CAUSALVERIFY: END-TO-END VERIFICATION OF CAUSAL ANALYSES BY LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models increasingly perform empirical analyses end to end, yet existing evaluations assess the written explanation or whether generated code executes, not whether the executed workflow recovers the intended causal estimand. We introduce CAUSALVERIFY, an execution-grounded benchmark for end-to-end causal analysis that follows a model from research-context interpretation to estimand recovery. It scores this workflow at four distinct layers: method recognition, design specification, executable implementation, and estimand recovery. It combines 259 real-paper contexts, 100 fixed-seed synthetic scenarios with executable reference estimates, and 23 paper–twin pairs in which a model commits to a design before seeing the data and its executed analysis is scored against a canonical estimator on the same realised dataset.

Execution is not correctness. Among 426 model-written workflows that run without error, 15.5% fail verification, and a keyword score of the effect direction stated in the text is a poor proxy for recovery. In a 23-pair, nine-model paired study, replacing a model's committed design with the reference design and its execution conventions raises joint recovery of the point estimate and standard error from 15.0% to 51.5%; yet 48.5% of eligible seeds still fail under the reference design. The direction replicates on six pairs built afterwards under a frozen construction protocol, although on the two newest pairs the gain is confined to models from the family that built the references. Design specification is consequential but not sufficient: a plausible method and runnable code do not guarantee recovery, and even supplying the reference design and its conventions leaves substantial downstream failure. CAUSALVERIFY evaluates the executed workflow rather than its surface plausibility, and every reported number is recomputed from frozen artifacts by a single script.

1 INTRODUCTION

Large language models (LLMs) are moving from commenting on empirical research to carrying it out (Korinek, 2023; Lu et al., 2026; Majumder et al., 2025; Chen et al., 2025): in causal analysis, reading an applied setting, committing to an econometric design, writing estimation code, and reporting a treatment effect. In this workflow, code that crashes announces itself. The dangerous failure is fluent (Zecevic et al., 2023): a response that names a sensible method, produces code that runs without error, and reports a confidently signed estimate of the wrong causal quantity. Evaluation practice has largely kept the instruments of text benchmarking: agreement with reference labels (Jin et al., 2023; 2024; Zhou et al., 2026; Lee et al., 2025), plausibility of the described design (Shi et al., 2026; Sawarni et al., 2026), or whether generated code executes (Chen et al., 2021; Liu et al., 2023; Lai et al., 2023). Each can tell us that a model chose a plausible method or produced runnable code; this paper asks whether these signals predict whether an executed workflow recovers the intended causal estimand.

A benchmark for this question must be realistic and verifiable at once, and no single source of tasks is both. Real papers provide authentic research contexts, but rarely a unique executable reference against which arbitrary model-written code can be checked (Silberzahn et al., 2018; Botvinik-Nezer et al., 2020; Aczel et al., 2026). Synthetic data provide that reference but give up realism. CAUSALVERIFY takes realism from real papers and verifiability from synthetic data, and its paired arm joins the two

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

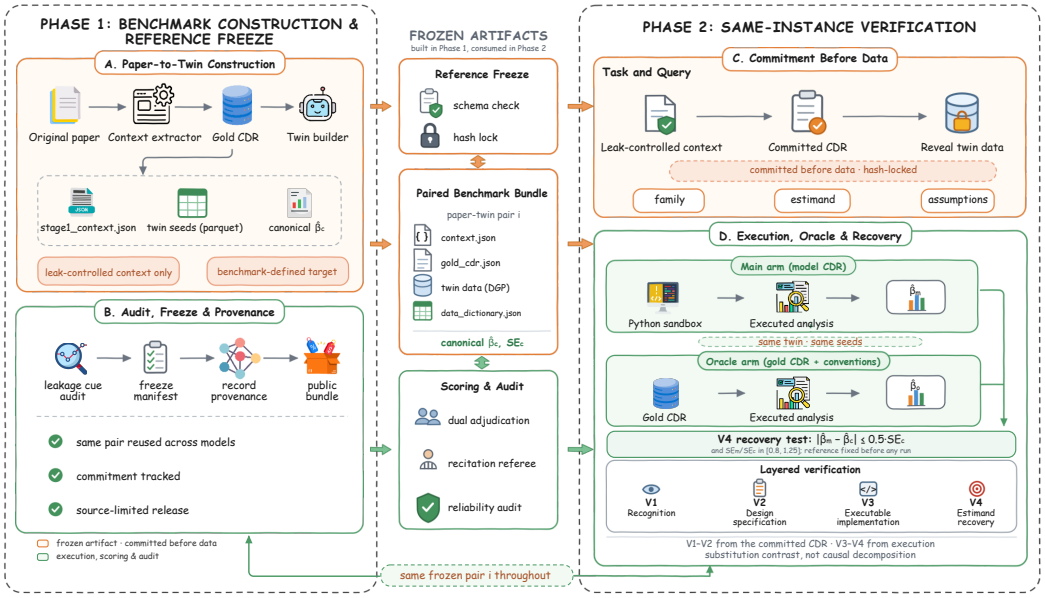


Figure 1: CAUSALVERIFY separates design commitment from executable verification. Phase 1 (left) builds a frozen bundle for each paper–twin pair: a leak-controlled research context, a gold causal-design representation (CDR), and a synthetic twin whose canonical estimate $\hat{\beta}_c$ and standard error are computed by benchmark-side estimators (**A**), then audited and hash-frozen (**B**). Phase 2 (right) evaluates a model against that bundle only: it commits to its own CDR before any data are revealed (**C**), implements it in a fixed sandbox on the twin, and is scored against the frozen reference (**D**); the oracle arm reruns the same trajectory with the gold CDR and its pinned execution conventions substituted. Layers V_1–V_2 are read from the commitment, V_3–V_4 from execution.

on the same item through a synthetic twin matched to a real paper's identification structure (fig. 1). In every executable task, a canonical estimator fixed by the benchmark defines the target estimate on the same realised dataset the model analyses, so that "the code ran" and "the estimate is right" are scored apart.

The paired arm also makes failures locatable: a model first commits to a structured causal-design representation (CDR) for a real-paper context, hashed before any data are revealed in the spirit of a pre-analysis plan (Casey et al., 2012; Olken, 2015); it then implements that commitment on the twin, and an oracle arm reruns the same trajectory under the reference design and its pinned execution conventions. Scoring is layered throughout: method recognition, design specification, executable implementation, and estimand recovery are distinct validity targets, scored separately and never aggregated, because passing one does not guarantee passing the next (section 2).

Execution is not correctness: among 426 model-written workflows that execute successfully on the synthetic scenarios, 15.5% still fail verification, roughly one in six analyses that look successful. Execution success remains informative at the model level, but textual agreement on effect direction is a poor proxy for estimand recovery (section 5). In a 23-pair, nine-model paired study, replacing the model's design with the reference design and its conventions raises joint recovery of the point estimate and its standard error from 15.0% to 51.5%, a substitution contrast on identical seeds rather than a decomposition of error. Yet 48.5% of eligible seeds still fail under the reference design. Design choice therefore matters substantially, but the reference design alone does not guarantee recovery (sections 6 and 6.2).

Our contributions are threefold. (1) An execution-grounded benchmark for causal-estimation workflows: 259 real-paper contexts, 100 fixed-seed synthetic scenarios, and 23 paper–twin pairs across four quasi-experimental design families; every task begins after the question, data, and setting have been specified (section 4). (2) A layered validity framework separating method recognition, design specification, implementation, and estimand recovery (section 2). (3) Evidence that neither runnable code nor the reference design alone guarantees recovery of the target estimate, together with an audit

Table 1: Four validity targets conflated by evaluations of AI-generated empirical research, and their measurement instruments in CAUSALVERIFY.

Layer	Validity target	Instrument
Method recognition	reference causal-design family	METHOD-MATCH vs. reference labels
Design specification	estimand, roles, assumptions	CDR fields vs. gold CDR
Executable implementation	declared analysis runs	EXECUTES (execution harness)
Estimand recovery	target quantity recovered	CORRECT vs. canonical estimate

of the benchmark's own construction: which parts were built by language-model agents or scored after results existed, and how far each moves the numbers (section 7).

2 A LAYERED VERIFICATION FRAMEWORK

An evaluation that reports a model "can do causal inference" asserts that its score measures the capability named (Raji et al., 2021). Producing a correct empirical causal result decomposes into capabilities that are conceptually distinct, separately measurable, and, as our experiments show, empirically separable (table 1).

Write $V_1, \ldots, V_4$ for passing the four layers of table 1 in order. The framework's central claim is one of *no general implication*: passing an upstream layer does not guarantee passing the next,

$$V_1 \not\Rightarrow V_2 \not\Rightarrow V_3 \not\Rightarrow V_4. \tag{1}$$

This is weaker than statistical independence (adjacent layers are strongly correlated: execution ranking tracks recovery at $\tau = 0.81$), and it is an empirical claim, tested rather than assumed. Section 5 supplies counterexamples to $V_3 \Rightarrow V_4$ at scale and shows that outer-layer rankings do not proxy for V_4; section 6 supplies same-instance evidence against $V_2 \Rightarrow V_3$ and $V_2 \Rightarrow V_4$. An evaluation that stops at V_1 measures recognition, one that stops at V_2 measures specification, and one that stops at V_3 establishes executability; none of these endpoints establishes that the target quantity was recovered.

Two design consequences follow. First, criterion validity at the recovery layer requires an *executable reference target*: a realised dataset on which a benchmark-fixed canonical estimator defines the target. The canonical estimator is a benchmark-fixed reference path, not a claim that one analysis is uniquely correct: the benchmark asks whether the model recovered its stated target, not whether that target is the only defensible specification (section 4). Textual reference answers cannot serve as that target (Silberzahn et al., 2018; Botvinik-Nezer et al., 2020; Aczel et al., 2026): our real-paper labels carry irreducible ambiguity (blinded audit: direction $\kappa = 0.29$). Second, the layers are scored *separately and never aggregated*. Models may be compared within each validity target; a composite would re-conflate what the framework separates and be dominated by whichever layer the task mix happens to weight (section 5). The deliverable is a profile, not a ranking.

3 RELATED WORK

Prior work evaluates important pieces of the LLM causal-analysis workflow (Ma, 2025; Yang et al., 2024), but existing benchmark families terminate at different points of the chain in table 1, and rarely at the executed causal estimate. Causal-reasoning benchmarks test textual reasoning about direction, counterfactuals, graphs, or identification (Kiciman et al., 2024; Jin et al., 2023; 2024; Chen et al., 2024; Chi et al., 2024; Zhou et al., 2026; Lee et al., 2025). Code and science-agent benchmarks test whether programs execute, pass functional tests, or deliver a final product (Chen et al., 2021; Lai et al., 2023; Jimenez et al., 2024; Zhuo et al., 2025; Jain et al., 2025; Huang et al., 2024; Majumder et al., 2025; Chen et al., 2025; Anthropic, 2026). Closest are end-to-end causal benchmarks. CauSciBench (Acharya et al., 2025) grades the final product on curated real, synthetic, and textbook tasks. Sawarni et al. (2026) disentangle identification from estimation on real datasets. CausalDS (Leban & Sun, 2026) spans Pearl's three rungs on synthetic structural causal models. InterveneBench (Shi et al., 2026) scores agreement between predicted and expert-verified study designs, while reproduction evaluations (Kohler et al., 2026; Nguyen et al., 2026) use agreement with

the original authors' results as the criterion. CAUSALVERIFY differs not in task breadth but in its unit and terminal criterion of verification: a design committed and hashed before any data are revealed, executed model-written code, a benchmark-fixed canonical estimate on the same realised dataset, and an oracle arm that substitutes the reference design and its execution conventions into the model's own trajectory. Design-layer and execution-layer failures thus separate on the same instance rather than across task pools. The distinction is not between textual and executable benchmarks, but between verifying an intermediate artifact and verifying the target quantity the workflow is meant to produce (extended comparison in section L, table 18).

4 BENCHMARK AND PROTOCOL

No single setting provides both ecological realism and executable reference estimates. Real papers test whether models parse applied research contexts, but their reference labels are interpretive: cleaned data, runnable specification and counterfactual effect are usually unavailable (section 2). Synthetic data-generating processes (DGPs) give up realism but make coefficient correctness executable. A paired arm links these two strengths on the same instance, allowing design interpretation and executed recovery to be evaluated together.

Scope. Every task begins after the research question, data description, and institutional context have been specified. The benchmark evaluates the downstream portion of empirical practice (design recognition through estimation and reporting) for four quasi-experimental families (Angrist & Pischke, 2009; Imbens & Rubin, 2015): difference-in-differences (DID) (Bertrand et al., 2004; Goodman-Bacon, 2021; Roth et al., 2023), event studies (ES), instrumental variables (IV), and regression discontinuity (RDD) (Calonico et al., 2014); the paired corpus also includes one randomized-encouragement design, scored under its IV family (Angrist et al., 1996). It does not evaluate problem formulation, data acquisition, design selection from ambiguous field constraints, or iterative robustness analysis, and it does not cover matching, propensity-score, doubly-robust, mediation, or heterogeneous-effect workflows. Competition-style evaluations that overweight modeling at the expense of subject-matter reasoning are "somewhat detached from the practice of causal inference" (Hernán, 2019); we claim nothing outside the stated boundary.

Executable arm: fixed-seed DGP scenarios. This arm isolates V_3 from V_4: it asks separately whether model-written code runs and whether the executed estimate recovers the benchmark target. It contains 100 fixed-seed scenarios (30 DID, 24 ES, 24 IV, 22 RDD) (Callaway & Sant'Anna, 2021; Sun & Abraham, 2021; MacKinlay, 1997; Imbens & Angrist, 1994; Angrist et al., 1996; Thistlethwaite & Campbell, 1960; Imbens & Lemieux, 2008; Lee & Lemieux, 2010). Realised CSV datasets are fixed before any model evaluation; canonical estimates are computed from those datasets alone by benchmark-side estimators that never see a model output (section 7 gives the scorer's history). Models receive the research question, data description, and a preview of the realised CSV, then produce runnable R code in one call with a 4,096-token output cap (pipeline in fig. 6). EXECUTES records whether the code executes in the fixed environment. CORRECT records whether the extracted treatment-effect estimate $\hat{\beta}_m$ matches the canonical estimate $\hat{\beta}_c$ computed on the *same realised dataset*:

$$|\hat{\beta}_m - \hat{\beta}_c| \,/\, |\hat{\beta}_c| \leq \tau, \qquad \tau = 0.50 \text{ by default.} \tag{2}$$

Comparing against the realised-data canonical estimate rather than the ideal DGP parameter avoids penalizing finite-sample deviation; the estimator's status as a benchmark-fixed reference path rather than a claim of unique scientific validity is stated in section 2. The tolerance is a permissive reporting default, not an adjudication rule: the model ranking is identical at every tested tolerance from 10% to 100% (section B). A text-direction diagnostic, DIRECTION-MATCH, is scored on the same scenarios against the known generating sign.

Real-paper contexts as text diagnostics. The breadth arm contains 259 archived economics papers, each represented by reconstructed research-question, data-description, and institutional-context fields with method names blacklisted. METHOD-MATCH and DIRECTION-MATCH are scored there against consensus reference labels from four LLM voters, two of which (GPT-4o, Gemini 2.5 Flash) are also evaluated (section 8); an internal audit, still open, found identification arguments or source results in more than half of 18 manually reviewed contexts. These labels diagnose the outer layers only and are

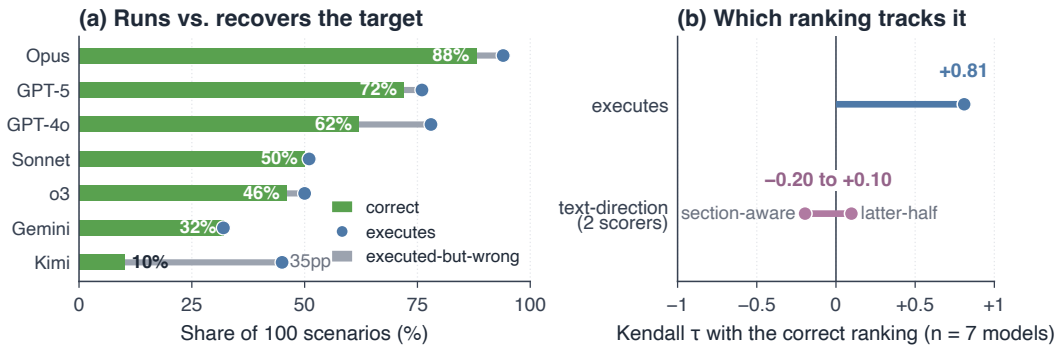


Figure 2: Executable arm, headline. **(a)** Per model, the share of 100 fixed-seed scenarios whose code runs (EXECUTES, dot) and whose executed estimate recovers the target within the 50% band (CORRECT, bar); the grey segment is executed-but-wrong. **(b)** Model-level rank agreement with the CORRECT ranking: ranking by EXECUTES tracks it (Kendall $\tau = 0.81$); the two keyword text-direction scorers do not ($\tau \in [-0.20, 0.10]$). Seven original panel models; open-weight members in table 5.

never correctness endpoints. A blinded 30-paper audit on a sample over-representing low-consensus papers agreed with consensus method labels at $\kappa = 0.61$ and direction labels at $\kappa = 0.29$, with agreement deteriorating under weak consensus (section K).

Paired bridge: design commitment before data. The paired protocol connects realism and executable verification on the same item. In Stage 1 a model reads a real-paper brief and commits to a structured causal-design representation (CDR: method family, estimand, variable roles, comparison group, timing, fixed effects, clustering, assumptions); the commitment is hashed before any data are revealed (Casey et al., 2012; Olken, 2015). In Stage 2 a fresh call hands the model its own committed CDR together with the data dictionary and column schema of a matched synthetic twin that preserves the paper's identification structure. The model writes code implementing its commitment; the script reads the twin data only inside the sandbox, and estimates and standard errors are scored against the twin's canonical values under frozen bands. An oracle arm reruns the same Stage-2 trajectory with the benchmark's gold CDR substituted for the model's commitment; gold CDRs were drafted and source-checked by language-model agents and approved by an author (section 7). The oracle view also fixes the execution conventions the gold implies (standard-error form, degrees of freedom, sample rules, units). Stage-1 scores are frozen and deviating trajectories remain in the analysis (frozen protocol in section D).

Controls, leakage, and scorer audits. Because CAUSALVERIFY is itself a measurement instrument, its construction and scoring pipeline is audited separately from model performance. Separate stages construct contexts, audit leakage, collect responses, extract coefficients, and aggregate labels, a role separation that reduces task-writer/respondent/grader circularity; calibration is elicited separately (Kadavath et al., 2022; Tian et al., 2023; Xiong et al., 2024). Templates, effect signs, sample sizes, seeds, realised files, and canonical estimators are benchmark-fixed, so a model cannot pass CORRECT by guessing signs or matching phrases. The coefficient-extraction judge is itself audited. In a blinded 50-cell human validation, on the 44 cells where the annotator found a treatment effect to compare, numeric agreement was 90.9% (40/44) and agreement on the induced pass/fail decision 88.6% (39/44); CORRECT is thus an audited instrument, not an oracle. The arms are never aggregated into a single leaderboard; table 19 in the appendix maps each control to the claim it protects.

5 EXECUTION RESULTS: RUNNABLE IS NOT RIGHT

The executable arm's two scores, EXECUTES and CORRECT, separate sharply, and they do so differently across models.

Execution and recovery are different capabilities. Across the seven original panel models, CORRECT pass rates span 10%–88% of 100 scenarios (fig. 2a, table 5); the two open-weight panel members, Qwen3-235B and DeepSeek-V4, reach 34 and 54, with robustness-only Llama-3.3-70B at 20 (section F). Of the 426 workflows that execute successfully in the original panel, 66 (15.5%; scenario-clustered 95% CI $[12.1\%, 19.2\%]$) fail verification: 40 (9.4%) return an estimate outside the tolerance band and 26 (6.1%) yield no extractable target coefficient. We call this union *executed-but-wrong*. Evaluation that stops at "the code ran" classifies all 426 as successes, and the share is not an artifact of a tight band: it is 27.2% at a 10% tolerance and still 14.1% at 100% (table 3).

Models fail at different layers. Conditional on executing, the panel splits into two regimes. Strong executors are near-ceiling once code runs (Sonnet 98.0%, Opus 93.6% conditional pass), so their losses concentrate upstream in execution failures. Gemini's 100% is not comparable: most of its upstream losses are truncated responses (table 5). Weaker executors more often produce workflows that run but fail downstream verification (Kimi 22.2% conditional pass: 35 of its 45 executing workflows fail verification). The open-weight families sharpen the contrast: conditional on execution, Qwen3 fails verification in 2.9% of cases and DeepSeek-V4 in 8.5%, while robustness-only Llama fails in 51.2%. These are independently developed families concentrating their failures at different layers of table 1 (section 2).

Execution ranking tracks recovery; text-direction ranking does not. Model-level rankings by EXECUTES track rankings by CORRECT at Kendall $\tau = 0.81$ (Spearman $\rho = 0.93$; 1,000 scenario-clustered bootstrap resamples, 95% CI $[0.62, 0.90]$), and the CORRECT ordering is unchanged at every tolerance from 10% to 100%. Leave-one-model-out panels preserve $\tau \in [0.733, 0.867]$ (leave-Gemini-out 0.867), and removing any one design family leaves the CORRECT ranking within $\tau \in [0.81, 1.00]$ of the full panel with the top-two and bottom-two model sets intact (section C). Execution is therefore informative at the model level, but not sufficient at the workflow level. A text-level signal behaves differently. We score each response's stated effect direction against the known generating sign with two deterministic keyword scorers, deliberately simple instruments of the kind a text-only evaluation would use, not validated direction classifiers. Their model rankings correlate with CORRECT at $\tau \in [-0.20, +0.10]$. At the level of the single workflow, the score agrees with the true sign for 67.8% of correct workflows and 60.0% of incorrect ones, so it carries little information about whether the computed estimate is right. We draw no conclusion from any individual model's text-direction rank, which rests on margins of a few scenarios.

6 PAIRED PAPER-TO-EXECUTION ANALYSIS

The paired study asks the same question one layer up, on the same item: when a model commits to a design for a real-paper context and then implements it on a matched twin, does the commitment carry through to the estimate? (fig. 1C,D; section D.) The study is reported as a layer profile with pair-clustered uncertainty, not as a leaderboard.

Pilot and prototype: where chains break. A six-pair, one-model pilot (section E.1) validated the harness: every committed CDR named the correct design family, so every failure occurred downstream of V_1. An earlier four-task prototype (section E.2) supplied the motivating case: on its RDD task all nine models executed, and two vendors' models returned the *same* wrong estimate, 5.075 against a canonical 1.428 ($3.55\times$), from an `rdrobust` call on truncated support, visible only by comparison against the canonical estimate. These cases motivated the full paired panel by showing that recognition and execution can both succeed while target recovery fails.

6.1 THE FULL PANEL: 23 PAIRS, NINE MODELS

The full panel runs all 23 pairs across nine models (table 2; fig. 3) under identical contracts: provider-default sampling, single-shot with no retry or repair, up to three calls per pair (Stage-1 commitment, main arm, oracle arm; 608 calls in all), an 8,192-token output cap, and an execution contract restricted to numpy, pandas, pyarrow and the Python standard library. Of 207 Stage-1 attempts, 13 produce no schema-valid commitment and are recorded as not scorable (never as design failures), leaving 194 committed designs. The cap binds: 35 calls stopped at it, accounting for 6 of the 13 invalid commitments and for 29 Stage-2 calls that executed no seed (per-model breakdown in section E).

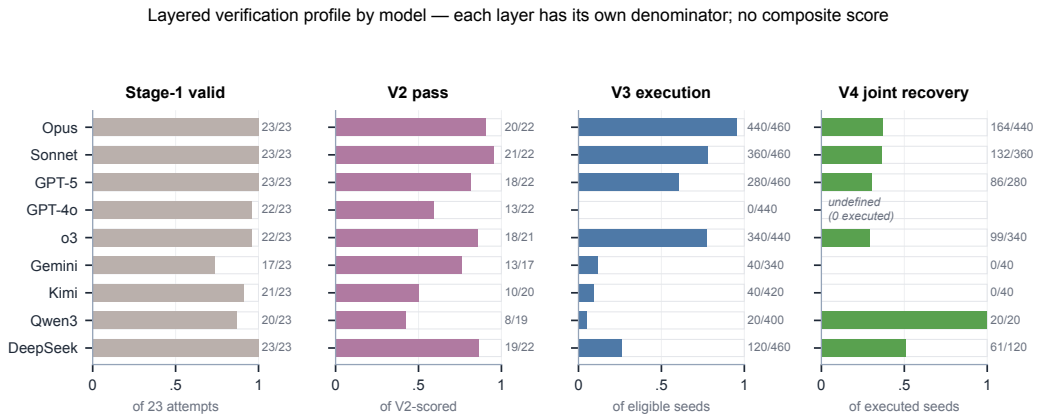


Figure 3: Layered verification profile across the nine-model panel. Each layer is reported against its own denominator (attempts; V2-scored commitments; eligible seeds; executed seeds) and no composite score is formed; per-model V2 values are a profile, not a ranking (section 7). GPT-4o's recovery layer is *undefined* (0 executed seeds after a uniform contract violation), which is distinct from the defined zeros of models that executed and failed to recover.

Design-specification verdicts were produced by language-model agents working under the authors' rulings, not by human raters. Adjudication was outcome-blind with sealed model identities, and a separate reference review checked every flagged gold claim against the archived source (section E). Mismatches concerning conventions absent from the model-facing context were removed from scoring as benchmark-underdetermined rather than charged as model error, uniformly and irrespective of how many models exhibited them. One pair shipped a defective gold and is excluded layer-scoped. A seeded 20% re-adjudication of eligible MATCH verdicts agreed at 97.6% and a 100% re-review of hedge/deferral verdicts at 17/17 (section E).

Design layer (V_2). The aggregate V_2 pass rate is descriptive, not a capability estimate: it depends on the adjudication instrument. Of 187 scored commitments (194 valid minus the excluded pair's seven), 140 pass (74.9%, pair-clustered 95% CI $[66.1, 82.9]$) and 47 (25.1%) carry at least one critical design mismatch attributable to the model on model-facing evidence: design-family recasts, misreads of explicitly stated estimands, violations of context-visible sample rules, uninvited transformations, and internally inconsistent commitments. Per-model pass rates span 0.42 (Qwen3) to 0.95 (Claude Sonnet 4.6). We rest no claim on its level: on the raw blinded verdicts, before any mismatch was voided, only 23 of 187 commitments (12.3%) carry no critical mismatch; 55 (29.4%) once the inference-convention field left the scored set; 140 only after the reference review voided 318 mismatches that the model-facing context did not determine, decisions taken after the verdicts had been read, uniformly across models.

Conditional downstream execution and recovery. At the model×pair unit (seeds are within-trajectory precision, never independent observations), V2-pass trajectories execute 47.9% of eligible seeds against 25.5% for V2-fail (+22.3pp, CI $[4.3, 40.4]$); among executed seeds, recovery of the canonical estimate is 38.7% vs. 17.1% (+21.6pp, CI $[-7.2, 45.1]$, on twelve executing V2-fail trajectories; directionally consistent, not resolved). Under a two-way bootstrap over models and pairs (Efron & Tibshirani, 1994) the execution difference's interval also includes zero ($[-13.4, 51.2]$). These are conditional associations: V_2 status is not randomized (fig. 7). Denominators stay separate throughout, and a model with no executed seeds has undefined recovery rather than zero (GPT-4o: 0/440 after a uniform contract violation; fig. 3).

Oracle substitution, common denominator. The oracle arm asks how downstream performance changes when the model's committed design is replaced by the reference design together with its pinned execution conventions, within the same model×pair trajectory. The substitution yields three quantities. Execution rises by 12.8pp (CI $[7.0, 18.8]$). On the 66 trajectories where both arms execute, conditional recovery rises from 35.9% to 98.5% (descriptive; selection-conditioned). The primary

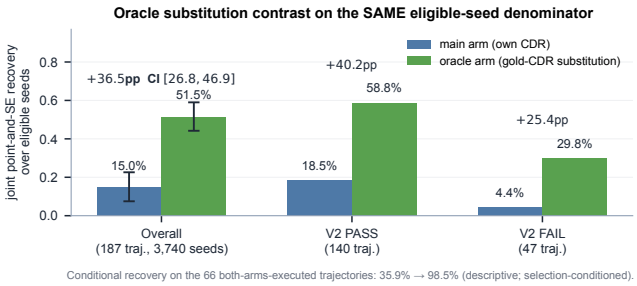


Figure 4: Oracle substitution contrast on the common denominator: joint point-and-SE recovery over *identical eligible seeds*, main arm (model's own committed design) vs. oracle arm (gold-CDR substitution), overall and by V2 stratum. Error bars are pair-clustered bootstrap 95% CIs (22 clusters, B=10,000); cvp_011 excluded (reference defect). Conditional recovery on the both-arms-executed subset appears only as a footnote because it is selection-conditioned.

comparison holds the denominator fixed: joint point-and-SE recovery over identical eligible seeds rises from 15.0% to 51.5% (+36.5pp, CI $[26.8, 46.9]$), and the contrast is larger among V2-pass trajectories (+40.2pp) than V2-fail (+25.4pp). The substitution effect shows that design information is consequential; the 48.5% of eligible seeds that still fail under the gold design show that it is not sufficient. Correcting the design does not remove the need for downstream verification. Because the oracle view carries the gold design *together with* its execution conventions, which the main arm must supply from its own commitment, the contrast bundles design and convention information, one more reason it is not a decomposition. The main arm's Stage-2 prompt carries no execution conventions, so its standard-error failures cannot be traced to an instruction the model saw; recovery of the point estimate alone rises from 21.1% to 53.6% (+32.6pp, CI $[23.0, 42.8]$), so the contrast does not rest on the standard-error conventions. The gold designs and oracle conventions were built with Claude-family agents, and the two Claude models in the panel recover best under them; without those two models the contrast is 9.3% to 39.4% (+30.1pp), so it does not depend on them. Leave-one-model and leave-one-pair ranges move the substitution contrast by at most $\pm$3.7pp.

6.2 Prospective replication on six new pairs

To test the substitution result prospectively, we built six further pairs only after the 23-pair panel had been analysed, under a stricter construction protocol frozen in advance, and ran the same single-shot contract on them (52 measured cells; one endpoint had been retired by its provider and was not substituted). Pooled over the six pairs, using each oracle arm's first run, main-arm joint recovery is 6.1% of 920 eligible seeds and oracle recovery 24.1%: the direction of the substitution contrast replicates on instances built after the claim was made. Two caveats are load-bearing. On four of the pairs the oracle arm was re-run after convention pins were added post hoc (39.8% instead of 22.2% on those pairs); only on the last two were all conventions fixed before any call. On those two pairs the whole oracle gain (1.1% to 28.6%) comes from the two Claude models, which recover every seed; without them it is 0.5% to 0.0%. Most other submissions fail to execute (9 of 46 calls stopped at the token cap). On these harder pairs we cannot separate stronger models from models that share conventions with the agent family that built the references (section H).

7 Threats We Checked

This section audits the benchmark as a measurement instrument rather than the models it evaluates. Every number reported here is recomputed from released files by the reproduction scripts of section A.

Benchmark built with language-model agents. The 23 gold CDRs, the synthetic twins, the design-layer adjudication and the reference review were produced by Claude-family agents under the authors' rulings and batch approval, not by independent human experts. The two blinded human audits of section K are the only human-rater instruments in this paper. The paired panel includes two Claude models, so we report no model ranking on the design layer or the oracle arm. We give the

substitution contrast without them: it survives on the 23 pairs (section 6.1), not on the two newest pairs (section 6.2). In the executable arm, the briefs of scenarios s31–s100 were worded by GPT-4o from benchmark-fixed templates; GPT-4o's gap on them is no larger than other models' (correct on 62.9% of the briefs it worded and 60.0% of the others, against 51.2% and 46.1% for the other six models).

Scoring choices made after outputs existed. The executable-arm scorer changed three times after most outputs had been collected: the reference moved from the planted parameter to the canonical estimate on the realised data (the two differ by 10% or more in 58 of 100 scenarios), regex coefficient extraction gave way to an LLM extraction judge (audited in section K), and event-study estimates were matched window-aware. Each step lowered the executed-but-wrong share (56.1%, 33.1%, 15.5%), yet the model ordering by CORRECT is identical under the first and last scorer (τ with EXECUTES: 0.81, 0.71, 0.81). Where canonical and planted effects agree (42 scenarios), the share is 14.9% ($\tau = 0.81$); without event studies, whose window rule is permissive, it is 15.9% (51/320). Seven RDD scenarios had their recorded sign corrected after outputs existed, when a generator bug surfaced (data unchanged); without them the text-direction correlation is still $\tau = 0.10$.

Growth and exclusions. Scenarios s31–s100 were generated after the first 30 had been run (executed-but-wrong: 12.2% on the first 30, 16.7% on the rest). The open-weight and robustness-only models were added after the seven-model results were known. An earlier pre-commitment prompting pilot is excluded from this paper because its responses were truncated at the token cap for two models and it was scored against a superseded reference. Boundary experiments are reported in sections I and J.

8 LIMITATIONS

Scope. The benchmark evaluates structured econometric estimation workflows after the question, data and setting are fixed, not problem formulation, data acquisition, or design selection in the sense of Hernán (2019). A model could excel here and fail at those stages (Raji et al., 2021). **Labels and construction.** Real-paper labels come from an LLM consensus whose pool overlaps the evaluated panel; the blinded audit bounds their reliability (direction $\kappa = 0.29$). The paired arm's gold CDRs and adjudication are agent-produced (section 7). Both are diagnostics or fixed references, not independent human ground truth. **Harness.** Output caps bind for models that spend tokens on hidden reasoning (55 of Gemini's executable-arm responses; 35 paired calls), so execution rates partly measure the harness, a scaffold effect documented across agent benchmarks by Zhang et al. (2026). **Synthetic structure and backend.** Executable reference estimates require synthetic twins covering four design families, not the space of applied practice; the executable arm runs R and the paired arm Python, so cross-arm comparisons carry a backend difference, and magnitudes should not be read as backend-general. **Scale.** The paired study covers 23 pairs and nine mainly closed or hosted models, with conditional rates whose pair-clustered recovery-difference interval includes zero. Rank statements are descriptive over the evaluated panel. Its endpoints are those callable between March and August 2026 (table 2); two have since been retired, and newer model generations are not evaluated. **Instrument.** Scoring depends on canonical specifications, frozen tolerances and an audited extraction judge. Two twin amendments (a unit-contract clarification; the repair of one task's Stage-1 context) and the design-profile instrument are not result-blind and are recorded as such. **Contamination.** Source papers are published, so pretraining familiarity cannot be excluded (Jain et al., 2025); identifiers are opaque and per-twin truths sealed. Section N expands each point.

9 CONCLUSION

CAUSALVERIFY treats the executed workflow as the unit of evaluation for LLM-assisted causal inference and instruments it in the four layers of table 1. On the executable arm, 15.5% of running workflows fail verification. The paired protocol makes the separation same-instance: across 23 pairs and nine models, main-arm joint recovery over eligible seeds is 15.0%; substituting the gold design and its conventions raises it to 51.5%, and 48.5% of eligible seeds still fail: design specification is consequential but not sufficient. CAUSALVERIFY measures whether a model-written analysis recovers a benchmark-fixed estimate; its oracle substitution is a measurement contrast, not a prescription for what should have been done. More generally, evaluation of quantitative scientific agents should validate the target quantity itself, and should be built so that others can verify that it did.

AI USE STATEMENT

Language models were used in three roles: as measurement instruments, as construction agents, and as coding and writing tools; the paper's claims are qualified accordingly (section 7). *As measurement instruments:* the reference labels of the real-paper arm are a consensus of LLM voters (four for the method family, three for direction); the executable arm's coefficient-extraction judge is an LLM (`claude-haiku-4-5`, the same vendor as two evaluated models), audited against blinded human annotation (section K); and the briefs of executable-arm scenarios s31–s100 were generated by GPT-4o from benchmark-fixed templates. *As construction agents:* Claude-family models produced the gold causal-design representations, the synthetic twins, the execution conventions, the design-layer adjudication verdicts and the reference review of the paired study, under protocols, rulings and batch approvals issued by the authors. The two blinded human audits in section K are the only independent human-rater instruments used to validate these components. *As tools:* AI systems assisted with the evaluation harness, the scoring and reproduction scripts, figure creation, and manuscript drafting and revision; the authors reviewed all code and text and are responsible for every claim. In the categories of the ICLR AI policy, this covers generating synthetic datasets, designing and giving feedback on methodology, implementing methods, creating figures, and drafting and editing the paper. Because two evaluated models belong to the same model family that constructed the paired references, we also report the substitution contrast with those models excluded.

ETHICS STATEMENT

The benchmark evaluates models on synthetic datasets and on research contexts reconstructed from archived research papers; it involves no human-subjects data. Paper sourcing carries a contamination surface, handled by opaque identifiers, sealed reference values, and a deliberate, versioned release decision rather than default publication. The intended use is measurement: evidence about where model-written causal analyses fail and what verification catches. Results should not be read as certifying any model for unsupervised empirical research, and the benchmark makes no judgement about which scientific claims a setting licenses: it measures recovery of benchmark-fixed estimates, nothing more.

REPRODUCIBILITY STATEMENT

The twin set is frozen under hashed manifests with a dated amendment record (v1.0–v1.6) in which every post-evidence change states its trigger, including two amendments that are explicitly not result-blind. Stage-1 inputs are rendered, frozen artifacts with record-level provenance to their source; model-facing task identifiers are opaque, with ground truth and identifier mappings held scorer-side and per-twin truths sealed. Executions run in a network-isolated container behind a runtime isolation gate that is tested, not asserted, before any live call; the scorer ships a self-test and a discrimination table; paid trajectories are archived with committed-design hashes, per-call usage ledgers, submitted scripts and per-seed numerics under a hashed evidence freeze, and the design-layer instrument publishes its full adjudication log. Deferred diagnostic defects are tracked in a public register and closed by post-hoc audits that may not alter frozen results: the two closed after unblinding (an infrastructure-sublabel taxonomy pass and a recitation-zone counterfactual audit) changed zero pass/fail statuses, and the one presentation-column defect they surfaced is published as a labeled post-unblind erratum beside the original rather than a silent correction. A metric crosswalk (section B) maps this paper's names to the released v1 artifacts.

Relation to the v1 release. The real-paper corpus (259 contexts), the 100-scenario executable arm and the confidence-elicitation diagnostic were released earlier as v1, with cached outputs and scoring tables; every executable-arm number in this paper is recomputed from those frozen files and is unchanged. New in this paper: the layered verification framework (section 2); the paired protocol and its 23-pair panel with the oracle substitution contrast (section 6); the prospective replication on six further pairs (section H); the repair pilot (section G) and the controlled paired-task repair (section I.3); the disclosure of construction and scoring choices (section 7); and the single-script reproduction harness (section A). The v1 text-agreement and calibration results are retained as diagnostics in section J.

REFERENCES

Sawal Acharya, Terry Jingchen Zhang, Andrew Kim, Anahita Haghighat, Xianlin Sun, Rahul Babu Shrestha, Maximilian Mordig, Furkan Danisman, Clijo Jose, Yahang Qi, Pepijn Cobben, Bernhard Schölkopf, Mrinmaya Sachan, and Zhijing Jin. CauSciBench: A comprehensive benchmark on end-to-end causal inference for scientific research, 2025. URL https://openreview.net/forum?id=uQzPkWvTyo. Submitted to ICLR 2026.

Balazs Aczel, Barnabas Szaszi, Harry T. Clelland, et al. Investigating the analytical robustness of the social and behavioural sciences. *Nature*, 652(8108):135–142, 2026. doi: 10.1038/s41586-025-09844-9. Many Analysts crowd-sourced study; 100+ contributing authors.

Joshua D. Angrist and Jörn-Steffen Pischke. *Mostly Harmless Econometrics*. Princeton University Press, 2009. doi: 10.1515/9781400829828.

Joshua D. Angrist, Guido W. Imbens, and Donald B. Rubin. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*, 91(434):444–455, 1996. doi: 10.1080/01621459.1996.10476902.

Anthropic. Evaluating claude's bioinformatics research capabilities with biomystery-bench. https://www.anthropic.com/research/Evaluating-Claude-For-Bioinformatics-With-BioMysteryBench, 2026. Research blog post.

Marianne Bertrand, Esther Duflo, and Sendhil Mullainathan. How much should we trust differences-in-differences estimates? *Quarterly Journal of Economics*, 119(1):249–275, 2004. doi: 10.1162/003355304772839588.

R. Botvinik-Nezer, F. Holzmeister, C. F. Camerer, et al. Variability in the analysis of a single neuroimaging dataset by many teams. *Nature*, 582(7810):84–88, 2020. doi: 10.1038/s41586-020-2314-9.

Brantly Callaway and Pedro H. C. Sant'Anna. Difference-in-differences with multiple time periods. *Journal of Econometrics*, 225(2):200–230, 2021. doi: 10.1016/j.jeconom.2020.12.001.

Sebastian Calonico, Matias D. Cattaneo, and Rocio Titiunik. Robust nonparametric confidence intervals for regression-discontinuity designs. *Econometrica*, 82(6):2295–2326, 2014. doi: 10.3982/ECTA11757.

Katherine Casey, Rachel Glennerster, and Edward Miguel. Reshaping institutions: Evidence on aid impacts using a preanalysis plan. *Quarterly Journal of Economics*, 127(4):1755–1812, 2012. doi: 10.1093/qje/qje027.

Mark Chen, Jerry Tworek, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. doi: 10.48550/arXiv.2107.03374.

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*, 2024. doi: 10.18653/v1/2024.findings-emnlp.363.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2410.05080.

Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. Unveiling causal reasoning in large language models: Reality or mirage? In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3064.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1994. doi: 10.1201/9780429246593.

Andrew Goodman-Bacon. Difference-in-differences with variation in treatment timing. *Journal of Econometrics*, 225(2):254–277, 2021. doi: 10.1016/j.jeconom.2021.03.014.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with APPS. In *Advances in Neural Information Processing Systems*, volume 34, 2021. doi: 10.48550/arXiv.2105.09938.

Miguel A. Hernán. Comment: Spherical cows in a vacuum: Data analysis competitions for causal inference. *Statistical Science*, 34(1):69–71, 2019. doi: 10.1214/18-STS684.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309. PMLR, 2024. doi: 10.48550/arXiv.2310.03302.

Guido W. Imbens and Joshua D. Angrist. Identification and estimation of local average treatment effects. *Econometrica*, 62(2):467–475, 1994. doi: 10.2307/2951620.

Guido W. Imbens and Thomas Lemieux. Regression discontinuity designs: A guide to practice. *Journal of Econometrics*, 142(2):615–635, 2008. doi: 10.1016/j.jeconom.2007.05.001.

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, Cambridge, UK, 2015. doi: 10.1017/CBO9781139025751.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2403.07974.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2310.06770.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. Can large language models infer causation from correlation? In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2306.05836.

Zhijing Jin et al. CLadder: Assessing causal reasoning in language models. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-1353.

Saurav Kadavath, Tom Conerly, Amanda Askell, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. doi: 10.48550/arXiv.2207.05221.

Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. doi: 10.48550/arXiv.2305.00050.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4110–4124, 2021. doi: 10.18653/v1/2021.naacl-main.324.

Benjamin Kohler, David Zollikofer, Johanna Einsiedler, Alexander Hoyle, and Elliott Ash. Read the paper, write the code: Agentic reproduction of social-science results. *arXiv preprint arXiv:2604.21965*, 2026. doi: 10.48550/arXiv.2604.21965.

Anton Korinek. Language models and cognitive automation for economic research. *National Bureau of Economic Research Working Paper 30957*, 2023. doi: 10.3386/w30957.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Scott Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. DS-1000: A natural and reliable benchmark for data science code generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18319–18345. PMLR, 2023. doi: 10.48550/arXiv.2211.11501.

Andrej Leban and Yuekai Sun. CausalDS: Benchmarking causal reasoning in data-science agents. *arXiv preprint arXiv:2607.08093*, 2026. doi: 10.48550/arXiv.2607.08093.

David S. Lee and Thomas Lemieux. Regression discontinuity designs in economics. *Journal of Economic Literature*, 48(2):281–355, 2010. doi: 10.1257/jel.48.2.281.

Donggyu Lee, Hyeok Yun, Meeyoung Cha, Sungwon Park, Sangyoon Park, and Jihee Kim. EconCausal: A context-aware economic reasoning benchmark for large language models. *arXiv preprint arXiv:2510.07231*, 2025. doi: 10.48550/arXiv.2510.07231.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. doi: 10.48550/arXiv.2205.14334.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.48550/arXiv.2305.01210.

Chris Lu et al. Towards end-to-end automation of AI research. *Nature*, 651:914–919, 2026. doi: 10.1038/s41586-026-10265-5.

Jing Ma. Causal inference with large language model: A survey. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. doi: 10.18653/v1/2025.findings-naacl.327.

A. Craig MacKinlay. Event studies in economics and finance. *Journal of Economic Literature*, 35(1): 13–39, 1997.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards data-driven discovery with large language models. In *International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2407.01725.

Bang Nguyen, Dominik Soós, Qian Ma, Rochana R. Obadage, Zack Ranjan, Sai Koneru, Timothy M. Errington, Shakhlo Nematova, Sarah Rajtmajer, Jian Wu, and Meng Jiang. ReplicatorBench: Benchmarking LLM agents for replicability in social and behavioral sciences. *arXiv preprint arXiv:2602.11354*, 2026. doi: 10.48550/arXiv.2602.11354.

Benjamin A. Olken. Promises and perils of pre-analysis plans. *Journal of Economic Perspectives*, 29 (3):61–80, 2015. doi: 10.1257/jep.29.3.61.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2 edition, 2009. doi: 10.1017/CBO9780511803161.

Inioluwa Deborah Raji, Emily M. Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. AI and the everything in the whole wide world benchmark. *arXiv preprint arXiv:2111.15366*, 2021. doi: 10.48550/arXiv.2111.15366.

Jonathan Roth, Pedro H. C. Sant'Anna, Alyssa Bilinski, and John Poe. What's trending in difference-in-differences? a synthesis of the recent econometrics literature. *Journal of Econometrics*, 235(2): 2218–2244, 2023. doi: 10.1016/j.jeconom.2023.03.008.

Ayush Sawarni, Jiyuan Tan, and Vasilis Syrgkanis. CausalReasoningBenchmark: A real-world benchmark for disentangled evaluation of causal identification and estimation. *arXiv preprint arXiv:2602.20571*, 2026. doi: 10.48550/arXiv.2602.20571.

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. *arXiv preprint arXiv:2603.15542*, 2026. doi: 10.48550/arXiv.2603.15542.

R. Silberzahn, E. L. Uhlmann, D. P. Martin, et al. Many analysts, one data set: Making transparent how variations in analytic choices affect results. *Advances in Methods and Practices in Psychological Science*, 1(3):337–356, 2018. doi: 10.1177/2515245917747646.

Liyang Sun and Sarah Abraham. Estimating dynamic treatment effects in event studies with heterogeneous treatment effects. *Journal of Econometrics*, 225(2):175–199, 2021. doi: 10.1016/j.jeconom.2020.09.006.

Donald L. Thistlethwaite and Donald T. Campbell. Regression-discontinuity analysis: An alternative to the ex post facto experiment. *Journal of Educational Psychology*, 51(6):309–317, 1960. doi: 10.1037/h0044319.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, 2023. doi: 10.18653/v1/2023.emnlp-main.330.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2306.13063.

Linying Yang, Vik Shirvaikar, Oscar Clivio, and Fabian Falck. A critical review of causal reasoning benchmarks for large language models. *arXiv preprint arXiv:2407.08029*, 2024. doi: 10.48550/arXiv.2407.08029.

Matej Zecevic, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. Causal parrots: Large language models may talk causality but are not causal. *Transactions on Machine Learning Research*, 2023. doi: 10.48550/arXiv.2308.13067.

Yonghong Zhang, Shadi Motaali, Vu Phong Dinh, Avin Piroutiniya, Jorge E. López de Vergara, Luis de Pedro, Ricardo Correia, Isabel M. Parra, and Yong Xie. The double measurement confound in agent benchmarks: De-scaffolding, ground-truth scoring, and reliability beyond the mean. *arXiv preprint arXiv:2609.09218*, 2026. doi: 10.48550/arXiv.2609.09218.

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

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2406.15877. Oral.

A Reproduction and Pinned Versions

What re-runs. The release carries a `reproduce/` directory. One command verifies the sha256 of every pinned input and then recomputes each number printed in this paper from the lowest-level released files (per-(scenario, model) score rows, per-call ledgers with per-seed numerics, blinded verdict files), comparing it with the printed value and failing on any difference. No model call is needed. The design-layer freeze, the unblinded table and the results freeze of the paired study are rebuilt from the verdicts and rulings, and every bootstrap interval is reproduced with its recorded seed.

Execution replication. Re-executing all 700 cached model-written R scripts of the executable arm in a fresh clone reproduced the EXECUTES flag for 700 of 700 cells and the extracted coefficient for 425 of the 426 executing cells (the one difference, 1.3%, comes from an unseeded script and does not change its verdict). The extraction judge's cached values are not re-run. The canonical point estimate and standard error of all 460 paired seed-level references re-derive from the released twin data and estimators.

What does not re-run. Model generations are cached, not regenerated: endpoints are provider aliases and would not return the same text. Three audit counts of the paired study (614 eligible verdicts, 3 disagreements, 17 of 17 hedge re-reviews) exist only in a frozen summary; the sampled worksheet was not archived. Sealed per-twin truths are withheld by design and are not needed to recompute any reported number.

Table 2: Pinned model endpoints. Dates are first and last cached call; paired-study ledgers carry no timestamps, so their dates are those of the commits that archived them.

Arm	Endpoint identifier	Calls	Dates (2026)
Executable (R; cap 4,096)	`claude-opus-4-6`, `claude-sonnet-4-20250514`, `gpt-4o`, `o3`, `gemini-2.5-flash`, `moonshot-v1-128k`	100 each	03-27 to 04-14
	`gpt-5`	100	04-27 to 04-29
	`meta-llama/Llama-3.3-70B-Instruct` (robustness)	100	05-01
	`Qwen/Qwen3-235B-A22B-Instruct-2507`, `deepseek-ai/DeepSeek-V4-Pro`	100 each	07-26
Real-paper	the seven executable-arm endpoints above	259 each	04-27 to 04-28
Paired (Python; cap 8,192)	`claude-opus-4-6`, `claude-sonnet-4-6`, `gpt-5`, `gpt-4o`, `o3`, `gemini-2.5-flash`, `moonshot-v1-128k`, `Qwen/Qwen3-235B-A22B-Instruct-2507`, `deepseek-ai/DeepSeek-V4-Pro`	608 total	08-21 to 08-22

Seeds and scorers. Bootstrap seeds: 20260507 (τ, B=1,000), 20260424 (executed-but-wrong share, B=2,000), 20260825 (paired study, B=10,000, 22 pair clusters). Executable-arm verdict: exit code 0 within 60 s, and relative error below 0.50 against the canonical estimate on the same realised dataset. Paired verdict: point within $0.5\times$ the canonical standard error and standard-error ratio in $[0.8, 1.25]$, twin-set freeze v1.6, 20 committed seeds per pair.

B TOLERANCE SWEEP AND METRIC CROSSWALK

The binary CORRECT endpoint uses a broad default relative-error tolerance of 50%. To check that the qualitative conclusions do not depend on this cutoff, primary-panel pass rates are recomputed at 10%, 25%, 50%, 75% and 100%; the sweep is a robustness diagnostic and the frozen headline results keep the 50% default. Table 3 and fig. 5 report all-scenario pass rates, so non-executing outputs remain failures. Rankings are essentially tolerance-invariant. The final row aggregates the executed-but-wrong phenomenon across the panel: $P(\text{wrong} \mid \text{executable})$ is 27.2% at the strict 10% tolerance and still 15.5% and 14.1% at 50% and 100%. The phenomenon is not an artefact of a tight band.

Table 4 fixes the correspondence between this paper's semantic metric names and the L-codes used by the released v1 artifacts and scoring scripts, so results can be cross-checked against the frozen records without a translation step. Appendix text carried over from the v1 release refers to the real-paper arm as Exp A and to the executable arm as Exp B.

C EXECUTABLE-ARM DETAILS

This appendix reports supporting diagnostics for the executable arm (section 4). Every table is derived from the frozen v1 scoring file for the seven-model v1 panel (700 model–scenario cells) and none replaces the frozen headline rates of section 5. Two instruments appear throughout: EXECUTES (legacy code L2b), whether the model-written code runs, and CORRECT (legacy code L2b$^+$), whether the extracted estimate falls inside the band of eq. (2) around the canonical estimate.

Table 3: Tolerance-sweep diagnostic for the executable arm. Each cell is the pass rate out of 100 scenarios; parentheses give the model rank at that tolerance. The 50% column is the frozen headline threshold. The final row pools the panel: the share of *executing* workflows whose estimate still falls outside the band. Top three per column in blue, purple, orange; the pooled row is bold and uncoloured.

Model	**10%**	**25%**	**50%**	**75%**	**100%**
PER MODEL					
Opus	72% (1)	84% (1)	88% (1)	88% (1)	88% (1)
GPT-5	64% (2)	71% (2)	72% (2)	72% (2)	72% (2)
GPT-4o	53% (3)	59% (3)	62% (3)	63% (3)	63% (3)
Sonnet	42% (4)	48% (4)	50% (4)	50% (4)	50% (4)
o3	38% (5)	44% (5)	46% (5)	47% (5)	47% (5)
Gemini	32% (6)	32% (6)	32% (6)	32% (6)	32% (6)
Kimi	9% (7)	9% (7)	10% (7)	13% (7)	14% (7)
POOLED OVER THE PANEL					
P(**wrong** | **exec.**)	**27.2%**	**18.5%**	**15.5%**	**14.3%**	**14.1%**

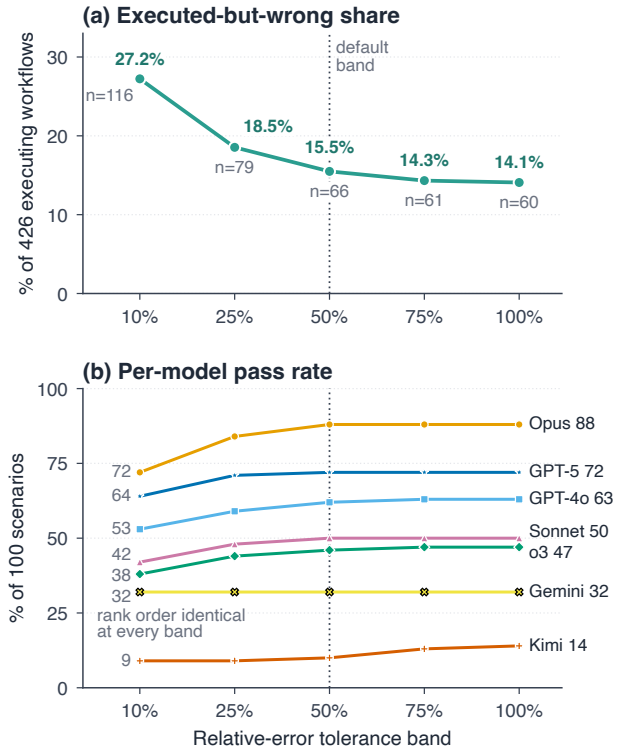


Figure 5: Tolerance sweep, drawn from table 3. **(a)** Executed-but-wrong share of the 426 executing workflows as the relative-error band widens from 10% to 100%; the default 50% band is the dotted line. **(b)** Per-model CORRECT rates at each band; the rank order is identical at every band.

C.1 CANONICAL ESTIMATORS PER DESIGN FAMILY

CORRECT compares the model-reported estimate to a fixed realised-data reference estimate. The canonical estimator is a benchmark-defined reference path for executable evaluation, not a claim that only one empirical analysis is scientifically defensible. It is also distinct from the structural DGP parameter used to simulate the data. Table 6 lists, per family, the reference estimate, the model-reported coefficients accepted as addressing it, and the coefficients most often reported instead.

Table 4: Metric crosswalk: semantic names used in this paper versus the L-codes of the released v1 artifacts.

This paper	v1 code	What it measures
METHOD-MATCH	L3	method-family agreement with reference labels
DIRECTION-MATCH	L4	sign/direction agreement with reference labels
EXECUTES	L2b	code executes in the fixed environment
CORRECT	L2b$^+$	executed estimate inside the frozen band

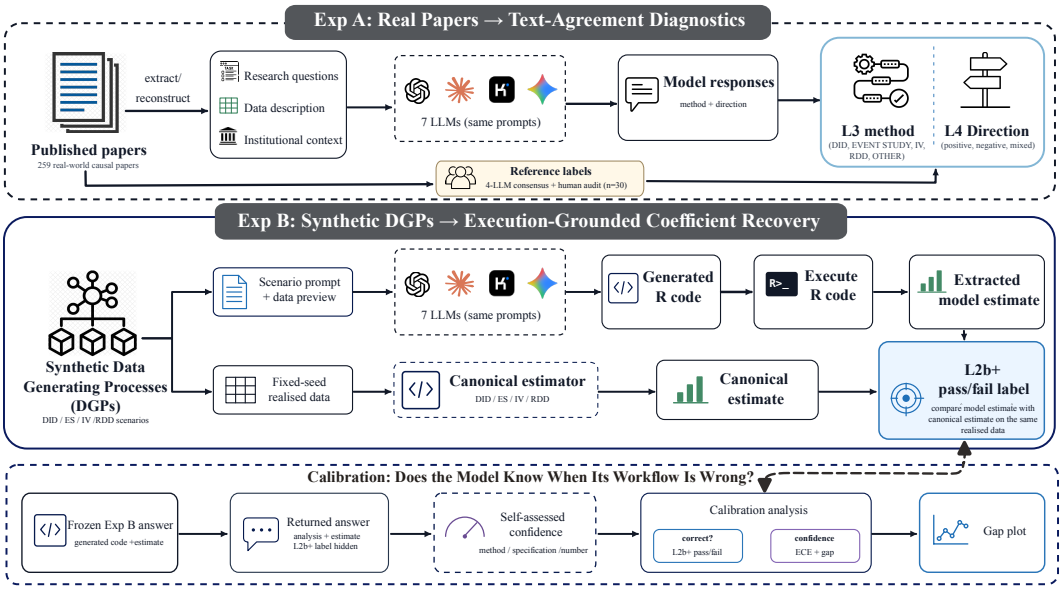


Figure 6: The v1 evaluation pipelines as released: the real-paper arm (top; 259 contexts, reference labels from a four-LLM consensus with a blinded human audit), the executable arm (middle; fixed-seed synthetic scenarios, model-written R code executed and compared with the canonical estimate on the same realised data) and the confidence-elicitation diagnostic (bottom). Labels use the v1 vocabulary (L3, L4, L2b$^+$); table 4 maps them to this paper's names. The paired protocol of section 6 is drawn in fig. 1.

Scope of each reference. Each reference is deliberately narrow. The DID target is a clean benchmark DID template, not a full benchmark of staggered or heterogeneous-treatment DID estimators (Callaway & Sant'Anna, 2021; Goodman-Bacon, 2021; Sun & Abraham, 2021). The event-study target is a fixed benchmark window mapping, not a universal event-study estimand (MacKinlay, 1997). The IV target tests implementation of the IV estimand (Imbens & Angrist, 1994; Angrist et al., 1996); it does not validate the exclusion restriction or weak-instrument diagnostics. The RDD target is a fixed benchmark RDD reference, not an exhaustive comparison of bandwidth or inference choices (Imbens & Lemieux, 2008; Lee & Lemieux, 2010; Calonico et al., 2014).

Because CORRECT requires code execution, the raw association between EXECUTES and CORRECT is partly structural. Table 7 therefore reports the conditional metric $P(\text{CORRECT} \mid \text{EXECUTES})$, which asks whether executed code computes the right coefficient.

C.2 FAILURE-TAXONOMY DIAGNOSTIC

We also audit the cells that fail CORRECT with a conservative rule-based taxonomy. The taxonomy is diagnostic rather than a scoring replacement: it distinguishes missing or unparseable code, execution failures, executed-but-unreported effects, wrong target coefficients, event-window or scale mismatches, IV stage errors, and RDD cutoff or sign-convention issues. Ambiguous cases are left uncategorized rather than forced into a narrow label. In table 8 the first two columns (230 execution

Table 5: Executable arm, per model (100 scenarios each). EXECUTES: code runs; CORRECT: executed estimate inside the 50% band; last two columns: how executed-but-wrong workflows fail. The two open-weight members were run later (section F). "Sonnet" here is `claude-sonnet-4-20250514`; the paired study uses `claude-sonnet-4-6`. Gemini's row is depressed by the harness: 55 of its 100 responses stopped at the 4,096-token output cap and 43 of those contain no code block (on its 45 complete responses it executes and recovers 25). Top three per column across the nine model rows in blue, purple, orange; for the two failure-count columns the three *lowest* are coloured, and tied values share a tier.

			CORRECT given	executed-but-wrong	
Model	EXECUTES	CORRECT	EXECUTES (%)	outside band	no coefficient
ORIGINAL PANEL (SEVEN)					
Opus	94	88	93.6	6	0
GPT-5	76	72	94.7	4	0
GPT-4o	78	62	79.5	5	11
Sonnet	51	50	98.0	1	0
o3	50	46	92.0	3	1
Gemini	32	32	100.0	0	0
Kimi	45	10	22.2	21	14
Pooled (seven)	**426**	**360**	**84.5**	**40 (9.4%)**	**26 (6.1%)**
OPEN-WEIGHT, RUN LATER					
Qwen3-235B	35	34	97.1	1	0
DeepSeek-V4	59	54	91.5	5	0

Table 6: Benchmark-defined canonical estimators used for CORRECT scoring on the executable arm.

Family	Canonical reference estimate	Accepted model-reported coefficient	Common non-target coefficients
DID	TWFE coefficient on `treat_x_post` after unit and period demeaning	`treat_x_post`, `treated:post`, or equivalent treatment-by-post effect	`treated` main effect, `post` main effect, placebo or pretrend coefficient
ES	Market-model coefficient on `event_day >= 0`; the final ES-aware scorer accepts frozen integer post-window conversions to this per-day scale	Explicit post-event AAR/CAR that can be deterministically converted to the canonical scale	Pre-event coefficient, arbitrary non-frozen window, diagnostic or placebo return
IV	Realised-data 2SLS coefficient on `x`, instrumented by `z`, with controls exogenous when present	Second-stage coefficient on `x` or the named endogenous variable	First-stage coefficient on `z`, reduced form, or control coefficient
RDD	Local-linear cutoff effect on the realised data: coefficient on `treated` near cutoff 0.5 with running-variable slope and interaction	Treatment, above-cutoff, or discontinuity coefficient consistent with benchmark direction	Density test, bandwidth statistic, unrelated global polynomial trend, placebo cutoff estimate

failures plus 44 no-code cells) are the 274 cells that fail EXECUTES, and the remaining six columns sum to the 66 executed-but-wrong cells of table 7.

C.3 RANK-STABILITY DIAGNOSTIC

The cross-layer rank correlations of section 5 are descriptive model-level diagnostics over the seven v1-panel models rather than population-level estimates. To assess panel sensitivity, Kendall τ and Spearman ρ between the EXECUTES ranking and the CORRECT ranking are recomputed after leaving out each *model* in turn. The leave-one-model-out Kendall range is 0.733–0.867 and the Spearman range is 0.886–0.943 (table 9). Adding Llama-3.3-70B-Instruct as a robustness-only eighth model

Table 7: Conditional correctness among executed outputs (counts out of 100 scenarios per model). "Wrong after exec." is the EXECUTES count minus the CORRECT count; the pooled row sums the seven models and is the source of the 15.5% (66/426) executed-but-wrong share in section 5. Top three per column in blue, purple, orange for the columns where higher is better (EXECUTES, CORRECT, and the conditional rate); the pooled row is left uncoloured.

Model	EXECUTES	CORRECT	Wrong after exec.	P(CORRECT | EXECUTES)	Gap
PER MODEL (100 SCENARIOS EACH)					
Opus	94	88	6	93.6%	6 pp
GPT-5	76	72	4	94.7%	4 pp
GPT-4o	78	62	16	79.5%	16 pp
Sonnet	51	50	1	98.0%	1 pp
o3	50	46	4	92.0%	4 pp
Gemini	32	32	0	100.0%	0 pp
Kimi	45	10	35	22.2%	35 pp
Pooled	426	360	66	84.5%	–

Table 8: Failure-taxonomy diagnostic on the executable arm, per v1-panel model. Cells show counts and (share of that model's cells failing CORRECT). Strong models execute correctly when they execute (Opus, Sonnet, Gemini conditional pass rates $\geq 93\%$; see table 7), while weaker models more often produce executed-but-wrong code (Kimi 22.2%, GPT-4o 79.5% conditional pass). Per-model rows sum to 340 failing cells, matching the aggregate decomposition (Exec fail 230, No code 44, Wrong coef 26, Coef-not-reported 15, Window/scale 11, IV stage 5, RDD 3, Other 6). Bold marks the largest failure category in each row.

Model	n	Exec fail	No code	Wrong coef	Coef N/R	Win/scale	IV stage	RDD	Other
Opus	12	**5 (41.7%)**	1 (8.3%)	2 (16.7%)	0	3 (25.0%)	0	0	1 (8.3%)
Sonnet	50	**49 (98.0%)**	0	1 (2.0%)	0	0	0	0	0
GPT-4o	38	**22 (57.9%)**	0	3 (7.9%)	1 (2.6%)	3 (7.9%)	4 (10.5%)	0	5 (13.2%)
o3	54	**50 (92.6%)**	0	2 (3.7%)	1 (1.9%)	1 (1.9%)	0	0	0
Kimi	90	**55 (61.1%)**	0	18 (20.0%)	13 (14.4%)	3 (3.3%)	1 (1.1%)	0	0
Gemini	68	25 (36.8%)	**43 (63.2%)**	0	0	0	0	0	0
GPT-5	28	**24 (85.7%)**	0	0	0	1 (3.6%)	0	3 (10.7%)	0

gives Kendall $\tau = 0.714$ and Spearman $\rho = 0.881$, while the ranked v1 panel remains the seven models.

Table 9: Rank stability of the model-level correlation between the EXECUTES ranking and the CORRECT ranking on the executable arm. The Llama row is robustness-only and never enters the ranking. Bold marks the headline full-panel row, with its coefficients in blue.

Diagnostic	N models	Kendall τ	Spearman ρ
SEVEN-MODEL V1 PANEL (RANKED)			
Seven-model baseline	**7**	**0.810**	**0.929**
Scenario-clustered bootstrap 95% CI (1,000 resamples)	7	[0.62, 0.90]	[0.75, 0.96]
Leave-one-model-out range	6 each	0.733–0.867	0.886–0.943
EIGHTH MODEL ADDED (ROBUSTNESS-ONLY, NOT RANKED)			
Seven plus Llama (robustness-only)	8	0.714	0.881

Leave-one-design-out. A second diagnostic, derived deterministically from the same frozen scoring file after the v1 release, removes one design *family* at a time and re-ranks the seven models by CORRECT on the reduced panel at the default 50% tolerance. The statistic is Kendall's τ_a between the CORRECT ranking on the *full* panel and the CORRECT ranking on the *reduced* panel (seven models, 21 pairs; a tie counts as neither concordant nor discordant), so it measures how stable the recovery ranking itself is to the family mix. It is a different quantity from the cross-layer τ(EXECUTES, CORRECT) of

table 9, and the two sets of values are not comparable: the DID-excluded value coincides numerically with the cross-layer baseline (both $17/21 = 0.810$) but correlates a different pair of rankings. In table 10, τ_a is 0.810, 0.905, 0.952 and 1.000; the top-two set {Opus, GPT-5} and the bottom-two set {Gemini, Kimi} are intact in all four reduced panels, and every movement is confined to ranks 3–5.

Table 10: Leave-one-design-out diagnostic on the executable arm (seven-model v1 panel, 50% tolerance). τ_a compares the reduced-panel CORRECT ranking with the full-panel CORRECT ranking; ranks 1–2 (Opus, GPT-5) and 6–7 (Gemini, Kimi) never change, so only ranks 3–5 are shown. "Wrong / exec." pools the seven models: executing workflows whose estimate falls outside the band, over executing workflows. Top three per column in blue, purple, orange (highest τ_a; lowest share); the full-panel reference row is in bold and is not ranked.

Panel	Scenarios	τ_a vs. full	Wrong / exec.	Share	Ranks 3–5 by CORRECT
REDUCED PANELS (ONE FAMILY EXCLUDED)					
Excluding DID	70	0.810	46/285	16.1%	o3 > GPT-4o > Sonnet
Excluding ES	76	0.905	51/320	15.9%	GPT-4o > o3 > Sonnet
Excluding IV	76	0.952	51/317	16.1%	GPT-4o = Sonnet > o3[†]
Excluding RDD	78	1.000	50/356	14.0%	GPT-4o > Sonnet > o3
SINGLE-FAMILY PANELS					
DID only	30	n/a	20/141	14.2%	n/a
ES only	24	n/a	15/106	14.2%	n/a
IV only	24	n/a	15/109	13.8%	n/a
RDD only	22	n/a	16/70	22.9%	n/a
Full panel	**100**	n/a	**66/426**	**15.5%**	**GPT-4o > Sonnet > o3**

[†]Exact tie, not a reversal: with the IV scenarios removed GPT-4o and Sonnet both pass 47 of 76, which is why $\tau_a = 20/21$ rather than 1.

The executed-but-wrong share is equally insensitive to the family mix: 14.0%–16.1% in every reduced panel against 15.5% on the full panel, and 13.8%–14.2% within DID, ES and IV, with RDD highest (22.9%) on the smallest executing base. Across tolerances the same pooled share is 116/426 (27.2%), 79/426 (18.5%), 66/426 (15.5%), 61/426 (14.3%) and 60/426 (14.1%) at 10%, 25%, 50%, 75% and 100%, completing the three values shown in the final row of table 3.

D PAIRED-ANALYSIS PROTOCOL

This appendix records the protocol behind section 6. Its normative documents are among the 17 set-wide artifacts hashed by the twin-set freeze; the panel ran under freeze v1.6 throughout.

Eligibility and gold annotation. Selection rules were frozen before any candidate was ranked (Casey et al., 2012; Olken, 2015); no model output or benchmark score may be consulted. A paper is eligible only if its bibliographic identity is externally verified, its inputs are legally reproducible, and a counterfactual perturbation can be built without changing the design class: its "operative identification strategy must belong to one of the four preregistered design families, and the synthetic twin must preserve that strategy without reinterpreting the study into another design class." An operative-design re-audit froze a shortlist of 23 (DID 6, IV 7, RDD 7, event study 3). The gold CDR describes the design, not the authors' code, with one scored primary estimand per task, chosen where the source designates none by a frozen hierarchy in which significance, magnitude and sign are never criteria; golds were drafted and source-checked by language-model agents, approved in batches by an author, and hashed before any twin is built.

Twin construction, seeds and sealed truth. The CDR is the only interface between paper and twin: the generator consumes the design payload and twin requirements and nothing else. A twin preserves the full design structure (instrument and first stage, adoption timing, running variable, compliance types); design-class flips are prohibited. Paper identity is conceded; protection is value-level. Each planted coefficient draws a regime (strong positive, strong negative, near null) by a sealed seed, block-balanced within family, and observed sign, significance and null status are not permitted inputs. The planted magnitude is $|\tau^*| = t^* \times SE_0$, with SE_0 the null-scaffold canonical SE and

$t^* \sim U(2.5, 6)$ (strong) or $U(0, 1)$ (near null); scales carry a per-twin factor from $U(0.25, 4)$, and τ^* is redrawn if within $2\times$ the scoring band of a comparable published value. Acceptance (recovery, coverage, SE calibration) uses at least 300 internal seeds per twin under pre-registered bands; *20 seeds per pair* are committed for evaluation. Each seed file's canonical estimate and SE sit in a sealed truth file, committed by hash and never exposed, and seed k is scored against seed k's values: match-to-canonical, not match-to-τ^*.

CDR schema. A CDR is a JSON object with exactly four top-level fields: `family` (`DID`, `IV`, `RDD`, `EVENT_STUDY`), `family_subtype`, `core` and a family-specific `family_module`. The `core` block has 13 required fields (`estimand`, `causal_exposure_concept`, `outcome`, `identifying_variation`, `sample_definition`, `unit_of_analysis`, `time_structure`, `validity_conditioning`, `fixed_effects`, `controls`, `clustering`, `target_coefficient`, `identification_assumptions`) and four optional ones (`comparison_group`, `variable_transformations`, `weights`, `auxiliary_reduced_form_estimand`). Every freeze version hashes schema v1.0; the v1.1–v1.2 revisions (nullable fields; an `OTHER` event-study subtype) postdate the panel evidence freeze and leave this inventory otherwise unchanged.

Critical fields for V_2. The design-layer instrument profiles 16 fields, each MATCH, MATERIAL MISMATCH, NOT APPLICABLE or UNRESOLVED, with no aggregate score. Seven are critical in the pilot, where any critical mismatch fails the run and a critical UNRESOLVED caused by benchmark ambiguity excludes it (CDR path read in brackets where not eponymous): `family` [mechanical], `estimand`, `target_coefficient`, `treatment_identity` [`family_module`], `transformations_and_scale` [`core.variable_transformations`], `comparison_assignment` [`core.comparison_group`, `core.sample_definition`] and `inference_convention` [`core.clustering`]; the other nine are profile-only. In the panel the frozen scoring rule names five scored critical fields, all adjudicated: `inference_convention` left the pass/fail set (rule R1 below) but stays in the critical field-verdict ledger, which thus spans six adjudicated fields per commitment. The mechanical `family` comparison remains scored but sits outside that ledger: label-only differences over a form-identical estimating equation are overridden to MATCH, and every remaining family mismatch coincides with a commitment already carrying countable recast mismatches.

Stage-1 commitment contract. Every provider call carries one system prompt ("You are an applied econometrician. You implement research designs exactly as specified and report what your implementation produces.") and one user message. At Stage 1 that message holds an instruction block ("Commit to a causal design. Say what you would estimate and how you would identify it, as a structured representation, BEFORE seeing any data."), three context fields (research question, data description, institutional context) and an answer contract. The context is a rendered, hashed per-task artifact written under two stop rules: it never asserts that identification assumptions hold and never reports a realised first stage, result direction or significance. A lint finds no method name in any of the 23 contexts, and task identifiers are opaque.

The answer contract asks for one JSON object in the schema's shape; the verbatim templates (the Stage-1 contract alone is 35 lines) are hashed with the prompt builder and are in the released archive. The validity gate is deliberately narrow (`family` is one of the four, `core` is an object, and `core.estimand`, `core.outcome` and `core.target_coefficient` are non-empty) so that any other omission is measured rather than rejected. A valid commitment is hashed at once (SHA-256 over canonical JSON) inside a function that resolves no seed path, data dictionary or data schema. There is no retry or repair: a gate-failing response is a recorded Stage-1 failure whose main arm does not run, for which the gold CDR is never substituted, and which is *not scorable* at V_2, never a design failure.

Stage-2 execution contract. Stage 2 is a second, stateless provider call, not a continued chat session: continuity is carried by the commitment, re-presented verbatim under the heading "THE DESIGN YOU COMMITTED TO" and recorded by hash on the main-arm result. The message adds the twin's data dictionary, a data schema (column names, dtypes and row count: the shape of the data, never its values) and the output contract. The model returns one self-contained Python script, run as `python your_script.py -data <one-seed.parquet>` separately on each of the 20 seed files,

which must exit 0 after printing one JSON object with `point_estimate` and `standard_error`. It may import numpy, pandas and pyarrow and nothing else beyond the standard library, must read only the file passed, must not require network access, and must run in under two minutes per seed. Every call has an 8,192-token output cap and the mode is single-shot: a script that fails to run is a recorded failure, not a retry. Scripts execute in a network-isolated, read-only container holding only an interpreter and those three libraries, and before any live call the runner executes an adversarial escape probe (sealed truth, host credentials, network) and refuses to proceed unless every vector is closed.

Oracle arm. The third call never invokes Stage 1 and runs whether or not the model's own commitment was valid. Under the heading "DESIGN SPECIFICATION" it substitutes the *model-facing view* of the gold CDR (the gold design payload, unchanged, plus an `execution_conventions` block) for the committed design; every other prompt section, the system prompt, token cap, model configuration, twin, seeds, sandbox and scorer are held fixed. One asymmetry belongs to the contrast: that block pins the standard-error form, degrees of freedom, sample rules and units, whereas a committed CDR carries conventions only insofar as the model wrote them.

Scoring: V_4 bands. The bands are frozen in the planting registry and read, as written there, `point_band`: $|\text{est} - \text{canonical}| < 0.5 \times SE_{\text{canonical}}$ and `se_ratio_band`: $SE_{\text{model}}/SE_{\text{canonical}} \in [0.8, 1.25]$; a seed is recovered only if both hold on that seed's canonical values. The point band sits strictly below the $2 \times SE$ displacement required of every registered wrong-specification probe; the 50% relative tolerance of the v1 release is not reused. Two registry-named variants exist: a diagnostic twin is scored on its standardized statistic ($|t_{\text{model}} - t_{\text{canonical}}| < 0.5$) plus 5%-level test agreement, and a two-endogenous twin on one coefficient only. A recitation diagnostic flags an estimate that misses the point band yet lies within $0.5 \times SE_{\text{canonical}}$ of a comparable published value.

Scoring: V_3 and denominators. A seed is *executed* when the script exits 0 and prints a JSON object with numeric `point_estimate` and `standard_error`. Four denominators stay separate: *attempts* (model $\times$ pair; 207); *scored commitments* for V_2 (194 schema-valid less the seven of the reference-defective pair; 187); *eligible seeds* for V_3 (20$\times$ the Stage-1-valid commitments; an arm that never ran counts as 20 failed seeds); and *executed seeds* for V_4, undefined rather than zero when nothing executed. These units were fixed before the first panel verdict; seeds are never independent observations.

V_2 adjudication: pilot instrument. Free-text fields take their verdict from an adjudication log recorded as data (agent-drafted rulings reviewed by an author), never from a model judging at run time; each of the 90 pilot rulings is hash-bound to the values it was made on, and each run is compared against the gold in force when it executed (freeze v1.3 for one run, v1.4 for five).

V_2 adjudication: outcome-blinded panel batches. The panel contract was frozen before the first verdict and leaves the instrument unchanged. The adjudicator reads batch files containing only the gold design payload and anonymised commitments (no model identity, no V_3/V_4, no oracle outcome, no usage), with the label-to-model mapping in a sealed file. There are 23 batches, one per pair, each with up to nine commitments in seeded-shuffled order, so that one gold's field semantics are ruled consistently across models. Uncovered questions are resolved together in a dated addendum of general rules, with equivalence argued on construction, never on reasonableness. Unblinding comes last, after the reliability audit and the V_2 profile freeze; later corrections must appear as labelled post-unblind errata.

V_2 adjudication: reference review. A field on which at least eight of nine commitments in a batch receive the same mismatch is automatically registered for review of the *gold*: an instrument-anomaly signal, never a verdict change; 30 flags were reviewed. Each source reviewer receives only the pair identifier, flagged field, gold claim, frozen Stage-1 context and archived source (never the commitments or mismatch counts) and returns quotations, not verdicts. Source fidelity (verified, defect, ambiguous) and model-facing determinability (determinate, underdetermined) are ruled in separate columns: a verified but underdetermined gold makes the field *not scorable*; a confirmed defect excludes the pair, layer-scoped, with no post-hoc gold repair. Verdict files are never rewritten.

V_2 adjudication: benchmark-underdetermined removals. The eight-of-nine threshold was a review trigger, not a scoring criterion: once review found mismatch classes arising from information absent from the model-facing context, the two-column rule was applied to every mismatch of the same semantic class however many models exhibited it ("disposition follows information availability, not flag frequency") under three rules frozen before model identities or downstream outcomes were accessed. R1: after verifying that none of the 23 frozen contexts pins an inference convention, that field exits V_2 pass/fail globally (V_4 still scores SE recovery). R2: each unflagged mismatch class receives its flagged counterpart's ruling item by item, with no blanket void. R3: every surviving critical mismatch is re-examined individually and is countable only on stated grounds (family recast; misread of a stated estimand; violated context-visible sample rule; internal incoherence; rewritten visible identification structure; ignored stated transformation). The ledger closes exactly: of $194 \times 6 = 1{,}164$ critical field-verdicts, 760 are matches, 75 countable mismatches, 318 voids (160 reference review, 46 R1, 112 R2/R3), 8 unresolved and 3 not applicable.

V_2 adjudication: re-adjudication audit. The 614 critical-field MATCH verdicts that remain scored after the overlays form the eligible population; a 20% random sample of 123 was re-adjudicated, with the seed (20260823) and sampling rule fixed at the two-batch checkpoint and the draw made only after every verdict was locked. Three of the 123 disagreed (97.6% agreement, 120 of 123), and all three were corrected to countable mismatches, included in the 75 above. A separately reported stress audit re-reviewed all 17 hedge or deferral matches and found no disagreement.

Amendment record. Superseded manifests are never rewritten. The initial freeze (v1.0) has six dated amendments. Four are dated 2026-08-18, the first three before any paired evaluation call: v1.1 removed a metadata block carrying a legacy identifier from all 23 model-facing views; v1.2 rewrote the institutional context of 21 of 23 tasks, found to state identification arguments or source results as premises, and made Stage-1 inputs hashed per-task artifacts; v1.3 implemented the three-call orchestration; v1.4, triggered by the first paid pilot run, changes only what is recorded, not what is measured. Two amendments declare themselves *not result-blind*: v1.5 (2026-08-20) adds a coefficient-scale pin to one task and a units pin to another after a paid pilot trajectory (excluded, never re-scored) exposed a source-to-twin unit ambiguity; v1.6 (2026-08-21) replaces one task's research question and data description after the pilot design-profile review found them describing a companion study. Neither alters a DGP, planted effect, canonical estimator, scoring band or archived trajectory. Separately, the design-profile instrument lies outside the twin-set freeze and is itself not result-blind for the pilot; its panel extension is outcome-blinded by construction.

E Paired Study: Additional Details

The protocol itself is drawn in fig. 1 (panels C and D); this appendix records the adjudication, model settings and pilot material behind section 6.1.

Adjudication of the design layer. The full panel runs all 23 pairs across nine models under twin freeze v1.6: 207 Stage-1 attempts, of which 13 produce no schema-valid commitment and are recorded as not scorable (never as design failures), leaving 194 committed designs. Design-specification verdicts were adjudicated outcome-blinded in per-pair batches with model identities sealed. A reference review by commitment-blinded reviewers (language-model agents, section 7) then verified every flagged gold claim against the archived source and the frozen Stage-1 context, ruling source fidelity and model-facing determinability in separate columns. For one pair (cvp_023) the archived document is a later replication study of the cited paper rather than the paper itself; its Stage-1 context was reconstructed from that document while its gold cites the original, and its unit-of-analysis field was recorded as a context-fidelity defect and removed from scoring. Mismatches that concerned conventions absent from the model-facing context (estimator-column anointments, footnote clustering keys, instrument lag structures) were removed from scoring as benchmark-underdetermined rather than charged as model error, uniformly and irrespective of how many models exhibited them; the inference-convention field left the scored set entirely, since no frozen context states one, and is reported as a descriptive alignment profile. One pair shipped a defective gold (a literal draft marker that also reached its oracle prompts) and is excluded layer-scoped. A seeded 20% re-adjudication of eligible MATCH verdicts agreed at 97.6% (120 of 123 re-adjudicated, from 614 eligible; three corrections, all adopted); a 100% re-review of hedge/deferral verdicts agreed at 17/17.

Models and inference settings. The primary panel evaluates nine models under identical contracts (endpoint identifiers as archived in the per-call ledgers): `claude-opus-4-6`, `claude-sonnet-4-6`, `gpt-5`, `gpt-4o`, `o3`, `gemini-2.5-flash`, `moonshot-v1-128k`, `Qwen/Qwen3-235B-A22B-Instruct-2507`, and `deepseek-ai/DeepSeek-V4-Pro`, all run in August 2026 with provider-default sampling settings, single-shot with no retry or repair, three calls per pair (Stage-1 commitment, main arm, oracle arm), an 8,192-token output cap, and an execution contract restricted to numpy/pandas/pyarrow and the Python standard library. Endpoint identifiers, per-call token ledgers, prompt hashes and submitted scripts are in the released trajectory archive; the identifiers are provider aliases and the ledgers carry no timestamps. Of the 35 paired calls that stopped at the 8,192-token output cap, 24 were Gemini, 9 GPT-5 and 2 o3; the latter two spent the whole budget on hidden reasoning tokens.

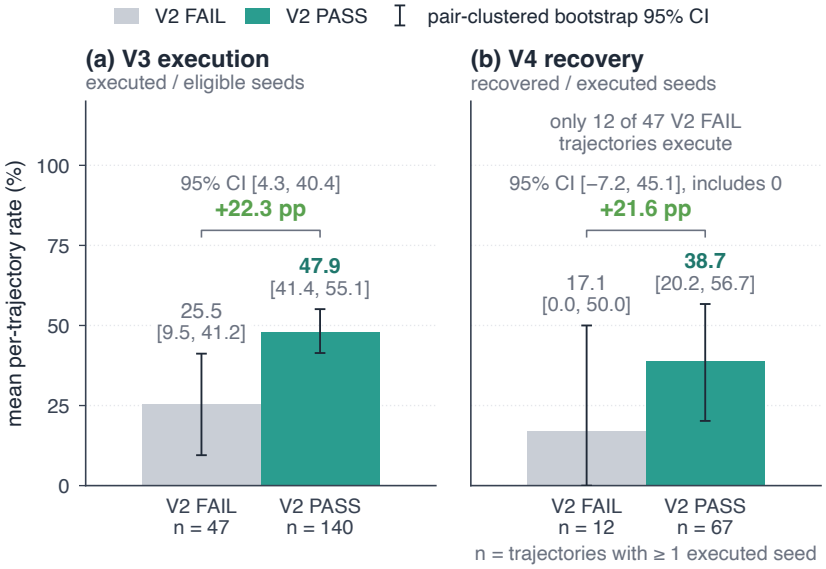


Figure 7: Conditional downstream execution (V_3) and recovery (V_4) by V2 status, with pair-clustered bootstrap 95% CIs (Efron & Tibshirani, 1994). The recovery difference's interval includes zero (12 executing V2-fail trajectories); the association is directionally consistent, not resolved. V_2 status is not randomized; these are conditional associations, not a causal decomposition.

E.1 A TWO-ARM PILOT ON SIX PAIRS

Before the full panel, six of the 23 pairs were run end to end with one model (`gpt-5.6-sol`, the archived ledger identifier for the pilot-era GPT-5.6 endpoint; single-shot, both arms; three provider calls per pair). Every trajectory is archived under a hashed evidence freeze; after the telemetry contract was expanded mid-programme, runs additionally preserve the committed CDR, call-level usage, the submitted scripts and per-seed numerics. The earlier Bernanke–Kuttner run predates that expansion and retains only the categorical verdicts of note a. Table 11 reports the result as a layer profile per pair, not a success rate: with six trajectories the deliverable is where each chain held or broke.

Three regularities in the profile are worth stating plainly, without generalizing beyond n=6. First, the *oracle arm is uniform*: with the gold CDR substituted, every seed of every pair executes and scores inside the frozen bands. Under this protocol that is a substitution contrast: the gold-CDR substitution changed downstream execution and recovery wherever the arms diverge. It is not a strict causal decomposition of error sources. Second, *family recognition is not the bottleneck*: all six committed CDRs name the correct design family; every failure occurs downstream of V_1. Third, the four distinct main-arm outcomes form a small taxonomy of *observed trajectory types*, two of which are not failures at all, rather than a rate of how often anything succeeds:

- **Full chain** (Bernanke–Kuttner, Oregon HIE): no critical-field mismatch in the committed CDR, execution on every seed, and recovery inside the frozen bands on every executed seed

Table 11: Layer profile of the six-pair pilot. V_1: committed family correct. V_2: critical-field mismatches in the committed CDR against the contemporaneous gold CDR (seven critical fields adjudicated MATCH/MISMATCH per field, non-critical profile mismatches in the notes, no aggregate score). V_3: seeds executed (main arm). V_4: executed seeds whose estimate and SE fall inside the frozen bands. The oracle arm reruns Stage 2 with the gold CDR.

Pair	V_1	V_2 profile	V_3 main	V_4 main	Oracle V_3 / V_4
Bernanke–Kuttner	✓	0 critical	20/20	20/20[a]	20/20
AJR	✓	0 critical[b]	19/20[c]	19/19	20/20
ADH	✓	excluded[d]	20/20	n/a[d]	20/20
Oregon HIE	✓	0 critical[b]	20/20	20/20	20/20
Dell (Mita)	✓	**4 critical**	20/20	**0/20**	20/20
KLM	✓	**5 critical**	**0/20**[e]	n/a	20/20

[a]Boolean verdicts preserved; this run predates the expanded telemetry contract, so its per-seed numerics are recorded as permanently unknown rather than backfilled. [b]Non-critical profile mismatches recorded (AJR 3, Oregon 2); see text. [c]One seed lost to a container abort: execution-conditioned success is 19/19 and end-to-end seed completion 19/20; the missing seed is infrastructure, not model behaviour. [d]Benchmark-side unit-contract ambiguity discovered by this run and resolved prospectively by a dated amendment; the trajectory is excluded from model-performance inference in both directions. [e]The model's own feasibility guard: its committed market-model estimation window (-250 to -31, $\geq$120 observations) cannot exist in a twin shipping 60 trading days per firm, and the script raised rather than substituting an uncommitted design.

($V_2 \to V_3 \to V_4$). For Bernanke–Kuttner the recovery claim is categorical (note a): the preserved verdicts, not per-seed values.

- **Design divergence, faithfully propagated** (Dell): the committed CDR differs from the gold on four critical fields at once (target specification, running variable, band and analysis unit, inference), adjudicated from the committed fields alone, before any reference to outcomes; the archived script implements that commitment on every seed and is wrong on every seed: sign-unstable across seeds, standard errors 16–28$\times$ the canonical. Surface-plausibility and runnability checks would not establish correctness here; reference-based numerical validation exposes the discrepancy.
- **Infeasible commitment, model-generated guard halt** (KLM): the commitment reproduces a recognizable source-paper event-study specification, including a pre-event estimation window, but diverges from the benchmark gold on five critical fields and names an estimation window the twin's data geometry cannot contain; Stage 1 closes before the data dictionary is visible, and the single-shot contract offers no renegotiation. A model-generated feasibility guard then caused a submission-level execution failure on every seed. The taxonomy records the mechanism (a guard, not crashing code) as a distinction of kind, not a credit.
- **Contract ambiguity, charged to the benchmark** (ADH): the model implemented exactly the per-\$1,000 estimand it committed, against a twin whose canonical execution treats the shipped exposure as an already-normalized coefficient unit, a correspondence the contract in force did not explicitly bridge; estimates and SEs scale together by $\sim$1,055$\times$ and the t-statistic is nearly unchanged, so every text-level check passes. The run is excluded from model-performance inference in both directions (it is neither a model failure nor, under the later clarified contract, retroactively a pass) and its lesson kept: the mismatch is invisible to any check short of comparing coefficient magnitude to an executable reference.

AJR sits between the taxonomy's poles and illustrates why V_2 is a profile rather than a score: the commitment adds conditioning (latitude, continent effects) that the gold's anointed specification excludes (three non-critical field mismatches and no critical-field mismatch), yet the twin ships no such columns, the implementation estimates the realizable core design, and recovery passes on every executed seed. A scalar V_2 score would have averaged this into noise; the profile keeps both facts.

The V_2 instrument and its 90-entry adjudication log were built after these outcomes were known and are recorded as such: not result-blind, with structural mitigations (field-semantics verdicts, rationales that stand without reference to V_4, no code path from the profiler to execution output, and per-run binding to the contract version in force at execution time).

E.2 THE FOUR-TASK PROTOTYPE ACROSS NINE MODELS

The pilot's protocol descends from a smaller prototype run earlier in the project: four tasks (one per family) under the same two-stage contract (brief, then a committed interpretation, then data, then code and execution) with Stage-1 scores frozen before Stage 2 and canonical results precomputed. Nine models completed all four tasks. The prototype is targeted validation at descriptive scale, and its value is the structure of its disagreements rather than its rates.

Results. Method recognition was near-ceiling (35 of 36 cells) and execution high (33 of 36), yet recovery separated: the event-study and DID tasks resolved cleanly (9/9 and 8/8 of executing models correct), while the RDD task executed for all nine models and recovered for six. The three misses are the instructive part: two vendors' models (GPT-4o and DeepSeek) produced the *same* wrong estimate, 5.075 against a canonical 1.428 ($3.55\times$, relative error 2.55), and a third missed at 0.59.

Case study: the $3.55\times$ estimate. Both models committed the same specification error, an `rdrobust` call (Calonico et al., 2014) on truncated support that leaves roughly 21 effective observations, and both returned an estimate that reads as an ordinary, confident RDD result. The workflow passes a text check and a does-it-run check; only the comparison against the canonical estimate exposes it, and the cross-vendor coincidence of the exact value is what marks it as a systematic specification failure rather than a sampling accident. This is the observation the paired protocol was then built to instrument at scale.

F OPEN-WEIGHT RUNS

This appendix documents the open-weight rows quoted in section 5: the protocol under which Qwen3-235B-A22B-Instruct-2507 and DeepSeek-V4 were run, one correction to the execution environment, and the full counts. Llama-3.3-70B-Instruct is listed alongside them as a robustness check only; it is never part of the model ranking, and the three rows are not a benchmark of open-weight models.

Protocol. The two additional families were run after the v1 release was frozen, under a protocol that was prespecified and frozen before the first full-panel generation call and before any scoring. The arms are `Qwen/Qwen3-235B-A22B-Instruct-2507` and `deepseek-ai/DeepSeek-V4-Pro` (reported as DeepSeek-V4), both served through a hosted inference endpoint whose resolved model string was checked against the requested one by a single liveness call per arm. Everything else is inherited unchanged from the v1 executable arm: the same 100 scenarios and realised datasets, the same system and user prompts, a 4,096-token output budget, provider-default sampling with no reasoning-mode parameters, a single shot per scenario, no retry on a substantive failure, and no per-model prompt adjustment. Scoring reuses the v1 chain without modification: R execution, coefficient extraction, comparison with the canonical estimate under eq. (2) at $\tau = 0.50$, and the event-study rule that accepts a cumulative window of up to 11 post-event days. Canonical estimates were read from the frozen v1 artifacts and never recomputed. The protocol fixed the endpoints (EXECUTES, CORRECT, and $P(\text{wrong} \mid \text{executable})$ per arm, with a by-family breakdown) and the wording of three possible readings in advance, and committed to reporting every arm whatever it showed. Coefficient extraction for the new outputs required new calls to the extraction model, because these outputs are absent from the v1 extraction cache.

Realised sample. All 200 generations completed (100 per arm) with no transient retries, no failures, and no truncated responses; every response contained an R code block, so each arm's denominator is 100 and every non-executing cell is code that was extracted and then failed at run time. The runs wrote only to their own directory, and the hashes of the two v1 score tables, recorded before the run, matched after each scoring pass.

Environment correction. The numbers in this paper come from the second of two scoring passes; the first is superseded. To run unvetted model-written R code safely, the two new arms were executed in a network-isolated container (R 4.4.2, no credentials mounted) rather than on the host R installation that scored the v1 panel. The first container was built from the *documented* v1 package list, which turned out to be narrower than the package set actually present when the v1 panel was

scored. In that first pass Qwen3 executed on 17 scenarios (16 correct; 1 of 17 wrong) and DeepSeek-V4 on 46 (42 correct; 4 of 46 wrong). An audit of every non-executing cell (83 and 54) found no code-extraction failures; 73 and 27 of them were "there is no package called" errors, 65 of those 100 for `readr` alone and the rest for other routine packages that, by the protocol record, v1-panel workflows had loaded successfully. This is a harness deviation, not model behaviour, and it made the first-pass counts non-comparable with the v1 panel. The remedy was written into the protocol record before re-scoring: a derived image adds exactly the eleven packages evidenced in the v1 environment (`readr`, `tidyverse`, `rddensity`, `lubridate`, `ivreg`, `plm`, `estimatr`, `quantreg`, `modelsummary`, `foreign`, `knitr`). `rdd` and `stargazer` were deliberately *not* added: they were absent from the v1 environment too (the v1 score table records 15 and 24 failures on them), so failing on them is a genuine failure under v1 semantics. Both arms were re-scored uniformly with the unchanged chain, no output was regenerated, and the first-pass scores, summary, and execution logs are archived. The correction is post hoc and not result-blind: it was triggered by the first-pass results, and it raises EXECUTES from 17 to 35 (Qwen3) and from 46 to 59 (DeepSeek-V4). It barely moves the conditional quantity the main text relies on (1/17 to 1/35; 4/46 to 5/59), and no cell that executed in the first pass changes status: the 18 and 13 newly executing cells add 18 and 12 correct workflows. The 26 missing-package failures that remain (25 Qwen3, 1 DeepSeek-V4) all name packages recorded as absent from the v1 environment as well (`lfe` 11, `ivmodel` 4, `stargazer` 4, `rdd` 4, `rddtools` 2, `ivpack` 1), so one iteration closed the gap. The v1 documentation's package list therefore understates the environment in which the v1 panel was actually scored.

Results. Table 12 gives the counts behind the main-text rates. The Llama row is not a new run: it is read directly from the frozen v1 score table, where Llama was scored in the v1 environment, and it was not re-executed in the container. The conditional rates rest on small executing denominators (35, 59, and 41 workflows), and the intervals in the last column are correspondingly wide; they support the main text's qualitative contrast (Qwen3 and DeepSeek-V4 lose most workflows before execution, Llama loses about half of its executing workflows after it) and no finer comparison.

Table 12: Open-weight runs on the 100-scenario executable arm. "Scen." is the number of scenarios; "Wrong" is the executed-but-wrong count (EXECUTES passed, CORRECT failed); $P(\text{wrong} \mid \text{exec.})$ divides it by the EXECUTES count. Brackets are Wilson 95% intervals. Qwen3 and DeepSeek-V4 were run after the v1 release was frozen (corrected environment). †Robustness only: the Llama row is read from the frozen v1 score table and is never part of the model ranking. Among the two ranked runs, the better value in each outcome column is in blue and the other in purple (higher is better for EXECUTES and CORRECT, lower for the two error columns); the robustness row is not coloured.

Model	**Scen.**	EXECUTES	CORRECT	**Wrong**	$P(\text{wrong} \mid \text{exec.})$
POST-FREEZE RUNS (CORRECTED ENVIRONMENT)					
Qwen3-235B-A22B-Instruct-2507	100	35	34	1	2.9% [0.5, 14.5]
DeepSeek-V4	100	59	54	5	8.5% [3.7, 18.4]
ROBUSTNESS ONLY (FROZEN V1 SCORE TABLE)					
Llama-3.3-70B-Instruct†	100	41	20	21	51.2% [36.5, 65.7]

By design family. Table 13 splits the same counts by design family (30 DID, 24 event-study, 24 IV, and 22 RDD scenarios). Cells are small and are reported for completeness, as the protocol required, not as a family-level comparison. Two patterns are visible without over-reading them: Qwen3's execution failures are concentrated in RDD (1 of 22 executes), and for both new arms the executed-but-wrong cases fall mainly in the event-study family (1 of 1 for Qwen3, 3 of 5 for DeepSeek-V4), whereas Llama's are spread across all four families.

G REPAIR PILOT

This appendix gives the design and full counts of the one-round repair pilot summarized in section I. It is a pilot (three models, one repair round, one fixed feedback sentence per failure type, several thin cells) run after the v1 release was frozen; it changes no v1 number.

Table 13: Open-weight runs by design family. Each cell is EXECUTES / CORRECT out of the n scenarios in that family; the difference between the two is the executed-but-wrong count. Highest CORRECT count per column in blue; the robustness-only Llama row is excluded from the comparison.

Model	**DID** (n=30)	**ES** (n=24)	**IV** (n=24)	**RDD** (n=22)
Qwen3	10 / 10	16 / 15	8 / 8	1 / 1
DeepSeek-V4	16 / 16	16 / 13	18 / 17	9 / 8
Llama (robustness only)	10 / 3	16 / 9	8 / 4	7 / 4

Sampling frame. The frame is every workflow in the frozen v1 score table that fails CORRECT for three models fixed in advance as a high, a middle, and a low ability band of the v1 panel (GPT-5, o3, and Kimi): 172 failed workflows in all (28, 54, and 90). It is stratified by design family (DID, event study, IV, RDD) and failure type: *non-executing* (EXECUTES failed) or *executed-but-wrong* (EXECUTES passed, CORRECT failed). Within each model $\times$ family $\times$ type cell all failures are taken up to a cap of 10; the five larger cells are subsampled with a fixed seed. The realised sample, frozen before any repair call, is 136 workflows: all 28 GPT-5 failures, 42 of o3's, and 66 of Kimi's; 100 non-executing and 36 executed-but-wrong. Every sampled workflow contains an extracted R code block, so "non-executing" uniformly means code that was present and failed at run time. Because of the cap, the rates below describe this stratified sample, not the population of all 172 failures.

The single repair round. Each case is one conversation: the original v1 prompt for that scenario, byte-identical; the model's original cached response, byte-identical; and then exactly one fixed sentence chosen by failure type. For a non-executing workflow the model is told, verbatim, "The submitted workflow failed execution. You may revise it once." For an executed-but-wrong workflow it is told "The workflow executed, but the extracted causal estimate did not satisfy the benchmark's numerical correctness criterion. You may revise it once." Nothing else is added: no error log, no canonical estimate, no tolerance, no sign or size of the error, no reference code, and no description of the scoring mechanism. Each case receives one repair response; only transport-level errors are retried (at most twice, with the identical request), and a case that still returns nothing is recorded as no response and counted as unrepaired.

Scoring rule. A workflow counts as *repaired* if and only if its revised response passes CORRECT under the unmodified v1 scoring chain: the revised R code is extracted and executed (60-second limit), the estimate is extracted by the same extraction model, and it must fall within the default band of eq. (2) ($\tau = 0.50$) around the canonical estimate, with the v1 event-study window rule applied to event-study cells. The primary endpoint is the share of originally failing workflows that are repaired, with Wilson 95% intervals. The protocol fixed four secondary cuts (model, failure type, design family, mechanical versus conceptual repair) and the reading of either a high or a low repair rate before any repair call (Olken, 2015).

Model by failure type, and incidents. Table 15 crosses the two cuts the main text reads together. The pooled asymmetry (22.2% against 33.0%) is not uniform across models: GPT-5 and o3 repair non-executing workflows more often than executed-but-wrong ones (o3 repairs none of its four), while Kimi shows the reverse on a larger cell. GPT-5 and o3 each have only four executed-but-wrong failures in the whole v1 panel, and the two pooled intervals in table 14 overlap, so the asymmetry is a pilot-level observation. Three execution facts are disclosed. (i) 132 of 136 repair calls returned a response, each containing R code; the other four (all o3, all non-executing: one RDD and three IV scenarios) failed at the transport level after both permitted retries and count as unrepaired, as the protocol required. (ii) The availability check that precedes the repair calls first misclassified o3 as unavailable, because its small output budget was consumed by the model's internal reasoning; the budget was raised, all three models passed, and the o3 cases were run later under the unchanged protocol. The protocol's literal rule cancels an arm that fails this check, so this is a recorded deviation. (iii) Unlike the open-weight protocol of section F, the freeze of this protocol and sample carries no independent timestamp; it is self-attested.

Mechanical versus conceptual repairs. Each of the 41 successful repairs is classified by a fixed syntactic rule applied to the original and the revised R code. A pattern set detects which

Table 14: Repair pilot: share of originally failing v1 workflows that pass CORRECT after one repair round, pooled and by the prespecified cuts. Brackets are Wilson 95% intervals. No-response cases count as unrepaired. Top three rates among the cut rows in blue, purple, orange; the bold pooled row is the headline and is not ranked.

Cut	Group	n	Repaired	Rate [95% CI]
Pooled	**all sampled failures**	**136**	**41**	**30.1% [23.1, 38.3]**
Model	GPT-5 (high)	28	18	64.3% [45.8, 79.3]
	o3 (middle)	42	15	35.7% [23.0, 50.8]
	Kimi (low)	66	8	12.1% [6.3, 22.1]
Failure type	non-executing	100	33	33.0% [24.6, 42.7]
	executed-but-wrong	36	8	22.2% [11.7, 38.1]
Design family	DID	38	10	26.3% [15.0, 42.0]
	event study	33	13	39.4% [24.7, 56.3]
	IV	28	7	25.0% [12.7, 43.4]
	RDD	37	11	29.7% [17.5, 45.8]

Table 15: Repair pilot: repaired / n by model and original failure type. Cells with $n = 4$ are the complete set of such failures for that model in the v1 panel. Largest fraction per column in bold.

Original failure type	GPT-5	o3	Kimi
Non-executing	**16 / 24 (66.7%)**	**15 / 38 (39.5%)**	2 / 38 (5.3%)
Executed-but-wrong	2 / 4 (50.0%)	0 / 4 (0.0%)	**6 / 28 (21.4%)**

estimator-call families appear in each script: fixed-effects panel calls (`feols`, `felm`, `plm`), plain `lm`, instrumental-variable calls (`ivreg`, `iv_robust`, `tsls`), regression-discontinuity calls (`rdrobust`, `RDestimate`) (Calonico et al., 2014), and staggered-DID package calls (`att_gt`) (Callaway & Sant'Anna, 2021). A repair is *conceptual* if the set of detected families changes between the two scripts and *mechanical* if it does not, which covers syntax, library, variable-name, and output-formatting fixes. The rule labels 32 repairs mechanical and 9 conceptual. The protocol allowed manual adjudication of ambiguous cases; the frozen labels coincide with the pattern rule in all 41 cases, so the split reported here is the rule's output. It is a coarse proxy in both directions: a changed call family need not be a changed identification strategy (in 8 of the 9 conceptual cases the only family detected in the revised script is plain `lm`), and an unchanged family does not rule out a changed specification. The split indicates, as the main text uses it, that one round of knowing-you-failed mostly fixes broken code; it does not measure how often models revise their identification reasoning.

H Prospective Replication: Six New Pairs

Construction and runs. The six pairs were built after the 23-pair panel had been analysed, under a construction protocol frozen in advance: a gold design frozen before the twin is built, a twin whose sealed one-shot acceptance run is never redrawn, and two independent re-implementations of the canonical estimator that must agree on all 20 seeds. Runs used the panel's contract unchanged (single-shot, no retry, up to three calls per cell, 8,192-token cap, same sandbox and scorer). Batch 1 (four pairs, nine models, 36 cells) ran in August 2026. Its oracle arm showed cross-vendor *identical* off-canonical estimates on two pairs, the signature of a convention the oracle view left free, and was re-run twice after pins restating frozen artifacts were added (a units pin for all four pairs, then an instrument-realisation pin and a correction-formula pin for the two affected pairs). Those pins are post hoc, so table 16, fig. 8 reports the oracle arm as first run, after the units pin, and under the frozen splice. Batch 2 (two pairs, 16 cells, 46 calls, September 2026) fixed every such convention in the oracle view *before* any call, from the same frozen bindings the independent re-implementers had received; its primary summary and reporting commitment were committed before the first call. One panel endpoint (`moonshot-v1-128k`) had been retired by its provider and returns 404; its two cells are not measured and no successor was substituted.

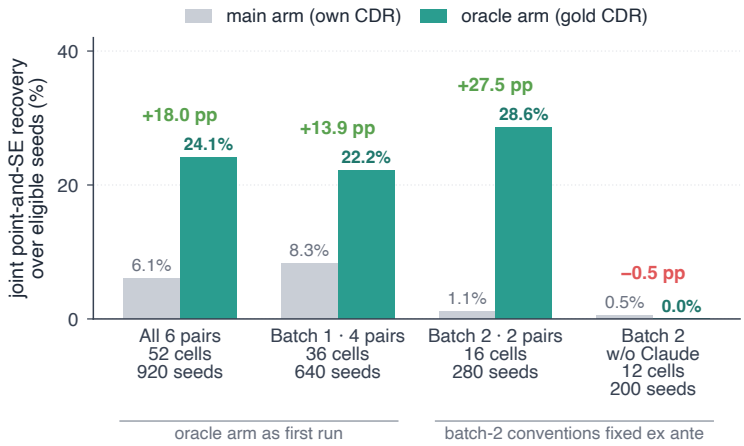


Figure 8: Prospective replication on six pairs built after the claim under a protocol frozen in advance (table 16): joint point-and-SE recovery over eligible seeds, main arm (grey) vs. oracle arm (teal), for all six pairs, batch 1 (four pairs, oracle arm as first run), batch 2 (two pairs, conventions fixed before any call) and batch 2 without the two Claude-family models, whose 20/20 oracle seeds on both pairs are the entire batch-2 gain.

Table 16: Joint point-and-SE recovery over eligible seeds (20× schema-valid commitments) on the six new pairs. "Without Claude" drops the two Claude models, whose family also built the references. Top three per recovery column among the batch rows in blue, purple, orange (ties share a tier); the pooled headline row is bold and uncoloured.

			all models		without Claude	
	cells	eligible	main	oracle	main	oracle
Batch 1, oracle as first run	36	640	8.3%	22.2%	2.7%	12.5%
Batch 1, oracle after units pin	36	640	8.3%	25.0%		
Batch 1, oracle frozen splice	36	640	8.3%	39.8%	2.7%	20.8%
Batch 2, conventions fixed ex ante	16	280	1.1%	28.6%	0.5%	0.0%
Six pairs, oracle as first run	**52**	**920**	**6.1%**	**24.1%**		

Batch 2 by cell. Of 16 cells, 14 produced a schema-valid commitment (Gemini and GPT-4o each failed Stage 1 on one pair). Among valid cells, 4 executed at least one main-arm seed and 6 at least one oracle seed. Main arm: Opus, Sonnet and o3 each recovered 1 of 20 seeds on one pair; nothing else recovered. Oracle arm: Opus and Sonnet recovered 20 of 20 seeds on both pairs; o3 and DeepSeek executed on one pair and recovered none; the remaining cells did not execute. Nine calls stopped at the token cap (GPT-5 and Gemini four each, o3 one), and Qwen3 again imported a package the execution contract forbids.

I Boundary Experiments

I.1 The failure mode is not R-specific

A matched Python-backend subset (30 scenarios; Opus, GPT-5, o3; run before the v1 release was frozen and not included in it) reruns the same tasks with a Python execution environment. The executed-but-wrong rate under Python is 14/58 (24.1%), against a matched-pair R baseline of 1/60 (1.7%) and the primary 100-scenario DGP panel's 15.5%; the pairwise 2×2 over 88 matched pairs is 29 both-pass, 15 Python-only, 30 R-only, 14 both-fail. The subset supports exactly one claim: the executed-but-wrong failure mode is present in both backends; it does not establish backend invariance, and with three models the cross-backend ordering carries no statistical meaning, so no ranking comparison is drawn from it.

I.2 One repair round is not enough

A protocol-frozen repair pilot (specification fixed before scoring) takes 136 failed v1 workflows and grants one oracle-flagged repair round: the model is told its output failed and may resubmit once. Pooled recovery is 41/136 (30.1%, Wilson 95% CI [23.1, 38.3]) and is sharply ability-graded: GPT-5 64.3%, o3 35.7%, Kimi 12.1%. Directionally, executed-but-wrong workflows are repaired *less* often than non-executing ones (22.2% versus 33.0%; the intervals overlap and one model shows the reverse, section G; o3 repaired 0 of 4 executed-but-wrong cases), and among the 41 successful repairs, 32 are mechanical fixes against 9 conceptual ones. A single round of knowing-you-failed is largely a mechanism for fixing broken code, not for fixing a wrong analysis. That is why the paired protocol's repair rule stays typed and why identification-grade feedback is left to the companion line rather than claimed here.

I.3 Controlled repair on paired-task failures

A second repair experiment, prespecified and committed before any call, takes the *census* of failing main-arm trajectories of the 23-pair panel (162: 108 that executed no seed, 54 that executed but did not recover on all seeds) and grants each exactly one stateless repair call with typed feedback only: the runtime error of the first failing seed, or the statement that the reported point estimate and standard error fell outside the acceptance bands on M of E executed seeds, with nothing about magnitude, sign or the canonical value. Endpoints, system prompt, 8,192-token cap, sandbox, seeds and scorer are those of the panel. Of the 162, 20 were not run because their endpoint (`moonshot-v1-128k`) has been retired by the provider, and one call never returned and is recorded as an infrastructure void; 141 are scored (141 paid calls).

After repair, joint point-and-SE recovery over eligible seeds is 14.3% (pair-clustered 95% CI $[8.1, 21.5]$): 16.1% among trajectories that had not executed and 11.1% among those that had executed without recovering. The prespecified directional hypothesis holds: at least one seed is recovered for 22.9% of previously non-executing trajectories against 7.1% of executed-none-recovered ones ($+15.7$pp, CI $[2.6, 29.3]$). Typed feedback repairs failures to *run* more often than failures to *recover*, as in the executable arm's pilot, now with an interval that excludes zero. Execution itself is restored for 36 of 70 non-executing trajectories but only 4 of the 20 whose original call had stopped at the token cap; 23 repair calls stopped at the cap again. Of the 20 trajectories that went from no recovered seed to at least one, 7 declared in the required header that the revised script deviates from the committed design and 2 carried no parseable declaration; this is self-report, and we make no claim about undeclared switching. One partly recovered trajectory lost recovered seeds.

J Text Proxies and Calibration

Text-agreement is label-noise-bound. On the real-paper arm, L3 method-family agreement is scoreable for 187 of the 259 real-paper contexts (a count unrelated to the paired panel's 187 scored commitments) and L4 direction agreement for 92 of 259; the unscoreable remainder is dominated by papers where the four-LLM consensus never converged on a single label. Model rankings do not collapse to one textual ability score: GPT-4o leads L3 (88.8%) while Sonnet leads L4 (85.9%), and fig. 10 shows the family breakdown, with Event Study lowest (64%, n=13; exploratory at that corpus size). What bounds the interpretation is the blinded 30-paper human audit, stratified by the strength of the consensus being audited: agreement with the consensus is 100% where at least three of the four labelers agree, 56% under a bare plurality, and undefined where the consensus produced no label (table 17; method $\kappa = 0.61$, direction $\kappa = 0.29$). Where the reference is solid, humans and the consensus coincide; where it is weak, they often do not, so cross-model variation in L4 partly measures reference-label noise rather than model behaviour. Exp A accordingly enters this paper as a paper-context design-recognition diagnostic under source-derived contexts, and no execution-level claim rests on it (section 8).

Self-reported confidence does not triage. Calibration asks each model to rate its own prior Exp B output without seeing the pass/fail label (Kadavath et al., 2022; Lin et al., 2022; Tian et al., 2023; Xiong et al., 2024). The elicitation asks for confidence that the estimate lies within 20% of the true effect, whereas correctness is scored at 50% of the canonical estimate, so the two are not

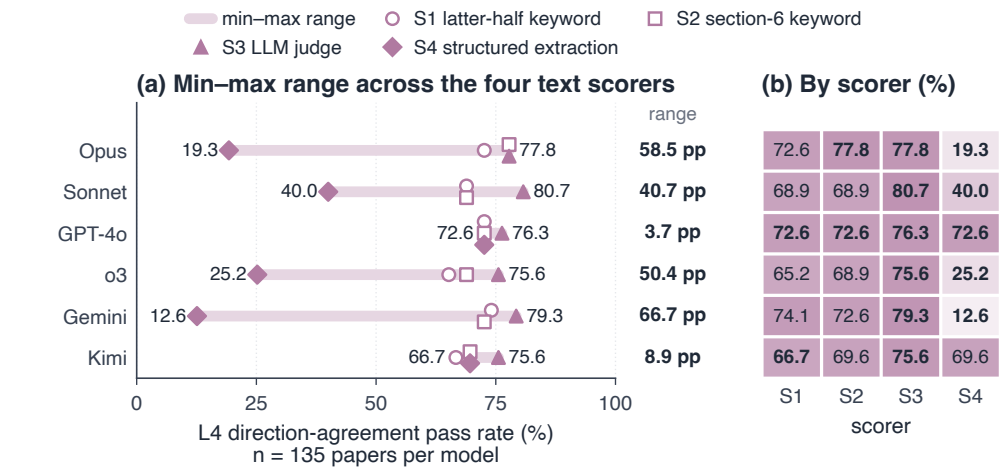


Figure 9: L4 scorer instability across frozen scorer variants (ported from the v1 release). Direction-agreement scores move materially under equally-defensible scorer variants, supporting the label-noise bound on text-layer diagnostics.

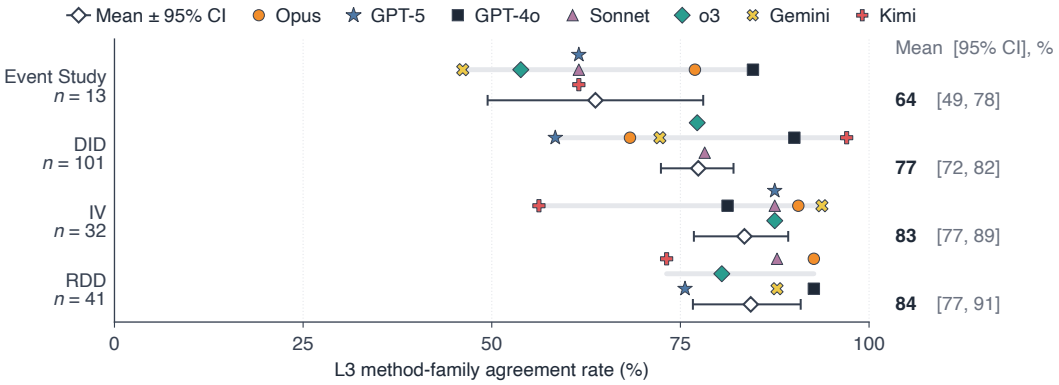


Figure 10: Exp A L3 method-family agreement by family. Markers show the seven model-level agreement rates; diamonds and whiskers the cross-model mean with paper-cluster bootstrap 95% intervals. Event Study has the lowest observed mean (64%, n=13) and is exploratory at that corpus size. L3 is a text-agreement diagnostic, not an executable correctness endpoint.

the same event. Actual correctness varies sharply across the panel (10.0% to 88.0% L2b$^+$), yet the confidence gap (mean confidence on correct minus incorrect outputs) stays within ± 0.075 for every main-panel model; the one larger gap (Gemini, $+0.234$) sits on the smallest usable sample (n=48). The asymmetry is the point: GPT-5 is second on execution-grounded correctness with a slightly *negative* gap, so ability and self-assessment dissociate in both directions (fig. 11). Under this retrospective protocol, self-reported confidence is not a usable deployment triage signal; the result does not rule out stronger calibration interfaces, and is evidence against assuming one exists by default.

Cross-vendor robustness. A single open-weights check (Llama-3.3-70B-Instruct, same 100 scenarios, same pipeline) reproduces the runnable-versus-correct gap: L2b $= 41\%$ against final L2b$^+$ $= 20\%$, placing it between Kimi and Gemini. Adding it as an eighth model lowers the cross-layer rank correlation to Kendall $\tau = 0.714$ without changing any qualitative ordering. This is a robustness check, not a benchmark of the open-weights ecosystem.

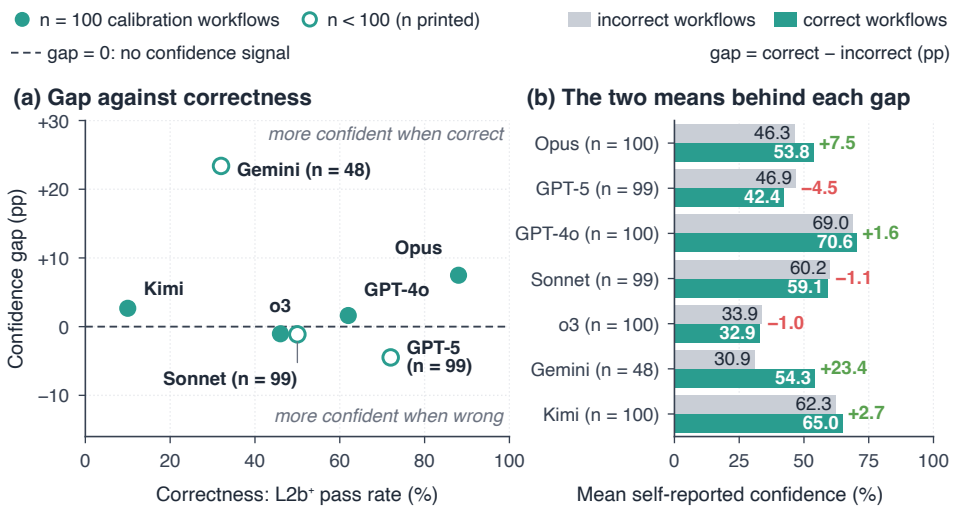


Figure 11: Correctness versus self-assessment. **(a)** Each point is one model: horizontal, execution-grounded L2b$^+$ correctness; vertical, the mean confidence gap between correct and incorrect workflows. **(b)** The two means behind each gap: mean self-reported confidence on incorrect (grey) and correct (teal) workflows, with the signed gap. High correctness does not imply a usable confidence signal.

K Human Annotation and Coefficient-Extraction Audits

Two blinded human audits bound the two instruments that are not executable by construction: the consensus reference labels of the real-paper arm (section 4) and the coefficient-extraction judge of the executable arm (section 4). Both were run after the automated benchmark freeze, and neither changes any frozen label, prompt, output, threshold, or headline rate.

K.1 Real-paper reference labels: 30-paper ambiguity audit

Protocol. The release contains a paper-native labeling protocol and a completed, blinded 30-paper slice of the real-paper arm. A single annotator labeled method family and effect direction from the original PDFs without access to the LLM votes, the consensus labels, or any model output; the consensus labels are the references against which METHOD-MATCH (legacy code L3) and DIRECTION-MATCH (legacy code L4) are scored. Direction labels rest on three of the four voters, since one voter casts no direction vote. One target paper was excluded because the available PDF was appendix-only, and a blinded replacement was drawn from the unsealed sample manifest, which contains only paper ID, difficulty tier, and PDF path. The human labels are an ambiguity check, not a training, prompting, or tuning input: they are not used to change the frozen consensus labels, construct prompts, select model outputs, or tune scoring thresholds. With one annotator and no relabel pass, every statistic below is agreement between human and consensus, not inter-annotator reliability. The slice is a stress test rather than a representative sample: 21 of its 30 papers (70.0%) lie in the low-consensus strata of the four-LLM vote (2-of-4 plurality, 2-of-4 tie, split), against 116 of 259 (44.8%) corpus-wide.

Agreement and denominators. Method-family labels match on 18 of the 30 papers (60.0%). Five of the 30 (three 2-of-4 ties, two splits) have no consensus method label and count as non-matches in that figure; on the 25 papers where both labels exist agreement is 18/25 (72.0%), and Cohen's $\kappa = 0.606$ is computed on those 25. Nine papers have no consensus direction label, leaving 21: direction agreement is 10/21 (47.6%) with $\kappa = 0.294$, where "unclear" is treated as a label (in 9 of the 11 non-matches the consensus says "unclear" and the human label does not). The 60.0% and 72.0% figures are thus the same 18 matches over two denominators. Table 17 gives the breakdown by consensus strength: agreement is complete where at least three of the four labelers agree and 56.2% under a bare plurality, while the tie and split rows are zero by construction, because there is no consensus label to match. The audit supports reading the real-paper arm as an agreement diagnostic with substantial ambiguity; it is not a full human re-adjudication of the 259-paper corpus.

Table 17: Method-family agreement between the blinded human labels and the four-LLM consensus, by strength of the consensus vote. Light-gray rows group the strata by whether a consensus method label exists; the bold row is the headline figure, and agreement values are left uncoloured because they are not a ranking.

Consensus stratum	Consensus label	Papers	Matches	Agreement
CONSENSUS LABEL EXISTS				
4 of 4 agree	yes	4	4	100.0%
3 of 4 agree	yes	5	5	100.0%
2 of 4, plurality	yes	16	9	56.2%
NO CONSENSUS LABEL (ZERO BY CONSTRUCTION)				
2 of 4, tie	none	3	0	0.0%
Split	none	2	0	0.0%
All audited papers		**30**	**18**	**60.0%**
Papers with a consensus label		25	18	72.0%

K.2 COEFFICIENT-EXTRACTION JUDGE: 50-CELL AUDIT

Protocol. The coefficient-extraction judge is a measurement instrument: it identifies the scalar treatment-effect coefficient from executed R code and its stdout before CORRECT (legacy code L2b$^+$) compares that coefficient to the canonical estimate. The annotation form of the blinded audit hides model identity, the CORRECT label, the canonical estimate, the judge's and the legacy regex extractor's effects, and all ranking information. The completed audit covers 50 v1-panel cells that pass EXECUTES (legacy code L2b), sampled with seed 20260502: DID 13, Event Study 12, IV 11, RDD 14. Sampling prioritizes cells whose pass/fail label changed between successive scorer generations (35 of the 50), so hard extraction cases are over-represented.

Results and denominators. All 50 cells were annotated. In 6 of them (DID 1, IV 1, RDD 4) the annotator found no treatment-effect coefficient in the output, so there is no human value to compare; in five of these the judge also extracted nothing, and in one RDD cell it extracted a value that passes. Both agreement rates are therefore out of the 44 comparable cells, not out of 50. Numeric agreement between judge and human is 40/44 (90.9%), where agreement means an absolute difference of at most 10^{-4} or a relative difference of at most 1%; agreement on the induced CORRECT pass/fail decision, applying the default 50% tolerance to the human-extracted value, is 39/44 (88.6%). The annotator flagged 15 of the 50 cells (30.0%) as having more than one plausible target coefficient. The audit summary flags 12 cells in all (DID 1, Event Study 6, IV 1, RDD 4): the six Event Study cells carry every disagreement among the 44 comparable cells (all four numeric and all five pass/fail disagreements), and the other six are the non-comparable cells. Most flags thus arise from two sources: (i) Event Study outputs that report a cumulative post-event window rather than the canonical per-day scale, and (ii) RDD outputs where printed signs, jump orientation, or robustness estimates make the main effect ambiguous. The audit supports the extraction instrument and does not change the frozen headline CORRECT rates; the flagged cases are why CORRECT is described as an audited measurement instrument rather than an infallible oracle.

L EXTENDED RELATED WORK

Prior work evaluates important pieces of the LLM causal-analysis workflow, but usually not the executed causal estimate (Yang et al., 2024; Ma, 2025). Causal benchmarks test textual reasoning about direction, counterfactuals, graphs, interventions, or identification (Jin et al., 2024; 2023; Chen et al., 2024; Chi et al., 2024; Kiciman et al., 2024). Code benchmarks test whether generated programs execute or pass functional tests (Chen et al., 2021; Hendrycks et al., 2021; Liu et al., 2023; Zhuo et al., 2025; Jain et al., 2025). Science-agent benchmarks evaluate broader discovery or tool-use workflows. CAUSALVERIFY differs in its verification target: the primary correctness signal is not a text label or a generic program test, but whether executed model-written analysis code recovers a benchmark-defined target estimate on the realised dataset.

Among end-to-end causal benchmarks, CauSciBench (Acharya et al., 2025) evaluates causal-analysis performance over curated real, synthetic, and textbook tasks, graded on the final product. CAUSALVERIFY asks a complementary measurement question: when a model appears to understand a causal task, *which layer of the workflow is externally verified*? It separates design-recognition diagnostics, controlled-DGP execution verification, and retrospective calibration rather than collapsing them into one causal-ability score. Its paired protocol adds an instrument no adjacent benchmark carries: a design committed and hashed before any data are revealed, then executed, so design-layer and execution-layer failures separate on the same instance rather than across task pools.

Concurrent work by Sawarni et al. (2026) disentangles causal identification from estimation on real-world datasets; its finding that strategy identification is easier than full specification is consistent with our name-versus-compute gap, and our six-pair pilot sharpens the same contrast to the single trajectory: a correctly named family whose committed specification diverges and whose faithful execution is therefore wrong on every seed. The methodological difference is the unit of verification: CAUSALVERIFY verifies generated and executed code on controlled DGP data, extracts the reported coefficient, and checks it against a realised-data canonical estimate under frozen bands. This isolates the failure mode where code runs but computes the wrong causal quantity. Recent workflow-oriented evaluations such as BioMysteryBench (Anthropic, 2026) place models in rich environments with real data, tool access, and objective final answers. CAUSALVERIFY is complementary: it fixes a canonical estimate on the realised data so that model-written causal code can be audited layer by layer rather than graded only on a final answer.

Three adjacent lines of work appeared around and after the v1 release. CausalDS (Leban & Sun, 2026) evaluates data-science agents across Pearl's three rungs (Pearl, 2009) on synthetically generated structural causal models, with ground truth derived from the known structures; CAUSALVERIFY occupies the narrower estimation slice at higher resolution: paper-grounded twins, canonical estimators under frozen tolerance bands, and a commitment-before-data protocol that separates the design layer on the same instance. InterveneBench (Shi et al., 2026) scales design-layer evaluation to 744 real policy studies, asking whether models can construct an appropriate causal study design from open-ended intervention contexts. Its STRIDES system uses simulated data and code execution internally to refine proposed designs, but the benchmark endpoint remains agreement between the predicted and expert-verified study design rather than numerical recovery of an externally fixed causal estimate. CAUSALVERIFY addresses the complementary downstream verification problem: once a model commits to a design, did it implement that design faithfully, and did the execution recover the benchmark-defined estimand? The two evaluations are end-to-end over different chains (study-design generation there, verification here) with different terminal endpoints. Closest in verification style, Kohler et al. (2026) reproduce social-science results from paper text and original data with deterministic cell-level comparison to the published outputs; the criterion there is agreement with the original authors, whereas CAUSALVERIFY's canonical estimates are benchmark-fixed on synthetic twins, so recovering the target is separable from matching a possibly imperfect original. Recitation of published values becomes a threat its referee zones must control rather than the success criterion. The oracle arm is likewise not a reproduction task: it substitutes the gold design against the model's *own prior commitment* on the same instance, with a twin-fixed target. ReplicatorBench (Nguyen et al., 2026) grades agents across the stages of a replication pipeline against human-verified replicable and non-replicable claims. It evaluates competence *at* replication; CAUSALVERIFY fixes the design layer via committed CDRs and verifies the executed estimate itself. Table 18 summarizes the affordances.

Table 18: Comparison with adjacent benchmarks. The table reports evaluation affordances rather than overall benchmark quality: ✓ marks an explicit mechanism for that dimension, not superiority; ∼ = partial; ✗ = absent. †EconCausal counts 10,490 triplets across 2,595 studies.

Benchmark	N	Target	Real ctx.	Exec. coeff.	Layered Dx	Self-assess.
CLadder / CORR2CAUSE (Jin et al., 2023; 2024)	11k / 400	Causal QA / direction	∼	✗	✗	✗
CausalBN-Bench (Zhou et al., 2026)	3	Graph + effect	✗	✗	∼	✗
EconCausal (Lee et al., 2025)	10,490†	Economic causal sign	✓	✗	✗	✗
CauSciBench (Acharya et al., 2025)	367	End-to-end CI pipeline	✓	∼	∼	✗
InterveneBench (Shi et al., 2026)	744	Intervention design	✓	∼	∼	✗
CausalReasoningBench (Sawarni et al., 2026)	173	ID + estimate	✓	✗	∼	✗
Code benchmarks (Chen et al., 2021; Jimenez et al., 2024)	–	Program tests	✗	∼	∼	✗
Discovery / science agents (Majumder et al., 2025; Chen et al., 2025)	264 / 102	Discovery workflow	✓	∼	∼	✗
CAUSALVERIFY	259 + 100 + 23	Causal workflow	✓	✓	✓	✓

M BENCHMARK CONTROLS

Table 19: Benchmark controls. Each component supports a different claim and protects against a different failure mode.

Component	Implementation	Claim supported
Real context	259 papers; reconstructed RQ/DD/IC; consensus reference labels	Text-level design-recognition diagnostics
Executable truth	100 fixed-seed DGPs and 23 paper-grounded twins; realised data; canonical estimators	Numerical coefficient correctness
CDR commitment	Stage-1 design hashed before any data are revealed; the main arm executes exactly the committed design	Same-instance separation of design from execution; no post-hoc design revision
Role separation	Separate stages for construction, response, labeling, extraction, calibration	Reduces circular grading
Leakage control	Method-name blacklist; prompt-surface leakage gates; opaque task identifiers; sealed truths	Prevents trivial label recovery
Scorer audit	Coefficient-extraction judge plus blinded human validation	Reduces brittle post-processing errors
Contract freeze	Hashed twin-set freeze; dated amendments that state their triggers, including when they are not result-blind	Distinguishes pre-specified structure from post-hoc repair
Claim boundaries	Exp A = recognition diagnostics; Exp B = execution; paired = same-instance cross-layer contrast; calibration = confidence alignment	Prevents overclaiming

N LIMITATIONS: EXTENDED DISCUSSION

Scope of the causal task. The benchmark evaluates structured econometric estimation workflows after the research question, the data and the institutional setting have been fixed. It does not evaluate problem formulation, data acquisition, design selection from ambiguous field constraints, or the broader causal-inference lifecycle in the sense of Hernán (2019); a model could excel here and still fail at the stages this benchmark deliberately holds constant.

Real-paper labels are diagnostics, not ground truth. Exp A's labels come from a four-LLM consensus whose pool structurally overlaps the evaluated panel. The blinded 30-paper ambiguity audit agrees with that consensus on 60.0% of method labels and 47.6% of direction labels (direction $\kappa = 0.294$). Exp A therefore supports design-recognition diagnostics under source-derived paper contexts; it is not evidence about unaided design inference from neutral institutional facts, and no execution-level claim rests on it. A separate construct audit of the legacy Exp A contexts remains open and is tracked in the released audit records.

Synthetic structure and backend. Executable reference estimates require synthetic twins; the DGPs cover four design families, not the space of applied practice. The v1 corpus fixes R as the execution backend while the paired arm executes Python, so cross-arm comparisons carry a backend difference as well as a protocol difference; v1's Python-backend replication shows the executed-but-wrong phenomenon is not R-specific, but magnitudes should not be read as backend-general.

Commitment before data. Stage 1 commitments close before the data dictionary is visible, and the single-shot contract offers no renegotiation. A commitment can therefore be faithful to the source literature and still diverge from the benchmark gold and fail to execute as committed against the twin's shipped geometry (section E.1). The protocol records this as a main-arm outcome by design; the one-round repair pilot recovers non-executing workflows more often than executed-but-wrong ones (33.0% versus 22.2%).

Panel scale and model coverage. The paired evidence now spans the full 23-pair, nine-model panel (section 6.1); the six-pair pilot (section E.1) remains as harness validation and taxonomy. Conditional downstream rates are reported with pair-clustered uncertainty, and the recovery-difference interval includes zero. The evaluated panels, v1 and panel, consist mainly of closed or hosted systems; rank statements are descriptive over the evaluated panel, not population claims over all models.

The instrument is audited, not infallible. Scoring depends on canonical specifications, frozen tolerances and coefficient extraction; individual-cell mistakes remain part of measurement uncertainty. Changes made after evidence existed are recorded as dated amendments that state their trigger, including two that are explicitly not result-blind (a unit-contract clarification and the repair of one task's Stage-1 context after the pilot), alongside a design-profile instrument that was itself built after pilot outcomes were known, so a reader can distinguish pre-specified structure from post-hoc repair rather than being asked to assume there was none.

Contamination and decay. The paired release re-keys every model-facing task to opaque identifiers; method family and difficulty appear in no filename, identifier or model-visible record, ground truth lives scorer-side, and per-twin truths remain sealed. Because source papers are published, familiarity from pretraining cannot be fully excluded on the real-paper side. Any released benchmark decays as later models train on its artifacts (Kiela et al., 2021; Jain et al., 2025); released results are versioned against hashed freezes, and sealed holdout material supports future re-keyed refreshes.

O Additional Panel and Legacy Diagnostics

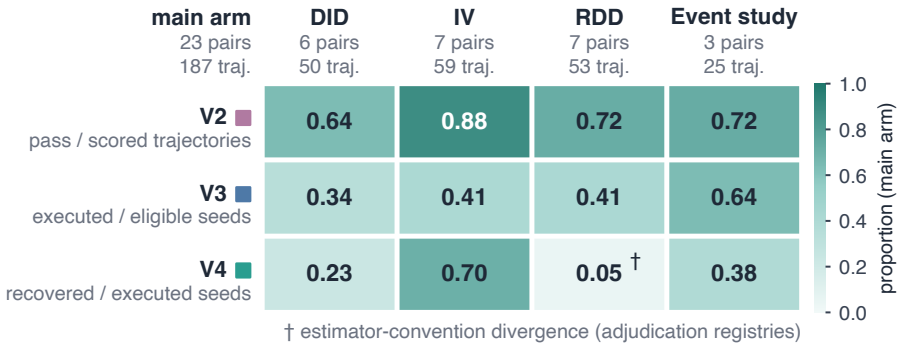


Figure 12: Panel profile by causal-design family (main arm, descriptive). Each layer keeps its own denominator; pairs per family are few, so no family ranking is implied. The RDD cell's low recovery is dominated by estimator-convention divergence documented in the adjudication registries.

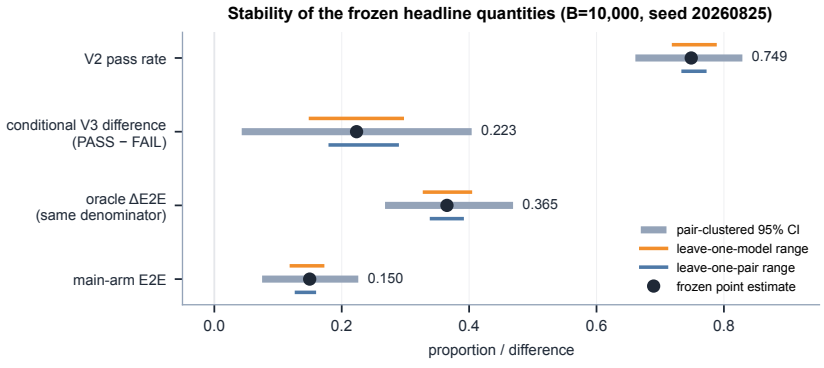


Figure 13: Stability of the frozen headline quantities: pair-clustered 95% bootstrap CIs with leave-one-model and leave-one-pair ranges. The substitution headline moves by at most ± 3.7pp under any leave-one-out; the other quantities move by up to 7.9pp.

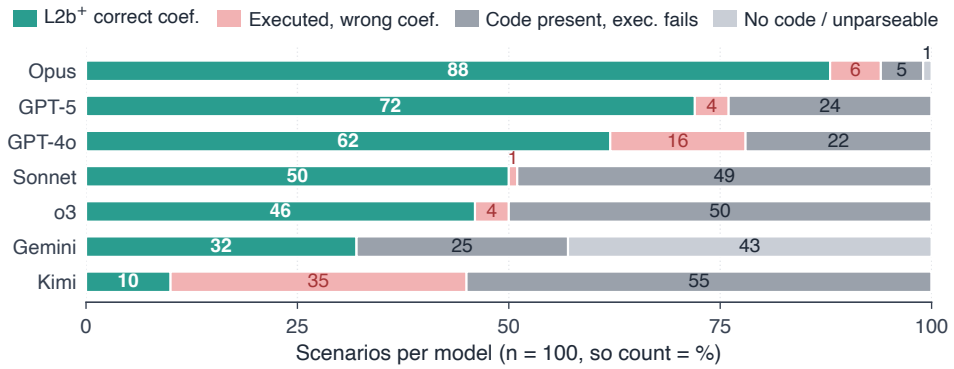


Figure 14: Legacy execution cascade from the v1 100-scenario DGP panel (demoted from the v1 main text): per-model composition of L2b$^+$ outcomes. Retained for continuity with the v1 release; the paired panel of section 6.1 supersedes it as the paper's core evidence.

$\geq$1-seed end-to-end figure (descriptive). 31 of 187 scored trajectories (15.0% of the 207 attempts) achieve Stage-1 validity, a V2 pass, at least one executed seed and at least one jointly recovered seed. The threshold was not pre-frozen; the continuous joint-recovery proportion over eligible seeds in section 6.1 is the primary end-to-end metric.