# petri-bench: Scoring the Scientific Method of LLM Agents on Procedurally Generated Causal-Discovery Tasks

Baris Sozudogru

June 2026

## Abstract

Benchmarks for AI "scientist" agents predominantly test the rediscovery of published results, which rewards memorization: a model can succeed by recalling the answer rather than earning it. We present petri-bench, a benchmark that scores the *scientific method* of an LLM agent — hypothesis isolation, controlled experimentation, inferential statistics, and calibrated claims — on procedurally generated causal-discovery tasks whose ground truth is created at generation time and therefore could not have appeared in any training corpus before evaluation. Each task is a seeded "mystery world" built on one of five deterministic multi-agent simulations: exactly one hidden parameter (two, at the hardest tier) is changed from a revealed control configuration, and the agent must identify the change through a budgeted, blind experiment harness that returns inferential statistics only. Scoring is objective and reproducible — no LLM judge in the headline metric — and decomposes into correctness, method rigor (whether the submitted conclusion is backed by a significant, isolating controlled experiment), and efficiency. A separate process-integrity audit flags p-hacking (submissions that survive only through uncorrected multiple comparisons after redundant testing) and probe-only "output matching." Tasks carry statistically grounded difficulty ratings, and the simulation layer is validated against established results from four literatures (19/19 checks). In a standardized 540-episode evaluation — six frontier models (GPT, Gemini, GLM, and three Claude tiers), three episodes per task, all through an identical agent loop — a one-factor-at-a-time reference sweep with curated test values outscored every model by 29 points or more (Wilcoxon signed-rank over paired tasks, all $p < 2\text{x}10^{-5}$), solved 100% of tasks, while no model solved above 50% (best: 45/90 episodes). An ablation locates the gap: the *same* sweep with test values drawn blindly from each parameter's legal range scores 40.5-47.7 across three independent value draws (mean 45.2) — below every model's mean in every draw — so the reference's advantage decomposes into informative intervention design *plus* disciplined procedure, and the models sit between the blind and informed variants: their intervention choices carry real information, but their procedure gives the advantage back. The procedure failures are direct, log-verified, and value-independent: in 89 episodes a model identified the changed parameter and submitted the wrong effect direction — 61% of those never directly measured the determining contrast, and the rest measured it and contradicted their own significant evidence. The benchmark's two axes separate cleanly: the model with the highest solve rate posts the lowest score (chronic over-testing, probe-only shortcuts), while the smallest Claude tier converts the lowest solve rate into a near-top score through clean process. A process-integrity audit catches eleven violations (p-hacking, probe-only "output matching") that headline scores alone would hide. All episode artifacts are self-contained and auditable; tasks, code, and results are archived (DOI: 10.5281/zenodo.20618024).

## 1. Motivation

Two observations motivate this benchmark.

First, evaluation is now the bottleneck for AI-driven science. Agent systems write papers that pass workshop peer review (Yamada et al., 2025), and public platforms run research agents as a product category. The open question is no longer whether such agents can produce scientific artifacts but whether they can

*do science*: choose the right experiment, control what must be controlled, reason correctly from noisy data, and claim only what their evidence supports.

Second, the dominant evaluation paradigm cannot answer that question. Benchmarks built from published studies — data-driven discovery suites and rediscovery tasks alike — are contaminated by construction: the target findings, and frequently the exact datasets, are in the training corpora of the models being evaluated (Reddy & Shojaee, 2024). The field's own surveys warn that rediscovery benchmarks reward memorization over method. Purpose-built discovery environments avoid contamination but are text-game-like, single-domain, or fixed-set (Jansen et al., 2024; Koblischke et al., 2025), so they measure exploration in one world rather than statistical method across domains.

petri-bench takes a third path: procedural generation of causal-discovery tasks over deterministic simulations, with the ground truth created — and empirically verified — at generation time. Because the hidden change is a seeded random draw, the answer to any specific task instance cannot have existed in any training corpus before its generation (and fresh instances regenerate on demand once a set is published; Section 3.2). Because the simulations are deterministic given a seed, every result is exactly reproducible. And because the agent's full call log is recorded, the *process* can be scored, not just the answer.

## 2. Related Work and Positioning

| Benchmark | Task substrate | Contamination resistance | Multi-domain | Statistical-method scoring | Process audit |
|---|---|---|---|---|---|
| ScienceWorld (Wang et al., 2022) | text game, elementary science | fixed public set | no (one world) | no | no |
| DiscoveryBench (Majumder et al., 2024) | published datasets/findings (+ synthetic split) | targets are published | yes | partial (workflow match) | no |
| ScienceAgentBench (Chen et al., 2024) | published data-driven tasks | targets are published | yes | output-based | no |
| DiscoveryWorld (Jansen et al., 2024) | virtual discovery game | novel worlds, fixed public set | themed worlds | partial | no |
| Gravity-Bench (Koblischke et al., 2025) | gravitational dynamics | simulated, fixed public set | no (one domain) | partial | no |
| **petri-bench** | 5 deterministic simulations | **procedural; fresh instances on demand** | **yes (5 domains)** | **yes (isolation + inference)** | **yes (p-hacking, output-matching)** |

Column definitions: *contamination resistance* describes whether task answers could, in principle, be recovered from training data at evaluation time — any *fixed* public set, including ours once frozen and released, is

recoverable by later models; the operative distinction is whether fresh, verified instances can be generated on demand (Section 9). *Statistical-method scoring* asks whether the score depends on how conclusions were established (isolation, inference), not only on the final answer. Characterizations follow the cited versions of each benchmark; we welcome corrections from their authors.

Law-discovery suites in single physical domains (e.g., symbolic-regression-style benchmarks) share the contamination-resistance of simulation but score the recovered equation, not the experimental method. petri-bench's contribution is the combination: procedurally generated ground truth across qualitatively different domains, scored on whether the agent practiced sound statistical method, with an explicit audit for the characteristic failure modes of LLM experimenters.

## 3. Benchmark Design

### 3.1 Model organisms

Tasks are built over five deterministic, headless simulations ("model organisms"), each from a different complex-systems literature, exposed to agents through the published `petri-labs-mcp` tool server:

| Simulation | Domain | Target metric | Replicates |
|---|---|---|---|
| market | heterogeneous-agent asset market | volatility | 12 |
| swarm | flocking / collective motion | polarization order parameter | 12 |
| origin | evolutionary population dynamics | final population | 12 |
| morph | Gray-Scott reaction-diffusion | pattern blob count | 12 |
| social | bounded-confidence opinion dynamics | opinion cluster count (polarization at L3) | 12 |

Each simulation is deterministic per seed; paired per-replicate seeds make every reported statistic bit-reproducible. Section 6 validates that these engines reproduce established results from their respective literatures.

### 3.2 Task generation

A task is a seeded *mystery world*. From a revealed control configuration, the generator draws one hidden parameter change (the *driver*) from a per-simulation pool, plus two *decoy* candidates verified inert at generation time: the driver's effect on the target metric must test significant (Mann-Whitney U with Holm correction across the metric vector) and each decoy's must not. A task therefore ships with an empirically verified ground truth — `{parameter, hidden value, direction of effect}` — that did not exist before generation. The contamination claim is time-scoped: the evaluations in this report ran before the frozen sets and their answers were published, so no evaluated model could have trained on them; once released, the frozen sets are recoverable like any public benchmark, and the durable defense is procedural — fresh, verified instances regenerate from new seeds on demand, and a leaderboard evaluation can hold its seeds private (Section 9). Tasks are frozen as JSON specs; regenerating any frozen task from its seed reproduces it byte-for-byte (enforced in CI).

Three difficulty tiers:

- **L1** — identify *which* of the candidate parameters changed and the *direction* of its effect on the target metric.
- **L2** — additionally classify the effect *magnitude* (small / medium / large by relative change in the metric mean, boundaries at 10–35–75%). The hidden value is drawn from a continuous range, so the class must be measured, not guessed from a finite menu.
- **L3** — *two* parameters change together (four candidates: the interacting pair plus two decoys). The agent must identify both and the *sign of their interaction*, defined by the 2x2 factorial contrast [metric(AB) − metric(A) − metric(B) + metric(control)], where metric(A) denotes the cell with *only* A changed from control. Generation gates on a sign-flip permutation test of per-seed contrasts (5,000 permutations, deterministic), so only tasks with a real, detectable interaction ship. One-factor-at-a-time experiments alone structurally cannot answer L3 — the interaction sign requires the combined cell — so the reference solver extends its sweep into a full 2x2 factorial at this tier (Section 7.1).

The frozen sets are `core-v0` (10 L1 tasks: five simulations x two seeds), `l2-v0` (10), and `l3-v0` (10), plus the original six-task `pilot-v0`.

### 3.3 The blind experiment harness

Agents interact with a task only through four tools, under a hard budget of 8 calls:

- `experiment(configA, configB, metric)` — a replicated controlled A/B between two configurations of the agent's choosing. Returns inferential statistics only: group means, relative change, Mann-Whitney U with Holm-adjusted p, significance verdict, and Cliff's delta effect size. The resolved configurations are never echoed back. (The unpaired Mann-Whitney U is the deliberately conservative default; the paired per-replicate seeds primarily serve bit-reproducibility, and a paired-test variant is future work.)
- `probe(guess, metric)` — compares the agent's guessed configuration against the *hidden* world, returning the same statistics. A non-significant result means the guess is metric-indistinguishable from the truth. Probes support hypothesis refinement but, deliberately, are confoundable: a decoy can reproduce the hidden world's metric level without being the changed parameter (Section 7.3).
- `claim(param, effect)` — an optional declared belief, graded for consistency with the agent's own logged evidence.
- `submit(...)` — the tier-appropriate final answer; submitting ends the episode.

Blindness (statistics only, no configuration echo) closes the leakage channel a naive harness would open; the budget makes experiment selection part of the measured skill. The budget is enforced by the harness itself — calls past the eighth return an error instead of running; the separate x0.6 scoring gate (Section 4) exists for episode logs produced outside the live harness, where over-budget calls could otherwise appear.

## 4. Scoring

Scoring is computed deterministically from the episode log; there is no LLM judge in the headline metric.

| Tier | Correctness | Method rigor | Efficiency |
|---|---|---|---|
| L1 | parameter 30 + direction 20 | 30 | 20 |
| L2 | parameter 25 + direction 15 + magnitude 20 (10 if adjacent class) | 25 | 15 |
| L3 | both parameters 30 (12 for one) + interaction sign 25 | factorial rigor 25 | 20 |

*Method rigor* is the axis this benchmark exists for: at L1/L2 it awards points only when the submitted parameter is backed by a significant experiment that isolates exactly that parameter on the target metric — a correct answer must be *earned*, not guessed or pattern-matched. At L3 the bar is a genuine 2x2 factorial: a significant isolating experiment on each parameter plus a combined-change experiment. Efficiency rewards solving within few calls (zero if no experiments were run, so an unbacked lucky guess scores at most the correctness points). Exceeding the budget multiplies the final score by 0.6. Declared claims inconsistent with the agent's own logged statistics are tracked as *claim validity*.

### 4.1 Process-integrity audit

A second lens is computed over each episode and deliberately kept out of the headline score, so the published reproduction numbers remain frozen while method red flags still surface. The audit re-applies Holm-Bonferroni *across the episode's entire family* of isolating tests on the target metric (the MCP server corrects within a single experiment, but not across an episode) and separates two things that are easy to conflate:

- **Family-robustness** (`backingSurvivesHolm`): does the submission's evidence still clear alpha after family-wide correction? A minimal one-test-per-candidate sweep whose driver is merely borderline fails this honestly — that is low statistical power, reported but not blamed. (The reference baseline's swarm episode is exactly this case: raw p ~ 0.017, Holm-adjusted across three tests 0.0502.)
- **p-hacking** (`pHacking`): the behavioral red flag. It fires only when the agent *fished* — re-tested a parameter repeatedly, or ran more isolating tests than there are candidates — and then submitted a lone-significant hit that fails the family-wide correction. The minimal complete design is never flagged, however borderline its result.

The audit also classifies each submission's support: backed by an isolating experiment, resting on probe output-matching only, or entirely unbacked.

## 5. Difficulty Calibration

Every task ships with a statistically grounded difficulty rating: the binding driver's Cliff's delta (effect size) and an *oracle power* — the Monte-Carlo probability that an optimal isolating sweep solves a random-seed instance of the task configuration. Concretely: from the per-seed raw metric values collected at generation for each cell (control and every candidate's treatment), draw 2,000 seeded subsamples without replacement at the bench's replicate count (12), re-run the Mann-Whitney/Holm decision on each, and count the fraction in which the driver tests significant and every decoy does not. Because the bench's experiments are deterministic at a fixed replicate count, spare budget buys no statistical power, so the one-factor-at-a-time baseline and the oracle coincide; oracle power is the task's solvability margin. Ratings band into easy (power ≥ 0.85), moderate (≥ 0.55), and hard (< 0.55): across the 30 frozen tasks, 14 rate easy, 9 moderate, and 7

hard. The swarm tasks dominate the hard band (weak drivers on a noisy order parameter); morph's regime-change drivers give near-perfect separation. Methodology details: `docs/petri-bench-difficulty.md`.

## 6. Validation of the Simulation Layer

A benchmark over toy worlds is only as credible as the worlds. Four of the five simulations are checked, headless and reproducibly on every release, against an established result from their literatures (19/19 checks passing); origin's check is outstanding (Section 9):

| Simulation | Reproduced result | Reference | Checks |
|---|---|---|---|
| swarm | order-to-disorder transition under alignment noise | Vicsek et al. (1995) | 5/5 |
| market | stylized facts: heavy tails, volatility clustering, no linear autocorrelation | Cont (2001) | 3/3 |
| social | bounded-confidence cluster scaling ~ 1/(2 eps); consensus transition | Deffuant et al. (2000); Hegselmann & Krause (2002) | 4/4 |
| morph | Gray-Scott phase structure: washout / spots / labyrinth across the feed-kill plane | Pearson (1993) | 7/7 |

Each validation records an explicit caveat where the implementation differs from the cited model (e.g., the swarm boids are not the Vicsek model, so the claim is the qualitative transition, not the critical exponent; morph uses rescaled diffusion on a 96x96 torus, so the claim is the phase structure, not Pearson's exact coordinates). The origin simulation is not yet covered by a literature check (Section 9). Reports and scripted figures: `docs/validation/`.

## 7. Reference Solvers and Preliminary Results

### 7.1 Solver suite

Four algorithmic reference solvers calibrate the score scale. `random`: a seeded guess with no experiments — the chance floor. `ofat`: one isolating test per candidate using a curated, known-informative test value, then submit the significant driver; at L3 the same solver extends into the full 2x2 factorial the tier requires (the table's L3 row reports this factorial extension, not bare OFAT). `adaptive`: the same sweep with early stopping. `ofat-rand`: the privileged-value ablation — the *identical* procedure with test values drawn uniformly at random from each parameter's legal [min, max] (seeded, deterministic), isolating how much of the reference's performance comes from knowing informative values rather than from the procedure.

A generic `llm` solver runs any model in an identical agent loop: the model receives the task brief and the four harness tools, the harness enforces the budget, and the full transcript becomes the episode log. Two transports plug into the same loop — direct HTTP (Anthropic- and OpenAI-compatible endpoints) and locally installed CLIs (driven headless over a strict one-JSON-tool-call protocol, executing in an empty read-only scratch directory so the model cannot access task files). All solvers are scored by the same code on the

same contract. The verbatim scaffold (system prompt, task brief template, tool descriptions) is reproduced in Appendix A.

### 7.2 The standardized frontier evaluation (sweep v1)

Six frontier models were evaluated on all 30 frozen tasks (five simulations x two seeds x three tiers), **three episodes per task**, through the *identical* agent loop: the same system prompt, the same four tools, the same budget, the same scoring. API-served and subscription-CLI models alike run behind the `LlmClient` seam; CLI transports execute in an empty scratch directory in read-only mode, so no model can access the task files. 540 episodes (90 per model), all committed as self-contained artifacts.

Per-tier means over 30 episodes per cell. An *episode pass* is one complete run over all 30 tasks (pass 1, 2, 3); parentheses give the range of the three per-pass means — an honest view of run-to-run variance:

| Model | L1 | L2 | L3 | Overall | Solve rate | Avg calls |
|---|---|---|---|---|---|---|
| adaptive (curated values) | 94.8 | 94.2 | 87.5 | 92.2 | 100% | 3.4 |
| ofat (curated values; factorial at L3) | 92.5 | 92.5 | 87.5 | 90.8 | 100% | 4.0 |
| glm-5.1 | 68.0 (61.5-75.3) | 63.9 (57.4-68.0) | 53.2 (47.2-56.6) | 61.7 | 46% | 4.3 |
| opus-4.8 | **72.3** (68.5-76.3) | 57.9 (54.3-64.6) | 55.1 (45.3-65.7) | 61.7 | 43% | 5.3 |
| haiku-4.5 | 63.1 (55.5-76.5) | 59.6 (50.8-66.3) | **60.8** (59.1-62.1) | 61.1 | 37% | 4.5 |
| codex-gpt-5.5 | 69.9 (66.5-72.5) | 53.0 (47.7-59.8) | 53.9 (50.3-58.4) | 58.9 | 42% | 4.8 |
| sonnet-4.6 | 59.3 (56.8-61.0) | 59.2 (54.4-62.3) | 56.3 (54.6-57.4) | 58.3 | 41% | 4.1 |
| gemini-3.1-pro | 64.4 (52.3-74.5) | **65.5** (59.6-69.1) | 42.0 (29.0-52.9) | 57.3 | 50% | 6.5 |
| ofat-rand (ablation: random values; lowest of 3 draws, mean 45.2) | 38.5 | 47.8 | 35.2 | 40.5 | 33% | 3.7 |
| random (chance floor) | 10.0 | 19.0 | 28.1 | 19.0 | 3% | 0 |

**The headline at n=3: the reference sweep outscores every frontier model by 29 points or more (Wilcoxon signed-rank over the 30 paired tasks: the reference is better on 25-29 of 30 against every model, all p < 2x10^-5), solves 100% of tasks — and no model solves above 50% (best: gemini, 45/90 episodes).**

**Where the gap comes from — the privileged-value ablation.** The reference knows one informative test value per candidate; models must choose their own. Re-running the *identical* procedure with test values drawn blindly from each parameter's legal range (`ofat-rand`) collapses the score from 90.8 to 40.5 — *below every model*. Statistics, stated with the same discipline the benchmark demands. The ablation arm is itself sensitive to its value draws: three independent draw seeds score 40.5, 47.5, and 47.7 (mean 45.2; solve rate

33-37%), and the committed draw is the lowest — a fact that would flatter any per-draw significance test of the models' margin. We therefore treat the model-vs-ablation separation as *descriptive*: every model's mean exceeds the ablation's best draw (57.3 vs 47.7, a 9.6-point floor), the direction is unanimous across all six models and all three draws, and against the committed draw each model wins 18-20 of 30 paired tasks (tie-corrected Wilcoxon p = 0.007-0.041 uncorrected; three of six survive Holm across the family, three land at p = 0.053). The inferential anchor of this report is the reference-vs-model family, which is unambiguous: all six Holm-corrected p < 10^-4. The decomposition is therefore explicit rather than assumed: intervention design (choosing informative values) and procedural discipline (isolation, inference, factorial structure) are both first-order terms. Models occupy the middle — their intervention choices carry real information, but their *procedure* gives back the advantage: the failures Section 7.3 documents (direction submitted without measuring it, missing factorials, unbacked and probe-only submissions, p-hacks) are value-independent and visible directly in the logs. We retract the earlier working-draft claim that value knowledge accounts for "not most" of the gap; the ablation shows it is a dominant term *for the reference*, while the models' deficit against the reference is jointly procedural and informational.

Five structural observations:

- **The field is tight; the reference gap is not.** The six models span 4.4 points overall (57.3-61.7) — model-vs-model differences of this size are not statistically distinguishable at n=3 and much of any single-episode ordering is variance; we caution against reading model rankings off n=1 evaluations of agentic benchmarks generally. The 29-point gap to the curated-value reference, by contrast, is significant against every model and stable in every episode pass.
- **Tier profiles differ descriptively.** opus-4.8 posted the best L1 mean (72.3), gemini-3.1-pro the best L2 (65.5), haiku-4.5 the best L3 (60.8). Per-pass ranges overlap, so these are descriptive leaders, not significant orderings — but the *shapes* differ visibly: haiku is flat where gemini swings 23 points between tiers.
- **Solve rate and score diverge — by design.** gemini-3.1-pro has the *highest* solve rate (50%) and the *lowest* overall score: it finds answers while paying constant method taxes (redundant testing in 49 of 90 episodes, four probe-only submissions, the most calls). haiku-4.5 converts the *lowest* solve rate (37%) into a near-top score through clean, fully backed process. petri-bench separates "gets answers" from "does science"; a correctness-only benchmark would invert these two models.
- **Stability is itself a differentiator.** haiku's L3 per-pass means span 3 points (59.1-62.1) and sonnet is similarly tight everywhere, while opus's L3 spans 20 points (45.3-65.7) and gemini's L1 spans 22 (52.3-74.5). gemini's L3 weakness is real (42.0, last by 11 points) but its single-pass floor of 29.0 was the variance floor, not the mean.
- **Failures track the difficulty rating.** In the first episode pass, both L1 tasks rated `hard` by oracle power (the swarm order-parameter tasks) were solved by zero of six models, with the easy band mostly at 5-6 of 6; one moderate task (market-202) also defeated all six. The power-based rating (Section 5) carries predictive signal about where models fail — independent evidence that it measures something real.

Caveats — what "identical loop" does and does not standardize: identical were the system prompt, task briefs, tool surface and protocol, budget, and scoring; *not* standardized were decoding parameters (subscription CLIs run at their defaults), model snapshots (identities as the CLIs report them, June 2026), and the transport's text-JSON tool protocol, which is uniform across models but may depress absolute scores relative to native function-calling APIs. The benchmark's scoring also rewards exactly the isolate-and-infer design its generator verifies tasks against; this is by construction — the reference solvers are reference *implementations of the demanded method*, not arbitrary competitors — but it means scores measure fidelity to that method, not open-ended creativity (held-out task families that break the OFAT template, such as the planned L4 boundary-location tier, are the designed counterweight).

### 7.3 Failure taxonomy

Because every episode log is committed, failure analysis is computable rather than anecdotal. Each episode is classified into objective, non-exclusive flags along two axes — *what* went wrong (wrong parameter, direction error, magnitude/interaction error, no answer) and *how the method failed* (unbacked submission, probe-only support, invalid claims, p-hacking, over-testing, budget misuse). Denominators: direction errors are counted among episodes that identified the correct parameter; flags are episode counts out of the stated set and may co-occur. On the 540 standardized episodes:

- The algorithmic solvers trigger none of the behavioral flags in any episode — the scale is calibrated. (The audit's separate *family-robustness* statistic is not a flag and is reported for all solvers; the reference's borderline swarm episode fails it by design, as Section 4.1 explains — low power, not misconduct.)
- **gemini-3.1-pro chronically over-tests**: redundant isolating tests in 49 of its 90 episodes (18/30 at L1, 24/30 at L2). The same signature appeared under its own scaffold in the heterogeneous pilot (5/6); two scaffolds and three passes are consistent with a model-level disposition for this snapshot (June 2026), though confirming a stable trait would take more prompts and model versions. It buys nothing: gemini spends the most calls (6.5) for the lowest score.
- **The dominant L2 failure, across every model, is misreading the direction of the effect**: 69 of 180 L2 episodes end with the right parameter and the wrong direction, against only 11 magnitude misclassifications. Decomposing all 89 L1+L2 direction errors against the logs answers *why*: 54 (61%) never ran the one *direct* measurement of the hidden direction (a probe of the control against the hidden world) — 37 of those ran no probes at all (the direction was pure assumption), 17 attempted indirect probe-based inference that never pinned the contrast; the remaining 35 (39%) *had* a significant direct measurement in their own log and submitted against it. Skipped or indirect measurement accounts for three-fifths of direction failures, misread evidence for two-fifths; both are failures the harness makes visible. opus-4.8 is the extreme case (16 of 30 L2 episodes; 2 of 30 solved).
- **The integrity audit caught eleven method violations**: audited p-hacks by opus-4.8 (3), glm-5.1 (2), and gemini-3.1-pro (1) — lone-significant submissions surviving only through redundant testing, failing the family-wide Holm correction — plus probe-only "output-matching" submissions by gemini-3.1-pro (4) and sonnet-4.6 (1). **haiku-4.5 and codex-gpt-5.5 ran 90 episodes each without a single behavioral flag** — in this evaluation, process discipline was not ordered by capability tier or price.
- **sonnet-4.6 is decoy-fooled most often** (wrong parameter in 14 of 30 L1 episodes), echoing its pilot profile (4/6) at scale.
- **random** profiles as designed: 100% unbacked submissions, solve rate at chance.

Each model failed differently in this evaluation — answer-finding without discipline (gemini), skipped direction measurements atop clean identification (opus), decoy susceptibility (sonnet), low-yield but unflagged process (haiku) — and these profiles were consistent across both scaffolds and all three passes. This per-model failure structure is precisely the axis petri-bench is built to measure. The full table regenerates with one command (`taxonomy`) and is committed at `results/taxonomy.md`.

### 7.4 The heterogeneous pilot (historical)

An earlier pilot ran six models on the six `pilot-v0` L1 tasks under each model's own CLI agent scaffold (heterogeneous prompts and tool surfaces). It produced the same headline (baseline 92.5 vs best model 74.2) and first exposed the over-exploration and probe-output-matching failure modes, but its per-model ordering is confounded by scaffold differences — gemini's pilot 74.2 vs standardized 57.3 illustrates how large the scaffold-plus-task-set term can be (the pilot used six easier L1 tasks). The pilot artifacts remain committed under `results/pilot-v0/` for comparison; the standardized sweep supersedes them for all claims in this report.

### 7.5 Running the evaluation

The sweep is one command, config-driven, and crash-safe: it resumes past completed episodes, skips models whose credentials are absent, and an episode interrupted by an infrastructure failure is retried on the next pass rather than scored. API keys are read only from named environment variables (HTTP models) or stay inside the local CLIs (subscription models); nothing is stored in the repository. Marginal cost of sweep v1 was zero API spend (subscription CLIs; simulation calls are CPU-cheap), bounded mainly by wall-clock at roughly 5-7 minutes per model per tier.

## 8. Reproducibility and Artifacts

- **Frozen tasks.** All task specs are committed JSON; CI regenerates each from its seed and asserts byte-identity, and re-runs the reference baseline asserting it re-earns its exact frozen scores (92.5 on every L1 task of the six-task `pilot-v0` set and on the L2 set; 87.5 on the L3 set).
- **Self-contained episodes.** Every artifact is `{task, log, score, audit, provenance}`; the provenance stamp records the benchmark, MCP-server, and all five engine versions, the git commit, and the runtime, so any number in this report traces to the exact code that produced it.
- **Objective scoring.** Score, audit, and taxonomy recompute from the logs alone; the aggregate tables regenerate with `report` and `taxonomy`.
- **Public surface.** Tasks compose only the published `petri-labs-mcp` tool server (npm); the benchmark adds no privileged engine access.
- **Archive.** Code and tasks are archived at DOI 10.5281/zenodo.20618024 (concept DOI; version DOIs per release).
- **LaTeX export.** Verified: `pandoc docs/petri-bench-report.md -s -o petri-bench-report.tex` exports cleanly (all tables as longtables); the published HTML and PDF render from the same source via `packages/hub/scripts/render-report.sh`.

## 9. Limitations

- **Model realism.** The simulations are minimal model organisms, validated for qualitative phenomena (Section 6), not calibrated digital twins; transfer from benchmark skill to laboratory skill is an open question shared by all simulated-science benchmarks.
- **Post-release contamination.** Publishing the frozen sets and their episode artifacts (including ground truth) makes *these* instances recoverable by future models; the time-scoped claim in Section 3.2 holds only for the evaluations reported here. The durable defenses are procedural: regenerate fresh verified instances from new seeds for any future evaluation, and keep leaderboard seeds private.
- **Probes in the headline setting.** `probe` is deliberately available because hypothesis refinement against the system under study is part of real method — but it permits the output-matching shortcut. The current design answers with the rigor axis (a probe-only submission forfeits all method-rigor points) and the audit (it is flagged), rather than by removing correctness credit; reasonable designers could choose more harshly, and the full-metric-vector identification planned below would close the confound at the source.
- **Single-metric tasks.** The target metric is revealed, and identification currently rests on one metric, which both narrows the search space and enables the probe output-matching confound; scoring identification on the full metric vector (a parameter's signature) is the designed response.
- **Coverage.** origin lacks a literature validation; three episodes per task bounds but does not eliminate sampling error in per-tier means (per-pass ranges are reported alongside every mean), and subscription-CLI models run at their CLIs' default decoding settings.
- **Scaffold sensitivity.** Even the standardized loop measures model+prompt, not the model in isolation; the scaffold is versioned with the benchmark and reproduced verbatim in Appendix A. The pilot-vs-

standardized comparison (Section 7.4) quantifies how large scaffold effects can be, and the text-JSON tool protocol (uniform, but not native function calling) may depress absolute scores.
- **Saturation path.** If frontier models clear core-v0, the designed responses are tighter budgets and effect sizes, the L3/L4 tiers, and re-headlining on process quality (p-hacking and unbacked-claim rates remain informative even at high solve rates).
- **Determinism boundary.** Determinism is enforced in the headless engines; browser-rendered visualizations are display-only and outside the reproducibility contract.

## 10. Roadmap

(1) The standardized frontier sweep and report-quality results tables; (2) full-metric-vector identification to defeat output matching; (3) blind-metric tasks (the target metric no longer revealed); (4) an L4 tier: estimate the *location of a phase boundary* (the validated transitions of Section 6 become the task substrate); (5) a public leaderboard backed by the auditable episode artifacts; (6) an optional, human-validated LLM-judge axis for reasoning quality, kept out of the headline metric.

## Appendix A: The Evaluation Scaffold, Verbatim

Every model received exactly this scaffold. System prompt:

> You are a careful experimental scientist solving a hidden-parameter mystery. Exactly one (or, at L3, two) of the candidate parameters was changed from a revealed control to make a hidden treatment world. Use the tools to run controlled experiments — change ONE parameter at a time and read the Holm-adjusted p-value and effect size — and reach an evidence-backed answer. Only conclude an effect when a result is significant. You have a limited budget of experiments; do not waste calls. When confident, call `submit`.

Task brief template (values filled per task):

```
model: <simulation id>
target metric: <metric>
control config: <JSON of the revealed control>
the changed parameter is one of: <candidate names>
experiment budget: 8 calls

<tier goal — L1: "ONE parameter was changed. Identify which, and whether it
pushes the target metric UP or DOWN." | L2: adds the magnitude classes by
relative change |deltaPct|: small 10-35%, medium 35-75%, large >=75% |
L3: "TWO parameters were changed together. Identify BOTH and the SIGN of
their interaction (positive if the combined effect exceeds the sum of the
individual effects, else negative).">
Run experiments (change ONE parameter vs the control), then submit.
```

Tool descriptions as presented: `experiment` — "Run a replicated A/B between two configs; returns statistics only (no configs echoed)"; `probe` — "Compare a guessed config against the hidden world; a non-significant result means your guess matches"; `claim` — "Record a belief about one parameter's effect (optional bookkeeping)"; `submit` — the tier-appropriate final answer (parameter + direction; + magnitude at L2; the parameter pair + interaction sign at L3). Config arguments are described as "parameter overrides on the control (omit a param to leave it at control)." Note that the brief does not disclose parameter ranges; models choose intervention values from their own priors.

The ablation's additional value draws are reproducible: `PETRI_OFAT_RAND_SALT=1` (and `=2`) re-runs `ofat-rand` with the alternative draw seeds cited in Section 7.2.

CLI transports deliver the scaffold and transcript over stdin each turn and require exactly one JSON tool call in reply (one in-protocol retry, then the episode ends as a logged no-submission); HTTP transports use the providers' native tool-calling. The sweep in this report used the CLI transport for all six models, so the protocol term is uniform.

## References


- Chen, Z. et al. (2024). ScienceAgentBench: Toward Rigorous Assessment of Language Agents for Data-Driven Scientific Discovery. arXiv:2410.05080.
- Cliff, N. (1993). Dominance statistics: Ordinal analyses to answer ordinal questions. Psychological Bulletin, 114(3), 494–509.
- Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. Quantitative Finance, 1(2), 223–236.
- Deffuant, G., Neau, D., Amblard, F., & Weisbuch, G. (2000). Mixing beliefs among interacting agents. Advances in Complex Systems, 3(1–4), 87–98.
- Hegselmann, R., & Krause, U. (2002). Opinion dynamics and bounded confidence: models, analysis and simulation. Journal of Artificial Societies and Social Simulation, 5(3).
- Holm, S. (1979). A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics, 6(2), 65–70.
- Jansen, P. et al. (2024). DiscoveryWorld: A Virtual Environment for Developing and Evaluating Automated Scientific Discovery Agents. arXiv:2406.06769.
- Koblischke, N. et al. (2025). Gravity-Bench-v1: A Benchmark on Gravitational Physics Discovery for Agents. arXiv:2501.18411.
- Majumder, B. P. et al. (2024). DiscoveryBench: Towards Data-Driven Discovery with Large Language Models. arXiv:2407.01725.
- Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. Annals of Mathematical Statistics, 18(1), 50–60.
- Pearson, J. E. (1993). Complex patterns in a simple system. Science, 261(5118), 189–192.
- Reddy, C. K., & Shojaee, P. (2024). Towards Scientific Discovery with Generative AI: Progress, Opportunities, and Challenges. arXiv:2412.11427.
- Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I., & Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. Physical Review Letters, 75(6), 1226–1229.
- Wang, R., Jansen, P., Cote, M.-A., & Ammanabrolu, P. (2022). ScienceWorld: Is your Agent Smarter than a 5th Grader? arXiv:2203.07540.
- Yamada, Y. et al. (2025). The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search. arXiv:2504.08066.