backdoor conditions: the models locate the latent confounder named in the story and declare the estimand non-identifiable, missing the frontdoor-type identification. Kimi's abstentions are hardness-triggered exits rather than up-front identification calls: it commits to estimates on both pairs under `clean` and `proxy`, and its `proxy_hard` trajectories show it fitting proxy-reconstruction models and implementing the estimators before submitting a null answer whose graph-level rationale contradicts its own attempts, mirroring the broader trend discussed in App. A.11.1.

Interval calibration and caveats. Kimi's per-row prediction intervals stay near their nominal 90% coverage across all three variants (88.2%–89.95%). Qwen achieves 89.6% on `clean` but produces no valid prediction intervals under either proxy view. Among submitted causal-effect intervals with nominal 95% coverage, neither model covers any target on `clean`; Kimi nevertheless covers two of three targets under `proxy`. Causal-interval undercoverage is therefore already present without the observation layer (cf. App. A.11.4). The ablation is deliberately small, supporting mostly a descriptive observation.

A.17 Limitations

Synthetic realism. Every scene is generated. Story fidelity to the hidden graph is audited by LLMs and confirmed post hoc by near-perfect structure recovery (App. A.4), and exam composition is grounded in real-study corpora (App. A.7), but no human study has rated the stories or data for domain plausibility, and the mechanism registries (App. A.5) span only the functional families we chose to implement. A model that performs well here may still fail on a real study, whose data, documentation, and confounding are not drawn from a known SCM; the benchmark measures causal data-science competence on problems with a certified answer, which is the price of having one.

Identifiability policy. Identifiability is decided at the graph level by `dowhy` and the `y0` *ID*/IDC** algorithms on the conceptual DAG. Partial identification, sensitivity analysis, and proximal or negative-control strategies are not credited: an agent that returns bounds or a proxy-based estimate on a non-identifiable target is graded as a missed abstention. This is a deliberate policy (App. A.9, App. A.6), and it makes the abstention axis a test of whether the agent adopts the benchmark's identification standard rather than of every defensible analysis.

Exam size and slices. Each exam is one draw of 100 or 200 tasks. The finite-exam intervals of App. A.12 are wide on the realistic exam — ranks two through seven are not separable — and many per-slice cells in Apps. A.10 and A.13 hold fewer than five tasks. The observation-variant slices are marginal views of exams whose other axes co-vary, so the per-variant comparisons in Tab. 18 and the additive adjustment of App. A.8 are descriptive; the matched-scene evidence (App. A.16) rests on ten pairs, two open-weight models, and one run each.

Scoring choices. CausalDSScore weights its three pools by task count and caps each interval term at $c{=}10$; both are design decisions. App. A.12 shows the leaderboard extremes are stable under the cap and the pool weights but that the middle reshuffles if the interval term is summarized by a median instead of the capped mean, and Composite Rank orders the hard-exam top four differently from CausalDSScore (Sec. 4). The permissive band for sign tasks and the null-equivalent labels accepted as abstention (App. A.9) are likewise conventions.

Agents and harness. All models run in one scaffold: a bash sandbox with a fixed task prompt, one reasoning setting per model, and call and cost limits (App. A.18). Tool-call and token counts are therefore properties of this harness and are not comparable to other benchmarks' (App. A.14). Provider-side refusals are graded as failures, one model's tool-output handling cost it valid submissions, and closed models were reached through a routing provider at a fixed point in time; the twelve models are a snapshot, and the three replications and the three-model, $N{=}10$ mini-exam bound how much we can say about reliability.

A.18 Technical details

Containerization and package pinning. Evaluation runs use a Singularity image built from Docker whose Python 3.11 stack is pinned to exact versions to avoid mistakes due to API changes: `pandas==3.0.2`, `numpy==2.4.4`, `scipy==1.17.1`, `scikit-learn==1.8.0`, `statsmodels==0.14.6`, `pyarrow==24.0.0`, `matplotlib==3.10.9`, `seaborn==0.13.2`, `networkx==3.6.1`, `xgboost==3.2.0`.

The benchmark task prompts list the versions so the agent does not waste budget probing the environment for the right API.

Execution limits. Per-task limits are 100 model queries for the realistic exam and targeted runs, and 150 for the hard exam, with tracked cost caps of $10–15. Each single shell command has a 120-second timeout. An attempt that reaches a limit is graded on whatever answer files it wrote and stays in the cost summaries of App. A.14; twelve did so, one on the realistic exam (K2-MoVA 36B-A4B, call limit) and eleven on the hard exam (K2-MoVA 36B-A4B six times at the call limit, Claude Fable 5.1 four times at the cost cap, Gemma 4 26B once at the call limit).

Workspace layout and answer validation. Each task starts a fresh model conversation; the container is reused within a replication, with scenes mounted read-only. Each command runs as a

separate shell process and state persists through files. Singularity is launched with `--no-home` and `HOME=/workspace` so that submitted answer files land in a path the grader will actually read. After each task the runner re-checks that the expected answer file exists under `/workspace/answers` and is minimally parseable as JSON or CSV; failures are recorded as `MissingAnswerAfterSubmit` rather than counting silently as submitted. When present, top-level scratch under `/workspace` is archived after each task to `<output_dir>/workspace_scratch/<scene_id>_<task_id>.tar.gz`, then removed before the next task starts. The given scenes (`scenes/`), the model's answer directory (`answers/`), and `INSTRUCTIONS.md` persist as the benchmark state.

Provider routing and inference settings. Five models are routed through OpenRouter with `provider.only` pinned: Claude Opus 4.8 and Claude Fable 5.1 to `anthropic`, Gemini 3.1 Pro to `google-ai-studio`, and GPT-5.5 and GPT-6 Astra to `openai`. All five use `allow_fallbacks=false` and request high reasoning effort; Opus additionally uses adaptive thinking in accordance with the benchmark's policy that runs models on high reasoning (or equivalent) and uses the defaults for the rest (high reasoning is the default for most models anyway). Seven models use self-hosted OpenAI-compatible endpoints: Qwen 3.6 35B, Kimi K2.6, Gemma 4 26B, GLM 5.3, DeepSeek V4.1 Flash, Kimi K3, and K2-MoVA 36B-A4B. GLM, DeepSeek, and Kimi K3 receive `reasoning_effort=high` through the extra request body. K2-MoVA receives high effort through its chat-template settings, with temperature 1.0 and top-$p = 0.95$. Qwen and Kimi K2.6 use thinking, while Gemma uses its non-thinking configuration, as that is its default. Token usage and tool-call counts are reported for all models to make inference-compute differences visible.

Retries and failure accounting. Model-query retries use exponential waits bounded between 4 and 60 seconds. Terminal endpoint failures, including service outages, are excluded from grading. Model-side failures, including refusals, malformed responses, and context overflow, remain graded.

Tool-output handling. One behavior specific to Qwen 3.6 35B partly explains its incomplete continuous submissions (Tab. 1): it manages tool output poorly, frequently emitting very long outputs — for instance, printing the entire provided dataset into the conversation instead of using `head`/`tail` or computing aggregates over it. The harness does not progressively trim accumulated context; when a *single* command's output exceeds a length threshold it instead shows only that command's head and tail, together with an explicit warning to use `head`/`tail`/`grep` or redirect to a file. Qwen repeatedly ignored this and lost the numeric values it needed on the hardest continuous estimands. We re-ran the exam with it at several output-budget thresholds (1k, 3.2k, and the default 10k characters used for the other models) and the behavior persisted, so we count these lost submissions as a genuine tool-use failure rather than a harness artifact. On the near one-shot submission style, Xu et al. (2026b) find that even after explicitly discouraging first-pass submissions via the prompt when using the GPT-family with `mini-swe-agent`, the behavior nonetheless persists, indicating that it is not purely harness-induced.

A.19 Prompt templates

A.19.1 Generation pipeline prompt templates

The generation pipeline (Sec. 3.2) uses separate LLM sessions for variable mapping, feasibility checking, consistency auditing, story generation, and story verification. Below we reproduce the core system and user prompt templates as used at the current benchmark version. The variable-mapping templates live in `causalds/schemas.py`; feasibility-check, consistency-audit, story-generation, and story-verification templates live in `causalds/pre_auditor.py`, `causalds/mapping_audit.py`, `causalds/verbalization_story.py`, and `causalds/verbalization_verify.py`, respectively. Runtime template variables (e.g., `{format_description}`, `{serialized_graph}`) are filled with the graph serialization and scene-specific details at generation time.

Variable mapping system prompt. Instructs the mapper LLM to choose a coherent domain and assign human-friendly variable names to abstract node identifiers. The prompt enforces that latent/unobserved nodes are correctly marked based on the graph's `observed_nodes` list. Mapping runs in unstructured (prompt) mode: `{output_format_block}` inlines the requested output format — JSON for the dataset presented — as a skeleton (`proposed_domain` plus one `variable_mapping` entry of `id` / `story_name` / `observed` / `type` / `unit` per node) followed by per-field instructions. The runtime template variables `{format_description}` and `{independence_note}` expand to a graph-format reminder and (optionally) a list of conditional independencies.

```
You are an expert in causal inference, domain modeling, and
statistical dependencies.
You will be given a set of abstract variables (possibly as a directed
acyclic graph). Your goal is to map these variables to a realistic,
coherent, and scientifically plausible domain (e.g., epidemiology,
economics, physics) that is consistent with the provided dependencies.

Format Specification:
    {format_description}
    {independence_note}

Guidelines:
1) Analyze the fixed nodes (if any) to infer the domain context.
2) Rename the target nodes to specific, measurable, and realistic
variables that fit this domain. Avoid generic names like "Factor A"
or "Variable X".
3) Ensure the 'unit' field contains realistic measurement units
(e.g., "mmHg", "years", "kg", "counts") and 'type' is appropriate
(e.g., "continuous", "binary"). This is not as crucial for the
UNOBSERVED variables.
4) The chosen variables must make sense together in a single scenario.
5) CRITICAL for "observed" field: Check the "observed_nodes" list in the
graph. Set observed=true ONLY for nodes in that list. Any node NOT in
observed_nodes is LATENT/unobserved - you MUST set observed=false
for these.

    {output_format_block}
```

Variable mapping user prompt template. Optional placeholders are filled by the grafting and anchor logic. The auxiliary stage substitutes `{additional_requirements_block}` with a directive to keep the shared anchor immutable, reuse its existing story name, and avoid meanings that would imply unintended direct links to existing variables; the final merged-graph audit substitutes a softer directive that lets minor wording refinements pass while preserving anchor identity, scope, type, and unit. The `{output_format_name}` expands to either `JSON` or `XML`.

```
You will be given a graph of variables represented as {format_label}.
Your task is to choose a coherent domain and assign human-friendly
names to node ids.{tool_note}
The graph representation follows below:
```

```
    {serialized_graph}

    {independence_section}
Context:
- Fixed nodes (already named): {fixed_nodes}
-> {fixed_nodes_instruction}
- Nodes needing names: {needs_names}
-> Rename these to fit the context of the fixed nodes.
    {fixed_name_assignments_block}{forbidden_story_names_block}
    {domain_hint_block}{additional_requirements_block}
    {existing_graph_mapping_block}{anchor_context_block}

Return ONLY the {output_format_name} object{extra_instruction}.
```

Variable mapping feedback prompt. Used in the audit/repair loop (Alg. 3) when the previous mapping fails consistency checks. `{output_format_name}` again expands to `JSON` or `XML`.

```
CRITICAL: Your previous mapping FAILED a causal-consistency audit.
You MUST revise variable meanings so that the following violations
no longer apply:
    {violation_block}

Especially: for every NON-EDGE pair, ensure there is NO plausible
direct causal link in either direction.
Return ONLY the {output_format_name} mapping object in the same
schema as before.
```

Pre-audit system prompt. The pre-auditor (Alg. 2) checks whether CauseNet-fixed node concepts can plausibly satisfy the graph's structural constraints before the mapper runs. The prompt defines a mediator-aware working definition of "direct causal link" and a HIGH/MEDIUM/LOW classification for non-edge inevitability that explicitly considers whether a mediator could absorb the effect. The block below is condensed: we omit the in-prompt salvageable / hopeless examples, the surrounding context notes, and the output-format instructions, and the inline `feasible=true/false` annotations summarize the prompt's decision-summary lines.

```
You are a causal structure feasibility checker.

You will be given:
1. A causal graph structure with FIXED NODE CONCEPTS (already named
from a knowledge base)
2. The structural constraints imposed by the graph:
- DIRECT EDGES: Causal relationships that MUST be plausible
- NON-EDGES: Pairs where NO direct causal link must exist

YOUR TASK: Decide whether there is at least one MAINSTREAM,
non-contrived interpretation under which the FIXED NODE concepts
satisfy the structural constraints. Ask: "Can a skilled mapper
typically find a coherent interpretation that makes the constraints
work?" This is a PRE-FILTER; a separate, thorough audit runs AFTER
mapping. Catch only OBVIOUS failures; default to letting borderline
cases through.

WORKING DEFINITION of DIRECT CAUSAL LINK:
'There exists an intervention on U that changes V while holding fixed
the other variables in the graph (especially the other nodes that
could mediate the effect).'

NON-EDGE CHECK:
For each NON-EDGE pair of fixed nodes, classify the inevitability of
a direct causal link in mainstream usage:
- HIGH: A direct effect would likely persist even after introducing
a mediator (residual direct pathway remains plausible).
-> feasible=false
- MEDIUM: A direct effect is plausible but a mainstream
```

```
operationalization (e.g., administrative proxy, eligibility rule,
time-indexed exposure) can reasonably remove it.
-> feasible=true, confidence="low"
- LOW: A direct effect is not a typical interpretation.
-> feasible=true

EDGE PLAUSIBILITY CHECK (apply very generously):
For each direct edge, accept ANY mechanism (even indirect, weak, or
context-dependent). Only flag as HOPELESS if the edge is physically
impossible or logically contradictory.

DECISION RULE: Default to FEASIBLE with confidence="low" when
uncertain; let the full audit catch actual problems.
```

Mapping audit system prompt. The mapping auditor enforces causal consistency of the proposed variable mapping against the graph structure. Non-edge violations are treated as hard constraints; direct-edge plausibility is judged leniently (only clearly implausible edges are flagged as violations); CI and type consistency are soft constraints. For grafted graphs, the prompt is rendered with a stage-specific instruction block (auxiliary-stage anchor immutability or final merged-graph leniency) that the same template substitutes in via `{stage_specific_instruction_block}`. The block below is lightly condensed: we omit the conditional interpretation-note inserts, the output-format instructions, and the JSON/XML schemas.

```
You are a strict causal consistency auditor.

    {stage_specific_instruction_block}You will be given:
1) DIRECT edges (u -> v) that must be plausible as direct causal effects.
2) NON-EDGE pairs (u, v) where NO plausible direct causal link may exist.
3) Optionally, conditional independence (CI) relations from the graph.
4) A proposed variable mapping from node ids to real-world meanings.
5) Optionally, expected node type hints from generation.

Your job:
1. For each NON-EDGE pair (U, V), decide if ANY DIRECT CAUSAL LINK
exists between the mapped variables. Working definition:
'There exists an intervention on U that changes V while holding
fixed the other variables in the graph (especially the other
nodes that could mediate the effect).'
You MUST include an entry in "non_edge_attestations" for EVERY
non-edge pair (no_direct_link true/false plus a brief
justification); if a direct link DOES exist, also record a
VIOLATION.

2. For each DIRECT edge (U -> V), decide if a plausible direct causal
link exists. Be GENEROUS: any reasonable mechanism (weak, partial,
or context-dependent) is sufficient. Only flag a VIOLATION if
clearly implausible.

3. For each CI statement, judge if plausible given variable meanings.
(Soft constraint.) When flagging a CI violation, name the variables,
explain why the chosen meanings make the independence implausible
(e.g., uncovered alternative pathways), and HINT at a reframing
that would restore it.

4. Type consistency: be lenient (treat close families as compatible,
skip when missing); only flag clear contradictions. (Soft.)

INTERPRETATION NOTES:
We mostly care about the NON-EDGE violations. Use the definition of
the DIRECT CAUSAL LINK above very strictly. For example, a verbal
aggregation of a detailed effect X -> M -> Y is NOT a separate
direct effect X -> Y! For that, we require a SEPARATE direct
pathway!

FEEDBACK QUALITY REQUIREMENT:
```

```
For every violation, the "explanation" must be actionable. End each
with a short 'HINT:' suggesting how it might be resolved. Do not
suggest changing the graph structure or fixed nodes; if the violation
is between two fixed nodes, say so explicitly so the caller can reject
the sample.
```

Story generation system prompt. Instructs the story generation LLM to produce a short narrative from the renamed graph and proposed domain. The runtime template variable `{format_system}` expands to a reminder of the graph serialization format used in the user prompt.

```
You are an expert in narrating a cohesive story based on a causal graph
with a domain and domain-relevant variables.
The graph will be provided in the following structured format:
    {format_system}

Your task is to write a SHORT, CONCRETE STORY (2-4 paragraphs) that:
- is situated in the provided PROPOSED DOMAIN
- uses ONLY the provided STORY NAMES from the VARIABLE MAPPING
(NEVER use raw node IDs like V0, V1, etc.)
- mentions EVERY mapped variable explicitly in the story,
including latent/background factors when present
- makes EVERY direct causal relationship in the graph explicit in
the story itself
- does NOT introduce unsupported extra direct causal relationships
- avoids graph jargon (no "edges", "nodes", "colliders", "DAG", etc.)
- is scientifically plausible and engaging
- may be slightly denser than usual if needed to cover the full
graph exactly

    You will be given a PROPOSED domain, a VARIABLE MAPPING (ids -> story
    names) in JSON, and a CAUSAL GRAPH in the format prescribed by the
    system prompt.

    Your task is to write a short, concrete STORY (2-4 paragraphs) using
    ONLY the provided story names from the VARIABLE MAPPING (no raw ids).
    The story needs to reflect the causal relationships in the graph and
    mention every mapped variable explicitly within the narrative, BUT
    WITHOUT using any graph jargon revealing its structure.
    Every direct edge in the graph must be stated or clearly implied in
    the STORY itself. Do not rely on the appendix/justifications to cover
    missing edges. Do not introduce unsupported extra direct causes.
    Retain the units and variable types (e.g., categorical, continuous)
    given by the mapping. Additionally return CAUSAL JUSTIFICATIONS
    explaining how the story reflects the causal structure.

    Return ONLY a JSON object: {"story": "...",
    "causal_justifications": "..."}

    Proposed domain:        {proposed_domain}
    Variable mapping (JSON):
        {variable_mapping_json}
    Graph structure:        {graph_structure}
        {extra_instruction}
```

Story audit system prompt. The story auditor verifies that the generated narrative faithfully represents the causal graph. Hard failures populate `violations`; lower-priority issues populate `warnings`; the auditor must additionally fill a `node_attestations` entry for every required variable and an `edge_attestations` entry for every direct edge. The block below is condensed: we omit the input list, the interpretation notes, the per-item sub-bullets, and the JSON/XML schemas.

```
You are a strict story-to-DAG auditor.

HARD REQUIREMENTS:
1. Every required variable must be explicitly mentioned in the STORY
```

```
using its provided story name. Latent/unobserved variables may be
narrated as hidden/background factors, but they still must be
mentioned. Emit one entry in "node_attestations" for EVERY
required variable.
2. Every DIRECT edge (U -> V) must be clearly supported by the STORY
itself. A path-level claim does NOT automatically cover every edge
on the path. Emit one entry in "edge_attestations" for EVERY
direct edge (supported / contradicted, with a justification).
3. If the STORY contradicts the stated direction of an edge, mark
that as a hard violation.

SOFT CHECKS:
4. Warn if the STORY introduces extra direct causal claims between
NON-EDGE pairs.
5. Warn if the STORY drifts from the proposed domain.
6. Warn if the STORY uses graph jargon.
7. Warn if the STORY is implausible, globally incoherent, or mixes
incompatible domains in a way that makes the scenario hard to
believe as a real setting.

DECISION RULE:
- Set "pass" to true only if all hard requirements are satisfied.
- Put ONLY hard failures in "violations"; put soft issues in
"warnings".
- Each issue must be actionable, ending in a "HINT:" clause that
suggests how to rewrite the story to remove it.
```

A.19.2 Benchmark task prompt templates

Every benchmark task is delivered to the agent as a fixed *system message* followed by a *user message* assembled by `build_single_task_prompt`. The user message wraps a family-specific body in shared scaffolding that is identical across tasks: a data-location header, the scenario `### Background` (the story), a `### Data` section, a shared `### Measurement Note` on data-backed tasks, the family-specific `### Question` and `### Output Format`, and a footer giving the answer-file path and submit instruction. To avoid repetition, we reproduce the shared blocks once below and then list only the per-family `### Question`/`### Output Format` bodies. Template variables (`{story}`, `{treatment_name}`, `{outcome_name}`, etc.) are filled from the scene at generation time; the formal JSON response schema is not sent to the model (the agent writes an answer file via `bash`), so each body shows only the inline JSON example. The full `(family, output variant)` catalog is enumerated in App. A.9.

Shared system message. Identical for every task; lists the available Python stack and the one-bash-call interaction contract.

```
You are an expert data scientist solving causal reasoning tasks.
You have Python 3.11 with: pandas 3.0.2, numpy 2.4.4, scipy 1.17.1,
scikit-learn 1.8.0, statsmodels 0.14.6, pyarrow 24.0.0,
matplotlib 3.10.9, seaborn 0.13.2, networkx 3.6.1, xgboost 3.2.0.
Do not assume unlisted packages are installed.

<rules>
1. EVERY response must include EXACTLY ONE bash tool call. No exceptions.
2. Before each tool call, briefly state what you are about to check and
   ↪ why.
3. For multi-line Python, use a heredoc:
python3 << 'PYEOF'
import pandas as pd
df = pd.read_parquet("scenes/<scene_id>/data.parquet")
print(df.describe())
PYEOF
4. To write files, use cat with heredoc:
cat > /workspace/answers/my_answer.json << 'EOF'
    {"key": "value"}
EOF
```

```
5. The grader reads only /workspace/answers. Do not write answers under
   ↪ /home.
6. When done, verify your answer file exists at the requested path, then
   ↪ submit: echo DONE
</rules>
```

Shared user-message skeleton. The data-location bullets list only the files released for the task: the `data.parquet`/`calibration.parquet`/`schema.json` lines appear only for data-backed tasks, and a `test_features.parquet` line is added for prediction. The story-only Rung-2/3 identification and graph-recovery tasks omit the `### Data` section and the Measurement Note.

```
# Task

**Data location:** `scenes/{scene_id}/`
- Story/context: `scenes/{scene_id}/story.md`
- Observational data: `scenes/{scene_id}/data.parquet`
   ↪ (data-backed only)
- Calibration subset (if present):
   ↪ `scenes/{scene_id}/calibration.parquet`   (data-backed only)
- Column metadata: `scenes/{scene_id}/schema.json`
   ↪ (data-backed only)
- Test features (no outcome column):
   ↪ `scenes/{scene_id}/test_features.parquet`   (prediction only)

### Background
    {story}

### Data
You are provided with observational data in `data.parquet`.
   ↪ (data-backed only)

    {### Measurement Note - shown below; data-backed tasks only}

    {family-specific ### Question and ### Output Format - see per-family
   ↪ blocks}

**Answer file:** `/workspace/answers/{answer_file}`

Write a JSON file matching the schema described above.   (CSV with
   ↪ columns `prediction`[, `lower`, `upper`] for prediction)

The grader reads only files under `/workspace/answers/`; do not write
   ↪ answers under `/home`.
When done, verify your answer file exists at the exact path above, then
   ↪ submit:
```bash
echo DONE
```
```

Shared Measurement Note (data-backed tasks). Inserted before the `### Question` on every task that ships a `.parquet` file. The first paragraph (released data scale) is always present; the remaining paragraphs appear only when the released observation variant contains noisy measurement columns.

```
### Measurement Note
The story names conceptual variables in the causal graph. Numeric values
   ↪ in the released `.parquet` files, including `data.parquet`, and any
   ↪ question-specified values are on the benchmark's released data
   ↪ scale for those conceptual variables. This scale may differ from
   ↪ the units, ranges, or examples implied by the story; values may be
   ↪ centered, standardized, or otherwise transformed. For answering
   ↪ data-backed questions, use the released data scale.

In this released dataset, some conceptual variables are not directly
   ↪ measured in `data.parquet`. Each such conceptual variable is
```

```
   ↪ replaced by a bundle of noisy measurement columns. Each bundle has
   ↪ exactly one conceptual parent and should not be treated as a set of
   ↪ separate causal variables. Examples in this scene: Covariate1:
   ↪ Covariate1_meas_a, Covariate1_meas_b; Mediator1: Mediator1_meas_a.

A smaller `calibration.parquet` contains the measurement columns
   ↪ together with gold-standard measurements of the corresponding
   ↪ conceptual variables for a subset of rows. Use the shared name
   ↪ stem, `schema.json`, and the calibration rows to determine which
   ↪ measurement columns belong to each conceptual variable.

For determining causal identifiability (when necessary), reason only
   ↪ over the conceptual variables described in the story. Same goes for
   ↪ any causal quantity. The observation layer affects solely the
   ↪ statistical estimation difficulty of the population quantity from
   ↪ provided data.

When estimating a quantity from data, if a named conceptual variable is
   ↪ measured by a bundle, use that bundle and the calibration data to
   ↪ estimate or reconstruct the conceptual variable before estimating
   ↪ the requested statistical or causal quantity.
```

Rung 1 (Associational). The bodies below give the `### Question` and `### Output Format` that replace the `{family-specific...}` placeholder above.

R1: Prediction (`point_predictor`; `prediction_interval` adds a central 90% interval). Prediction is the one Kaggle-style family: it writes a CSV of held-out predictions rather than a JSON answer, so its body replaces `### Question`/`### Output Format` with an `### Objective`. The `prediction_interval` variant appends the `### Uncertainty Quantification` block.

```
### Data
You are provided with training data in `data.parquet` containing the
   ↪ following columns:
"Treatment", "Outcome", "Covariate1", "Covariate2"

### Objective
Build a predictive model for the conceptual outcome **Outcome** using
   ↪ the variables available in the released data.
If the conceptual outcome is measured by a bundle in the released data,
   ↪ use the calibration rows to learn how those measurements relate to
   ↪ the conceptual outcome.

### Uncertainty Quantification (Prediction Intervals)
   ↪ [prediction_interval variant only]
Also produce a **central 90% prediction interval** for each prediction.

### Notes
- Evaluation uses held-out test data.
```

R1: Association — sign. The `effect_size_point` variant instead asks for the numeric Pearson strength (`{"value": 0.45, "method": "pearson"}`); it is explicitly framed as "not a causal-effect estimate."

```
### Question
Estimate the sign of the observational association between the
   ↪ conceptual variables **Treatment** and **Outcome**.

Report `"+"` if higher values of **Treatment** are associated with
   ↪ higher values of **Outcome**, `"-"` if higher values of
   ↪ **Treatment** are associated with lower values of **Outcome**, and
   ↪ `"unknown"` only if the empirical association is too unclear to
   ↪ determine.
```

```
### Output Format
Provide your answer as a JSON object:
```json
    {
    "sign": "+",  // or "-" or "unknown"
    "stat": {"method": "correlation", "value": 0.45}  // optional,
  ↪ encouraged
}
```
```

R1: Conditional association — sign before/after conditioning. Same `association` family; the `delta_point`/`delta_sign_only` variants ask for the magnitude/sign of the *change* in association after conditioning, and `argmax_change` asks which of several offered conditioning variables changes it most.

```
### Question
Consider the association between **Treatment** and **Outcome**.

1. What is the sign of the marginal (unconditional) association?
2. What is the sign of the association after conditioning on the
  ↪ conceptual variable **Covariate1**?

Conditioning here means statistical conditioning, not intervention.

### Output Format
Provide your answer as a JSON object:
```json
    {
    "sign_before": "+",  // marginal association sign: "+", "-", or
  ↪ "unknown"
    "sign_after": "+",   // association after conditioning: "+", "-", or
  ↪ "unknown"
    "conditioning_var": "Covariate1",
    "explanation": "Brief explanation of why the association changes or
  ↪ stays the same"
}
```
```

R1: Collider phenomenon (explaining away). The `induced_association_sign_only`/ `induced_association_strength_point` variants instead ask for the sign / numeric strength of the association induced by conditioning on the collider.

```
### Question
Consider the conceptual variables **Treatment**, **Outcome**, and
  ↪ **Collider1**.

After conditioning on the conceptual variable **Collider1**, is there a
  ↪ nonzero observational association between **Treatment** and
  ↪ **Outcome**?

### Output Format
Provide your answer as a JSON object:
```json
    {
    "association_present": true,  // whether a nonzero association is
  ↪ present after conditioning on Collider1
    "explanation": "..."
}
```
```

Rung 2 (Interventional).

R2: Causal sketch — directed edges (`skeleton_edges` drops direction). A story-only task (no `### Data`/Measurement Note); the prompt states the exact variable count.

```
### Question
Based only on the scenario description, identify the conceptual causal
  ↪ variables it describes and the direct causal relationships among
  ↪ them.

The conceptual causal graph contains exactly 5 causal variables.

Important:
- Include only direct causal effects, not indirect effects through
  ↪ intermediate variables.
- A causes B means that an intervention on A would directly change B
  ↪ while holding fixed other variables that could mediate the effect.
- Do not include associations caused only by common causes, selection,
  ↪ or conditioning.
- Include every causal variable the story describes, including any that
  ↪ is described only as a hidden, background, or unmeasured factor.

### Output Format
Provide your answer as a JSON object (use the exact variable names as
  ↪ they appear in the story):
'''json
    {"edges": [{"from": "Variable1", "to": "Variable2"}, {"from":
  ↪ "Variable1", "to": "Variable3"}]}
'''
The skeleton_edges variant returns undirected pairs instead:
  ↪ {"skeleton_edges": [{"a": "Variable1", "b": "Variable2"}]}.
```

R2: Identification — method label. A story-only task: the graph drives a population-ATE identifiability judgment and no estimator is run. Sibling variants are `identifiable_boolean` (`{"identifiable": true}`) and the adjustment-set queries `one_valid_adjustment_set` (`{"adjust": [...] | "no_backdoor" | "non_id"}`), `minimal_adjustment_set_size` (`{"k": ...| "no_backdoor" | "non_id"}`), `n_valid_adjustment_sets` (`{"n": ...| 0 | "no_backdoor" | "non_id"}`), and `all_minimal_adjustment_sets` (`{"adjustment_sets": [...] | "no_backdoor" | "non_id"}`). For the all-minimal variant, `[[]]` denotes the empty adjustment set as the only minimal valid set; for the one-valid variant, `[]` denotes that no adjustment is needed. The size, count, and all-minimal variants list the offered covariates under an `### Available Conceptual Covariates` header, while the one-valid variant names them in the question prose.

```
### Conceptual Observed Variables
The following conceptual variables are observed for
  ↪ causal-identification purposes, in no particular order:
  ↪ "Treatment", "Outcome", "Covariate1", "Covariate2", "Mediator1"

### Question
Is the population Average Treatment Effect (ATE) of **Treatment** on
  ↪ **Outcome** identifiable from the observational distribution over
  ↪ the conceptual observed variables? Return the first applicable
  ↪ identification label.

Target estimand:
ATE = E[Y | do(Treatment=x1)] - E[Y | do(Treatment=x0)]

Treat the story as specifying the qualitative conceptual causal graph.
  ↪ Identifiability is a property of that graph and the observational
  ↪ distribution over the observed conceptual variables; do not assume
  ↪ parametric forms, effect homogeneity, or access to the structural
  ↪ model.

Return the first applicable label:
1. "trivial_zero": Treatment has no directed causal path to Outcome, so
  ↪ the population ATE is identifiable as zero.
```

```
2. "backdoor": Otherwise, a valid backdoor adjustment set among the
   ↪ observed conceptual variables identifies the population ATE.
3. "frontdoor": Otherwise, a valid front-door formula using observed
   ↪ conceptual variables identifies the population ATE.
4. "other_id": Otherwise, the population ATE is identifiable by another
   ↪ valid do-calculus / ID argument.
5. "none": The population ATE is not identifiable.

### Output Format
Provide your answer as a JSON object:
```json
    {"method": "backdoor"}
```
```

R2: Effect estimate — population ATE (point). Data-backed (Measurement Note shown). Variants share this stem and swap the output contract: `ate_sign_only` (`{"sign": "+"/"-"/"0"/"unknown"}`), `ate_vs_assoc_sign_match` (`{"matches": true|false|null}`), and `ate_uq_95`, which appends "also provide a central 95% confidence interval" and returns `{"ate": .., "ci_lower": .., "ci_upper": ..}`. Non-identifiable scenes (e.g. bare-IV) require the `null`/`unknown` abstention value.

```
### Data
You are provided with observational data in `data.parquet` containing
   ↪ the following columns:
"Treatment", "Outcome", "Covariate1", "Covariate2"

### Question
Estimate the population Average Treatment Effect (ATE) of the conceptual
   ↪ treatment **Treatment** on the conceptual outcome **Outcome**.

Use treatment levels x0=0.0 and x1=1.0.

ATE = E[Y | do(Treatment=1.0)] - E[Y | do(Treatment=0.0)]

### Output Format
Provide your answer as JSON:
```json
    {"ate": 1.23}
```
If the population ATE is not identifiable from the observational
   ↪ distribution over the conceptual variables, return:
```json
    {"ate": null}
```
```

R2: Bias diagnostic — collider bias. The `forbidden_controls_list` variant instead lists the offered conceptual covariates and asks which must *not* be conditioned on, returning `{"forbidden": [...] | [] | "no_backdoor" | "non_id"}` (set-F_1 graded, with the non-identifiable branch routed to abstention). The empty list means that adjustment is available and no listed variable must be excluded.

```
### Question
A researcher wants to estimate the population causal effect of
   ↪ **Treatment** on **Outcome**.
They propose to condition on the conceptual variable **Collider1** in
   ↪ their analysis.

Would conditioning on **Collider1** introduce collider bias or otherwise
   ↪ open a noncausal path between treatment and outcome?

Answer using the story-implied conceptual causal graph. The released
   ↪ data may help inspect associations, but the bias judgment is causal
   ↪ and graph-based. Measurement columns are measurements of conceptual
   ↪ variables, not separate controls.
```

```
### Output Format
Provide your answer as a JSON object:
```json
    {"bias_present": true, "explanation": "..."}
```
```

Rung 3 (Counterfactual).

R3: Counterfactual identification — ETT (`identifiable_boolean`). A story-only judgment of whether the target counterfactual estimand is identifiable; instantiated for ETT, NDE, and NIE, swapping in the corresponding estimand name, formula, and mediator list.

```
### Question
Treat the story as specifying the qualitative conceptual causal graph.

Target estimand: **Effect of the Treatment on the Treated (ETT)**

Formal notation:
ETT(x1, x0) = E[Y_{x1} - Y_{x0} | X=x1], with x0=0, x1=1

Is the effect of treatment on the treated for **Treatment** on
  ↪ **Outcome** identifiable from the observational distribution under
  ↪ the story-implied conceptual graph?

Identifiability of this counterfactual estimand is a property of the
  ↪ graph and the population observational distribution alone; do not
  ↪ assume parametric forms, effect homogeneity, monotonicity, or
  ↪ access to the structural model.

### Output Format
Provide your answer as a JSON object:
```json
    {"identifiable": true, "explanation": "..."}
```
```

R3: Counterfactual effect — ETT (point). Data-backed. The `sign_only` variant returns `{"sign": "+"/"-"/"0"/"unknown"}`, and `effect_uq_95` appends "estimate the population ETT and provide a central 95% confidence interval", returning `{"value": .., "ci_lower": .., "ci_upper": ..}`. A non-identifiable ETT requires the `value=null` (or `sign="unknown"`) abstention value.

```
### Question
Estimate the population Effect of the Treatment on the Treated (ETT) of
  ↪ the conceptual treatment **Treatment** on the conceptual outcome
  ↪ **Outcome**.

Use treatment levels **x0=0** and **x1=1**.

ETT(x1, x0) = E[Y_{x1} - Y_{x0} | X=x1]

If the ETT is not identifiable from the observational distribution over
  ↪ the conceptual variables, return `{"value": null}`.

### Output Format
Provide your answer as a JSON object:
```json
    {"value": 1.23, "explanation": "Briefly state the identifiability
  ↪ judgment and estimation approach."}
```
```

R3: Mediation effect — NDE (point). Shown for the NDE point estimate; the NIE prompt is analogous with $\text{NIE}(x_1, x_0) = E[Y_{x_0, M_{x_1}} - Y_{x_0, M_{x_0}}]$. For each estimand, `sign_only` returns `{"sign": "+"/"-"/"0"/"unknown"}` and `effect_uq_95` appends a central 95% confidence interval (`{"value": .., "ci_lower": .., "ci_upper": ..}`); a non-identifiable effect requires the `value=null` / `sign="unknown"` abstention.

```
### Question
Estimate the population Natural Direct Effect (NDE) of the conceptual
  ↪ treatment **Treatment** on the conceptual outcome **Outcome**, with
  ↪ mediator(s): **Mediator1**.

Use treatment levels **x0=0** and **x1=1**.

NDE(x1, x0) = E[Y_{x1,M_{x0}} - Y_{x0,M_{x0}}]

If the NDE is not identifiable from the observational distribution over
  ↪ the conceptual variables, return `{"value": null}`.

### Output Format
Provide your answer as a JSON object:
```json
    {"value": 0.85, "explanation": "Briefly state the identifiability
  ↪ judgment and estimation approach."}
```
```

The `direct_vs_indirect_dominance` variant instead asks which component (NDE or NIE) has larger absolute magnitude, returning `{"dominant": "direct"/"indirect"/"tie"}` (or `null` when either the NDE or NIE is non-identifiable).