CAUSALDS: BENCHMARKING CAUSAL REASONING IN DATA-SCIENCE AGENTS

Andrej Leban
Department of Statistics
University of Michigan
Ann Arbor, MI, United States
leban@umich.edu

Yuekai Sun
Department of Statistics
University of Michigan
Ann Arbor, MI, United States
yuekai@umich.edu

Abstract

Large language models (LLMs) increasingly act as integrated *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 a principled causal data-generating structure. Furthermore, 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 realistic domain. We optionally ground the *composition* of the benchmark components in empirical distributions obtained from real-world datasets, thus retaining empirical structure while reducing the "causal parrot" risk through completely synthetic generation. From each scene, we then derive tasks spanning all three of Pearl's rungs, with typical data-science prediction tasks appearing as Rung 1. 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 the frequent presence of imperfect observations, which are generated by an observation model. Additionally, recognizing when a question admits no warranted answer and abstaining is treated as a first-class scored outcome. The benchmark thus jointly evaluates symbolic causal reasoning, data science, uncertainty quantification, abstention, and tool use/coding.

1 Introduction

Modern LLMs are increasingly powerful in agentic settings and are routinely used in data-science workflows (Jing et al., 2025; Chan et al., 2025; Gu et al., 2024; Majumder et al., 2025). Their actual *causal* reasoning capabilities, however, remain contentious (Zecevic et al., 2023; Jin et al., 2023). 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[1] 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 standard 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 released data and assumptions. The fifth is tool use and coding: carrying the analysis out through code in an

[1] github.com/andleb/causalds

agentic, file-backed environment. 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; Qiang et al., 2025; Gu et al., 2024; Majumder 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; Sheth et al., 2024; Chen et al., 2024; Zhou et al., 2024; Du et al., 2026). 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). Concurrent structure-preserving probes make this concern concrete: merely anonymizing the semantic variable names, with the causal task held fixed, sharply reduces causal-benchmark accuracy (Yu & Zhou, 2026). Concurrent interactive-discovery environments address another part of the gap by sampling hidden SCMs and letting agents intervene (Yang et al., 2026); they target experimental mechanism recovery rather than causal data-science analysis over released observational files.

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. We evaluate contemporary agents on a realistically grounded CausalDS exam and find that the axes do not collapse to a single capability: models differ in content correctness, uncertainty quantification, abstention recognition, and tool-use efficiency.

Main Contributions

- **SCM-grounded synthetic scenes with empirically anchored composition.** We generate hidden causal graphs, instantiate SCMs, synthesize observational data, and compute private ground truth for evaluation. The scene generator is fully synthetic, while benchmark composition can be anchored along empirical axes such as variable type, graph structure, identifiability, mechanism profile, and observation complexity.
- **Graph-faithful free-form verbalization.** CausalDS maps abstract graph nodes to coherent domain variables (partially seeded from CauseNet (Heindorf et al., 2020)) and generates natural-language stories describing the resulting causal setting. Both the variable mapping and the final story are audited against the hidden graph and overall narrative coherence, so that the benchmark can combine realistic prose with controlled causal structure.
- **A separate observation layer for data-analysis difficulty.** We distinguish the conceptual SCM from the data view released to the agent. In addition to clean observations, CausalDS can replace conceptual variables with bundles of noisy observations of the latter. This lets the benchmark vary the data-science difficulty without changing the causal aspects, such as the target causal estimand or its identifiability status.
- **A broad causal data-science task suite.** From each scene, CausalDS derives tasks across Pearl's hierarchy, including prediction, association, graph recovery, identification, effect estimation, bias diagnostics, and counterfactual reasoning. Because tasks share a hidden SCM, they connect language interpretation, statistical estimation, and formal causal reasoning.
- **Abstention-aware deterministic evaluation.** CausalDS scores submitted answers against hidden ground truth, including deliberately non-identifiable causal queries where the correct behavior is to abstain. We evaluate contemporary agents on an exam compositionally grounded in real-world corpora and show that model performance dissociates along the five axes above.

2 Background

We briefly recall the formalism the benchmark relies on, the identifiability questions it tests, and what we mean by a *data-science agent*.

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 back-door 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.

Recognizing the non-identifiable case is itself a benchmark-relevant skill, and the CausalDS scoring rules (Sec. 3.7) treat abstention on non-identifiable estimands as a first-class outcome — non-identifiable problems can appear in all settings (cf. Sec. 3.4) where non-identifiability is a possibility. 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. Identifiability is always relative to the conceptual variables, as opposed to their noisy observations, which are discussed in Sec. 3.2.

Data-science agents. By a *data-science agent* we mean an LLM that interacts with a sandboxed environment by reading benchmark-provided files, executing code, inspecting intermediate outputs, and writing answer files. Each CausalDS task therefore mixes language understanding, code-driven estimation, and basic tool use.

3 The CausalDS Benchmark

A CausalDS dataset instance is a *scene* consisting of a narrative story, a tabular dataset, a lightweight data schema, and a list of tasks; ground truth quantities and a hidden test split are stored separately for scoring. Fig. 1 sketches the generation pipeline, executed in order: (i) sample a causal graph, optionally enlarged by anchor-based *grafting* of auxiliary motifs through shared anchor nodes; (ii) instantiate a structural causal model under typed continuous and binary mechanism profiles, generate observational data (Sec. 3.1), optionally using an additional *observation model* (Sec. 3.2); (iii) map nodes to variable names grounded in a plausible domain and generate a verified narrative story (Sec. 3.3); (iv) compute tasks and ground truth (Sec. 3.4); and (v) package public and private scene artifacts (Sec. 3.5). Larger production runs use a blueprint-driven path in which a *composition configuration* specifies distributions over composition axes: motif, graft complexity, identifiability regime, treatment/outcome type, SCM profile, and released observation model variants. 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 presents an example of a *scene*.

3.1 Graphs and SCM-generated tabular data

We sample DAGs starting from canonical *motifs* (chain, fork, confounding, and others; full catalog is reproduced in App. A.5, 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. The shared node is the only overlap between the existing graph and the new fragment. When grafting

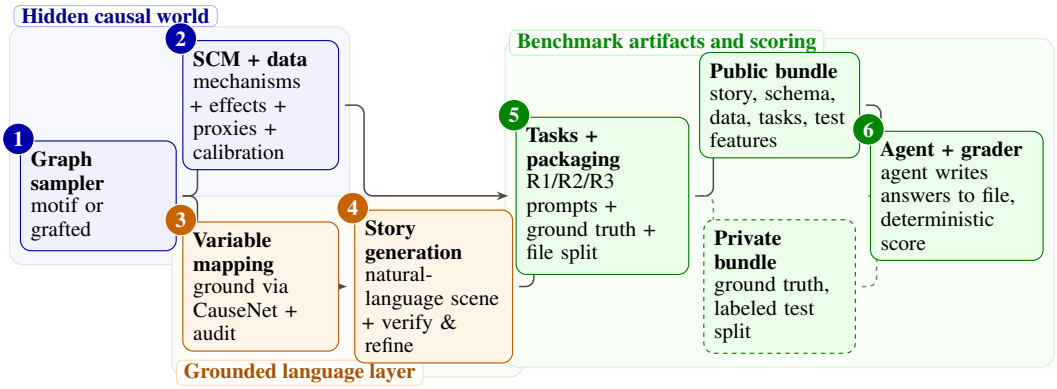


Figure 1: *CausalDS pipeline*: from hidden causal structure to benchmark execution.

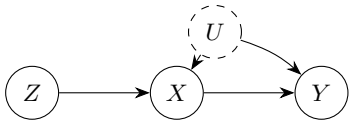
Scene example (R2 bias diagnostic, forbidden-controls output variant; `iv` motif; hard observation variant).

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*: story, public dataset with observed measurements, hidden DAG, and the posed question.

is active, we restrict both the main graph and the auxiliary graph motif pools toward motifs that remain stable under grafting. Anchor grafting thus allows increasingly complex structures while preserving narrative coherence by the use of the shared (already verbalized) anchor, and limiting the variable mapping problem to isolated subgraphs. Some motifs include latent variables (e.g., unobserved confounders), enabling stress tests for identifiability and adjustment reasoning. Graph properties are calibrated against empirical histograms from real-world causal datasets (App. A.7). This is similar in principle to approaches such as Nemotron Personas (Meyer & Corneil, 2025); to our knowledge, this is the first application in this type of benchmark.

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_scm_profile` and a `binary_scm_profile` drawn from independent registries, so mechanism choice is settled marginally per node type. Each registry mixes *empirical profiles* whose mean-function families derive from real biochemical-equation and Boolean-rule

corpora with *synthetic profiles* that exercise additional regimes (e.g., neural network mixture-noise, sharp-threshold, etc.) for additional diversity; noise is kept additive (continuous) or single-link (binary) so that the same SCM remains suitable for Monte Carlo evaluation of treatment effects, mediation, and counterfactuals. The full registry, mechanism families, and noise options are listed in App. A.2.

3.2 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 conceptual variables are distinct from true latent SCM nodes: hidden confounders are unobserved everywhere, whereas measured variables remain part of the narrated graph, but are observed directly only in a small *calibration* sample provided to the model.

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 $W_{jr} \mid Z_j \sim \text{Bernoulli}(p_{jr}(Z_j))$. Because the underlying SCM is unchanged, the conceptual causal estimands are preserved. 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, mediator, etc., and never flips an identifiability label.

This 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.

In implementation, several observation variants are released per scene; for example—`clean` (no measurement layer), `proxy`, and `proxy_hard` (more measured non-outcome nodes, stronger corruption, smaller calibration split). When generating the variants, candidates are screened by the bundle Fisher information $I_j(z) = \sum_r I_{jr}(z)$ across the realized support of Z_j to ensure that the measurement does not corrupt the value to the extent of making inference impossible. Admissibility thresholds, recoverability diagnostics, and resulting private observation diagnostics are provided in App. A.6.

3.3 Mapping variables and story generation

Abstract node identifiers are mapped to domain-relevant variable names (e.g., `v1` $\rightarrow$ `Exercise`) using an LLM.[2] Optionally, a knowledge-grounding step seeds names for a subset of nodes using CauseNet (Heindorf et al., 2020), inducing the synthetically generated remaining nodes to conform to them. Thus, we introduce real-world diversity into the entire verbalization. In the results presented in this work, we adopted a policy of seeding the outcome node and one of its neighbors. 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 holds the graph, SCM, data, task, and private ground truth fixed while varying only the seeded story and variable names shows stability for strong-enough models (App. A.12).

To reduce semantic drift and "accidental edges" implied by naming choices, the mapping is then refined iteratively using a *mapper-auditor* loop. For grafted graphs, we apply the same mapper/auditor machinery stage-wise: the main graph is mapped first; each auxiliary fragment is then grounded separately with the shared anchor immutable and duplicate names forbidden; finally the merged full

[2]Kimi 2.5 for the data used to generate the results presented.

graph is audited globally. Given the renamed graph and the matching proposed domain, an LLM generates a narrative that describes the variables and their causal relations. The draft is checked by another auditor and revised iteratively until it passes. The generation-time LLM clients can also use web search to pin down realistic variable names, units, and story details. Alg. 1 summarizes the top-level control flow. The subroutines invoked by Alg. 1 are listed in App. A.3; further details are in App. A.4, and the generation-side prompt templates are reproduced in App. A.16.1.

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).

3.4 Task suite

We use a 3-level taxonomy: **Rung** ($R1$–$R3$) places the task on Pearl's hierarchy (associational, interventional, counterfactual); **question family** is the semantic intent / target quantity (e.g., prediction, identification, effect estimation) and expands on CLadder's collection; **output variant** is the required output type within the family (e.g., sign vs. strength; point vs. interval). We denote tasks as `Rung::family::variant`. Tab. 1 summarizes the question families; the per-family output variants and their scoring rules are enumerated by rung in Tabs. 8–10 (App. A.9).

Some variants are available only in compatible scenes (interval prediction requires continuous outcomes, collider-related tasks require colliders, etc.). 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[3].

3.5 Scene format

Public scene artifacts comprise the *story*, relevant dataset *parquet* files, and the dataset schema. For scenes with a measurement observation model, we additionally include a *calibration set*, a smaller table containing both measurement columns and gold conceptual-variable values. 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, each with three possible observation model variants.

3.6 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

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

Table 1: 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.

inside a sandboxed Docker (or similar) container with network access disabled. The container ships the standard data-science stack; each command runs as a separate shell process and state persists through files. Each task is run as a fresh agent conversation with configurable step and cost limits, keeping per-task contexts small and recording full trajectories for reproducibility and error analysis. After the run completes, the grader parses the agent's answer files and scores them against private ground truth (Sec. 3.7); we additionally log efficiency signals (tool-call counts and API-reported token usage) to support joint evaluation of reasoning quality and tool use. Technical details (containerization, available packages, etc.) are given in App. A.15.

3.7 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. This 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.

Each task emits one atomic score with a metric name; the per-metric formulas (RMSE, ROC-AUC, Brier, log-loss, F_1, the interval score (IS) (Gneiting & Raftery, 2007) ...) and the task-specific scoring rules by rung are listed in App. A.9 (Tabs. 8–10).

For every variant where the answer might not be identifiable, the grader uses *mutually-exclusive routing*: a task is graded as a binary 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):

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

with $(n_i, s_i) = (\mathrm{n}_{\mathrm{test}}, \mathrm{sd}(Y_{\mathrm{test}}))$ for prediction point tasks (using $\sqrt{\mathrm{Brier}}$ as the numerator for binary outcomes), and $(n_i, s_i) = (1, |\tau_i|)$ for scalar point estimates, where τ_i is ground-truth scalar

target (e.g., the ATE). For interval scores, s_i is one of the above depending on whether we are dealing with prediction intervals or effect intervals. The headline columns are then defined in Eq. 2:

$$\begin{aligned} \text{PassRate} &= |D|^{-1} \sum_{i \in D} \mathbf{1}[\hat{y}_i = y_i], \\ \text{Med. NRel. Err} &= \text{med}_{i \in T_{\text{NR}}} \text{NRelErr}_i, \\ \text{Med. F}_1\text{-Loss} &= \text{med}_{i \in T_{F_1}} (1 - F_{1,i}). \end{aligned} \tag{2}$$

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

$$\begin{gathered} S_{\text{NR}} = \frac{|T_{\text{NR}}^{\text{pt}}| m_{\text{pt}} + |T_{\text{NR}}^{\text{int}}| m_{\text{int}}}{|T_{\text{NR}}|}, \; m_{\text{pt}} = \text{med}_{i \in T_{\text{NR}}^{\text{pt}}} \text{NRelErr}_i, \\ m_{\text{int}} = \text{mean}_{i \in T_{\text{NR}}^{\text{int}}} \min(\text{NRelIS}_i, c), \end{gathered} \tag{3}$$

with $c = 10$, keeping point-error tasks median-robust while ensuring that the models do not get away with submitting overly-confident confidence intervals by using a **capped mean** (App. A.9) for the intervals (as the IS is strictly proper in expectation, the mean is the more appropriate aggregator).

For an absolute single-number summary we define the **CausalDS** score (Eq. 4):

$$\text{CausalDSScore} = w_p(1 - \text{PassRate}) + w_r S_{\text{NR}} + w_f \, \text{Med. F}_1\text{-Loss}, \tag{4}$$

with $w_p = N_p / (N_p + N_r + N_f)$, $w_r = N_r / (N_p + N_r + N_f)$, and $w_f = N_f / (N_p + N_r + N_f)$, where $N_p = |D|, N_r = |T_{\text{NR}}|$, and $N_f = |T_{F_1}|$ are the per-pool task counts; *lower is better*.

A CausalDS score of 0 thus requires perfection on every column; while most scores should land in the $[0, 1]$ range, the score can exceed 1 when the NRel. Err median is large. 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). This does not transfer across exams of similar composition; 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. 5 we report both the intermediate (the three components of the CausalDS score) and atomic metrics, whenever appropriate.

4 Related work

Tab. 2 presents an overview across the main design axes for the most closely-related work.

Besides the works summarized in Tab. 2—*CauSciBench*, *CausalReasoningBenchmark*, *CausalProfiler*, *CausalGame*, *CausaLab*, and *CLadder* (Acharya et al., 2026; Sawarni et al., 2026; Panayiotou et al., 2026; Chen et al., 2026; Yang et al., 2026; Jin et al., 2023)—CausalDS is related to three broader lines of work.

Real-study causal analysis benchmarks. The closest real-data benchmarks evaluate causal analysis over scientific or policy studies. *CauSciBench* and *CausalReasoningBenchmark* test end-to-end causal analysis on curated datasets, with emphasis on method choice, identification, estimation, and uncertainty; *InterveneBench* instead emphasizes intervention-centered study-design reasoning in real social systems without predefined graphs or structural equations (Shi et al., 2026). These benchmarks provide strong external realism, but their instances are derived from existing papers, datasets, or study designs, which 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* constructs natural-language causal QA examples from small graph motifs and oracle-generated answers spanning Pearl's ladder; *CausalGraph2LLM*,

Table 2: Related benchmark overview.

Work	Synth. inst.	Gen. graph	Free-form story	File / code	SCM GT	Obs. layer	R1–R3 tasks	Non-ID / abstain	Comment
CausalDS	✓	✓	✓	✓	✓	✓	✓	✓	(**Ours.**) Integrated causal data-science agent benchmark. Generated hidden SCMs; synthetic tabular data; graph-audited free-form stories; clean and noisy-measurement observation variants; file-backed tool use; R1–R3 tasks; scored abstention.
CauSciBench	◐	⚠	◐	✓	⚠	–	⚠	–	End-to-end causal-effect analysis benchmark. Real-paper/textbook-derived tasks plus synthetic scenarios; natural-language problem setup; variable/method selection; statistical implementation; R2 effect-estimation focus; no generated graphs or observation layer.
CausalReasoning Benchmark	–	–	◐	✓	–	–	⚠	–	Identification/estimation benchmark on real datasets. Curated papers and datasets; structured identification specs; point estimates and standard errors; R2-focused; no generated graph, hidden SCM, synthetic story, or measurement layer.
CausalProfiler	✓	✓	–	–	✓	–	✓	✓	Synthetic causal-ML data generator; no LLM-eval involved. Random causal models, data, queries, and ground truth; observational/interventional/counterfactual coverage; in- and out-of-identification regimes; no language scenes or file-backed LLM agents.
CausalGame†	✓	⚠	◐	◐	✓	◐	◐	–	Interactive causal-agent game benchmark. SCM-driven game environments; hidden confounders, noisy measurement, selection bias; active data collection; small scenario family; not tabular scene files or broad R1–R3 task generation.
CausaLab†	✓	✓	⚠	◐	✓	–	◐	–	Interactive causal-discovery environment. Freshly sampled hidden SCMs per episode; budgeted interventions, held-out-target prediction, and DSL-based mechanism recovery; fixed lab narrative rather than per-graph stories; no tabular scene files, counterfactual tasks, or abstention.
CLadder	✓	⚠	⚠	–	◐	–	✓	–	Canonical causal reasoning QA benchmark. Synthetic graph/query pairs; motif-style causal graphs; templated natural-language questions; Bernoulli CBNs; R1–R3 symbolic QA; no tabular data, tool use, measurement layer, or non-ID abstention.

Legend. ✓ = central feature; ◐ = partially present; ⚠ = narrow, probe-level, or adjacent; – = absent or N/A. †Concurrent under a May 1, 2026 cutoff. The columns correspond to: **Synthetic benchmark instances:** instances generated fresh, not curated from existing sources. **Generated causal graph:** (option of) dynamically sampled graphs, not a fixed motif family or hand-authored design. **Free-form story:** natural-language verbalization tied to the graph, beyond templates. **File / code analysis:** agent operates over files, code, and tools, not single-prompt QA. **SCM ground truth:** true causal structure, mechanisms, and effects are known. **Observation layer:** data can reflect the conceptual causal variables through a noisy measurement. **R1–R3 tasks:** does the benchmark cover Pearl's hierarchy? **Non-ID / abstain:** deliberately includes non-identifiable cases requiring abstention.

CLEAR, and *CausalBench* focus on graph-centric or format-dependent tests of causal understanding (Sheth et al., 2024; Chen et al., 2024; Zhou et al., 2024). Other benchmarks target narrower failure modes: *CausalPitfalls* stresses classical statistical traps such as confounding, Simpson's paradox, and selection bias, while *Executable Counterfactuals* focuses on code-mediated counterfactual reasoning (Du et al., 2026; Vashishtha et al., 2026). On the generation side, *CausalProfiler* samples causal models, data, queries, and ground truth, and *Language Models as Causal Effect Generators* introduces sequence-driven SCMs for treatment-effect benchmarking (Bynum & Cho, 2025). Concurrent work extends this line: *NoisyCausal* evaluates natural-language causal QA under structured noise (distractors, perturbed values, latent confounders, partial masking), and *ReplaySCM* scores executable Boolean-SCM mechanism induction from finite interventional evidence via replay on held-out interventions (Xu & Fu, 2026; Batzoglou, 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.

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 engineering, 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), while *CausalGame* evaluates causal thinking through interactive scientific-discovery games (Chen et al., 2026). Concurrent work *CausaLab* evaluates LLM agents in an interactive synthetic laboratory where each episode hides a freshly sampled SCM: the agent intervenes under a budget, predicts a held-out target, and is scored on both task success and the fidelity of the recovered mechanism (Yang et al., 2026). CausaLab thus asks whether an agent can act as an experimental causal discoverer; CausalDS asks whether it can act as a causal data scientist over narrated, file-backed observational scenes. CausalDS is thus complementary to the above works: it benchmarks agents over scene directories where causal reasoning, data-science skills, and tool use all matter simultaneously.

Overall, prior and concurrent work covers some individual ingredients of CausalDS, but to the best of our knowledge no benchmark integrates graph-audited free-form scenes, SCM-generated data files, a noisy-measurement observation layer, complete Pearl-ladder estimand tasks with deterministic scoring, and first-class non-identifiability abstention into a single generator.

5 Main Results

Table 3: Realistic-exam leaderboard. Sorted by ascending CausalDSScore (Eq. 4). *Pass Rate* combines binary content and abstention; S_{NR} is the relative numeric component (Eq. 3). *Comp. Rank* is the rank-version of CausalDS score over the same components. *Valid cont. answers* counts valid submissions among continuous-style tasks. *Tok./task* counts the mean total (prompt+completion) tokens per task. Best per column in bold. ↓ — lower is better, ↑ — higher is better.

#	Model	Valid cont. ans. ↑	CausalDSScore ↓	Comp. Rank ↓	Pass Rate ↑	Med. NRel. Err ↓	S_{NR} ↓	Tok./task
1	Claude Opus 4.8	38/39 (97.4%)	**0.2780**	**0.200**	**82.4%**	**0.179**	**0.566**	17.7k
2	Gemini 3.1 Pro	38/39 (97.4%)	0.3703	0.367	76.5%	0.231	0.754	145.6k
3	Qwen 3.6 35B	31/39 (79.5%)	0.4474	0.567	63.2%	0.276	0.934	140.7k
4	Kimi K2.6	38/39 (97.4%)	0.4754	0.567	65.7%	0.230	0.935	266.4k
5	GPT-5.5	37/39 (94.9%)	0.5610	0.533	**82.4%**	0.224	1.324	12.9k
6	Gemma 4 26B	**39/39 (100.0%)**	0.6442	0.767	55.9%	0.313	1.267	32.4k

Realistic-composition exam. We report results on a 100-scene realistic-composition exam sampled from a 953-scene dataset. Task-family weights are tilted towards real-world distributions (App. A.7), hence we term this the *realistic exam*. Exam contains 28 R1, 51 R2, and 21 R3 tasks; observation variants split 48/32/20 across the three increasing difficulties (`clean`/`proxy`/`proxy_hard`); outcomes split 59 continuous / 41 binary. The motif breakdown (with grafted scenes split by graft count) is in

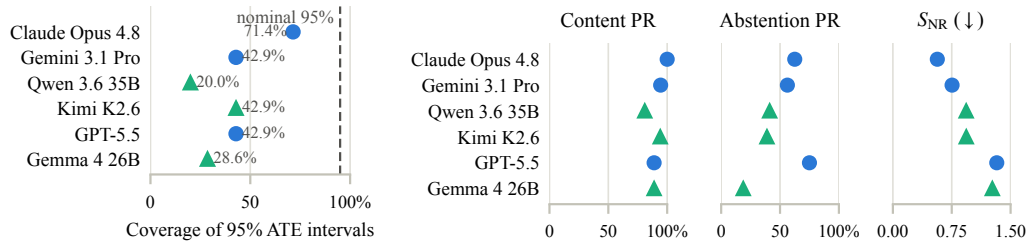
App. A.10. We evaluate six models: three frontier closed models (Claude Opus 4.8, Gemini 3.1 Pro, GPT-5.5), each run at high reasoning effort, and three open-weight models (Qwen 3.6 35B, Kimi K2.6, Gemma 4 26B), run at their model defaults. The step limit was 100 and per-task cost limit was $10[4]. For the specific experimental settings, refer to App. A.15.

Headline results Table 3 presents the aggregate metrics and is ranked according to the *CausalDS* score (Eq. 4). Claude Opus 4.8 leads the field (0.278) and sweeps the quality column s: it is best or tied-best on Pass Rate (82.4%), Med. NRel. Err (0.179), and S_{NR} (0.566). Two frontier reasoning models take the top two composite spots (Claude Opus 4.8 then Gemini 3.1 Pro at 0.370); the open models occupy a narrow middle band — Qwen 3.6 35B (0.447) and Kimi K2.6 (0.475) separated by less than 0.03 — with the smaller Gemma 4 26B last (0.644). The most informative

[4]Where available, never hit in practice.

dissociation is GPT-5.5: it ties Claude Opus 4.8 for the best raw Pass Rate (82.4%) yet ranks only fifth on CausalDSScore (0.561), recovering to third on the exam-relative Composite Rank (0.533). GPT-5.5 spotlights the difference between the composites: its continuous estimates are the worst-calibrated among the models ($S_{\mathrm{NR}} = 1.324$; mean NRel. Err on `proxy_hard` of 3.10, Tab. 6). CausalDSScore sums the pools by magnitude, so that heavy tail dominates and drops it to fifth; Composite Rank averages the ranks, so the tail has less of an effect (per-axis rank breakdowns: Tab. 14, App. A.10). Gemma 4 26B commits an answer on every continuous task (39/39 valid), but pays with the highest normalized error (Med. NRel. Err 0.313) and the lowest Pass Rate, landing last; Qwen 3.6 35B has lowest valid-continuous answer rate (79.5%) by failing more of the hardest estimands — mostly through its own poor tool-output management: it has a tendency to dump the whole dataset to output instead of using `head`/`tail`. This persists even with several re-runs where we manually truncate its tool output (App. A.15).

For this exam, the differentiation is concentrated at Rung 2 and on uncertainty quantification: R2 Pass Rate spans 28.6% (Gemma 4 26B) to 92.9% (GPT-5.5) (App. A.10, Tab. 13) and *identification* Pass Rate ranges from 28.6% (Gemma 4 26B) to 100.0% (Claude Opus 4.8); empirical coverage of 95% ATE intervals collapses to 20.0%–71.4% — that is, the models are *overconfident* (Fig. 3a, Tab. 6), with Claude again performing the best. This is consistent with recent uncertainty-elicitation benchmarks: nominal 95% credible intervals from GPT-5-family models cover only 9–44% of ground truth in Bayesian-elicitation evaluations (Hobor et al., 2026), nominal 99% intervals cover 65% on Fermi-style estimation (Epstein et al., 2025), and elicited probabilistic priors are systematically overconfident (Renda et al., 2025).



(a) (b)

Figure 3: (a) Empirical coverage of the nominal 95% ATE intervals per model over the identifiable `ate_uq_95` tasks (n=7 per model; 5 for Qwen 3.6 35B). (b) Content pass rate, abstention pass rate, and S_{NR} (Eq. 3; lower is better) per model. Models are ordered by CausalDSScore (best at top); blue circles — frontier (closed) models, green triangles — open-weight models.

Tool-use strategy splits the field (Fig. 4): GPT-5.5 (2.1 calls/task) and Claude Opus 4.8 (3.4) are near one-shot, while Gemini 3.1 Pro, Kimi K2.6, and Qwen 3.6 35B iterate heavily (11–18 bash calls per task, App. A.10, Tab. 18). Recent agent benchmarks likewise find that the number of tool calls a model makes per task varies strongly across model families and benchmarks, and is not predicted by overall capability (Wang et al., 2025; Xu et al., 2026a; Zhang et al., 2026; Xu et al., 2026b). Token usage spans more than an order of magnitude, from 12.9k tokens/task (GPT-5.5) to 266.4k (Kimi K2.6). Notably, the two model classes are linearly separable in the cost–quality plane (the dashed line in Fig. 4): at comparable token budgets, the closed models outscore the open ones.

Content correctness and abstention separate. Pass Rate decomposes into content and abstention slices (Tab. 4, Fig. 3b). Content correctness is high across the whole field (81.0%–100.0%); the differentiating axis is abstention, which spans 18.8%–75.0%. The split tracks the **frontier/open-weights model divide**: the three frontier models lead abstention — GPT-5.5 (75.0%), Claude Opus 4.8 (62.5%), Gemini 3.1 Pro (56.2%) — while the open models trail (Qwen 3.6 35B 41.2%, Kimi K2.6 38.9%, Gemma 4 26B 18.8%), with the smaller Gemma especially struggling. The two extremes are instructive: Claude Opus 4.8 is the only model with perfect content (100.0%, 18/18) and pairs it with solid abstention, whereas Gemma 4 26B over-commits, matching the field on content (88.9%) but declining almost nothing when it should (18.8% abstention) and finishing last overall. Thus knowing when to abstain is a separate, advanced skill. As discussed in Sec. 3.7, pool sizes vary across

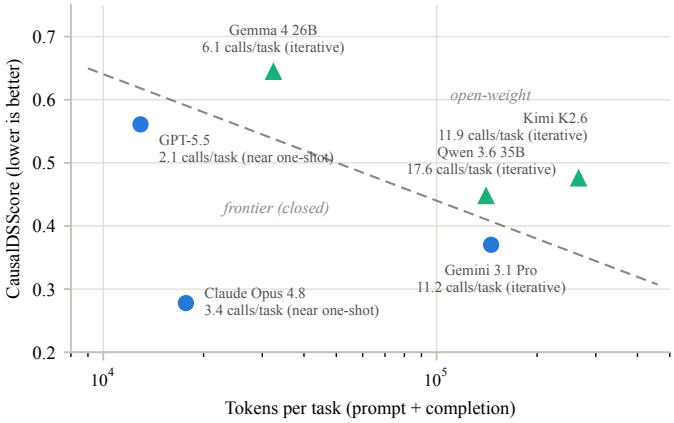


Figure 4: Tokens per task (log scale) versus CausalDSScore (lower is better) on the realistic exam. Each point is one model, annotated with its mean bash calls per task and interaction style; the dashed line separates the frontier (closed) models from the open-weight ones.

models because routing into the abstention pool depends on the model's own answer. Abstention is the live discriminator on both upper rungs, which is presented in App. A.10.

Table 4: Pass Rate decomposition. Abstention pass rate (PR) is over tasks where either the ground truth is non-identifiable *or* the model chooses to abstain - hence the number (N) of tasks is dynamic.

Model	Pass Rate	Content PR (N)	Abstention PR(N)
Claude Opus 4.8	**82.4%**	**100.0% (18)**	62.5% (16)
GPT-5.5	**82.4%**	88.9% (18)	**75.0% (16)**
Gemini 3.1 Pro	76.5%	94.4% (18)	56.2% (16)
Kimi K2.6	65.7%	94.1% (17)	38.9% (18)
Qwen 3.6 35B	63.2%	81.0% (21)	41.2% (17)
Gemma 4 26B	55.9%	88.9% (18)	18.8% (16)

Imperfect observation and uncertainty quantification drive a second axis of failure. Tables 5 and 6 present the discrete and continuous aggregates separated by the observation layer hardness (with `clean` denoting no observation layer — all conceptual variables directly measured), as well as the performance on the 95% confidence interval for the ATE subset of questions. We observe the expected trends: increasing observation difficulty tends to increase the difficulty, and disproportionally so for the weaker models.

The results also reveal a failure mode invisible in the aggregate statistics, which takes the median. Under the hardest observation the *mean* NRel. Err runs far above the median for most models — Gemma 4 26B blows up to a mean of 4.59 against a median of 0.348 (a $\sim 13\times$ gap), and GPT-5.5 to 3.10 (vs. 0.222): a handful of estimates on noise-corrupted data miss by orders of magnitude.The clear exception is again Claude, whose mean stays controlled at 0.557 (against a median of 0.183) A similar miscalibration also shows up in interval coverage, where empirical coverage of the 95% ATE intervals collapses to 20.0%–71.4% against the nominal 0.95, with Claude Opus 4.8 the best (71.4%) and Qwen 3.6 35B the worst (20.0%).

However, the per-observation layer hardness results in Tabs. 5 and 6 are naturally aggregated over *different* sub-exam slice compositions (cf. App. A.7). A controlled ablation that re-evaluates the *same* conceptual scene/task under all three released observation views (App. A.13) confirms the hardness ordering within matched scenes and separates two effects: the estimates that do get produced become substantially worse, and the harder views additionally cause outright failures to answer — context-window exhaustion, unwarranted abstention — alongside more calls per task.

Table 5: Pass Rate by observation variant. Bracketed numbers are pool sizes: a fixed count in a column header applies to every row, while *(N)* marks columns whose pool size varies by model, with the per-row count shown next to each value.

Model	PR_{clean} (N)	PR_{proxy} (13)	PR_{hard} (N)
Claude Opus 4.8	**88.9%** (18)	**84.6%**	33.3% (3)
Gemini 3.1 Pro	83.3% (18)	76.9%	33.3% (3)
Qwen 3.6 35B	63.2% (19)	**84.6%**	16.7% (6)
Kimi K2.6	63.2% (19)	76.9%	33.3% (3)
GPT-5.5	**88.9%** (18)	76.9%	**66.7%** (3)
Gemma 4 26B	50.0% (18)	69.2%	33.3% (3)

Table 6: Normalized error by observation variant and uncertainty quantification (*ATE-UQ95 cov.* — empirical coverage of 95% ATE intervals). Pool-size conventions as in Tab. 5.

Model	Med. $\text{NRel}_{\text{clean}}$ (N)	Med. $\text{NRel}_{\text{proxy}}$ (7)	Med. $\text{NRel}_{\text{hard}}$ (N)	Mean $\text{NRel}_{\text{hard}}$ (N)	ATE-UQ95 cov. (N)
Claude Opus 4.8	0.134 (16)	**0.099**	0.183 (15)	**0.56** (15)	**71.4%** (7)
Gemini 3.1 Pro	**0.122** (16)	0.301	0.238 (15)	1.41 (15)	42.9% (7)
Qwen 3.6 35B	0.223 (15)	0.381	0.276 (9)	1.58 (9)	20.0% (5)
Kimi K2.6	0.223 (15)	0.302	**0.173** (16)	1.49 (16)	42.9% (7)
GPT-5.5	0.171 (16)	0.224	0.222 (14)	3.10 (14)	42.9% (7)
Gemma 4 26B	0.212 (16)	**0.099**	0.348 (16)	4.59 (16)	28.6% (7)

Repeated-attempt stability, pass@k and pass^k–like metrics. We bracket each *open-weight model*'s behavior over k=3 independent restarts by simplified *pass@k*– and *pass^k*–like metrics. Namely, instead of estimating the population parameter as in Chen et al. (2021); Yao et al. (2024), we look at whether the task is solved by *any* of the k or, correspondingly, *all* of the k repeats; more broadly, we take the *best* or the *worst* case, respectively.

In this case, the scoring uses *ground-truth-fixed pools* (a deviation from the answer-dependent main routing) so that the runs are compatible: only ground-truth non-identifiable cases go into the "abstention pool" ; thus k=1 matches the main result run; further details are presented in App. A.11.

For the binary outcomes (Tab. 7, n=34), pass@3 reaches 76.5–91.2% — Kimi K2.6 91.2%, numerically matching the best frontier single-shot rate on the same pool — while pass^3 is within 50.0–61.8%. Thus, a 26–29-point gap exists between the best-case and worst-case performance.

Perhaps a more interesting metric is the variance of the CausalDS score itself on repeated attempts as the latter adds continuous-like answers to the binary ones captured by pass@k/pass^k. That score-level variance (7–27% of the mean; Tab. 7) is not benchmark instability: the point estimates reproduce near-exactly across restarts (median per-task NRelErr SD ≤ 0.002 over tasks answered in all 3 runs). It comes instead from a relatively few uncertainty-quantification tasks and from abstention-decision flips (App. A.11, Tab. 20). For the former, a few effect-interval estimands swing between near-correct and the cap across restarts, and because intervals enter the score through a capped *mean*, the swings pass through to the aggregate; for the latter, each flipped commit/abstain decision is re-scored in a small pass/fail pool, so when several flips land the same way within a run, the Pass Rate shifts noticeably instead of averaging out.

While we report the mean here (with Kimi significantly improving its first-try performance, mainly by getting a few of the above catastrophic mistakes right), we use the first try for the main results Table 3 so that the comparison with the frontier models is fair.

Table 7: pass@k and pass^k (solved by all k), *CausalDSScore* variability on the realistic exam for the open-weight models.

	pass@k (n=34)			pass^3	CausalDSScore		
Model	k=1	k=2	k=3	(all 3)	mean	SD	rel. SD
Kimi K2.6	67.6%	85.3%	**91.2%**	**61.8%**	0.375	0.102	27.1%
Qwen 3.6 35B	70.6%	73.5%	82.4%	55.9%	0.419	0.040	9.5%
Gemma 4 26B	55.9%	70.6%	76.5%	50.0%	0.646	0.045	7.0%

6 Discussion

CausalDS pairs hidden SCMs and their synthetic, graph-faithful verbalizations with SCM-derived ground truth — including each target's identifiability status — and an observation model that corrupts the released data view; thus, each exam scores abilities that existing benchmarks often isolate or conflate: symbolic causal reasoning, quantitative estimation, uncertainty quantification, epistemic abstention, and tool-use efficiency.

On the presented exam these axes demonstrably dissociate; however, we argue that all are integral to a well-rounded *causal data-science agent*.

Symbolic causal reasoning is the axis on which the field mostly converges: all six models recover structure essentially perfectly, and make the symbolic identification calls mostly without error. The dissociation opens downstream, along several fault lines. The frontier–open-weight separation lives on the epistemic axes — uncertainty quantification and abstention — and on tool use/efficiency: the frontier models lead abstention outright, and under the hardest observation variant, only Claude Opus 4.8 keeps the *mean* normalized error controlled. In tool use/efficiency, the frontier is literal: the two model classes are linearly separable in the cost–quality plane (Fig. 4). The reasoning–non-reasoning line likewise shows up on the epistemic axes while sparing the numerics: Gemma 4 26B, the one non-reasoning model, stays genuinely competitive at point estimation — it leads the open-weight models — and is exposed specifically on abstention and interval calibration. Finally, capability and reliability separate — given three attempts, the open-weight models can reach frontier territory, yet fail to do so consistently — and this line, too, runs along the epistemic axes: the run-to-run variance concentrates in interval estimands and abstention flips, while repeated point estimates reproduce closely.

The main future work is therefore to expand the benchmark along the many axes it already exposes: larger and more varied exam compositions, deeper trajectory-level failure taxonomies, and more systematic stress tests via targeted exams for abstention, quantitative skills, and counterfactual reasoning.

CausalDS thus poses a question that neither symbolic causal benchmarks nor data-science benchmarks alone can answer: given a story, the structure it implies, and an imperfect view of the data, what is an agent's answer worth? On the presented exam, the models largely master the parts — reading the structure, producing the estimates — and part ways on the whole: knowing how good the estimate is, and whether any answer is licensed at all. A competent causal data-science agent must clear every axis at once, and CausalDS is built to notice when it does not. That causal data-science competence decomposes this way is the empirical finding, with Claude Opus 4.8 being the closest to a well-rounded *causal data-science agent*.

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, 2026. URL https://openreview.net/forum?id=uQzPkWvTyo. Submitted to ICLR 2026.

Serafim Batzoglou. ReplaySCM: A Benchmark for Executable Causal Mechanism Induction from Interventions, May 2026. URL http://arxiv.org/abs/2605.08197. arXiv:2605.08197 [cs.LG].

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. *EMNLP*, pp. 2096–2115, 2025. doi: 10.18653/v1/2025.emnlp-main.107. URL https://doi.org/10.18653/v1/2025.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].

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 2026. doi: 10.48550/arXiv.2607.04293. URL http://arxiv.org/abs/2607.04293. arXiv:2607.04293 [cs]. Oral presentation.

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.

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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.815. URL https://aclanthology.org/2024.findings-emnlp.815/.

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. 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, 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*, 2024. 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. URL https://openreview.net/forum?id=vyflgpwfJW.

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.

Yev Meyer and Dane Corneil. Nemotron-Personas-USA: Synthetic personas aligned to real-world distributions, June 2025. URL https://huggingface.co/datasets/nvidia/Nemotron-Personas-USA.

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].

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'. 2026.

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. 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.

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, Beicheng Huang, Jibin Wu, Liang Feng, and Kay Chen Tan. CausalBench: A Comprehensive Benchmark for Causal Learning Capability of LLMs. arXiv, September 2024. doi: 10.48550/arXiv.2404.06349. URL http://arxiv.org/abs/2404.06349. arXiv:2404.06349 [cs].

A Appendix

A.1 Code and data availability

The CausalDS source code — the scene-generation pipeline, the evaluation harness, and the grader — is available at `github.com/andleb/causalds`. The same repository also provides the entire datasets used for the main exam and the ablations presented in this work.

A.2 Scene generation details

Noise families and link functions. The continuous and binary SCM profile registries are documented in App. A.7. For the noise: continuous noise families include Gaussian, Laplace, Student-t, and Gaussian mixtures; binary mechanisms produce a Bernoulli output through a logistic, threshold, or noisy-gate link with no explicit noise term — the Bernoulli draw is the source of stochasticity.

A.3 Scene synthesis sub-algorithms

The top-level scene-synthesis control flow (Alg. 1) calls three sub-routines: PREAUDIT (Alg. 2) for the LLM-based feasibility check on CauseNet seeding, MAPSTAGE (Alg. 3) for naming the (sub)graph variables with audit/repair, and VERBALIZE (Alg. 4) for story generation with a story-to-DAG verifier loop.

Algorithm 2 LLM pre-audit for CauseNet seeding

Input: stage graph H with CauseNet-fixed nodes F.
Output: feasibility verdict (`feasible`, `confidence`, `reason`).
Extract directed edges E, all unordered non-edges $\mathcal{N}$, and the fixed–fixed subset $\mathcal{N}_F \subseteq \mathcal{N}$.
Prompt with fixed nodes, remaining placeholders, E, $\mathcal{N}_F$, and $\mathcal{N}$.
Ask whether a skilled mapper could interpret the fixed concepts and name placeholders so the graph constraints remain plausible.
Reject iff a fixed–fixed non-edge has an unavoidable mainstream direct-cause reading, or a fixed–fixed edge is impossible.
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)
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.
The mapper proposes M under fixed-name, completeness, and uniqueness constraints.
for $t = 1$ to T **do**
    The auditor checks edge plausibility, absence of plausible direct links for non-edges, and conditional independence/type-consistency signals.
    if no violations remain **then**
        return M
    end if
    if a violation cannot be fixed without changing fixed names **then**
        return failure
    end if
    The mapper regenerates M from a feedback prompt provided by the auditor, listing the violations.
end for
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.
Build yellow-flag context Φ by scanning M for derived-name markers and restrictive qualifiers (cf. Sec A.4)
The story writer drafts σ from (D, M, Φ).
for $s = 1$ to S **do**
    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.
    if there are no hard violations **then**
        return σ
    end if
    The story writer revises σ from the verifier's feedback listing the hard violations, warnings, and Φ.
end for
return failure.

A.4 Mapping and narration audit details

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 restrictive qualifiers (e.g. "based only on", "not influenced by other factors"), and appends a yellow-flag block pointing the auditor at the suspicious nodes. This targets a recurring failure mode in which a broad latent construct is narrated as a direct cause of an administrative artifact (e.g. a construct narrated as directly causing a record ID meant to keep track of it).

A.5 Motif catalog

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

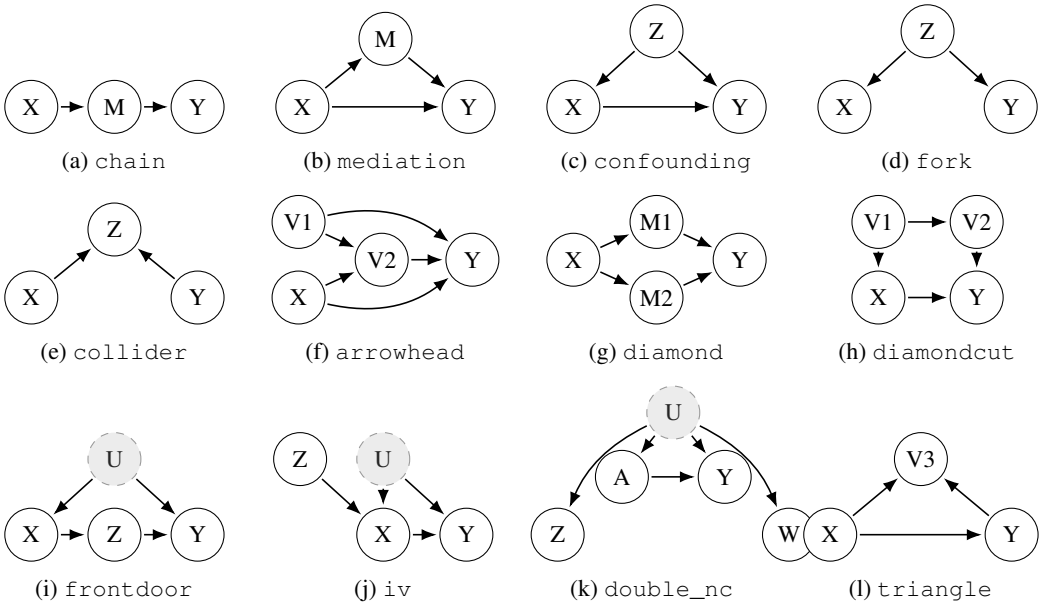


Figure 5: Base motif templates used by the sampler. Latent variables are shown as gray dashed nodes.

A.6 Observation-model implementation details

Mechanism families and noise. Continuous measurement mechanisms h_r are drawn from the same families used for SCM nodes: a handcrafted form $h_r(z) = wz + b + w_{\text{nl}} \tanh(z - c) + w_{\text{pair}}\, z^2$ (linear plus shifted nonlinearity plus interaction) and a small spectral-normalized neural network. The shift $c \sim U(-1.5, 1.5)$ is drawn independently per measurement and moves the cancellation point of the $\tanh$ derivative away from the realized treatment-contrast region, where $h_r'(z) \approx 0$ would otherwise create a flat region with vanishing Fisher information (cf. "*Admissibility checks*" below). The continuous noise family and binary link function selections are drawn from the same pool as for the main SCM (cf. App. A.2). When the noise is chosen to be *heteroscedastic*, σ is replaced with $\sigma(\mathbf{z}) = \text{softplus}(w_\sigma^\top \mathbf{z} + b_\sigma)$, clamped to $[\sigma_{\min}, \sigma_{\max}]$, so the Fisher denominator is location-dependent: $I_r(z) = (h_r'(z))^2 / (\sigma_0\, \sigma(z))^2$.

Admissibility checks on candidate bundles. For each candidate measurement bundle on a latent Z, we evaluate $I(z) = \sum_r I_r(z)$ on a grid over the realized support of Z and accept the bundle only if: (i) $\min_z I(z) \geq \tau_{\min}$ (no dead zones), (ii) $\bar{I} \geq \tau_{\text{avg}}$ (sufficient average), and (iii) lower-tail quantiles ($I_{0.10}$, $I_{0.25}$) clear separate thresholds. Scenes that also place a measurement bundle on the outcome apply the same checks with stricter thresholds. The resampler retries the mechanism draw up to a configured budget; persistently failing scenes are rejected before any downstream verbalization. The setting used for the results presented is $\tau_{\min} = 0.03$.

Recoverability diagnostics in the private bundle. Beyond Fisher-information filtering, we privately characterize each scene by recoverability diagnostics computed on a shared holdout, expressed as R^2 for continuous conceptual variables and AUC for binary ones so that scores are comparable across scenes. The *upper bound* (intrinsic recoverability) trains a gradient-boosting model on all d_j measurements jointly using the full conceptual-plus-measurement dataset—labels no benchmark agent ever sees—giving an *oracle-like* ceiling. The *lower bound* (naive baseline) fits the best single-measurement *linear* model from the public calibration set only (OLS for continuous targets, logistic regression for binary). The gap between the two bounds decomposes scene difficulty into a misspecification component and a multi-measurement information-gain component:

$$\Delta \;=\; \underbrace{(\mathcal{L}_{\text{naive}} - \mathcal{L}_{\text{single,oracle}})}_{\text{misspecification gap}} + \underbrace{(\mathcal{L}_{\text{single,oracle}} - \mathcal{L}_{\text{cal}})}_{\text{multi-measurement information gain}}\;, \tag{5}$$

where $\mathcal{L}_{\text{single,oracle}}$ is the best single-measurement *nonlinear* loss. A large gap implies large potential gains from a skilled data-science agent; a small gap means a single column captures most of the recoverable signal. The upper bound itself caps even an oracle's downstream performance. Both bounds and their per-variable gaps are recorded for analyis. However, these recoverability scores are diagnostic summaries rather than hard acceptance criteria; admissibility is still determined at the mechanism level, as described above. Fig. 6 shows the resulting per-scene summary on a representative scene.

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. The key design choice is to ground *axes*, not to imitate any one source wholesale: we derive axis-specific empirical anchors from several benchmark corpora and structural companions, weight them by how directly they expose the quantity of interest, and then allow small benchmark-design tilts when needed for difficulty, coverage, or stress-testing. 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 which scenes get sampled and built; exam axes (P(QuestionType | structural label), P(OutputVariant | QuestionType), P(observation variant | QuestionType)) shape which tasks and which released measurement view are selected per scene at exam time. This split lets the same scene support multiple released exams without re-running expensive verbalization.

Because we cannot estimate the full joint $P(\text{motif}, \text{question type}, \text{output variant}, \text{observation variant})$ from available data, we factor the *exam side* through the question type. Conditional on the question

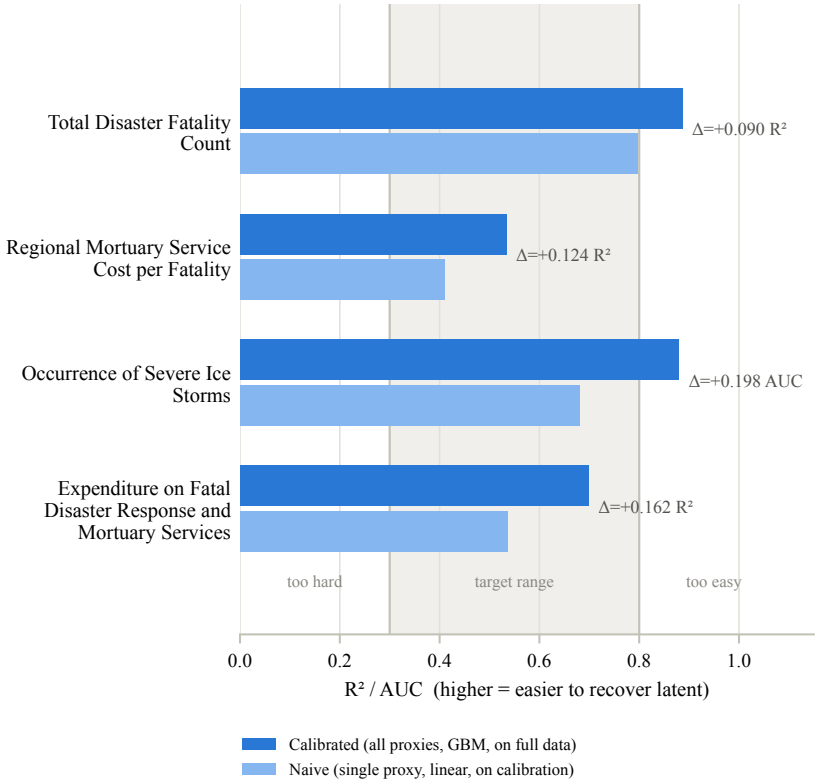


Figure 6: Per-scene recoverability diagnostic for the `proxy_hard` variant of one scene with four measured conceptual variables (three continuous, one binary). Light blue bars show the naive lower bound (best single-measurement linear/logistic fit on the public calibration set); dark blue bars show the calibrated upper bound (gradient-boosting fit on the full latent-plus-measurement data the agent never sees). The shaded band marks the target window (0.3–0.8) used during scene generation: below it is "too hard," above it "too easy." The Δ next to each pair is the upper-minus-lower gap (Eq. 5).

type Q, the output variant is drawn from the answer-contract families compatible with Q and the observation variant from the per-question-type observation distribution, with the two choices treated as independent given Q:

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

The corresponding graphical factorization is shown in Fig. 7.

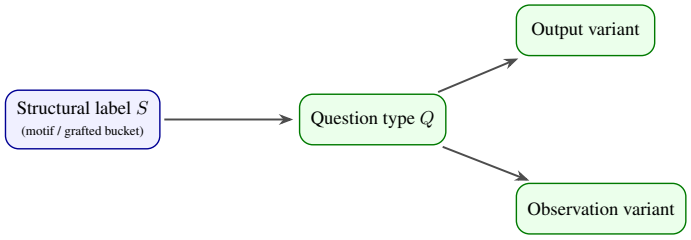


Figure 7: Exam-composition factorization. Conditional on the structural label S (motif or the synthetic `grafted` bucket), the exam builder draws the question type Q, then independently draws the output variant and the observation variant given Q. Each conditional corresponds to one exam-side axis: $P(Q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

For **variable types**, we treat treatment and outcome separately and estimate only the coarse continuous-versus-non-continuous split that matters for scene generation. The empirical anchor

comes from row-level causal benchmarks whose treatments and outcomes can be typed from data columns, metadata, or variable descriptions: *EconCausal*, *CauSciBench* , *CausalReasoningBenchmark*, and *CauSciBench*'s QR/textbook source (Lee et al., 2026; Acharya et al., 2026; Sawarni et al., 2026; Liu et al., 2024). We then combine those sources using credibility weighting based on, e.g., how well the dataset's constructs map to ours. In production we nudge this empirical mixture slightly toward more continuous variables: some high-cardinality discrete variables are better modeled as effectively continuous in downstream estimation, and a modest tilt increases data-analysis difficulty without changing the underlying causal semantics.

For **motif composition and grafting**, real-study causal benchmarks (*CausalReasoningBenchmark*, *CauSciBench*, *EconCausal*, and *InterveneBench*) typically expose *design-family* labels (e.g., adjusted observational, DiD/event-study, RD, IV, matching) rather than explicit graph motifs. We therefore map those design-family histograms into our motif inventory through an explicit crosswalk, producing source-specific motif distributions first and deferring any global mixture to a separate weighting step (Sawarni et al., 2026; Acharya et al., 2026; Lee et al., 2026; Shi et al., 2026). We use real-paper graph corpora (*ReCITE*) differently: they are not benchmark-side design-family sources, but they do expose explicit graph structure, which lets us calibrate the split over observed-effect motifs, auxiliary motif tendencies, and graft-complexity indicators (Saklad et al., 2025). Synthetic motif benchmarks (*CLadder*) can still be useful here, but only as additions to ensure coverage, rather than frequency priors (Jin et al., 2023). The auxiliary motif pool and graft-count distribution are derived separately. The graft-count prior is hand-authored with a small empirical tail-pull rather than taken from corpus frequencies, as the available real-paper structural complexity indicator (*ReCITE*) violates $P(2 \text{ grafts}) < P(1 \text{ graft})$, which we believe to be structurally justified.

For **identifiability**, real-world data on non-identifiable cases is limited: empirical benchmark sources (*EconCausal*, *CauSciBench*'s real-study subset) deal almost entirely with identifiable cases (99.6% when pooled), with the small remainder unresolved rather than explicitly non-identifiable (Lee et al., 2026; Acharya et al., 2026). Challenge suites (*CausalGame*, *CausalPitfalls*) inform the *kinds* of identification obstacles worth representing, but not their field frequency (Chen et al., 2026; Du et al., 2026). `double_nc` graphs carry latent treatment–outcome confounding, and a generic instrumental variable is not accepted as identifying the population ATE; their frequency follows the motif priors above. Second, we request a deliberate *non-identifiable slice* (10% of generated scenes for the dataset presented) on otherwise-identifiable motifs, realized by rejection sampling: we inject latent confounding $U \to \{X, Y\}$ and accept only graphs whose population-ATE identifiability check returns `False`. Together the two routes make roughly 30% of realistic-exam scenes non-identifiable, keeping abstention a first-class evaluation axis.

SCM mechanism profiles are derived from available empirical evidence: real biochemical-equation and Boolean-rule corpora (*BioModels*, *SABIO-RK*, *BiodivineBooleanModels*, and the *Kadelka DesignPrinciples* corpus) (Malik-Sheriff et al., 2020; Wittig et al., 2018; Pastva et al., 2023; Kadelka et al., 2024); these are the only available corpora that expose per-equation or per-rule mechanism rows, so they alone contribute counts (with a partial-export weight applied where coverage is incomplete). Source-native classes are filtered through a small support floor and collapsed into a compact registry of three continuous profiles (`linear_additive`, `interaction_response`, `symbolic_transform`) and two binary profiles (`single_regulator_logistic`, `multi_input_logic_gate`), so the registry stays domain-neutral despite a biology-heavy source pool. The resulting empirical distribution is heavily concentrated on multi-parent interaction shapes and multi-input gates, so the production mixture is the convex combination $\alpha\,\hat{p}_{\text{emp}} + (1-\alpha)\,p_{\text{cov}}$ with $\alpha = 0.3$ and a uniform synthetic coverage prior p_{cov} over additive, handcrafted, neural-black-box, mixture-noise, and heteroscedastic regimes (continuous), plus logistic, sharp-threshold, noisy-OR, noisy-AND, and signed-logic regimes (binary); a sixth continuous regime (coarsened-continuous) is implemented and configured but carries zero weight in the presented benchmark, as low-cardinality coarsened nodes proved brittle under the measurement-information admissibility check. The empirical component is deliberately demoted due to both the narrow domain concentration, as well as limited mechanism diversity . The two distributions are determined marginally by node type and are independent of motif class, since no public artifact ships row-level joint labels of mechanism by motif.

On the exam side, the **question-type prior** $P(Q \mid S)$ (**Exam Axis 1**) is built as a Bayes-style combination of an *empirical marginal* $\hat{p}_{\text{emp}}(Q)$ over task families sourced from real-world causal benchmarks that expose family labels: *CausalReasoningBenchmark*, *CauSciBench*, *EvidenceInfer-*

ence, *EconCausal*, and a synthetic *structural lift* derived from *CLadder* (Sawarni et al., 2026; Acharya et al., 2026; DeYoung et al., 2020; Lee et al., 2026; Jin et al., 2023), which is the only available source that ships joint graph × query type rows. The empirical marginal is mixed with a uniform deployable-family floor so that no rung-1 or rung-3 family is collapsed to zero mass under R2-heavy real-world empirical distribution; the structural lift is applied with a small exponent so that *CLadder*'s synthetic balance does not overpower the empirical ordering. Motifs absent from *CLadder* (notably the grafted bucket) fall back to the deployable base bucket.

For the **output-variant prior** $P(\text{OutputVariant} \mid \text{QuestionType})$ (**Exam Axis 2**), we keep an explicit manual crosswalk between source-native answer contracts and our internal output-variant set. The weighted conditional is driven only by sources with multiple possible output variants: *CausalReasoningBenchmark* and *CauSciBench* (both its real-study and its textbook-derived QR subsets) (Sawarni et al., 2026; Acharya et al., 2026). In practice this covers three question families (`causal_sketch`, `identification`, `effect_estimate`); for every other family no empirical row exists, and the exam builder draws the output variant uniformly over that family's compatible variants.

For the **observation-model variant**, the deployed three-class set $\{\texttt{clean}, \texttt{proxy}, \texttt{proxy\_hard}\}$ is grounded against real-data sources that expose per-row accessibility fields. *ReCITE* (article-level `explicitness` on causal-graph rows) and *EvidenceInference* (per-prompt valid-evidence location on biomedical RCT prompts) provide direct rows for graph-recovery and effect-estimation question families; *CausalReasoningBenchmark* (paired precise/vague prompt rows) and *TellMeWhy* (Answerable / Not Answerable real why-questions) act as additional companions (Saklad et al., 2025; DeYoung et al., 2020; Sawarni et al., 2026; Lal et al., 2021). A four-level ordered source vocabulary ($\texttt{clean} < \texttt{partly\_explicit} < \texttt{document\_context} < \texttt{external\_implicit}$) collapses these accessibility fields and bridges to the deployed three-class set; the two hardest source levels both map onto the deployed `proxy_hard` class. The exam-side prefill $P(\text{observation variant} \mid \text{QuestionType})$ uses direct rows where available, explicit borrowed rows for nearby families, and the empirical marginal $\hat{P}(\text{observation variant})$ as a fallback for question types without direct source labels.

A.8 Exam-side difficulty knob

After the priors of App. A.7 are built, a separate **difficulty knob** $d \in [0, 1]$ tilts 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 an editorial within-family difficulty rank $r_q(v)$ with midpoint c_q, and index observation variants by $o(\text{obs}) \in \{0, 1, 2\}$ for $\{\texttt{clean}, \texttt{proxy}, \texttt{proxy\_hard}\}$. Each tilt is multiplicative on the prefill before renormalization:

$$\begin{aligned} \hat{p}_d(Q \mid S) &\propto \hat{p}(Q \mid S) \exp\big(\beta\,(2d{-}1)\,(r(Q){-}2)\big), \\ \hat{p}_d(v \mid q) &\propto \hat{p}(v \mid q) \exp\big(\delta\,(2d{-}1)\,(r_q(v) - c_q)\big), \\ \hat{p}_d(\text{obs} \mid Q) &\propto \hat{p}(\text{obs} \mid Q) \exp\big(\gamma\,(2d{-}1)\,(o(\text{obs}){-}1)\big), \end{aligned} \tag{6}$$

In Eq. 6, $d = 0.5$ is a no-op, $d = 0$ pulls toward R1 / easier contracts / `clean`, and $d = 1$ pulls toward R3 / harder contracts / `proxy_hard`. The strength constants β, δ, and γ are kept separate so rung pressure, output-variant pressure, and observation pressure can be calibrated independently (all default to 1); single-variant families (e.g. `counterfactual_identification`) are unchanged on the output-variant axis. This is an editorial dial rather than a frequency claim: Pearl rung is only a coarse difficulty indicator and the within-family rank is editorial, while the observation ordering $\texttt{clean} < \texttt{proxy} < \texttt{proxy\_hard}$ is on firmer ground. The realistic exam uses $d = 0.5$, so its composition is determined entirely by the empirical priors of App. A.7 without any difficulty-knob tilt; sweeping d away from 0.5 is the intended way to produce easier or harder exam mixes from the same underlying priors.

A.9 Scoring rules by rung

The abstain convention from Sec. 3.7 applies whenever the target estimand is non-identifiable or the model abstains (routing details below); otherwise, the task-specific metric below is used. GT denotes SCM-derived ground truth. For continuous outcomes we use RMSE; for binary outcomes

ROC-AUC alongside the Brier score $\frac{1}{n}\sum(\hat{p}_i - y_i)^2$ and log-loss; for set- and graph-recovery tasks we use $F_1 = 2|\hat{S} \cap S|/(|\hat{S}| + |S|)$ (with $F_1 = 1$ when both sets are empty); for interval forecasts the Gneiting–Raftery proper interval score $\mathrm{IS}_\alpha(\hat{\ell}, \hat{u}; \tau) = (\hat{u} - \hat{\ell}) + \frac{2}{\alpha}(\hat{\ell} - \tau)\mathbf{1}[\tau < \hat{\ell}] + \frac{2}{\alpha}(\tau - \hat{u})\mathbf{1}[\tau > \hat{u}]$ (Gneiting & Raftery, 2007); and for the remaining content variants either exact match or absolute error. The interval score is strictly proper, with expected minimum at the true central (equal-tailed) $(1 - \alpha)$ predictive interval: the agent picks its own estimator and the resulting $(\hat{\ell}, \hat{u})$ are scored against the scalar parameter τ (or row-wise y_j for prediction intervals), so no reference interval is computed or stored. For pooling, each interval task contributes the normalized interval score $\mathrm{NRelIS}_i = \mathrm{IS}_\alpha^{(i)}/(1 + s_i)$, with the same normalizer s_i as in Eq. 1: $s_i = \mathrm{sd}(Y_{\mathrm{test}})$ for prediction intervals (row-wise scores averaged over the test set, $\alpha = 0.1$ for the requested central 90% interval) and $s_i = |\tau|$ for effect intervals ($\alpha = 0.05$ for the central 95% interval). Interval scores are aggregated by means at every level (over test rows within a task, and across tasks via the capped mean discussed below), since averaging preserves propriety.

Table 8: Rung 1 scoring rules.

Task	Outcome 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 pool routing. 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, replacing the per-variant content metric with a single uniform abstention binary $\mathrm{AbstMatch}_i$ (Sec. 3.7). This covers every variant whose estimand can be non-identifiable: the rung-2 `identification` family (both `identifiable_boolean` and `method_label`, and the four backdoor-adjustment-set variants), the rung-2 `effect_estimate` family, `bias_diagnostic.forbidden_controls_list`, and the rung-3 `counterfactual_identification`, `counterfactual_effect`, and `mediation_effect` variants. For the identification-type families this routing changes no score — the identifiability judgment is the content — it only assigns the task to the abstention pool so the same binary is counted once, uniformly across families; for the estimate-type families it replaces the content metric, which would be meaningless on a non-identifiable estimand. Each task therefore lives in exactly one pool: a wrong abstention call collects a 0 on the abstention binary but does not also collect a substituted content score (preventing partial-credit hallucinations on non-identifiable estimands). The R3 abstention check uses the per-estimand R3 identifiability (`counterfactual_identification`.⟨ETT,NDE,NIE⟩); for the

Table 9: Rung 2 scoring rules.

Task	Outcome 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/IV), 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.

dominance variant we require both NDE and NIE to be identifiable. Null-equivalent labels (none, null, unknown, missing field, empty string) are accepted interchangeably.

Every task variant feeds an aggregate metric. The benchmark currently samples 33 active (TaskType, OutputVariant) pairs across the rung-1/2/3 tables above, and every active variant contributes to at least one of the three leaderboard metrics (Pass Rate, Med. NRel. Err, Med. F_1-Loss). Pass Rate pools the 18 exact_match-graded content variants together with the abstention binary. Med. NRel. Err pools the 11 continuous-output variants in three groups: the two prediction variants (prediction.point_predictor, whose binary branch contributes $\sqrt{\text{Brier}}/(1 + \text{sd}(Y_{\text{test}}))$, and prediction.prediction_interval via the normalized interval score); the six effect variants (effect_estimate.{ate_point, ate_uq_95}, counterfactual_effect.{effect_point, effect_uq_95}, and mediation_effect.{effect_point, effect_uq_95}, with the three _uq_95 intervals scored by NRelIS); and the three association-strength points (association.{effect_size_point, delta_point} and collider_phenomenon.induced_association_strength_point), whose absolute errors are normalized by $1 + |\tau|$. Med. F_1-Loss pools the four F_1-graded variants: causal_sketch.{edges_only, skeleton_edges}, identification.

Table 10: Rung 3 scoring rules.

Task	Outcome 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 |NDE| exceeds |NIE|, whether |NIE| exceeds |NDE|, or whether they are tied within tolerance. If non-identifiable, 1 iff `dominant=null`.

`all_minimal_adjustment_sets`, and the identifiable branch of `bias_diagnostic.forbidden_controls_list`. The three pools cover all 33 pairs ($18 + 11 + 4$).

Aggregator choice for S_{NR}. The aggregator S_{NR} and its point/interval pools $T^{\mathrm{pt}}_{\mathrm{NR}}, T^{\mathrm{int}}_{\mathrm{NR}}$ are defined in Eq. 3. Point-graded tasks emit naturally bounded normalized errors, and a median over $T^{\mathrm{pt}}_{\mathrm{NR}}$ tracks typical-scene performance without being tail-driven. Interval-graded tasks emit a strictly proper score whose miss-penalty multiplier $2/\alpha$ is unbounded by design: a *single overconfident miss* can reach NRelIS values an order of magnitude above the typical-scene level, which is precisely the signal the score is meant to expose. Aggregating those by median would discard exactly that information; aggregating by raw mean would let one catastrophic miss dominate the composite. The *capped mean* $\mathrm{mean}\ \min(\mathrm{NRelIS}_i, c)$ with $c = 10$ preserves the proper-score-induced overconfidence penalty for typical misses (a $1 \times |\tau|$ miss incurs ≈ 4, well below the cap) while bounding any single task's contribution to c, so cross-exam stability does not depend on the single most extreme task.

A.10 Realistic-exam breakdowns

Tables in this section report the exam composition together with input-mode (symbolic vs. data-backed), pass-rate, per-rung, per-task-family, per-motif, and abstention-only views of the run summarized in Sec. 5 (the observation-variant view is located in the main text, Tabs. 5 and 6). *n* is the number of tasks in the slice; *Cont. PR* restricts to `exact_match`-graded discrete tasks; *Abst. PR* restricts to the abstention pool. Best per column in bold. All slices are marginal views of the same 100-task exam: the axes are not varied independently, so a difference along one axis can also reflect the task families, output variants, and observation variants that happen to fall into that slice.

Table 11: Composition of the realistic exam (100 scenes) by structural motif and task family. `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**

Table 12: Abstention by task family on the realistic exam: abstention pass rate (per-family abstention pool size in parentheses). "—": empty pool. Pool sizes vary across models because routing depends on the model's own answer. Best per row in bold.

Family	Claude Opus 4.8	Gemini 3.1 Pro	GPT-5.5	Qwen 3.6 35B	Kimi K2.6	Gemma 4 26B
`identification`	**100.0%** (5)	60.0% (5)	80.0% (5)	40.0% (5)	60.0% (5)	0.0% (5)
`bias_diagnostic`	0.0% (3)	66.7% (3)	**100.0%** (3)	33.3% (3)	0.0% (3)	0.0% (3)
`effect_estimate`	0.0% (3)	0.0% (3)	**100.0%** (3)	25.0% (4)	0.0% (3)	33.3% (3)
`counterfactual_id`	**100.0%** (3)	**100.0%** (3)	66.7% (3)	66.7% (3)	66.7% (3)	66.7% (3)
`counterfactual_effect`	**100.0%** (2)	50.0% (2)	0.0% (2)	50.0% (2)	66.7% (3)	0.0% (2)
`mediation_effect`	—	—	—	—	0.0% (1)	—

Abstention by family (Tab. 12). The frontier models perfectly solve abstention on several families while the open models do on none; the average rates likewise reflect this split. GPT-5.5 is perfect on the R2 IV / forbidden-control families (`effect_estimate`, `bias_diagnostic`) and Claude Opus 4.8 on `identification` and both counterfactual families, whereas Gemma 4 26B fails to abstain correctly on every single question for `identification`, `bias_diagnostic`, or `counterfactual_effect`.

Table 13: Per-rung headline aggregates over the realistic exam. R1 has no abstention pool. Best per column in bold.

	Rung 1 ($n = 28$)			Rung 2 ($n = 51$)				Rung 3 ($n = 21$)			
Model	PR	Cont. PR	NRel. Err	PR	Cont. PR	Abst. PR	NRel. Err	PR	Cont. PR	Abst. PR	NRel. Err
Claude Opus 4.8	**100.0%**	**100.0%**	0.319	57.1%	**100.0%**	45.5%	0.175	**100.0%**	**100.0%**	**100.0%**	0.028
Gemini 3.1 Pro	**100.0%**	**100.0%**	0.301	57.1%	**100.0%**	45.5%	0.248	88.2%	91.7%	80.0%	**0.027**
Qwen 3.6 35B	75.0%	75.0%	**0.276**	46.7%	**100.0%**	33.3%	1.061	73.7%	78.6%	60.0%	0.099
Kimi K2.6	**100.0%**	**100.0%**	0.277	42.9%	**100.0%**	27.3%	0.275	77.8%	90.9%	57.1%	0.118
GPT-5.5	**100.0%**	**100.0%**	0.287	**92.9%**	**100.0%**	**90.9%**	**0.171**	70.6%	83.3%	40.0%	0.050
Gemma 4 26B	**100.0%**	**100.0%**	0.361	28.6%	**100.0%**	9.1%	0.596	70.6%	83.3%	40.0%	0.049

Per-rung aggregates (Tab. 13, Fig. 8). Rung-2 content is solved by every model (Cont. PR 100% across the field), so the entire R2 Pass-Rate spread (28.6–92.9%) is carried by the abstention pool: the identifiability call, not the execution of an identified analysis, is what separates the models. R1 is saturated for five of the six models (Qwen 3.6 35B at 75.0%), leaving the prediction-error band (0.276–0.361) as the only R1 signal. At R3 the ordering inverts for GPT-5.5 — best R2 abstention (90.9%) but joint-worst R3 abstention (40.0%) — while Claude Opus 4.8 is the only model perfect on the binary R3 columns; the surviving R3 point estimates are quite accurate for every model (NRel. Err 0.027–0.118)[5].

[5]Note that this can be due to other factors, such as the task family, outcome variant, and observation-layer difficulty drawn for the R3 pool: the per-rung slices differ in composition, not only in rung.

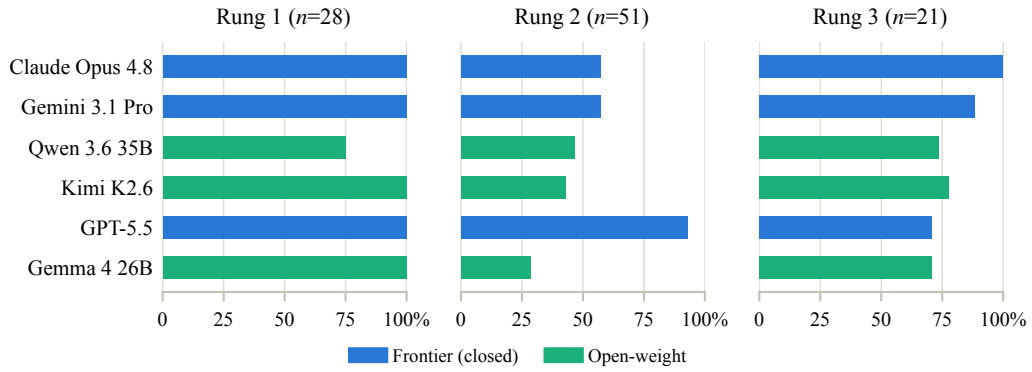


Figure 8: Pass Rate per rung and model on the realistic exam; pool sizes n in the panel titles. Models are ordered by CausalDSScore (best at top). Blue — frontier (closed) models; green — open-weight models.

Table 14: Per-axis **Composite Rank** for the realistic exam. Within each axis, models are reranked over the tasks in that slice and the per-column average rank is normalized to $[0, 1]$ (with 0 being best), using the same three pools as CausalDSScore (Pass Rate, S_{NR}, Med. F_1-Loss). *Overall* repeats the exam-wide Composite Rank of Tab. 3; rows are sorted by it (tie broken by CausalDSScore). Best (lowest) per column in bold.

	Rung			Observation variant			
Model	R1	R2	R3	clean	proxy	proxy_hard	Overall
Claude Opus 4.8	0.500	**0.267**	**0.233**	0.400	**0.200**	**0.333**	**0.200**
Gemini 3.1 Pro	**0.300**	0.400	**0.233**	**0.300**	0.633	0.400	0.367
GPT-5.5	0.567	0.500	0.667	0.333	0.500	0.500	0.533
Qwen 3.6 35B	0.567	0.567	0.633	0.733	0.533	0.700	0.567
Kimi K2.6	0.633	0.500	0.633	0.467	0.567	0.467	0.567
Gemma 4 26B	0.433	0.767	0.600	0.767	0.567	0.600	0.767

Per-axis Composite Rank (Tab. 14). The frontier leaders split the axes by difficulty: Claude Opus 4.8 is best-ranked on the harder slices (R2, R3, `proxy`, `proxy_hard`), Gemini 3.1 Pro on the easier ones (R1, `clean`, tying on R3), and no open-weight model takes any column. With sketch F_1 saturated for every model, the remaining rank differentiator is the interval-sensitive S_{NR} pool, where GPT 5.5's poorly calibrated intervals cost it (cf. Tab. 6).

Table 15: Per-model performance by task input mode on the realistic exam (*symbolic* vs. *data-backed*; see text). Pass Rate pools content `exact_match` and abstention binaries; symbolic tasks carry no continuous estimate (NRel. Err undefined) and data-backed tasks carry no structure-recovery F_1 (Med. F_1-Loss undefined).Task counts in parentheses; best per column in bold. Invalid continuous answers are excluded.

	Symbolic ($n = 49$)		Data-backed ($n = 51$)	
Model	Pass Rate	Med. F_1-Loss	Pass Rate	Med. NRel. Err
Claude Opus 4.8	86.4% (22)	0.000 (27)	**75.0%** (12)	**0.179** (38)
Gemini 3.1 Pro	81.8% (22)	0.000 (27)	66.7% (12)	0.231 (38)
Qwen 3.6 35B	72.7% (22)	0.000 (27)	50.0% (16)	0.276 (31)
Kimi K2.6	72.7% (22)	0.000 (27)	53.8% (13)	0.230 (38)
GPT-5.5	**90.9%** (22)	0.000 (27)	66.7% (12)	0.224 (37)
Gemma 4 26B	54.5% (22)	0.000 (27)	58.3% (12)	0.313 (39)

Symbolic vs. data-backed (Tab. 15). The tasks can also be divided into those that require the use of the provided data (*data-backed*) and those that can be solved symbolically: `causal_sketch`, `identification`, `counterfactual_identification`, and the forbidden-controls `bias_diagnostic` variant are solvable from the story-implied graph alone. Symbolic reasoning is led by GPT-5.5 (90.9% Pass Rate), Claude Opus 4.8 (86.4%), and Gemini 3.1 Pro (81.8%), with the open models trailing (Qwen 3.6 35B and Kimi K2.6 tie at 72.7%, Gemma 4 26B last at 54.5%). On data-backed tasks Claude Opus 4.8 leads both Pass Rate (75.0%) and continuous-estimate error (lowest at 0.179 NRel. Err), with Gemini 3.1 Pro and GPT-5.5 tied at 66.7% and the open models between 50.0% and 58.3%. The cross-mode anti-correlation seen among the open models — Gemma 4 26B is symbolic-worst yet edges Qwen and Kimi on data-backed Pass Rate — does not extend to the frontier models, which are strong on both, Claude Opus 4.8 most of all. Structure recovery is saturated (Med. F_1-Loss $= 0$ for all models), so symbolic differentiation comes entirely from `identification`/`counterfactual_identification` correctness.

Table 16: Per-task-family aggregates on the realistic exam. Rows group models whose displayed pass rates agree, continuous columns then report the group's min–max range. Within each family, rows are sorted by Pass Rate; best per column within each family in bold.

Family (n)	Model	PR	Cont. PR	Abst. PR	Med. NRel. Err	Med. F_1-Loss
`prediction` (15)	all but GPT-5.5	—	—	—	0.530–0.621	—
	GPT-5.5	—	—	—	0.954	—
`association` (11)	all six	**100.0%**	**100.0%**	—	0.001	—
`collider_phenomenon` (2)	all but Qwen 3.6 35B	**100.0%**	**100.0%**	—	0.009–0.164	—
	Qwen 3.6 35B	50.0%	50.0%	—	—	—
`causal_sketch` (17)	all six	—	—	—	—	**0.000**
`identification` (14)	Claude Opus 4.8	**100.0%**	100.0% (2/2)	**100.0%** (5/5)	—	**0.000**
	GPT-5.5	85.7%	100.0% (2/2)	80.0% (4/5)	—	**0.000**
	Gemini 3.1 Pro / Kimi K2.6	71.4%	100.0% (2/2)	60.0% (3/5)	—	**0.000**
	Qwen 3.6 35B	57.1%	100.0% (2/2)	40.0% (2/5)	—	**0.000**
	Gemma 4 26B	28.6%	100.0% (2/2)	0.0% (0/5)	—	**0.000**
`effect_estimate` (13)	GPT-5.5	**100.0%**	—	**100.0%** (3/3)	**0.171**	—
	Gemma 4 26B	33.3%	—	33.3% (1/3)	0.596	—
	Qwen 3.6 35B	25.0%	—	25.0% (1/4)	1.061	—
	Claude Opus 4.8 / Gemini 3.1 Pro / Kimi K2.6	0.0%	—	0.0% (0/3)	0.175–0.275	—
`bias_diagnostic` (7)	GPT-5.5	**100.0%**	100.0% (1/1)	**100.0%** (3/3)	—	**0.000**
	Gemini 3.1 Pro	75.0%	100.0% (1/1)	66.7% (2/3)	—	**0.000**
	Qwen 3.6 35B	50.0%	100.0% (1/1)	33.3% (1/3)	—	**0.000**
	Claude Opus 4.8 / Kimi K2.6 / Gemma 4 26B	25.0%	100.0% (1/1)	0.0% (0/3)	—	**0.000**
`counterfactual_identification` (12)	Claude Opus 4.8	**100.0%**	**100.0%** (9/9)	**100.0%** (3/3)	—	—
	Gemini 3.1 Pro	91.7%	88.9% (8/9)	**100.0%** (3/3)	—	—
	GPT-5.5 / Kimi K2.6 / Qwen 3.6 35B	91.7%	**100.0%** (9/9)	66.7% (2/3)	—	—
	Gemma 4 26B	83.3%	88.9% (8/9)	66.7% (2/3)	—	—
`counterfactual_effect` (7)	Claude Opus 4.8	**100.0%**	**100.0%** (2/2)	**100.0%** (2/2)	0.034	—
	Gemini 3.1 Pro	75.0%	**100.0%** (2/2)	50.0% (1/2)	**0.020**	—
	Kimi K2.6	60.0%	50.0% (1/2)	66.7% (2/3)	0.147	—
	Qwen 3.6 35B	40.0%	33.3% (1/3)	50.0% (1/2)	0.099	—
	GPT-5.5 / Gemma 4 26B	25.0%	50.0% (1/2)	0.0% (0/2)	0.080–0.083	—
`mediation_effect` (2)	Claude Opus 4.8 / Gemini 3.1 Pro / Gemma 4 26B	**100.0%**	**100.0%** (1/1)	—	0.003–0.034	—
	Qwen 3.6 35B	50.0%	50.0% (1/2)	—	—	—
	GPT-5.5	0.0%	0.0% (0/1)	—	0.019	—
	Kimi K2.6	0.0%	—	0.0% (0/1)	0.012	—

Per-family commentary on Tab. 16. Three families do not discriminate: `association` and `causal_sketch` are solved by all six models, and `prediction` mostly sits close to the noise-bounded floor (five models within 0.530–0.621 NRel. Err, GPT-5.5 the outlier at 0.954). The R2 families separate the models almost purely through the abstention call — their identifiable discrete queries are answered correctly by everyone — and no model makes that call reliably across families: Claude Opus 4.8 declines every non-identifiable `identification` and counterfactual query yet catches none of the `effect_estimate` or forbidden-control (`bias_diagnostic`) abstentions, while GPT-5.5 is close to the mirror image — perfect on those two families, but the only model besides Gemma 4 26B to catch no `counterfactual_effect` abstention. Estimation accuracy on the identifiable `effect_estimate` queries — a wrong commit on a non-identifiable estimand

is scored only as a missed abstention and never enters the error pool — splits along the same line: the frontier models stay accurate (NRel. Err 0.171–0.248), whereas Qwen 3.6 35B and Gemma 4 26B blow up (1.061 and 0.596; Kimi K2.6 is the open-weight exception at 0.275). On the counterfactual families Claude Opus 4.8 is alone at ceiling ($12/12$ and $4/4$); `mediation_effect` ($n = 2$) is too small to read.

Table 17: Per-motif aggregates on the realistic exam, with grafted scenes split by graft count. Motifs with ≤ 2 tasks are not broken out. Best per column within each motif in bold.

Motif (n)	Model	PR (n_{disc})	Cont. PR	Abst. PR	Med. NRel. Err
`iv` (27)	GPT-5.5	**78.6%** (14)	—	**78.6%**	0.052
	Claude Opus 4.8	57.1% (14)	—	57.1%	0.052
	Gemini 3.1 Pro	50.0% (14)	—	50.0%	**0.026**
	Qwen 3.6 35B	42.9% (14)	—	42.9%	0.052
	Kimi K2.6	35.7% (14)	—	35.7%	0.052
	Gemma 4 26B	21.4% (14)	—	21.4%	0.052
`confounding` (25)	Claude Opus 4.8	**100.0%** (6)	**100.0%**	—	0.260
	Gemini 3.1 Pro	**100.0%** (6)	**100.0%**	—	**0.243**
	GPT-5.5	**100.0%** (6)	**100.0%**	—	0.840
	Qwen 3.6 35B	85.7% (7)	85.7%	—	0.276
	Kimi K2.6	85.7% (7)	**100.0%**	0.0%	0.302
	Gemma 4 26B	83.3% (6)	83.3%	—	0.374
`grafted_1` (16)	Claude Opus 4.8	**100.0%** (5)	**100.0%**	**100.0%**	**0.112**
	Gemini 3.1 Pro	**100.0%** (5)	**100.0%**	**100.0%**	0.197
	GPT-5.5	80.0% (5)	75.0%	**100.0%**	0.164
	Kimi K2.6	80.0% (5)	75.0%	**100.0%**	0.360
	Qwen 3.6 35B	66.7% (6)	60.0%	**100.0%**	0.634
	Gemma 4 26B	60.0% (5)	75.0%	0.0%	0.250
`grafted_2` (5)	all but Gemini 3.1 Pro	**100.0%** (2)	**100.0%**	—	0.246–0.357
	Gemini 3.1 Pro	50.0% (2)	50.0%	—	0.250
`chain` (5)	all six	**100.0%** (1)	**100.0%**	—	0.301–0.530
`fork` (4)	all six	**100.0%** (1)	**100.0%**	—	0.002
`collider` (3)	all six	**100.0%** (2)	**100.0%**	—	—
`frontdoor` (3)	Claude Opus 4.8	**100.0%** (1)	**100.0%**	—	0.421
	Gemini 3.1 Pro	**100.0%** (1)	**100.0%**	—	**0.224**
	Gemma 4 26B	**100.0%** (1)	**100.0%**	—	0.237
	Qwen 3.6 35B	50.0% (2)	**100.0%**	0.0%	0.417
	GPT-5.5	0.0% (1)	0.0%	—	**0.224**
	Kimi K2.6	0.0% (1)	—	0.0%	**0.224**
`mediation` (4)	all six	—	—	—	0.008

Per-motif commentary on Tab. 17. The two large motifs probe complementary skills. On `iv`, every discrete query lands in the abstention pool — by design, a generic instrument does not identify the population ATE (App. A.7) — so the motif ranks models purely on the abstention call, with a wide spread (GPT-5.5 78.6% down to Gemma 4 26B 21.4%), while the median error on the continuous tasks is small and nearly uniform (0.026–0.052). `confounding` is the mirror image: its queries are identifiable (the lone abstention entry is Kimi K2.6 wrongly abstaining on one), the frontier models pass every discrete query, and the error column separates them instead — 0.243–0.374 for most models against GPT-5.5's 0.840. Grafting does not degrade the frontier: on `grafted_1`, Claude Opus 4.8 and Gemini 3.1 Pro stay perfect with the lowest errors, whereas Qwen 3.6 35B drops to 66.7% with 0.634 NRel. Err and Gemma 4 26B misses its abstention call. A second graft (`grafted_2`) adds no further degradation: all models pass every query except a single unanswered counterfactual-identification query from Gemini 3.1 Pro. The simple identifiable structures act as a sanity floor — `chain`, `fork`, `collider`, and `mediation` are passed by all six models, with only the `chain` estimation spread (0.301–0.530) separating them. On `frontdoor`, the motif-defining front-door ATE query is estimated accurately by four models, while Qwen 3.6 35B wrongly abstains on it and Claude Opus 4.8 leaves it unanswered; the row's discrete pass rates instead reflect the accompanying mediation sign query, which GPT-5.5 leaves unanswered and Kimi K2.6

wrongly abstains on. Only the five motifs with ≤ 2 tasks that are not broken out (`arrowhead`, `diamondcut`, `triangle`, `diamond`, `double_nc`) carry too few tasks to support conclusions in either direction.

Table 18: Token usage and efficiency on the realistic exam. Tokens are prompt+completion; *Tokens/*$(1-\text{Score})$ is a measure of efficiency (lower is better) — the table is sorted by this metric. *Calls/Task* is the mean number of bash invocations per task. *CausalDSScore* is repeated from Tab. 3. Rows sorted by Tokens/$(1-\text{Score})$. Best per column in bold.

Model	Total Tokens	Tokens/Task	CausalDSScore	Tokens/$(1-\text{Score})$	Calls/Task	Strategy
Claude Opus 4.8	1.77M	17.7k	**0.2780**	**2.45M**	3.4	near one-shot
GPT-5.5	**1.29M**	**12.9k**	0.5610	2.95M	**2.1**	near one-shot
Gemma 4 26B	3.24M	32.4k	0.6442	9.10M	6.1	iterative
Gemini 3.1 Pro	14.56M	145.6k	0.3703	23.13M	11.2	iterative
Qwen 3.6 35B	14.07M	140.7k	0.4474	25.45M	17.6	iterative
Kimi K2.6	26.64M	266.4k	0.4754	50.78M	11.9	iterative

Efficiency. Tool-use strategy splits the field (Fig. 4). GPT-5.5 (2.1 calls/task) and Claude Opus 4.8 (3.4) are near one-shot — they read the data, write the entire analysis as one script and submit — while the other four iterate, exploring the data, fitting several candidate models, and refining: Gemma 4 26B (6.1), Gemini 3.1 Pro (11.2), Kimi K2.6 (11.9), and Qwen 3.6 35B (17.6). Calls and token usage do not track each other: Kimi K2.6 spends the most tokens (266.4k/task, 26.6M total) despite making fewer calls than Qwen 3.6 35B (140.7k/task), reflecting a longer reasoning process, while GPT-5.5 is the leanest overall (12.9k/task) despite the high reasoning setting, with Claude Opus 4.8 close behind (17.7k/task). Gemma, the one non-reasoning model, follows behind at 32.4k/task. Normalized by quality, Tokens/$(1-\text{Score})$ ranks Claude Opus 4.8 most token-efficient (2.45M) and Kimi K2.6 least (50.8M) — a $21\times$ spread. Token usage is the efficiency axis comparable across all six models: five are served locally or through unpriced routes, and only Gemini 3.1 Pro is billed at a published per-token price, its full run costing \$20.02 (\$0.20/task). These interaction styles echo trajectory-level findings from recent agent benchmarks. On RoadmapBench,

Claude Opus 4.7 attains the highest resolved rate with the fewest tool calls and the lowest exploration ratio (Xu et al., 2026a) — the same coupling of a targeted trajectory with top quality that we observe for Claude Opus 4.8. The pattern is not family-general: the same benchmark reports GPT-5.4 making the *most* tool calls without converging ("analysis paralysis"), while AutoLab reports the opposite failure for the same model — submitting after minimal exploration despite substantial remaining budget (Xu et al., 2026b); on MCP-Bench, GPT-5 is among the heavier tool users (Wang et al., 2025), and DeepPlanning finds that more tool use generally improves long-horizon planning (Zhang et al., 2026). Tool-use strategy is thus strongly model- and benchmark-specific. We also note that call counts are not commensurate across benchmarks: a single bash call in our harness can execute a complete analysis script, so the comparable quantity is the interaction style (single-pass script submission versus multi-round orchestration) rather than the raw counts.

A.11 Per-axis pass@k breakdowns

Setup. The headline pass@k / pass^k table is in the main text (Tab. 7); here we give the method, the continuous analogue (Tab. 19), and the per-axis breakdowns. We draw up to $k{=}3$ independent restarts per task, each a fresh single-shot trajectory. Unlike the answer-dependent main routing, each task is assigned to a scoring pool by its private ground truth: non-identifiable targets are binary abstention tasks (a committed answer is wrong, not incomplete), identifiable targets keep their native metric, and abstaining on an identifiable target is an ineligible candidate. This gives fixed pools — binary $n{=}34$, point $n{=}25$, interval $n{=}14$, graph/set $n{=}27$ — common to all three models. Because point and interval losses are pooled separately here, while the leaderboard's Med. NRel. Err mixes them under the answer-dependent routing, the continuous medians below are not directly comparable to Tab. 3. pass@k (Chen et al., 2021) is the direct empirical estimator: a task is covered if any of its first k restarts passes, and pass^k requires all k to pass. For continuous targets the analogue is the oracle best-of-k loss (Brown et al., 2024) — the lowest realized loss among the k valid candidates — with worst-of-k its pass^k counterpart. By construction binary coverage is nondecreasing in k and each

task's best-of-k loss nonincreasing (the medians below are taken over the valid subsets, which grow with k, so they need not be monotone).

Table 19: Continuous analogue of pass@k (open-weight models): oracle best-of-k and worst-of-k median loss; cells show median (valid/total). Graph/set $1-F_1$ stays 0.000 across restarts (omitted). Sorted by best@3 within each pool.

Pool	Model	best@1	best@2	best@3	worst@3
point (NRelErr)	Gemma 4 26B	0.066 (25/25)	0.052 (25/25)	0.034 (25/25)	0.091 (24/25)
	Kimi K2.6	0.076 (24/25)	0.053 (25/25)	0.052 (25/25)	0.109 (24/25)
	Qwen 3.6 35B	0.090 (20/25)	0.090 (24/25)	0.073 (24/25)	0.076 (18/25)
interval (NRelIS)	Kimi K2.6	1.61 (14/14)	1.43 (14/14)	1.43 (14/14)	1.81 (14/14)
	Qwen 3.6 35B	1.76 (11/14)	1.60 (12/14)	1.60 (12/14)	1.76 (9/14)
	Gemma 4 26B	1.79 (14/14)	1.79 (14/14)	1.79 (14/14)	2.33 (13/14)

Three observations from Tab. 19. Qwen 3.6 35B's missing submissions are largely transient: a second attempt lifts its valid point coverage from $20/25$ to $24/25$ (interval $11/14$ to $12/14$), so the tool-output failures behind its low single-run valid-answer rate (App. A.15) are recoverable by restarting. This is the source of our judgement in the main text, where we attributed the invalid submissions to the model and not the benchmark. Kimi K2.6 suffers from outliers: its best@3 is competitive in both pools (0.052 point, 1.43 interval — the latter the best) yet it posts the worst point-pool worst@3 (0.109) and by far the largest interval run-to-run SD (0.058; Tab. 20) — a few estimands flip between near-correct and far-off across restarts. Gemma 4 26B is surprisingly competitive: it leads the point pool at every k (best@3 0.034) despite the worst leaderboard Med. NRel. Err (Tab. 3) — its weakness is intervals (flat at 1.79 across k, worst worst@3 at 2.33), not point estimation.

Table 20: Run-to-run stability and variance on the complete 3-run grid (open-weight models). *Binary pool* (n=34): pass@3 (any of 3), passˆ3 (all 3), *unstable* (some but not all), stable-fail (all fail), and mean per-run success $\bar{\hat{p}}$.
Run-to-run SD: median per-task SD of NRelErr / NRelIS over the tasks answered validly in all 3 runs; cells show median (valid/total). Best per column in bold.

	Binary pool (n=34)					Run-to-run SD	
Model	pass@3	passˆ3	unstable	stable-fail	$\bar{\hat{p}}$	NRelErr	NRelIS
Kimi K2.6	91.2%	**61.8%**	29.4%	**8.8%**	**77.5%**	0.00035 (24/25)	0.058 (14/14)
Qwen 3.6 35B	82.4%	55.9%	26.5%	17.6%	67.6%	**0.00024** (18/25)	0.019 (9/14)
Gemma 4 26B	76.5%	50.0%	26.5%	23.5%	64.7%	0.0015 (24/25)	**0.006** (13/14)

Run-to-run stability and variance. For the CausalDSScore variability reported in the main text (Tab. 7), we score each of the three restarts as a complete exam run through the same scoring pipeline as the leaderboard (not the ground-truth routing used for pass@k above); run 1 therefore reproduces the leaderboard values (Tab. 3). The reported SD — 0.102 (Kimi K2.6), 0.040 (Qwen 3.6 35B), 0.045 (Gemma 4 26B) — is the sample standard deviation of the three per-run scores. Taking the SD of the complete scores, rather than combining the variances of the score's components, keeps the covariance between components in the measurement. The variability is dominated by the tail-sensitive S_{NR} component (per-run SD 0.11–0.23 across models), while the F_1-Loss component is constant across runs. One caveat: whether a task is graded on content or as an abstention depends on the model's own answer, so the number of graded continuous submissions varies slightly across runs (e.g., Qwen 3.6 35B submits 31, 35, and 31 valid continuous answers, of 39 continuous tasks, over the three runs), and the per-run scores aggregate over slightly different denominators.

On the complete 3-run grid, all-restarts consistency (passˆ3) sits well below best-of-3: 50.0–61.8% against pass@3 76.5–91.2% (Tab. 20). The gap is the *unstable* band — 26–29% of the binary pool passes on some but not all restarts — while stable failures (0/3) are 8.8% (Kimi K2.6), 17.6% (Qwen 3.6 35B), and 23.5% (Gemma 4 26B), and mean per-run success is 64.7–77.5%. The continuous metrics are far steadier across restarts: the median per-task run SD is 2.4×10^{-4}–1.5×10^{-3} NRelErr for point estimands — roughly two orders of magnitude below the point-loss medians themselves, i.e.,

the agents refit the same released data to near-identical estimates — against 0.006–0.058 NRelIS for intervals, the more variable construction. That median understates the interval term: most intervals reproduce tightly, but a few effect-interval estimands (`ate_uq_95`) swing between ≈ 0 and the cap (10) across restarts (per-task NRelIS SD 2.5–5.5), and because S_{NR} aggregates intervals with a capped *mean*, those few tasks — not the typical interval, and not the point estimates — drive the run-to-run S_{NR} swing.

Table 21: Pass-rate lift ($\mathrm{pass@1} \rightarrow \mathrm{pass@3}$, percent) on the binary pool, sliced by rung, observation variant, and input mode (open-weight models, ground-truth-fixed pools). The R1 and `proxy_hard` slices are small (n=3).

Slice	Kimi K2.6	Gemma 4 26B	Qwen 3.6 35B
Rung			
R1	100 → 100	100 → 100	100 → 100
R2	42.9 → 78.6	28.6 → 57.1	50.0 → 64.3
R3	82.4 → 100	70.6 → 88.2	82.4 → 94.1
Observation variant			
clean	66.7 → 94.4	50.0 → 72.2	66.7 → 77.8
proxy	76.9 → 92.3	69.2 → 84.6	84.6 → 92.3
proxy_hard	33.3 → 66.7	33.3 → 66.7	33.3 → 66.7
Input mode			
symbolic	72.7 → 90.9	54.5 → 72.7	72.7 → 86.4
data-backed	58.3 → 91.7	58.3 → 83.3	66.7 → 75.0

Pass-rate lift by slice. The binary $\mathrm{pass@1} \rightarrow \mathrm{pass@3}$ lift is largest at Rung 2 (e.g., Kimi K2.6 from 42.9 to 78.6%) and Rung 3, while Rung 1 is saturated from the start — its binary slice holds only n=3 tasks, solved by every model in every run (Tab. 21); by observation variant the relative gain is largest on `proxy_hard`, as expected (all three roughly double off a low base); and data-backed tasks gain more than symbolic ones for Kimi K2.6 and Gemma 4 26B. The clearest pattern is abstention (Tab. 22): the open models recover much of the abstention they miss first-try — `identification` abstention reaches 100% (Kimi K2.6) and 80% (Qwen 3.6 35B) by k=3, and every model reaches at least 66.7% on `counterfactual_identification`. Even Gemma, which get abstention entirely wrong on three task families, shows some ability given three tries. The `bias_diagnostic` (forbidden-control) family is the exception, staying low ($\leq 33.3\%$) even at k=3. The per-family abstention pools are small (n = 2–5), so these indicative only. Nonetheless, the open-weights models are *able* to distinguish non-identifiable scenes, just not *reliably so*.

Table 22: Abstention recoverability ($\mathrm{pass@1} \rightarrow \mathrm{pass@3}$, percent) on the non-identifiable targets, by task family (open-weight models). Each cell is the fraction of the family's non-identifiable targets correctly abstained on within k restarts. Per-family pools are small (n = 2–5); `mediation_effect` has no non-identifiable target.

Family	Kimi K2.6	Gemma 4 26B	Qwen 3.6 35B
`identification`	60.0 → 100	0.0 → 40.0	40.0 → 80.0
`effect_estimate`	0.0 → 66.7	33.3 → 66.7	33.3 → 33.3
`bias_diagnostic`	0.0 → 33.3	0.0 → 33.3	33.3 → 33.3
`counterfactual_id`	66.7 → 100	66.7 → 66.7	66.7 → 100
`counterfactual_effect`	100 → 100	0.0 → 100	50.0 → 100

A.12 Robustness to the CauseNet-seeded verbalization

CauseNet seeding (Sec. 3.3) injects recognizable 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 released structure 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, 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 ten motifs and both symbolic and data-backed families. We evaluate two open-weight agents, Kimi K2.6 and Qwen 3.6-35B.

Diagnostic loss. To place heterogeneous task outputs on one scale we use a "diagnostic loss" (higher is worse): $1 -$ score for boolean, F1, and abstention tasks; the reported normalized relative error, RMSE, or interval score for continuous tasks; and 1 for a missing or unparseable answer. 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 23: Matched verbalization-swap ablation: within-bundle range of the diagnostic loss (max$-$min over the four verbalizations of one fixed formal problem; higher means the outcome moved more when only the story and variable names changed). The lower block summarizes each model over the ten bundles.

Bundle	Motif	Rung	Task family	Kimi K2.6	Qwen 3.6-35B
1	arrowhead	R2	effect (ATE)	3.55×10^{-11}	0.00312
2	chain	R2	sketch	0.0	0.0
3	collider	R1	association	2.65×10^{-11}	8.20×10^{-5}
4	confounding	R2	identification	0.0	0.0
5	diamond	R1	prediction	0.00618	0.107
6	diamondcut	R2	bias diagnostic	0.0	0.0
7	frontdoor	R3	counterfactual effect	0.892	0.614
8	double_nc	R3	counterfactual ident.	0.0	1.00
9	collider	R1	collider phenomenon	0.0	0.0
10	mediation	R2	effect (ATE)	0.0	0.00299
Bundles with nontrivial spread (> 0.001)				2/10	5/10
Bundles with large spread (> 0.05)				1/10	3/10
Mean within-bundle loss SD				0.052	0.086
Mean / max within-bundle range				0.090 / 0.892	0.173 / 1.00

The stronger agent is more stable. Table 23 shows that most bundles are flat: every discrete-output bundle outside Rung 3 (sketch, identification, bias diagnostic, collider phenomenon) 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 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.

The flips are story-linked, not run noise. 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 unparseable answer) — 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/back-door 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, 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: because the formal problem is held fixed, a story-dependent answer can only arise from importing semantics that the graph and data do not support, and the matched design isolates precisely that. Kimi's single unstable bundle shows the same susceptibility in trace form: the failure mode recedes with capability but does not vanish.

A.13 Observation-layer hardness under matched scenes

The observation layer (Sec. 3.2) 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. The per-variant results in the main text (Tabs. 5 and 6) are scored on disjoint subsets of the exam. This appendix, conversely, reports the controlled companion: 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 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.12). The paired deltas are descriptive contrasts on these ten selected pairs; they are 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 24: Matched observation-layer ablation: paired capped-loss contrasts over the ten matched scene/task pairs (mean $\pm$ SD across pairs; paired-bootstrap 95% CI; *worse* — the number of pairs scoring with $\Delta > 0$). The survivor-only column conditions on a valid numeric answer in both arms, with the number of contributing pairs in parentheses.

Model	Contrast	Mean $\pm$ SD	95% CI	Worse	Survivor-only (n)
Kimi K2.6	proxy $-$ clean	-0.02 ± 0.45	$[-0.293, +0.240]$	7/10	-0.02 (10)
Kimi K2.6	proxy_hard $-$ clean	$+0.20 \pm 0.27$	$[+0.053, +0.373]$	8/10	$+0.14$ (8)
Qwen 3.6-35B	proxy $-$ clean	$+0.20 \pm 0.42$	$[+0.003, +0.498]$	5/10	$+0.01$ (5)
Qwen 3.6-35B	proxy_hard $-$ clean	$+0.33 \pm 0.47$	$[+0.070, +0.626]$	6/10	$+0.08$ (4)

Table 25: Matched observation-layer ablation: per-variant behavior. *False abstentions* counts abstentions for the tasks, all of which were designed to have identifiable targets; *context exits* counts trajectories terminated by a context-window overflow ; *median survivor loss* is the median capped loss over valid numeric answers only.

Model	Behavior	`clean`	`proxy`	`proxy_hard`
Kimi K2.6	valid answers (of 10)	10	10	8
	false abstentions (of 6)	0	0	2
	context exits	0	0	0
	mean calls per task	16.5	26.7	26.0
	median survivor loss	0.006	0.039	0.152
Qwen 3.6-35B	valid answers (of 10)	8	6	4
	false abstentions (of 6)	2	1	2
	context exits	0	5	5
	mean calls per task	13.3	19.6	19.7
	median survivor loss	0.005	0.030	0.163

The matched contrast confirms the hardness ordering. Table 24 shows both models scoring worse under `proxy_hard` (per-instance results in Fig. 9), with paired-bootstrap intervals excluding

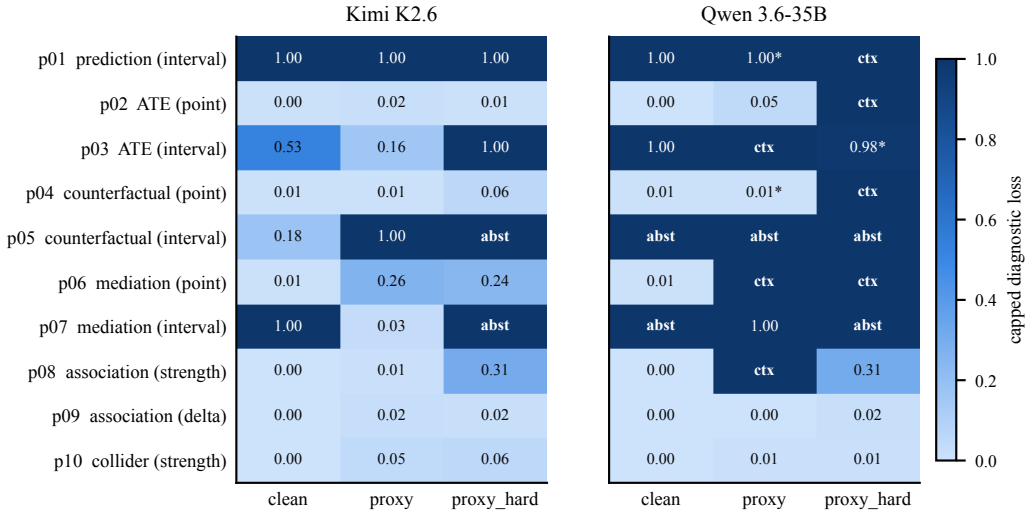


Figure 9: Matched observation-layer ablation: capped diagnostic loss for all 60 instances (10 matched scene/task pairs $\times$ 3 observation variants $\times$ 2 models). Cell shading and the printed value give the capped loss; failed cells are labeled by cause (*abst* — false abstention, *ctx* — context-window failure with no gradable answer), and an asterisk marks an answer graded despite a subsequent failure.

zero (Kimi $+0.20$, $8/10$ pairs worse; Qwen $+0.33$). The middle tier separates the models: Qwen already degrades at `proxy` ($+0.20$, CI $[+0.003, +0.498]$), while Kimi's `proxy` contrast is flat on average (-0.02 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): the mean contrast is carried by a subset of pairs that fail outright under the harder view.

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 $0.006 \rightarrow 0.152$ (Kimi) and $0.005 \rightarrow 0.163$ (Qwen) from `clean` to `proxy_hard` (Tab. 25), and Kimi's survivor-only Δ_{hard} is $+0.14$ with $7/8$ pairs worse — and since it is exactly the failing instances that leave the survivor pool, these conditional numbers if anything understate the effect.

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 $8/6/4$ (Qwen). The underlying driver is workload: both models work harder as the view hardens (Kimi $16.5 \rightarrow \sim 26$ calls 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: every one of its failures is a context-window overflow (ten overflow exits across the two proxy tiers). 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. 10 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 $+0.11$ 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. On the main exam's `proxy_hard` slice (caveat: different tasks), the frontier models show no comparable failures: counting both missing answers and false abstentions, Claude Opus 4.8 and Gemini 3.1 Pro drop 5% of tasks and GPT-5.5 10%, which is comparable to their overall rate, with no false abstentions at all.

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.

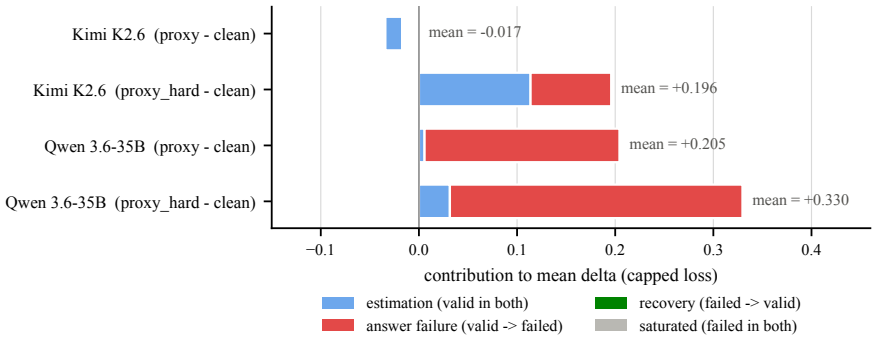


Figure 10: Exact decomposition of the mean paired capped-loss delta over the ten matched pairs, by what happened to answer validity between the `clean` arm and the comparison arm: pairs valid in both arms contribute their estimation-error change; pairs valid under `clean` but failed under the comparison contribute $1 - L_{\text{clean}}$; pairs without a valid answer in either arm are saturated at the cap and contribute zero.

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 — including a front-door ETT estimator, and, on the NIE pair, writing the abstention, deleting it, and attempting two further numerical approaches — before submitting a null answer whose graph-level rationale contradicts its own attempts. The stated non-identifiability argument is available in every arm; what changes under the hardest view is the model's willingness to stand behind an estimate.

Interval calibration and caveats. The interval tasks split into the same two regimes as the main exam: per-row prediction intervals stay near their nominal 90% coverage under every variant (0.86–0.90), while the single intervals for causal effects almost never cover their nominal 95% — the causal-uncertainty overconfidence of Tab. 6 is not created by the observation layer. The ablation is deliberately small: ten pairs, two open-weight agents, and one run per cell, so per-family splits are not especially meaningful.

A.14 Per-trajectory failure modes on the scene presented in Fig. 2

As shown in Fig. 2, the running example (`scene_000154`) is an instrumental-variable scene narrated as an agricultural study: an Irrigation Activation Subsidy Z instruments whether an Irrigation Event is Executed (X, binary), which in turn drives the Field Soil Erosion Rate Y, while an unobserved Subsurface Soil Permeability U confounds X and Y. The instrument Z, treatment X, and outcome Y are each observed only through independent noisy measurement bundles (the `proxy_hard` variant). Because U is unmeasured and the only identifying route is the instrument, the *population* ATE is not point-identified: Z identifies at most a local, compliance-restricted effect, not $E[Y \mid \text{do}(X{=}1)] - E[Y \mid \text{do}(X{=}0)]$ (`y0.id`/`dowhy` both output the same verdict). In the presented exam the harness administers this scene as a forbidden-controls bias diagnostic: with the subsidy instrument the only observed variable besides the treatment and outcome, an agent must either list the observed variables that are forbidden controls or declare the population effect not identifiable, which is the correct answer. Table 26 shows that only two of the six models—Gemini 3.1 Pro and GPT-5.5—return it. The other four answer `no_backdoor`: they correctly observe that no observed backdoor adjustment set exists, but then assert that the population ATE is nonetheless identifiable, treating the subsidy as an instrument that recovers it. That is precisely the failure mode the scene targets—the instrument identifies only a local, compliance-restricted effect, not the population ATE—so `no_backdoor` over-claims identifiability and is scored as a failed abstention. The trajectories show that the four `no_backdoor` agents reach the same over-claim along very

different routes—from deriving the correct answer in full and then discarding it in favor of guessed grader convention (Kimi K2.6) to never questioning the instrument-identifies-ATE step at all (Gemma 4 26B)—while GPT-5.5 abstains without any exploration whatsoever (Tab. 26).

[6]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.

Table 26: Per-agent responses on the forbidden-controls bias diagnostic for the running example `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.

Model	Answer	Diagnosis
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".[6]
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.

A.15 Technical details

Containerization and package pinning. Evaluation runs use a Docker / Singularity image 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.

Workspace layout and answer validation. 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. 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 so that scratch artifacts cannot leak across the benchmark's per-task fresh-conversation contract; The given scenes (`scenes/`), the model's answer directory (`answers/`), and `INSTRUCTIONS.md` persist as the benchmark state.

Provider routing, rate limits and harness details The three frontier models are routed through OpenRouter with `provider.only` pinned (to `anthropic`, `google-ai-studio`, and `openai` respectively) and `allow_fallbacks=false`, so failures appear as observable retry warnings rather than silently rerouting to a different endpoint or quantization; each is run at high reasoning effort (Claude Opus 4.8 additionally with adaptive thinking) as this is the closest to a "default" for two of the models and hence gives the most comparable results. The three open-weight models (Qwen 3.6 35B, Kimi K2.6, Gemma 4 26B) are served locally through an OpenAI-compatible (vLLM) endpoint. We used their default serving configuration, as no standardized reasoning-effort interface exists: the harness sends no sampling or reasoning parameters, so sampling follows each model's released generation defaults. In practice the two reasoning-capable models (Qwen 3.6 35B, Kimi K2.6) emit thinking traces by default, while Gemma 4 26B has no reasoning mode. We instead report token usage and tool-call counts for all models to make inference-compute differences visible. One behavior specific to Qwen 3.6 35B partly explains its lower valid-continuous rate (79.5%, Tab. 3): 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.16 Prompt templates

A.16.1 Generation pipeline prompt templates

The generation pipeline (Sec. 3.3) 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.16.2 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 Sec. 3.4 and 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. Include your reasoning as text BEFORE making the tool call.
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_` 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 $\mathrm{NIE}(x_1, x_0) = E[Y_{x_0, M_{x_1}} - Y_{x_0, M_{x_0}}]$. 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).