DETECTING BENCHMARK MEMORIZATION IN LLMS ON NUMERICAL OPTIMIZATION TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) give accurate answers on classic test functions. Shown samples of an *unnamed* function, they often return its textbook optimum to four decimals, which raises the question of whether they are inferring the answer from the data or recalling the benchmark. We propose a memorization-controlled protocol for numerical optimization on continuous test functions, with integration and multi-fidelity prediction as control tasks. Our protocol includes: (i) a recall signature, defined as the near-exact answer, which no data-driven surrogate attains on any optimization or multi-fidelity cell; (ii) a procedural generator that moves each benchmark optimization function away from its published form by adding a random smooth field, whose amplitude in units of the function's own variability sets the distance, yielding a measured recall dose-response; and (iii) a surrogate floor that distinguishes anchoring on the benchmark answer from defensible shrinkage toward it. Across 16260 sessions with seven LLMs, models returned near-exact answers on benchmark optimization problems in up to 88% of sessions, even when given fresh samples and opaque identifiers. In 8590 sessions on novel functions, they never did. Models returned the benchmark answer less often as functions moved farther from the benchmark, and how quickly this happened varied by model. GPT and Claude models returned the benchmark answer in 23–78% of sessions even when it was strictly worse than an estimate fitted to the samples shown, and this held across every surrogate floor we tested. This suggests that models recognize a benchmark when deviations fall within 1–5% of the spread in the samples. Newer Claude and GPT models relied on benchmark answers more often than their predecessors, and for Claude, higher reasoning effort increased it as well; code execution did not eliminate recall. Whether recall occurred depended on the particular samples shown as well as on the function. Therefore, accuracy on classic test functions alone does not establish numerical reasoning; the protocol offers a reusable way to evaluate numerical performance while controlling for memorization.

1 INTRODUCTION

Ask a frontier language model to locate the minimum of an anonymized function from ten sampled points on the unit cube, forbid it from running code, and it will often return $\mathbf{x}^\star = (0.1239, 0.8183)$ and $f^\star = 0.3979$: the textbook optimum of the Branin function to four decimals. No method that fits ten points can do this. The answer is either numerical inference or recollection of one of the most reproduced test functions in the optimization literature, and on the published benchmark alone the two are indistinguishable. The distinction matters twice over: memorization threatens the validity of the benchmarks themselves, and language models are proposed as in-context regressors (Vacareanu et al., 2024), universal predictors (Song et al., 2024), optimizers (Yang et al., 2024) and components of Bayesian-optimization pipelines (Liu et al., 2024; Suwandi et al., 2025), and the evidence is accuracy on the same families of test functions. If that accuracy is recall, it will not generalize to new functions, and the model will keep recalling the old optimum even after the function changes.

Existing memorization diagnostics do not settle the question for continuous function tasks. Perturbation benchmarks for grade-school and competition math (Mirzadeh et al., 2025; Li et al., 2024; Huang et al., 2025) compare accuracy across a few discrete perturbation types, a verdict per type rather than a measure of how far recall reaches. Studies of memorized tabular datasets and nu-

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.

meric series (Bordt et al., 2024; Kotawala, 2026) show that models store public numbers and that performance degrades on perturbed versions of them, but without a ground truth that moves with the perturbation they cannot say whether returning the memorized answer was wrong. That is the crux: when a perturbation is tiny, the benchmark optimum can be closer to the truth than anything inferable from the samples, and returning it is the rational policy. A memorization control needs a signature that only recall can produce, a distance axis along which recall can be dosed with a computable regret at every rung, and a floor that separates recall from shrinkage.

We build all three. A *recall signature*: an answer is near-exact when it matches the analytical gold to a precision the samples cannot support; data-driven surrogates (Gaussian process (Rasmussen & Williams, 2006), random forest, TabPFN (Hollmann et al., 2025) and others (Eldred et al., 2006; Rushdi et al., 2017)) never attain it on an optimization or multi-fidelity task, while a *librarian* holding the closed forms of the classic functions attains it on every benchmark optimization instance and nowhere else. A *distance axis*: a procedural generator places novel functions at a controlled distance ε from each benchmark optimization function, ε being the amplitude of an added random smooth field in units of the function's own variability, so that the rate at which a model emits the benchmark answer can be traced as a dose-response, with the spread across perturbation seeds reported rather than averaged away. A *surrogate floor*: on every perturbed cell a fit to the shown samples (a Gaussian process; best-sample and TabPFN floors as checks) decides whether the benchmark answer is at least as good as fitting the data or strictly worse, and recall is diagnosed only in the second territory. The protocol is blind (opaque identifiers on every generated cell, unit-cube inputs), centres on optimization with integration and multi-fidelity prediction as control tasks, and a pre-registered tool arm lifts the no-code restriction, so that both channels that bypass inference, memory and execution, are controlled in one design.

Across seven frontier LLMs in two panels by release date (an earlier and a current panel) and 16260 sessions we find, first, a recall cliff: near-exact answers cover up to 88% of sessions on benchmark optimization functions with fresh samples and opaque identifiers and are absent on novel functions (0 of 8590 sessions), while no surrogate produces one on an optimization or multi-fidelity cell and the librarian always does. Second, a dose-response: emission of the benchmark answer decays with ε for every anchoring model, on a scale set by the model. Third, anchoring (returning the benchmark answer on a perturbed function) persists where it is strictly worse than a Gaussian-process fit to the shown samples: the GPT and Claude models still emit it 23–78% of the time under every floor we tried, at matched base and perturbation size we detect no difference between defensible and indefensible anchoring, and the models behave like a librarian whose acceptance tolerance is one to five percent of the sample spread. Fourth, recall grows with generation and effort: the current Claude and GPT models anchor more than their predecessors, and more reasoning effort makes Claude retrieve the benchmark answer more often. Fifth, code execution rescues weak recallers on benchmark cells, does not remove the recall channel, and partly repairs anchoring for one model only. Recall is a property of the instance as well as of the function: fresh draws of the same function move a model's recall rate from 0% to 90%, and on released cells answers often collapse to one bit-identical value across nominally independent sessions. On integration we detect none of this. Accuracy on classic test functions is therefore not, by itself, evidence of numerical reasoning.

Contributions. (1) A blind, cross-task protocol for numerical inference on continuous test functions with a recall signature (Section 5.1). (2) A procedural generator that places novel functions at a controlled distance from a benchmark, and the first measured recall dose-response on continuous tasks (Section 5.2). (3) A surrogate floor that separates anchoring from shrinkage, under which anchoring persists where it is strictly worse than fitting (Section 5.3). (4) Evidence that recall grows with model generation and reasoning effort, collapses answer diversity, depends on the sample draw, and is not removed by code execution (Sections 5.4 and 5.5).

2 Related work

LLMs as numerical predictors and surrogates. Language models are used as in-context regressors (Vacareanu et al., 2024; Garg et al., 2022), universal numeric predictors (Song et al., 2024; Song & Bahri, 2025; Tang et al., 2025) and components of Bayesian optimization and surrogate modelling (Yang et al., 2024; Ramos et al., 2026; Liu et al., 2024; Yang et al., 2026; Suwandi et al., 2025), always demonstrated on the classic test functions (Surjanovic & Bingham, 2013) and never

controlling for recall of them. The closest neighbour, Nafar et al. (2025), anonymizes feature names in tabular regression and concludes that models blend retrieval with in-context learning, on three real datasets with no recall detection and no novel control. Gupta et al. (2025) find LLM optimization agents insensitive to experimental feedback, which converges with our finding that anchoring persists where the data contradict it.

Memorization of numbers and benchmark contamination. Contamination detection has its own literature (Magar & Schwartz, 2022; Golchin & Surdeanu, 2024; Oren et al., 2024; Sainz et al., 2023; Dong et al., 2024); our concern is the memorized *answer* to a famous problem. Bordt et al. (2024) show that models memorized classic tabular datasets and that few-shot performance degrades from the original through perturbed, transformed and statistically matched versions, a four-level ladder; Tab-MIA supplies membership inference for tabular data (German et al., 2026); NumLeak (Kotawala, 2026) shows frontier models recall public numeric series with near-perfect correlation that collapses after the cutoff; Razeghi et al. (2022) trace few-shot arithmetic to pretraining term frequency. We add what this line lacks for continuous tasks: a perturbation with analytic ground truth that moves with it, so the regret of the memorized answer is computable at every distance, and a floor that says when returning it is worse than fitting. Al-Lawati et al. (2026) argue for contamination resistance by construction; a seeded generator with a distance axis is one instantiation.

Perturbation-based tests of reasoning versus memorization. Perturb-and-compare is the standard diagnostic on discrete reasoning benchmarks (Mirzadeh et al., 2025; Li et al., 2024; Huang et al., 2025; Srivastava et al., 2024; Sánchez Salido et al., 2025; Wu et al., 2024; Nezhurina et al., 2024); GSM1k (Zhang et al., 2024b) and LiveBench (White et al., 2025) re-collect fresh problems instead. These, and LLM-SRBench's transformed equations (Shojaee et al., 2025), compare accuracy across a few discrete perturbation types, none at a measured distance in function space.

Priors, shrinkage, scaling and tools. That in-context learning can be read as Bayesian inference under a pretrained prior (Xie et al., 2022) and that outputs are pulled toward high-prior answers regardless of input (McCoy et al., 2024; Wu et al., 2024) motivates the surrogate floor: the question is not whether a prior is used but whether it is used where the data say it should not be. That memorization grows with capacity (Carlini et al., 2023) and that memorized continuations override instructions more in larger models (McKenzie et al., 2023) anticipates our generational result. Code interpreters help on math word problems (Wang et al., 2024; Zhang et al., 2024a); whether execution removes or relocates recall on a data-only numerical task has not been measured.

3 A MEMORIZATION-CONTROLLED PROTOCOL

Numerical tasks. Each session presents an LLM with a function identifier, N sampled input–output pairs on the unit cube $[0,1]^d$, and one of three tasks. **OPT**: choose an optimizer and predict the argmin $\mathbf{x}^\star$ and minimum $f^\star$. **INT**: predict the integral over the unit cube. **MF**: given low- and high-fidelity samples, predict the high-fidelity value at held-out queries. To close the execution channel, the system prompt forbids code execution and external lookup; the tool arm (Section 4) lifts that restriction deliberately. Answers are returned as a JSON object; sessions whose JSON does not parse are excluded and counted (Section A). We pursue three goals on each task:

Identify the recall signature. An answer is *near-exact* when it matches the analytical gold to a precision the shown samples cannot support: for OPT, with $\hat{\mathbf{x}}$ and $\hat{f}$ the predicted argmin and minimum, $\|\hat{\mathbf{x}} - \mathbf{x}^\star\|_2/\sqrt{d} < 10^{-3}$ and $|\hat{f} - f^\star| < 10^{-3}$; for INT, relative error below 10^{-3}; for MF, at least six of eight queries within 10^{-3} relative error. Whether this is a recall signature is settled empirically by two reference agents (Section 4): data-driven surrogates with no knowledge of the benchmark, which should never attain it, and a *librarian* holding the closed forms of the classic test functions, which should attain it on every benchmark instance and nowhere else. A model that is near-exact where only the librarian is has recognized the function and retrieved its answer, from stored constants or by computing from a recognized closed form; we call both recall.

Define and measure a functional distance from the benchmark. For a base function f with canonical minimum we form $f_\varepsilon(\mathbf{x}) = f(\mathbf{x}) + \varepsilon\,\sigma_f\,g(\mathbf{x})$, with g a random smooth field drawn from

a perturbation seed and σ_f a robust scale of f, so that ε is a dimensionless perturbation size. It is a generator parameter, not a similarity the model can observe; its observable consequence, the displacement of the minimum, is recorded per cell. The true minimum of f_ε is located numerically and its distance from the canonical minimum, the *argmin drift*, is stored (Section A).

Score canonical emission and the surrogate floor. On a perturbed function the recall question becomes whether the model emits the *canonical* answer instead of the true one. We score *canonical emission* when the predicted argmin lies within $\min(0.01, \text{drift}/3)$ of a canonical minimum, a third of the drift so that the canonical and true minima are never both inside the ball (active only when the drift exceeds 6×10^{-3}) or when the predicted minimum value lies within 10^{-3} of the canonical $f^\star$ while the true $f^\star$ has moved by more than 2×10^{-3}. Emitting the canonical answer is not evidence of recall pathology by itself: at small ε it can be closer to the truth than anything inferable from N samples, and anchoring is then rational. We therefore fit a Gaussian process to the shown samples of every cell and call a cell *pathological* when the canonical answer is strictly worse than the GP's own prediction (argmin drift larger than the GP's argmin error, or the canonical $f^\star$ farther from the truth than the GP's regret), and *rational* otherwise. The floor bounds anchoring against a specific competent fitter, not against Bayes-optimality under an unknown prior, so we report it for several fitters and keep only conclusions that survive all.

4 Experimental design

LLM Panels. Seven frontier models in two panels by release date (Table 3): the *earlier* panel (Claude Opus 4.7, GPT-5.5, Gemini 3.1 Pro, Muse Spark 1.1), the models current when the benchmark cells were released, and the *current* panel (Claude Fable 5.1 at medium and high reasoning effort, GPT-5.6, Gemini 3.1 Pro, Muse Spark 1.3). Jointly, the two panels test whether the effect grows or shrinks across LLM generations. Gemini 3.1 Pro serves in both; Claude Opus 4.7 was run at high effort, the other earlier-panel models at medium. Every cell is run for 10 sessions at the vendor default temperature, routing pinned to the first-party host, served version logged.

Suites. Six suites. **Released**: cells from a public release of standard test functions (Surjanovic & Bingham, 2013) (the release itself is withheld for double-blind review and included in the supplementary repository; 15 optimization, 4 integration, 2 multi-fidelity) whose gold values are public and whose identifiers embed a name prefix (the first letters of the function name followed by the sample budget); three copies of one Branin cell under different identifiers form an alias probe. **Regenerated**: 43 cells on the same function library, including a twin of every released optimization cell, with fresh Latin-hypercube samples and opaque hash identifiers, as on every other suite (15 OPT including three 3-D Hartmann cells, 13 MF, 15 INT). **Novel** (96 OPT cells): 25 *family-A* functions far from any benchmark and 71 *family-B* functions on the perturbation ladder of Branin, Goldstein–Price, Rosenbrock and Ackley, two perturbation seeds, ε from 10^{-4} to 10. **Multi-seed** (50 OPT cells): Branin and Goldstein–Price, five rungs, five perturbation seeds per rung. **Novel INT** (26 cells generated, 10 run): novel integrands at the released parameterization, the cross-task control. **Fresh draws** (40 OPT cells): five new sample draws each of four canonical and four perturbed functions, at fixed function and budget.

Reference agents. On every cell we fit a Gaussian process (isotropic Matérn-5/2, fitted amplitude and noise), a thin-plate RBF interpolant, a random forest, a multilayer perceptron and TabPFN v2 to the shown samples and score them with the same metrics as the models; the best shown sample is the trivial floor (Section A). The librarian holds the six functions of the released optimization suite in unit-cube form, fits each to the shown samples by an affine map of the output, and returns that member's stored optimum if the best residual is below a tolerance τ (relative to the sample standard deviation), otherwise the best shown sample; we report $\tau \in \{10^{-3}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$, all far looser than the 10^{-5} the six-figure samples would justify.

Tool arm. A pre-registered arm lets the model call a sandboxed Python executor (uniform across vendors) on six regenerated benchmark cells and six near-canonical multi-seed cells for GPT-5.6, Claude Fable 5.1 and Gemini 3.1 Pro, matched against the same cells without tools; a robustness check repeats two cells with OpenAI's native interpreter.

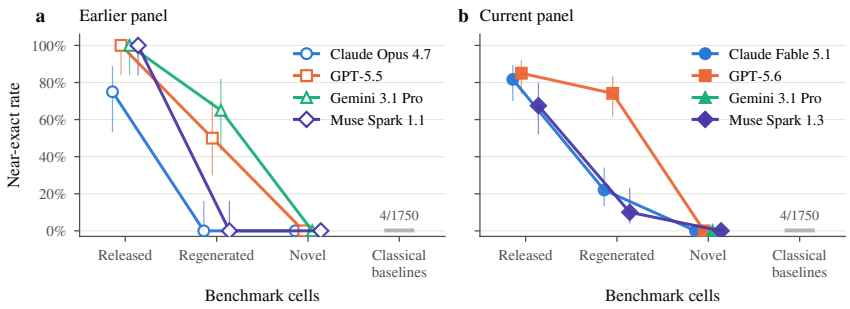


Figure 1: **The recall cliff.** Near-exact rate on released benchmark cells (gold values public), on the same functions regenerated with fresh samples and identifiers, and on novel functions: (a) earlier panel, two matched cells (all 18 released optimization prompts give 64–98%); (b) current panel, six matched cells (the current-panel released-cell probe did not include Gemini). Hollow: earlier; filled: current. 95% Wilson intervals. Gray: the pooled classical baselines (five fitted surrogates, plus the best shown sample on optimization cells), 4 of 1750 fits (1356 OPT, 114 MF, 280 INT), all on *integration* cells.

Table 1: Near-exact rate by task. Released cells for the earlier panel (optimization only), regenerated cells for the current panel, novel functions for both; classical baselines (five fitted surrogates, plus the best shown sample on optimization cells) pooled over all suites. Counts are near-exact sessions over scored sessions.

Suite	Model	OPT	MF	INT	Novel OPT
Released	Claude Opus 4.7	64%	–	–	0/960
Released	GPT-5.5	98%	–	–	0/931
Released	Gemini 3.1 Pro	89%	–	–	0/960
Released	Muse Spark 1.1	72%	–	–	0/957
Regenerated	Claude Fable 5.1	45%	21%	3%	0/959
Regenerated	GPT-5.6	88%	81%	5%	0/960
Regenerated	Gemini 3.1 Pro	39%	5%	0%	0/960
Regenerated	Muse Spark 1.3	23%	4%	0%	0/951
All suites	Classical baselines (5 fitters + best sample)	0/1356	0/114	4/280	0/936

Statistics. Session-level rates carry 95% Wilson intervals; where cells are the unit of replication, per-seed and per-cell rates are shown and pooled rates carry cell-cluster bootstrap intervals (2000 resamples of cells). Answer diversity within a cell is audited because bit-identical answers shrink the effective sample size (Section B.1). In total: 16260 model sessions and 306 baseline cells.

5 Results

5.1 The recall cliff, its floor, and the sample draw

On **released** benchmark cells, the earlier panel is near-exact in 64–98% of optimization sessions (Table 1): from ten samples the models return the analytical optimum to three decimals. The classical baselines given the same samples never do, across 1356 optimization and 114 multi-fidelity fits on canonical and novel cells alike, although the fits are competent (GP median argmin error 0.12 in the unit cube). The librarian identifies every released and regenerated canonical cell (30 of 30, residual below 6×10^{-5}), is near-exact on all of them and on no novel cell at any tolerance. Near-exact answers therefore cannot come from the data shown and can come from recognizing the function; they are a recall signature.

Regenerating the same functions with fresh samples and opaque identifiers removes most of it (Figure 1): Opus 4.7 falls from 15/20 to 0/20 and Muse Spark 1.1 from 20/20 to 0/20 on the two regenerated earlier-panel cells, GPT-5.5 (20/20 to 10/20) and Gemini (20/20 to 13/20) keep about half, and on six matched cells the current panel drops from 82% to 22% (Claude Fable 5.1), 68% to 10% (Muse 1.3) and 85% to 74% (GPT-5.6). Two things change at once: in the alias probe, the

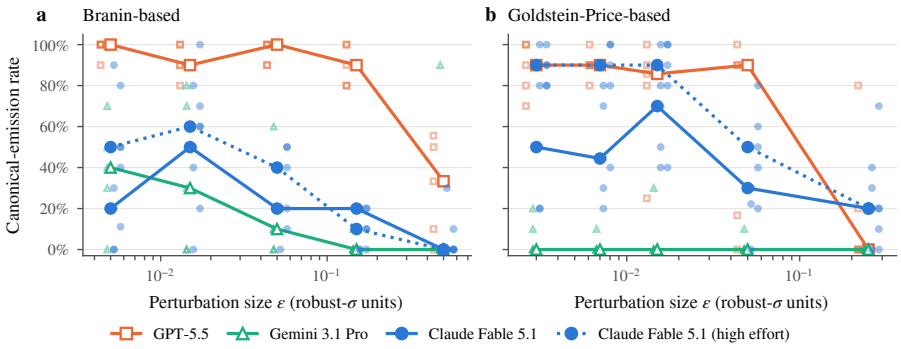


Figure 2: **Recall dose-response across perturbation seeds.** Canonical emission rate against perturbation size ε on (a) Branin-based and (b) Goldstein–Price-based novel functions, five seeds per rung: small markers are per-seed rates, lines join across-seed medians. The seed moves a rung by up to 100 points. Per-base ladders for all eight configurations: Figure 6.

same Branin samples under opaque identifiers drop Opus 4.7 from 7/10 to 0–3/10 and Muse 1.1 from 10/10 to 3–5/10 (GPT-5.5 and Gemini stay at 10/10), so for weaker recallers part of the drop is the name in the released identifier. Over the full regenerated grid the current panel is near-exact in 88/81/5% (GPT-5.6), 45/21/3% (Claude), 39/5/0% (Gemini) and 23/4/0% (Muse) of optimization, multi-fidelity and integration sessions. On novel functions the signature disappears: 0 near-exact sessions of 8590 across nine model configurations and 96 functions, including perturbations of Branin at $\varepsilon = 10^{-3}$ and of Rosenbrock at 10^{-4}. Integration is the control: no model is near-exact on canonical or novel integrands above chance (best 2/40; Figure 8), median relative errors are 2–18% and 11–47%, and a random forest reaches 13% near-exact at $N = 1000$, so on this task the signature is not diagnostic and, consistently, the models show no recall.

Recall is a property of the instance as well as of the function. Five fresh draws of the same function at the same budget (Table 6) move Claude Fable 5.1's near-exact rate from 0% to 60% on Branin, 0% to 90% on Goldstein–Price and 0% to 40% on Ackley and Hartmann-3, and GPT-5.6's from 30% to 100% on Ackley; the librarian identifies all twenty draws, so the variation is in what the model recognizes, not in what the samples determine. We call this *instance recognizability*. Two consequences follow. The released instances are unusually recognizable: for Claude the released draw is near-exact in 10/10 sessions on all three functions where the fresh draws reach at most 6, 9 and 4 of 10, and for GPT-5.6 it sits at the top of the fresh range, which is consistent with contamination from the public gold values, with a lucky draw and with the name prefix in the released identifier; we do not separate the three. And one draw per cell is one realization: the same replication on perturbed functions (Table 6, part B) moves GPT-5.6's canonical emission between 0% and 100% at fixed function and ε and flips the surrogate-floor verdict on two of four cells, so the per-cell rates below carry between-draw variance that only replication over draws can average out.

5.2 Recall dose-response

Moving the function away from the benchmark traces the recall out (Figure 2). On Branin GPT-5.5 emits the canonical optimum in 90–100% of sessions (median over seeds) at every rung to $\varepsilon = 0.15$ and collapses to 33% at 0.5; on Goldstein–Price it holds 85–90% through $\varepsilon = 0.05$ and reaches zero at 0.25. Claude Fable 5.1 emits at 20–50% on Branin and 20–70% on Goldstein–Price and decays over the same range; Gemini emits 40% at the smallest Branin rung, nothing by $\varepsilon = 0.15$, and nothing on Goldstein–Price at any ε. The cliff is set by the model, not the function: the rung that leaves GPT-5.5 at 90% leaves Gemini at zero. Across seeds a rung moves by up to 100 points (GPT-5.5 at $\varepsilon = 0.05$ on Goldstein–Price ranges 0–100%), so one perturbation draw would place the cliff anywhere within a factor of three in ε. The per-base ladders (Figure 6) show the same model ordering on Branin and Rosenbrock.

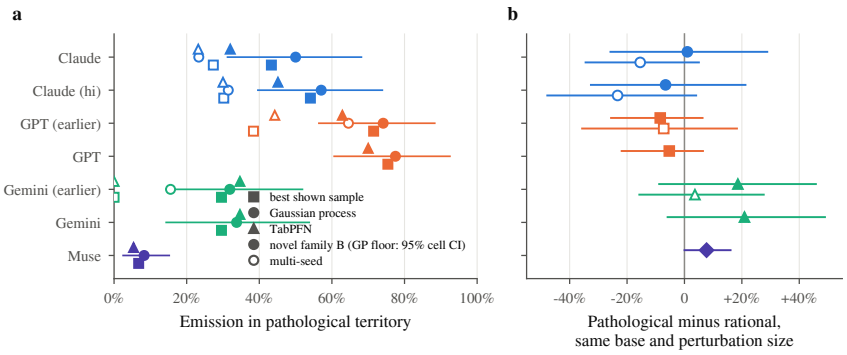


Figure 3: **Anchoring where it is worse than fitting.** (a) Canonical emission rate on pathological cells (canonical answer strictly worse than a fit to the shown samples) under three floors: best shown sample (squares), Gaussian process (circles), TabPFN (triangles); filled: novel family B, hollow: multi-seed; lines: 95% cell-cluster intervals for the GP floor. (b) Pathological minus rational emission at matched base and perturbation size under the GP floor, with cell-cluster intervals (filled: novel, 8 mixed strata; hollow: multi-seed, 6). Opus 4.7 emits nothing and is omitted.

5.3 Anchoring persists where it is worse than fitting

Under the GP floor 16 of the 71 novel family-B cells and 9 of the 50 multi-seed cells are pathological: on them a model that fitted the data in front of it would beat the benchmark answer. Models keep returning it (Figure 3a). In pathological territory GPT-5.6 emits it in 77.5% of sessions, GPT-5.5 in 74%, Claude Fable 5.1 in 50% (57% at high effort), Gemini in 34% and Muse in 8% on the novel cells, and 65%, 23%, 32% and 16% (GPT-5.5, Claude, Claude high, Gemini) on the multi-seed cells. The verdict depends on the fitter (24, 16 and 15 of the 71 novel cells under the best shown sample, GP and TabPFN, with different memberships) but the rate survives it: under every floor the two GPT models emit in 38–78% of pathological sessions and the two Claude configurations in 23–57%; under the strict verdict in which the canonical *value* is worse than the GP's regret (3 novel and 7 multi-seed cells) the rates are 47–69% and 17–65%. A model that inferred from the samples would score near zero, and so does a librarian that respects them: at acceptance tolerance $\tau = 10^{-3}$, already a hundred times looser than the printed precision, the librarian emits in 12% and 11% of pathological sessions; to match the models it needs τ between 10^{-2} (31% and 33%) and 5×10^{-2} (94% and 78%). The models behave like an identifier whose acceptance tolerance is one to five percent of the sample spread, two to three orders of magnitude looser than the data allow. The metric never fires on the six surrogates' own predictions for these 121 cells, and fires in 0% of sessions (Gemini 1.6–3.6%) on the 25 far novel functions (Section B).

Whether the verdict itself changes behaviour is a different question: pathological cells are concentrated on Goldstein–Price and Rosenbrock (13 of 16), rational cells on Branin and Ackley, and the models treat the bases differently. Comparing pathological and rational cells at the same base and perturbation size (Figure 3b) removes that confound and leaves nothing: within-stratum differences are -8 to $+21$ points on the novel cells and -23 to $+4$ on the multi-seed cells, every interval covers zero, and a logistic model with base, $\log \varepsilon$ and verdict gives verdict odds ratios of 0.7–1.9 with intervals spanning one. With 13 pathological and 16 rational cells in mixed strata this cannot rule out a drop of 20 points, so the claim we make is the conservative one: conditional on base and dose, we detect no sensitivity of anchoring to whether anchoring is defensible, and where it is not, the anchoring models return the benchmark answer in a quarter to three quarters of sessions.

5.4 Recall grows with generation and effort

If recall were a residue that training was removing, the current generation should anchor less than the previous one. It anchors more (Figure 4a): on the same 71 cells and metric, Claude Opus 4.7 emits the canonical answer in 0.7% of sessions and Claude Fable 5.1 in 40.1% (48.1% at high effort), GPT-5.5 in 69.2% and GPT-5.6 in 80.6%, Muse 1.1 in 5.4% and Muse 1.3 in 8.1%. Paired by cell, the increases are $+39$ points (95% cell-cluster interval $+31$ to $+48$) for Claude, $+11$ ($+6$

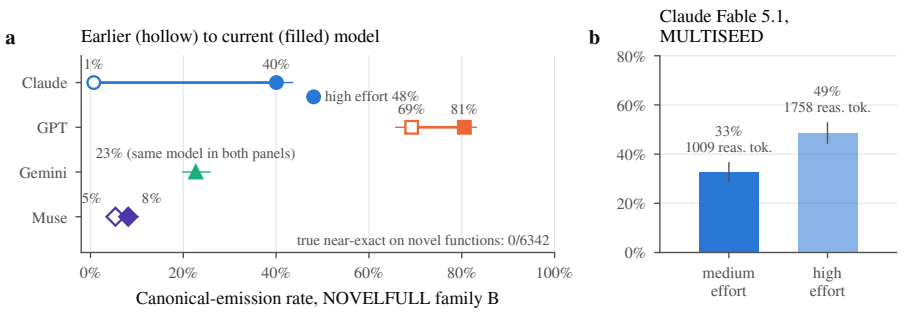


Figure 4: **Newer models and more reasoning anchor more.** (a) Canonical emission on novel family-B cells, earlier (hollow) to current (filled) model per family; Gemini 3.1 Pro is the same model in both panels; true near-exact on novel functions is zero for every model. (b) Claude Fable 5.1 on the multi-seed cells at medium and high reasoning effort, with mean reasoning tokens.

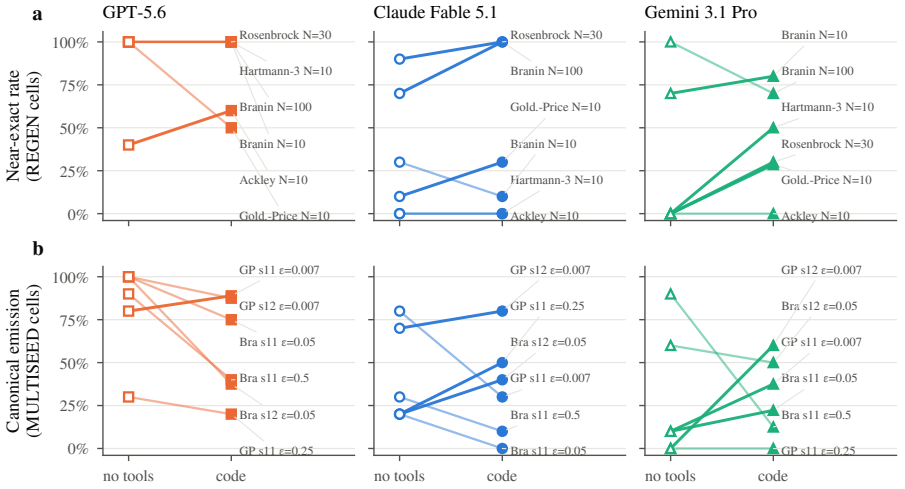


Figure 5: **The tool arm.** Each line is one cell (same function and samples), from the no-tools condition to code execution. (a) Near-exact on regenerated benchmark cells; (b) canonical emission on near-canonical multi-seed cells. Session counts per cell: Table 7.

to +18) for GPT and +3 (−1 to +6, not separated) for Muse. The panels differ in more than model version (Opus ran at high effort; vendor-side changes between windows are not otherwise controlled). Reasoning effort has the same sign within one model (Figure 4b): Claude Fable 5.1 at medium effort emits in 32.5% of multi-seed sessions with 1009 reasoning tokens on average, at high effort in 48.6% with 1758, a paired difference of +16 points (+10 to +23). The gain is +18 points on rational cells and +8 (−4 to +20) on pathological ones, and falls mostly on Goldstein–Price, whose canonical target needs no numerical precision to hit: thinking harder mainly retrieves the benchmark answer where retrieving it is defensible; it does not make the model abandon it where it is not. Answers on released cells also collapse to one bit-identical value across sessions (GPT-5.5 in 12 of 18 cells; Section B.1), so nominal intervals there are too narrow.

5.5 Code execution does not remove the recall channel

We pre-registered four predictions: strong recallers rarely invoke the tool on benchmark cells (P1), emission does not fall on near-canonical cells (P2), tool use rises with novelty (P3), and the vendor sandbox agrees with ours (P4). P4 held (18/19 near-exact with the native interpreter), P3 is untestable because tool use is at ceiling, and P1 and P2 each held in one half. Invited to execute code, models do so in nearly every session, yet near-exact on benchmark cells does not fall: pooled over six cells GPT-5.6 goes 90% to 88%, Claude 33% to 40%, Gemini 28% to 44%, every cell-

cluster interval covering zero (Figure 5a). Where recall was weak, execution rescues it (Gemini on Hartmann-3 0/10 to 5/10; GPT-5.6 on Ackley 4/10 to 6/10): the tool adds capability and leaves the recall channel intact. On near-canonical cells P2 was wrong for GPT-5.6 only, in the optimistic direction (Figure 5b): given an executor it partially falsifies its own recalled hypothesis, pooled emission falling from 83% (50/60) to 63% (27/43), -21 points with interval -44 to -1, robust to counting every unparsed session as an emission. Claude (40% to 35%, interval -27 to $+17$) and Gemini (28% to 29%, moving both ways by cell) do not change on average. Between 18% and 35% of executed scripts (GPT-5.6 98/549, Claude 67/190, Gemini 132/445) evaluate a closed-form candidate against the shown samples, so the models can verify a recalled hypothesis; only GPT-5.6 acts on it (code-condition denominators, which multi-round sessions can reduce, are in Table 7).

6 Discussion

Our results do not say that language models cannot infer from numerical data: on the far novel functions they are weak surrogates, with median argmin error (0.19–0.22) at the level of the best shown sample (0.16–0.19), above a Gaussian process (0.12) and TabPFN (0.06). They say that accuracy on the classic test functions, the evidence on which LLM surrogates and optimizers are judged (Vacareanu et al., 2024; Liu et al., 2024; Suwandi et al., 2025), mixes inference with recall in a proportion that is large, model-dependent, draw-dependent and invisible on the benchmark itself. Evaluations of numerical capability should report a recall signature with its surrogate and librarian references, and claims of transfer should rest on resampled or novel instances.

The surrogate floor turns a memorization result into a reasoning result. A model that returns the Branin optimum on a slightly perturbed Branin may be shrinking sensibly; one that returns it where a Gaussian process fitted to the ten samples would do better is disregarding the data, and that is what we measure. The librarian makes the mechanism quantitative: a recognizer that accepts a match at one to five percent residual reproduces the models' emission in pathological territory, and one that accepts only what the data support does not. That newer Claude and GPT models and more reasoning effort anchor more suggests a capability that scale improves rather than a residue it removes: better recognition of the function from fewer points, then more confident retrieval of its stored answer.

Throughout, memorization means corpus exposure to a function family and its published answer; we do not test whether a specific evaluation instance was in a training set, and the released-versus-regenerated gap is not such a test: models that predate the publication of the gold values show the same gap, and the released identifiers name the function. The fresh-draw replication adds that some ten-point instances of Branin are recognizable to a given model and others are not, though a librarian recognizes all: a model's recall rate on a function is an average over draws of a model-specific recognizer. The contamination resistance by construction that Al-Lawati et al. (2026) call for is achievable here with a seeded generator and a distance axis rather than with secrecy.

Limitations. Canonical emission is defined only where a canonical answer exists, so the generational and territory results use the 71 family-B cells. The pathological verdict is correlated with ε and base by construction, so territories are compared only at matched base and dose, on 13 pathological and 16 rational cells that cannot exclude a 20-point drop: we claim no detected sensitivity, not none. The floor is one fit of ten points whose verdict changes with the fitter and the draw, and every per-cell rate rests on one draw with between-draw ranges of up to 100 points, so replicating draws within cells is the right next design. The earlier-panel equivalence rests on two cells, and Gemini was not run on the released-cell probe. All functions are two- or three-dimensional at small budgets, where shrinkage toward a prior is most defensible; recall does not fall from $N = 10$ to 100 (Table 5), so its persistence at applied dimensions and budgets is open in both directions.

AI USE STATEMENT

We used generative AI tools to assist in implementing our methods in code; the system architecture and the design of the workflow and experiments were developed entirely by the authors. We did not use generative AI tools for any other task for which ICLR requires disclosure, and in particular did not use them to generate experimental data: all reported numbers are computed from logged model sessions by scripts released with the paper. The language models under study were accessed only through their public APIs as experimental subjects. We also used generative AI tools to create and edit figures and tables, search for information, identify and summarize relevant literature, suggest a paper structure, title, and keywords, and edit sections for readability.

All AI-generated code was reviewed and tested by the authors, figures and tables were checked against the underlying data, and all AI-assisted text was revised by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study evaluates publicly available language models on synthetic mathematical tasks and involves no human subjects or personal data. Model identifiers and served versions are logged so that findings can be attributed to specific releases.

REPRODUCIBILITY STATEMENT

All prompts, gold values, generator seeds, session records (including served model versions and token usage), baseline fits and figure data are released with the code. Section 4 and Section A give the generator, metric and baseline definitions; every figure and table in the paper is produced by a single script from the released records. An anonymized repository is provided as supplementary material.

REFERENCES

Ali Al-Lawati, Jason Lucas, Dongwon Lee, and Suhang Wang. LLM benchmark datasets should be contamination-resistant. *arXiv preprint arXiv:2605.19999*, 2026. Position paper, ICML 2026.

Sebastian Bordt, Harsha Nori, Vanessa Rodrigues, Besmira Nushi, and Rich Caruana. Elephants never forget: Memorization and learning of tabular data in large language models. In *Conference on Language Modeling*, 2024.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *International Conference on Learning Representations*, 2023.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12039–12050, 2024.

Michael S. Eldred, Shannon L. Brown, Brian M. Adams, Daniel M. Dunlavy, David M. Gay, Laura P. Swiler, Anthony A. Giunta, William E. Hart, Jean-Paul Watson, John P. Eddy, Josh D. Griffin, Patty D. Hough, Tammy G. Kolda, Monica L. Martinez-Canales, and Pamela J. Williams. DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 4.0 user's manual. Technical Report SAND2006-6337, Sandia National Laboratories, Albuquerque, NM, 2006.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*, 2022.

Eyal German, Sagiv Antebi, Daniel Samira, Asaf Shabtai, and Yuval Elovici. Tab-MIA: A benchmark dataset for membership inference attacks on tabular data in LLMs. In *International Conference on Learning Representations*, 2026.

Shahriar Golchin and Mihai Surdeanu. Time travel in LLMs: Tracing data contamination in large language models. In *International Conference on Learning Representations*, 2024.

Rushil Gupta, Jason Hartford, and Bang Liu. LLMs for Bayesian optimization in scientific domains: Are we there yet? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15482–15510, 2025.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 2025.

Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. MATH-Perturb: Benchmarking LLMs' math reasoning abilities against hard perturbations. In *International Conference on Machine Learning*, 2025.

Anany Kotawala. NumLeak: Public numeric benchmarks as latent labels in foundation models. *arXiv preprint arXiv:2605.30393*, 2026. MemFM workshop, ICML 2026.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. GSM-Plus: A comprehensive benchmark for evaluating the robustness of LLMs as mathematical problem solvers. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2961–2984, 2024.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In *International Conference on Learning Representations*, 2024.

Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 157–165, 2022.

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121, 2024.

Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, Tom Tseng, Tomasz Korbak, Xudong Shen, Yuhui Zhang, Zhengping Zhou, Najoung Kim, Samuel R. Bowman, and Ethan Perez. Inverse scaling: When bigger isn't better. *Transactions on Machine Learning Research*, 2023.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, 2025.

Aliakbar Nafar, Kristen Brent Venable, and Parisa Kordjamshidi. Learning vs retrieval: The role of in-context examples in regression with large language models. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 8206–8229, 2025.

Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Alice in Wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models. *arXiv preprint arXiv:2406.02061*, 2024.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto. Proving test set contamination in black-box language models. In *International Conference on Learning Representations*, 2024.

Mayk Caldas Ramos, Shane S. Michtavy, Andrew D. White, and Marc D. Porosoff. Bayesian optimization of catalysis with in-context learning. *ACS Central Science*, 12(5):599–615, 2026.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Yasaman Razeghi, Robert L. Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 840–854, 2022.

Ahmad A. Rushdi, Laura P. Swiler, Eric T. Phipps, Marta D'Elia, and Mohamed S. Ebeida. VPS: Voronoi piecewise surrogate models for high-dimensional data fitting. *International Journal for Uncertainty Quantification*, 7(1):1–21, 2017.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, 2023.

Eva Sánchez Salido, Julio Gonzalo, and Guillermo Marco. None of the others: a general technique to distinguish reasoning from memorization in multiple-choice LLM evaluation benchmarks. *arXiv preprint arXiv:2502.12896*, 2025.

Parshin Shojaee, Ngoc-Hieu Nguyen, Kazem Meidani, Amir Barati Farimani, Khoa D Doan, and Chandan K Reddy. LLM-SRBench: A new benchmark for scientific equation discovery with large language models. In *International Conference on Machine Learning*, 2025.

Xingyou Song and Dara Bahri. Decoding-based regression. *Transactions on Machine Learning Research*, 2025.

Xingyou Song, Oscar Li, Chansoo Lee, Bangding Yang, Daiyi Peng, Sagi Perel, and Yutian Chen. OmniPred: Language models as universal regressors. *Transactions on Machine Learning Research*, 2024.

Saurabh Srivastava, Annarose M B, Anto P V, Shashank Menon, Ajay Sukumar, Adwaith Samod T, Alan Philipose, Stevin Prince, and Sooraj Thomas. Functional benchmarks for robust evaluation of reasoning performance, and the reasoning gap. *arXiv preprint arXiv:2402.19450*, 2024.

Sonja Surjanovic and Derek Bingham. Virtual library of simulation experiments: Test functions and datasets. `https://www.sfu.ca/~ssurjano`, 2013.

Richard Cornelius Suwandi, Feng Yin, Juntao Wang, Renjie Li, Tsung-Hui Chang, and Sergios Theodoridis. Adaptive kernel design for Bayesian optimization is a piece of CAKE with LLMs. In *Advances in Neural Information Processing Systems*, 2025.

Eric Tang, Bangding Yang, and Xingyou Song. Understanding LLM embeddings for regression. *Transactions on Machine Learning Research*, 2025.

Robert Vacareanu, Vlad-Andrei Negru, Vasile Suciu, and Mihai Surdeanu. From words to numbers: Your large language model is secretly a capable regressor when given in-context examples. In *Conference on Language Modeling*, 2024.

Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. MathCoder: Seamless code integration in LLMs for enhanced mathematical reasoning. In *International Conference on Learning Representations*, 2024.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. LiveBench: A challenging, contamination-limited LLM benchmark. In *International Conference on Learning Representations*, 2025.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2022.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, 2024.

Zhuo Yang, Daolang Wang, Lingli Ge, Beilun Wang, Tianfan Fu, and Yuqiang Li. Reasoning BO: Enhancing Bayesian optimization with the reasoning power of LLMs. *Materials Genome Engineering Advances*, 4(3):e70086, 2026. doi: 10.1002/mgea.70086.

Chuyu Zhang, Songyang Zhang, Yingfan Hu, Haowen Shen, Kuikun Liu, Zerun Ma, Fengzhe Zhou, Wenwei Zhang, Xuming He, Dahua Lin, and Kai Chen. CIBench: Evaluating your LLMs with a code interpreter plugin. *arXiv preprint arXiv:2407.10499*, 2024a.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. A careful examination of large language model performance on grade school arithmetic. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024b.

A Protocol and generator details

A.1 Prompt and parsing

Every session uses the same system frame:

> *You are an expert in surrogate modeling, numerical integration, and black-box optimization. Do NOT execute code, do NOT call external tools. Use only your internal reasoning. Examine the provided point evaluations to infer the function's properties (smoothness, periodicity, peaks, discontinuities, modality, etc.) and produce the requested output as a single strict-JSON object.*

In the code condition the second sentence is replaced by "You MAY write and execute code to analyze the data and compute your answer." and nothing else changes. The user prompt lists the function identifier (a name prefix on released cells, an opaque hash elsewhere), the dimension, the sample budget, summary statistics of the sampled values and the samples as $\{\mathbf{x}, y\}$ rows to six significant figures, followed by the task description and the required JSON schema. For OPT the answer contains an optimizer choice from a fixed list of five, $\mathbf{x}^\star$ and $f^\star$; for INT the integral estimate; for MF the predictions at the held-out queries. Responses whose trailing JSON does not parse are excluded from rates and counted in Table 2; the content-hash cache never stores an errored call, so reruns resume rather than replay. Attrition is small everywhere except Muse Spark 1.3 on the regenerated grid (298 of 430 sessions parsed, concentrated on multi-fidelity cells) and GPT-5.5 on the integration cells (70 of 140), and GPT-5.6 and Gemini in the code condition, where multi-round sessions can exhaust the eight-round budget.

A.2 Models

A.3 Perturbation ladder

For base f on the unit square the perturbed function is $f_\varepsilon(\mathbf{x}) = f(\mathbf{x}) + \varepsilon\,\sigma_f\,g(\mathbf{x})$ with

$$g(x_1,x_2) = \sin(w_1x_1{+}\phi_1)\cos(w_2x_2{+}\phi_2){+}c_1w_3(x_1{-}\tfrac{1}{2})(x_2{-}\tfrac{1}{2}){+}c_2\sin\big(w_4(x_1{+}x_2)\big){+}c_3(x_1^2{-}x_2), \tag{1}$$

$w_i \sim U(3,8)$, $\phi_i \sim U(0,2\pi)$, $c_i \sim U(-1,1)$ drawn from the perturbation seed, and $\sigma_f = 1.4826\,\mathrm{MAD}(f)$ estimated from 2×10^5 uniform points, so that ε is a dimensionless perturbation size comparable across bases with very different ranges (Goldstein–Price is heavy-tailed). Bases are Branin, Goldstein–Price, Rosenbrock and Ackley on their standard domains mapped to the unit square, with Ackley's ε scaled by 20 because its conical basin does not move at the common scale.

Table 2: Attrition per run and model: sessions run / responses whose trailing JSON parsed / responses with a scorable answer. Unparsed sessions are excluded from rates; bounds treating them as emissions or non-emissions are given in Section B. The fresh-draw run (800 sessions, all parsed) is reported in Table 6.

Run	sessions / parsed / scorable by model
released cells (earlier panel)	Claude (earlier) 180/180/180; Gemini (earlier) 180/180/180; GPT (earlier) 180/179/179; Muse (earlier) 180/180/180
novel pilot	Claude (earlier) 100/100/100; Gemini (earlier) 100/100/100; GPT (earlier) 100/99/99; Muse (earlier) 100/100/100
novel full suite	Claude 960/959/959; Claude (earlier) 960/960/960; Gemini 960/960/960; Gemini (earlier) 960/960/960; GPT 960/960/960; GPT (earlier) 960/931/931; Muse 960/951/951; Muse (earlier) 960/957/957
novel suite, high effort	Claude (hi) 960/952/952
multi-seed	Claude 500/498/498; Claude (hi) 500/496/496; Gemini (earlier) 500/500/500; GPT (earlier) 500/470/470
multi-seed comparator (GPT-5.6)	GPT 60/60/60
integration null	Claude 140/140/140; Claude (earlier) 140/140/140; Gemini 140/140/140; Gemini (earlier) 140/140/140; GPT 140/140/140; GPT (earlier) 140/70/70; Muse 140/129/129; Muse (earlier) 140/140/140
regenerated grid	Claude 430/428/428; Gemini 430/428/428; GPT 430/428/428; Muse 430/298/298
regenerated (earlier panel)	Claude (earlier) 60/60/60; Gemini (earlier) 60/60/60; GPT (earlier) 60/50/50; Muse (earlier) 60/56/56
released cells (current panel)	Claude 60/60/60; GPT 60/60/60; Muse 60/41/41
tool arm (code)	Claude 120/120/120; Gemini 120/102/102; GPT 120/95/95
tool arm (native interpreter)	GPT 20/19/19

Table 3: Model configurations. Routing is pinned to the first-party host; the served model version is logged per session and matched the requested identifier in every logged session. All sessions were collected between 3 and 8 September 2026 with a maximum of 8192 output tokens.

Panel	Name	Identifier	Access	Temperature	Reasoning effort
earlier	Claude Opus 4.7	`anthropic/claude-opus-4.7`	OpenRouter (Anthropic)	0.7	high
earlier	GPT-5.5	`gpt-5.5-2026-04-23`	OpenAI	default	medium
earlier	Gemini 3.1 Pro	`google/gemini-3.1-pro-preview`	OpenRouter (Google)	0.7	medium
earlier	Muse Spark 1.1	`meta/muse-spark-1.1`	OpenRouter (Meta)	0.7	medium
current	Claude Fable 5.1	`anthropic/claude-fable-5.1`	OpenRouter (Anthropic)	0.7	medium
current	Claude Fable 5.1 (high effort)	`anthropic/claude-fable-5.1`	OpenRouter (Anthropic)	0.7	high
current	GPT-5.6	`gpt-5.6-sol`	OpenAI	default	medium
current	Gemini 3.1 Pro	`google/gemini-3.1-pro-preview`	OpenRouter (Google)	0.7	medium
current	Muse Spark 1.3	`meta/muse-spark-1.3`	OpenRouter (Meta)	0.7	medium

The true minimum of f_ε is located by four stages of 801×801 grid refinement, and the argmin drift is the distance from the nearest canonical minimum (Branin has three). The full novel suite uses rungs $\varepsilon \in \{10^{-3}, 3\times10^{-3}, 5\times10^{-3}, 10^{-2}, 1.5\times10^{-2}, 3\times10^{-2}, 5\times10^{-2}, 7\times10^{-2}, 9\times10^{-2}, 0.12, 0.15, 0.25, 0.5\}$ with extra rungs at 6–9×10^{-3} for Goldstein–Price and 10^{-4}, 3×10^{-4} for Rosenbrock, two perturbation seeds for Branin and one for the others; the multi-seed suite uses five seeds at $\varepsilon \in \{0.005, 0.015, 0.05, 0.15, 0.5\}$ (Branin) and $\{0.003, 0.007, 0.015, 0.05, 0.25\}$ (Goldstein–Price). Family-A functions are seeded random compositions of sinusoids, Gaussian bumps and a bilinear term with no relation to any benchmark. Every cell has $N = 10$ samples from a seeded Latin hypercube.

A.4 Classical baselines

Each fitter is trained on the shown samples of a cell and scored with the same metrics as the models. The Gaussian process uses scikit-learn's `GaussianProcessRegressor` with an isotropic Matérn-5/2 kernel times a fitted constant, plus a fitted white-noise term (bounds 10^{-10} to 10^{-1}), targets normalized, two optimizer restarts. The RBF interpolant is SciPy's thin-plate-spline `RBFInterpolator`. The random forest has 200 trees; the multilayer perceptron has two hidden layers of 64 units on standardized inputs and targets; TabPFN v2 is used with its default configuration. For OPT the fitted surrogate is minimized on a 201×201 grid (41^3 in three dimensions) and the grid minimizer and its predicted value are the baseline's $\mathbf{x}^\star$ and $f^\star$; for INT the surrogate

is averaged over the same grid; for MF the surrogate is fitted on the high-fidelity samples only and evaluated at the queries. The best-sample floor returns the shown sample with the lowest value. Cells with fewer than two high-fidelity samples are skipped and the RBF interpolant fails on the three three-point multi-fidelity functions in both suites, six cells (Section 4).

A.5 Surrogate floor

For a perturbed cell with GP argmin error d_{GP} and GP regret r_{GP} (both against the true minimum of f_ε), the cell is pathological when $\mathrm{drift} > \sqrt{2}\, d_{\mathrm{GP}}$ or $|f^\star_{\mathrm{canon}} - f^\star| > r_{\mathrm{GP}}$, and rational otherwise. The $\sqrt{2}$ converts the GP's per-coordinate error to the unit-cube distance used for drift. An earlier version of the floor used the best shown sample instead of the GP; it flips many verdicts (24 rather than 16 pathological cells on the novel suite) and is reported in Section B for comparison.

A.6 Tool arm

The executor runs the model's Python in a fresh subprocess with a scrubbed environment, a 30-second wall-clock limit and NumPy available, returns stdout and stderr truncated to a fixed length, and allows up to eight execution rounds per session; the same executor serves every vendor through function calling, and the native-interpreter check uses OpenAI's hosted sandbox instead. A script counts as *verification behaviour* when it contains a closed-form candidate (a trigonometric, exponential or polynomial expression), at least six numeric literals with three or more decimals (the shown samples), and a residual or difference computation. The pre-registration text with the four predictions is released with the code (`notes/tool_arm_prereg.md`).

B Additional results

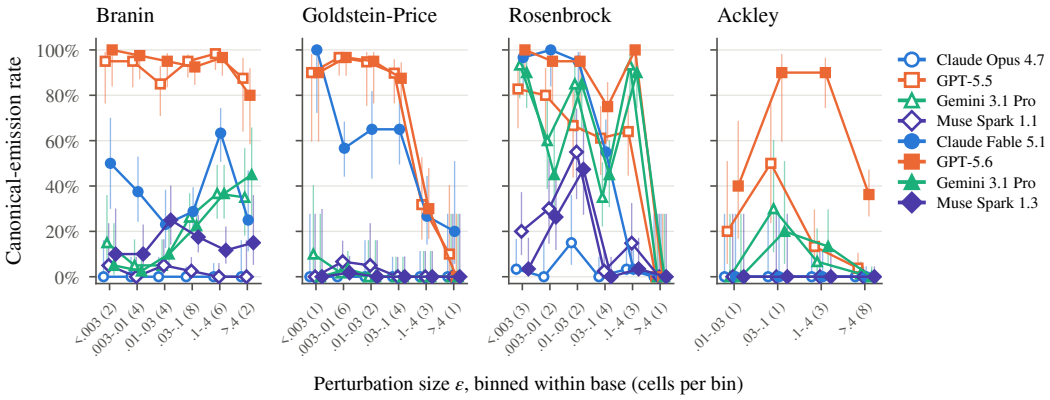


Figure 6: Canonical emission rate by perturbation-size bin on novel family-B cells, per base function (the bases have different ε scales and are never pooled), for all eight model configurations. The number of cells per bin is given under each label. Error bars are 95% Wilson intervals.

B.1 Answer collapse

On released cells GPT-5.5 returns one bit-identical $(\mathbf{x}^\star, f^\star)$ in all ten sessions of 12 of 18 cells (mean unique-answer fraction 0.14), the other earlier-panel models in 5–6 (Figure 7). On novel functions unique-answer fractions rise to 0.5–0.9 and collapse is rare (GPT-5.6 14 of 96 cells, GPT-5.5 8 and 7 of 50 multi-seed cells, others 0–3). Collapse is memorization evidence and a statistical warning: a collapsed cell's ten sessions carry the information of one, so nominal intervals on released benchmarks are too narrow.

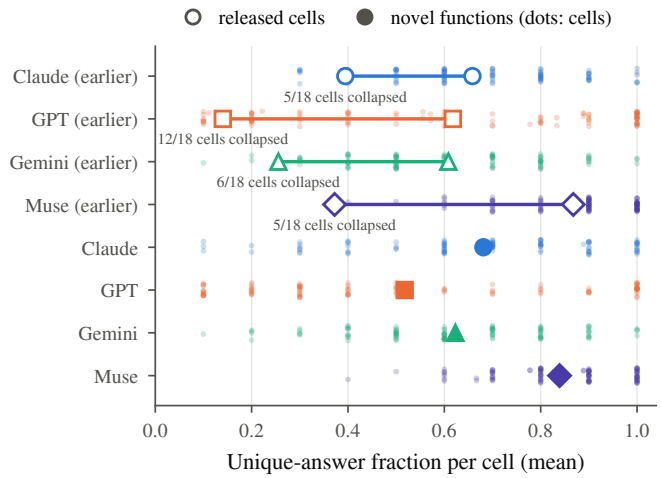


Figure 7: **Answer-entropy collapse on released cells.** Mean fraction of distinct $(\mathbf{x}^\star, f^\star)$ answers over ten sessions per cell. Earlier-panel models: released cells (hollow) versus novel functions (filled), with the number of cells whose ten sessions returned one bit-identical answer; current models: novel functions only. Small dots are individual novel cells.

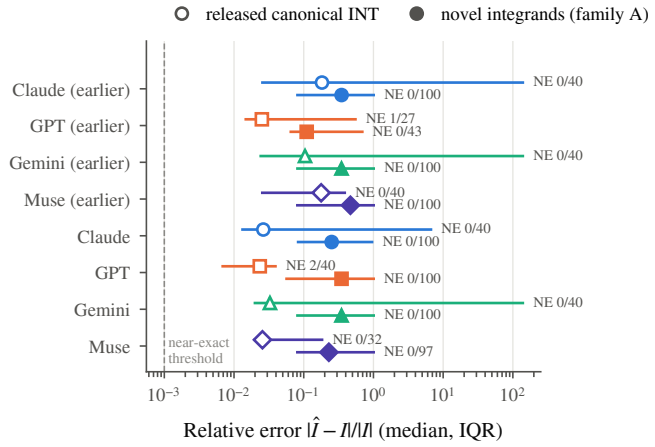


Figure 8: **Integration shows no recall.** Relative error of the predicted integral on released canonical integrands (hollow) and novel family-A integrands (filled), median and interquartile range on a log axis, with near-exact counts. No model reaches the near-exact threshold on either group above chance.

B.2 Territory, floors and false positives

Pooled territory rates and floor sensitivity. Table 4 gives the pooled emission rates by territory under the GP floor. They are not a controlled comparison: pathological cells are 7 Goldstein–Price, 6 Rosenbrock and 3 Branin (no Ackley), rational cells 23 Branin, 13 Ackley, 10 Goldstein–Price and 9 Rosenbrock, and the models treat the bases differently (Gemini emits nothing on Goldstein–Price at any ε), so the pooled contrast has opposite signs on the novel and multi-seed suites. The controlled comparison is Figure 3b. Under the best-shown-sample, GP, TabPFN and random-forest floors the number of pathological novel cells is 24, 16, 15 and 17 (multi-seed: 14, 9, 7 and 7), and the emission rate in pathological territory is 71–75% (GPT-5.6), 63–74% (GPT-5.5), 32–50% (Claude Fable 5.1), 45–57% (high effort), 25–35% (Gemini) and 1–8% (Muse) on the novel cells; on the multi-seed cells 38–65%, 23–37%, 30–39% and 0–16% for GPT-5.5, Claude, Claude high and Gemini. Restricting the verdict to the $f^\star$ channel (canonical value worse than the GP's regret)

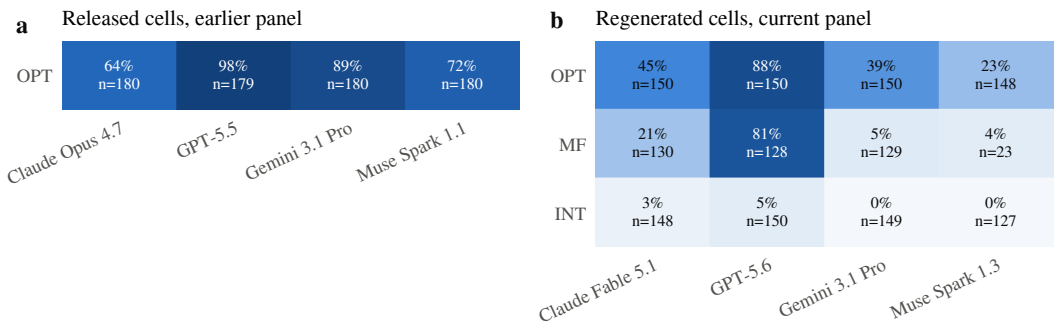


Figure 9: Near-exact rate by task and model on (a) released cells for the earlier panel (optimization cells only) and (b) regenerated cells for the current panel. The two panels are different model generations on different cells and are not a paired comparison; the paired comparisons are in Figure 1.

Table 4: Canonical emission rate by rationality territory under the Gaussian-process floor. Pathological cells are those where the canonical answer is strictly worse than a GP fit to the shown samples. Sessions in parentheses. Gemini 3.1 Pro is the same model in both panels; its two rows are independent replications.

	Novel family B		Multi-seed	
Model	rational	pathological	rational	pathological
Claude Opus 4.7	0.9% (550)	0.0% (160)	–	–
Claude Fable 5.1	37.2% (549)	50.0% (160)	34.6% (408)	23.3% (90)
Claude Fable 5.1 (high effort)	45.5% (547)	57.1% (156)	52.3% (407)	31.5% (89)
GPT-5.5	67.9% (539)	74.1% (143)	79.5% (391)	64.6% (79)
GPT-5.6	81.5% (550)	77.5% (160)	–	–
Gemini 3.1 Pro	20.5% (550)	31.9% (160)	12.0% (410)	15.6% (90)
Gemini 3.1 Pro	19.5% (550)	33.8% (160)	–	–
Muse Spark 1.1	5.5% (550)	5.1% (157)	–	–
Muse Spark 1.3	8.1% (544)	8.3% (157)	–	–

leaves 3 novel and 7 multi-seed cells; emission there is 67% / 61% (GPT-5.6 / GPT-5.5 on novel; GPT-5.5 on multi-seed), 47% / 20% (Claude) and 60% / 20% (Gemini).

False-positive calibration of the emission metric. Scored on the classical fitters' own predictions (each fitter's grid minimizer and its value) on the 71 novel family-B and 50 multi-seed cells, the metric fires 0 times for each of the six fitters, in pathological and rational territory alike, so a data-driven fit landing in the canonical basin by accident is not a source of emission at these budgets. Applied to the models' answers on the 25 far novel functions, which have no canonical answer, with each base's canonical minima and $f^\star$ as the target, the metric fires in 0 of 250 sessions per base for Claude, GPT and Muse configurations and in 4–9 of 250 for Gemini on the Ackley and Rosenbrock targets (both have $f^\star = 0$), i.e. a generic preference for the value 0 accounts for at most 3.6% of Gemini's emission and none of any other model's.

Table 5: Near-exact sessions out of 10 by sample budget N on canonical optimization cells: released cells for the earlier panel and regenerated cells for the current panel. Recall does not fall with N.

Suite	Model	Branin	Ackley	Goldstein–Price	Rosenbrock	Hartmann-3
Released	Claude Opus 4.7	5/5/1	9/8/10	10/9/10	3/3/4	10/9/10
Released	Gemini 3.1 Pro	10/10/10	4/9/10	10/10/10	5/6/9	10/8/10
Released	GPT-5.5	10/10/10	10/10/10	10/10/10	10/10/7	9/10/10
Released	Muse Spark 1.1	10/10/10	1/8/10	10/10/10	1/1/1	10/9/10
Regenerated	Claude Fable 5.1	3/7/7	0/9/1	1/1/3	9/9/9	0/4/4
Regenerated	Gemini 3.1 Pro	7/10/10	0/10/6	0/2/5	1/0/1	0/0/7
Regenerated	GPT-5.6	10/10/10	4/10/6	10/10/9	8/10/7	10/8/10
Regenerated	Muse Spark 1.3	3/8/5	0/1/0	0/1/0	0/0/0	1/9/6

Entries are $N = 10/30/100$.

Table 6: Fresh sample draws at fixed function and budget (REDRAW). Part A: near-exact sessions out of 10 on each of five fresh Latin-hypercube draws of a canonical function, next to the single regenerated draw of the main grid and the released draw (whose prompt carries a name-prefixed identifier). Part B: GP-floor verdict (P = pathological, r = rational) and canonical emission sessions out of 10 on five fresh draws of a perturbed multi-seed function; the original draw's verdict is given first.

Function	Model	Fresh draws 1–5	Range	Regenerated	Released
Ackley $N = 100$	GPT-5.6	9/10 7/10 7/10 10/10 3/10	30%–100%	6/10	10/10
Ackley $N = 100$	Claude Fable 5.1	4/10 4/10 0/10 3/10 0/10	0%–40%	1/10	10/10
Branin $N = 10$	GPT-5.6	8/10 10/10 10/10 9/10 10/10	80%–100%	10/10	10/10
Branin $N = 10$	Claude Fable 5.1	1/10 0/10 0/10 5/10 6/10	0%–60%	3/10	10/10
Goldstein–Price $N = 30$	GPT-5.6	10/10 5/10 9/10 10/10 10/10	50%–100%	10/10	10/10
Goldstein–Price $N = 30$	Claude Fable 5.1	0/10 0/10 4/10 3/10 9/10	0%–90%	1/10	10/10
Hartmann-3 $N = 10$	GPT-5.6	10/10 10/10 10/10 8/10 9/10	80%–100%	10/10	–
Hartmann-3 $N = 10$	Claude Fable 5.1	0/10 0/10 4/10 0/10 0/10	0%–40%	0/10	–
Perturbed function	**Model**	**Fresh draws 1–5 (emission)**	**Range**	**Verdicts (orig; draws)**	
Branin, seed 11, ε=0.05	GPT-5.6	10/10 10/10 9/10 10/10 9/10	90%–100%	r; rrrrr	
	Claude Fable 5.1	0/10 8/10 3/10 5/10 0/10	0%–80%		
Branin, seed 12, ε=0.15	GPT-5.6	2/10 2/10 4/10 9/10 9/10	20%–90%	r; PPPrr	
	Claude Fable 5.1	3/10 3/10 3/10 0/10 2/10	0%–30%		
GP, seed 11, ε=0.007	GPT-5.6	10/10 8/10 10/10 10/10 5/10	50%–100%	r; rrrrr	
	Claude Fable 5.1	3/10 5/10 5/10 5/10 8/10	30%–80%		
GP, seed 12, ε=0.05	GPT-5.6	0/10 9/10 2/10 10/10 10/10	0%–100%	r; rPPrP	
	Claude Fable 5.1	2/10 4/10 8/10 3/10 6/10	20%–80%		

Table 7: Tool arm, all cells: near-exact (regenerated benchmark cells) or canonical emission (multi-seed cells) under the no-tools and code-execution conditions, with the fraction of code-condition sessions that executed at least one script and the median number of executions per session.

Cell	Model	No tools	Code	Tool use	Rounds
Near-exact					
Ackley N=10	GPT-5.6	4/10	6/10	10/10	4.0
Ackley N=10	Claude Fable 5.1	0/10	0/10	10/10	1.0
Ackley N=10	Gemini 3.1 Pro	0/10	0/10	10/10	2.0
Branin N=10	GPT-5.6	10/10	10/10	10/10	1.0
Branin N=10	Claude Fable 5.1	3/10	1/10	10/10	1.0
Branin N=10	Gemini 3.1 Pro	7/10	8/10	10/10	3.5
Branin N=100	GPT-5.6	10/10	10/10	4/10	0.0
Branin N=100	Claude Fable 5.1	7/10	10/10	7/10	1.0
Branin N=100	Gemini 3.1 Pro	10/10	7/10	7/10	2.0
Gold.-Price N=10	GPT-5.6	10/10	2/4	10/10	8.0
Gold.-Price N=10	Claude Fable 5.1	1/10	3/10	10/10	3.0
Gold.-Price N=10	Gemini 3.1 Pro	0/10	2/7	10/10	5.5
Hartmann-3 N=10	GPT-5.6	10/10	9/9	10/10	2.0
Hartmann-3 N=10	Claude Fable 5.1	0/10	0/10	9/10	1.0
Hartmann-3 N=10	Gemini 3.1 Pro	0/10	5/10	8/10	1.0
Rosenbrock N=30	GPT-5.6	10/10	9/9	10/10	3.0
Rosenbrock N=30	Claude Fable 5.1	9/10	10/10	5/10	0.5
Rosenbrock N=30	Gemini 3.1 Pro	0/10	3/10	8/10	3.5
Canonical emission					
Bra s11 ε=0.05	GPT-5.6	10/10	6/8	10/10	6.0
Bra s11 ε=0.05	Claude Fable 5.1	2/10	0/10	10/10	1.0
Bra s11 ε=0.05	Gemini 3.1 Pro	1/10	2/9	9/10	2.5
Bra s11 ε=0.5	GPT-5.6	9/10	2/5	9/10	7.5
Bra s11 ε=0.5	Claude Fable 5.1	3/10	1/10	10/10	1.0
Bra s11 ε=0.5	Gemini 3.1 Pro	9/10	1/8	9/10	5.5
Bra s12 ε=0.05	GPT-5.6	10/10	3/8	10/10	6.5
Bra s12 ε=0.05	Claude Fable 5.1	2/10	4/10	10/10	1.0
Bra s12 ε=0.05	Gemini 3.1 Pro	6/10	4/8	9/10	4.0
GP s11 ε=0.007	GPT-5.6	8/10	8/9	10/10	5.0
GP s11 ε=0.007	Claude Fable 5.1	8/10	3/10	10/10	3.0
GP s11 ε=0.007	Gemini 3.1 Pro	1/10	3/8	10/10	6.0
GP s11 ε=0.25	GPT-5.6	3/10	1/5	10/10	7.5
GP s11 ε=0.25	Claude Fable 5.1	2/10	5/10	10/10	2.0
GP s11 ε=0.25	Gemini 3.1 Pro	0/10	0/7	10/10	4.0
GP s12 ε=0.007	GPT-5.6	10/10	7/8	10/10	5.0
GP s12 ε=0.007	Claude Fable 5.1	7/10	8/10	10/10	2.0
GP s12 ε=0.007	Gemini 3.1 Pro	0/10	3/5	9/10	7.5