Can Language Models Learn a PDE? A Benchmark for Scientific Reasoning through Experimental Design

Joseph K. Miller [1]

Abstract

We study how language models choose and use measurements when identifying an unknown differential equation. In this benchmark, a model selects weak-form test functions under a query budget and uses numerical tools to fit and check the hidden coefficients. The interaction records measurement choices, verification, error reduction, and confidence alongside final accuracy. We analyze 180 scored runs from six models on 30 shared tasks. On medium tasks, Gemini 3.1 Pro and Claude Opus 4.6 both achieve 9/10 success, but their mean consistency-check counts are 2.3 and 7.6. Equal aggregate success thus coexists with different patterns of tool use. Numerical controls provide a reference: a fixed sine design solves all 30 reconstructed tasks and succeeds on 92%, 91%, and 98% of 100 fresh tasks in each of three settings at the minimum query budget. The resulting records support a study of experimental behavior, while the benefit of adaptive measurement selection remains an open question. We release the environment, analysis, and numerical controls.

1. Introduction

Scientific investigation requires choosing what to measure, interpreting the result, and deciding what to do next. We study these decisions in a small mathematical environment: a language model probes an unknown elliptic equation, uses numerical tools to estimate its coefficients, and checks or revises the estimate. The model selects the experimental sequence; the tools perform the linear algebra.

An answer-only score leaves much of this process unobserved. A model may recover the coefficients on its first solve, substantially improve an initially poor estimate, or repeatedly check an inaccurate answer. Our benchmark records the measurements and tool interactions alongside a mathematically defined error. It was introduced in the Metacognition track of Kaggle's Measuring Progress Toward AGI competition (Miller, 2026; Burnell et al., 2026). The central question is how models with different or similar outcomes use the same experimental interface.

We make three contributions: a tool-based environment connecting measurement choices to known coefficient truth; an analysis of outcome and behavioral profiles for six models on 30 shared tasks; and numerical controls that establish a reference for interpreting those profiles. In the archived runs, Gemini 3.1 Pro and Claude Opus 4.6 have equal medium-task success counts but differ in verification frequency and expression-family diversity. Examining tool use, error change, and reported confidence makes these distinctions visible without combining them into a single cognitive score.

The numerical reference is also informative. A fixed sine query sequence solves all 30 reconstructed archived tasks, showing that adaptive measurement selection is not necessary for recovery on this set. The model experiments therefore study how agents navigate a controlled investigation. The controls and identifiability analysis clarify how coefficient recovery depends on the inverse problem and its numerical implementation.

2. Task and Benchmark Construction

2.1. An Unknown Equation and a Measurement Budget

The hidden equation is

$$\begin{aligned} -(a(x)u'(x))' + b(x)u'(x) + c(x)u(x) &= f(x), \\ x \in (0,1), \qquad u(0) &= u(1) = 0. \end{aligned} \tag{1}$$

The diffusion, transport, reaction, and source fields a, b, c, f are represented as $q(x) = \sum_{j=0}^{N-1} \theta_{q,j} P_j(x)$, where P_j are shifted Legendre polynomials and the coefficients are sampled. A seed fixes the coefficient vectors and the numerically computed response u. The model can inspect u and its derivative but must recover the hidden fields. The reference prompt supplies the operator family and parameter count, while the numerical solver handles the basis representation.

[1]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: Joseph K. Miller <jkm314@mit.edu>.

Preprint. November 2, 2026.

Table 1. The agent's interface. Stored queries consume the query budget; the other tools remain subject to the turn limit.

Action	Role in the investigation
Evaluate	Inspect u and u' at chosen locations.
Query	Measure $\int f\varphi$; append a solver row.
Inspect terms	Examine diffusion, transport, and reaction weights.
Solve	Fit coefficients using the stored measurements.
Check	Compare a prediction with a new weak-form measurement.
Submit	Return the latest coefficient estimate for scoring.

"Learning" means parameter recovery within a session. In one spatial dimension the task is an ODE boundary-value problem representing an elliptic PDE setting, not a test of multidimensional PDE structure.

An experiment consists of choosing a test function φ that vanishes at both endpoints. Integration by parts gives

$$\int_0^1 au'\varphi' \,\mathrm{d}x + \int_0^1 bu'\varphi \,\mathrm{d}x + \int_0^1 cu\varphi \,\mathrm{d}x = \int_0^1 f\varphi \,\mathrm{d}x. \tag{2}$$

The query returns the source moment on the right. Internally, the tool uses u and the coefficient basis to assemble a row of the linear inverse problem. Accumulated measurements support least-squares estimates of the operator coefficients and a separate fit of the source. For example, $\varphi(x) = \sin(\pi x)$ probes one source moment; changing its frequency or shape changes the constraint imposed on the unknown equation.

This construction gives experimental choices a concrete mathematical role. Test functions that look different need not provide independent constraints, and an ill-conditioned set of measurements can fit well while recovering coefficients poorly. The question for the agent is what to do with the feedback it receives. The environment permits choosing more measurements, inspecting term contributions, resolving the system, and checking a prediction.

2.2. The Interaction Loop

Figure 1 shows the experimental loop; Table 1 summarizes the interface. The model can use polynomial, trigonometric, exponential, and localized expressions, subject to the parser's numerical and endpoint checks. The numerical solver handles coefficient recovery; the model supplies the experimental sequence.

A turn is one model response and may contain several free commands, but at most one stored query. A solve is permitted after $4N$ stored queries, where N is the number of basis coefficients per field. The reference prompt encourages repeated solving and checking, asks the model to explain its choices, and requests confidence reports after solves. It also provides a hint that coefficient recoverability depends on the relative contributions of the equation's terms and the chosen test functions. This assistance is part of the reference protocol; Appendix A.2 gives the wording and the limits of historical prompt provenance.

Table 2. Settings recorded in the scored archive. The full generator and numerical parameters are given in Appendix A.

	Easy	Medium	Hard
Basis coefficients per field N	2	3	4
Total hidden coefficients	8	12	16
Minimum queries before solving	8	12	16
Stored-query budget	12	18	24
Turn limit	36	54	72

The budget counts stored solver rows. A free consistency check also reveals a source moment, so two runs with the same query count may have acquired different amounts of information. We retain check and diagnostic counts as part of the behavioral profile rather than treating the stored-query count as the entire cost of an investigation.

2.3. A Ground-Truth Outcome

For field q, let θ_q be its true coefficient vector. The score is

$$E_q = \max_j |\widehat{\theta}_{q,j} - \theta_{q,j}|, \qquad E = \sum_{q\in\{a,b,c,f\}} E_q. \tag{3}$$

We use $E < 0.01$ as the primary success event and $E < 0.1$ as a threshold sensitivity check. This is error in the prescribed coefficient coordinates; it is distinct from a weak residual or a held-out-grid function error. The continuous errors remain available to the evaluator for studying changes within a run. Ground-truth coefficient errors are not tool feedback shown to the model.

3. Evaluation Protocol and Behavioral Measures

3.1. Shared Tasks and Archived Runs

The version-1 Kaggle archive contains 180 scored runs from six models on three settings and ten seeds per setting. A task is one generated coefficient tuple and its numerical solution; a run is one model's interaction with that task. All six models share the same 30 tasks. Each record contains the transcript, parsed actions, measurements, solve outputs, errors, and available confidence reports. We verify cell sizes, shared truths, finite scores, and agreement with the saved sweep summary. The run-level figures use locally retained records whose regenerated summaries match the saved archive analysis; file provenance and validation details are given in Appendix C.

All runs record the `standard` condition, called *enhanced*

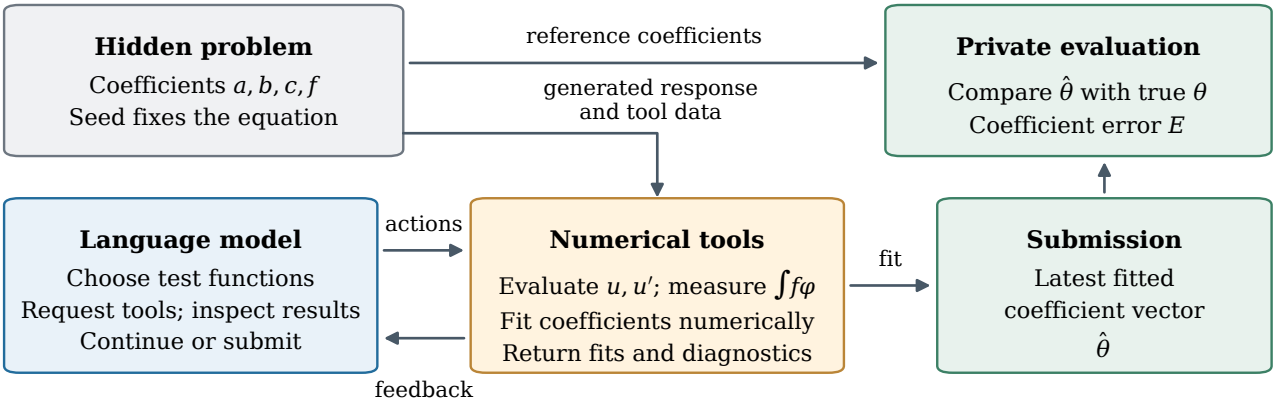


Figure 1. The agent chooses measurements and numerical tools fit the hidden coefficients. Submissions are scored against ground truth. Checks reveal extra source moments without using the stored-query budget.

in the Kaggle write-up. The archive has no baseline-prompt ablation, repeated model draws per task, or complete history of failed attempts and retries. Exact provider IDs and recorded settings are in Appendix C. The available reference implementation describes the intended assistance and interface, but the archive does not pin the full source and system prompt used for every run. Consequently, the analyses describe the retained scored interactions rather than end-to-end provider reliability.

Comparisons use the shared task as the statistical unit. We report Wilson 95% intervals for success proportions and paired task-level comparisons. With ten tasks per model and setting, these comparisons have limited resolution. Numerical controls use reconstructed tasks and a fixed sine policy on 300 fresh tasks; they make no new model calls and are kept separate from archived outcomes.

3.2. Behavioral Measures

We examine four observable aspects of the session, motivated by questions about planning, within-session improvement, diversity, and monitoring. The metrics retain distinct units and interpretations.

Tool use. Counts of solves, consistency checks, and term-integral calls describe how a model allocates interaction to fitting, verification, and inspecting the equation. We summarize each action type separately. Counts measure activity under a prompt that encourages iteration; they do not by themselves establish the quality of a plan.

Error change. For runs with at least two recorded solves, $R = E_{\text{first}}/E_{\text{last}}$ describes the net error reduction. All 159 repeated-solve runs have valid ratios; the 21 single-solve runs are reported separately. A large ratio can reflect a poor starting estimate, so we interpret improvement together with the endpoint. The metric measures within-session error change; strategic adaptation requires further evidence from the intervening actions.

Expression-family diversity. Normalized entropy $H = -\sum_i p_i \log_2(p_i)/\log_2 5$ summarizes the frequencies of five expression categories: polynomial, trigonometric, exponential, localized, and other. This is a heuristic classification of expression forms. Independence and usefulness are properties of the resulting measurement matrix, so entropy is not an information-gain measure.

Confidence and correctness. The reference prompt asks for a probability that each coefficient vector is "within 0.1 of the true value," without specifying a norm. We operationalize that event as maximum absolute coordinate error $E_q < 0.1$, for $q = a, b, c$. We focus on the operator coefficients: the separately fitted source f satisfies $E_f < 0.1$ at all 687 recorded solves, so including it would add an always-correct outcome class. Each report is matched to the most recent solve returned on a strictly earlier turn. Reliability curves pool reports within confidence bins. Run-balanced Brier scores and signed confidence-minus-correctness gaps provide complementary summaries; negative gaps mean average underconfidence for this event. Uncertainty estimates resample whole runs, retaining all reports and coefficient blocks within each run. Reporting stages vary across mod-

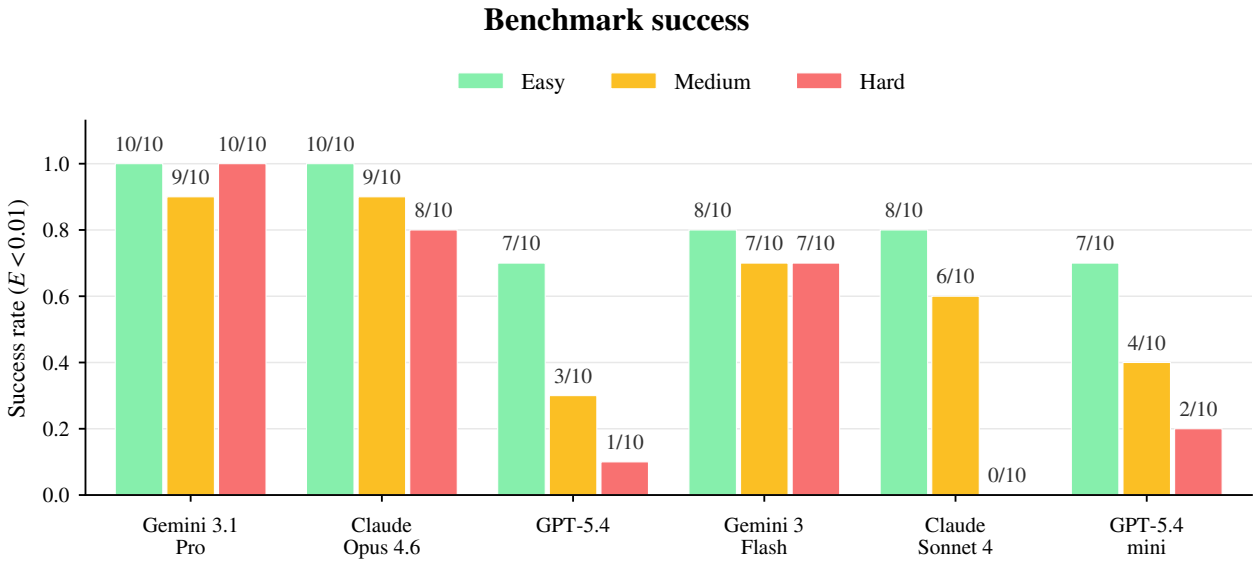


Figure 2. Coefficient recovery across models and difficulty settings. Each bar shows the fraction of ten tasks with final error $E < 0.01$; labels give the success counts. Wilson intervals and paired comparisons are in Appendix C.

els, so these summaries characterize confidence under the observed protocol rather than a matched test of intrinsic metacognitive ability.

4. Model Outcomes and Experimental Behavior

4.1. Coefficient Recovery

Figure 2 compares success across the six models and three settings. At $E < 0.01$, Gemini 3.1 Pro succeeds on 10/10, 9/10, and 10/10 easy, medium, and hard tasks; Claude Opus 4.6 achieves 10/10, 9/10, and 8/10. GPT-5.4 drops sharply: 7/10, 3/10, and 1/10. Success is not monotone across settings for every model, and the difficulty labels change several generator parameters together. With ten tasks per cell, even 10/10 has a Wilson 95% interval of approximately [0.72, 1.00]. Appendix C gives these intervals and paired comparisons; the released numerical summaries also report the relaxed threshold $E < 0.1$.

4.2. Tool Use

Models with similar success rates use the interface differently (Figure 3). Gemini 3.1 Pro and Claude Opus 4.6 both achieve 9/10 medium successes: eight tasks are solved by both, with one success unique to each model. Yet Pro makes an average of 2.3 consistency checks and 3.1 solves, compared with 7.6 checks and 4.0 solves for Opus.

More activity also coexists with poor recovery. On hard tasks, GPT-5.4 makes an average of 20.6 term-integral calls and 6.5 solves, compared with 0.9 and 3.0 for Gemini 3.1 Pro, while their success counts are 1/10 and 10/10. These counts describe how models allocate their interaction budget; they do not establish a causal effect of checking or diagnostic activity.

4.3. Within-Session Improvement

The improvement profiles vary substantially across models (Figure 4). Claude Sonnet 4 reduces error by a median factor of approximately 193 on medium tasks, yet succeeds on only 6/10. Its four failed runs all improve, by factors of 3.1–24.4, while still finishing above the success threshold. Error reduction and final accuracy answer different questions.

Means and medians also tell different parts of the story. Gemini 3.1 Pro has 10/10 hard successes and a median improvement ratio of 1.0, but a mean of approximately 833, driven by a few large reductions. The individual points show this variation directly. Ratios compare recorded solves; the submitted errors agree with last-solve errors to within 1.34×10^{-8}, with no change in success classification. Appendix D gives the sample and metric definitions.

4.4. Query Diversity

Successful models can use quite different mixtures of test functions (Figure 5). On medium tasks, Gemini 3.1 Pro and Claude Opus 4.6 have median family entropies of 0.38 and 0.67, despite equal success counts. On hard tasks, Pro's median entropy falls to 0.24 while it solves all ten tasks.

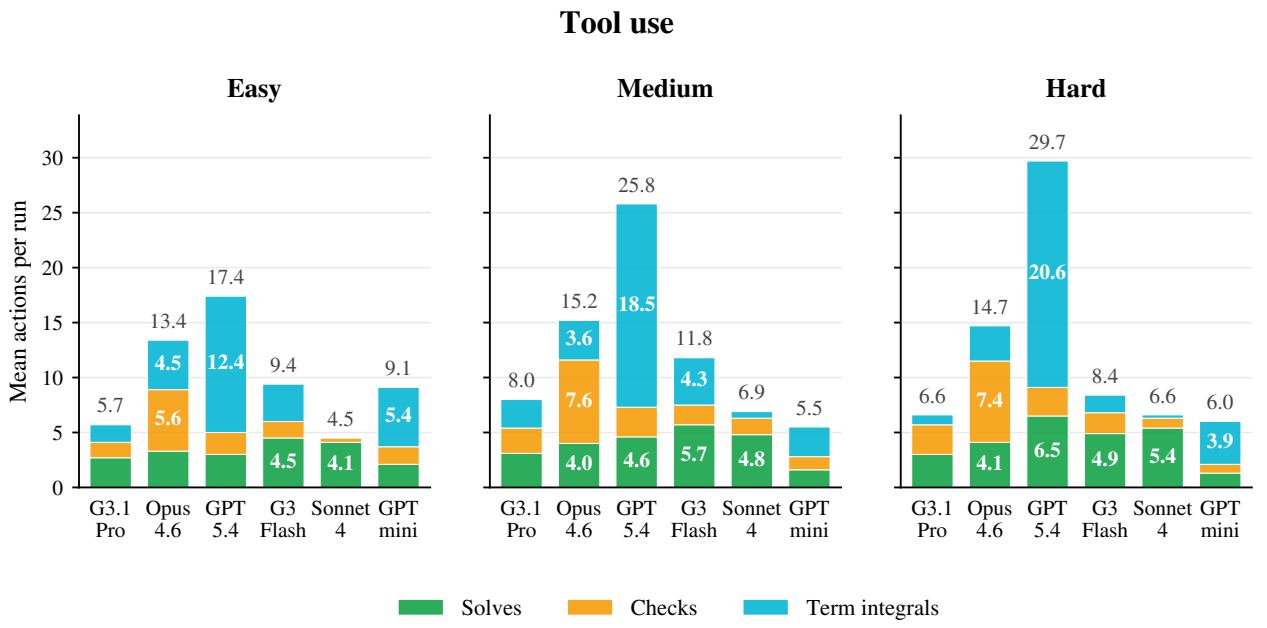


Figure 3. How models use the tools. Stacked bars show mean solves, checks, and term-integral calls per run, with ten runs per model and setting. Mean counts add to the mean combined count of these three action types. Model order follows Figure 2; more calls need not produce a better answer.

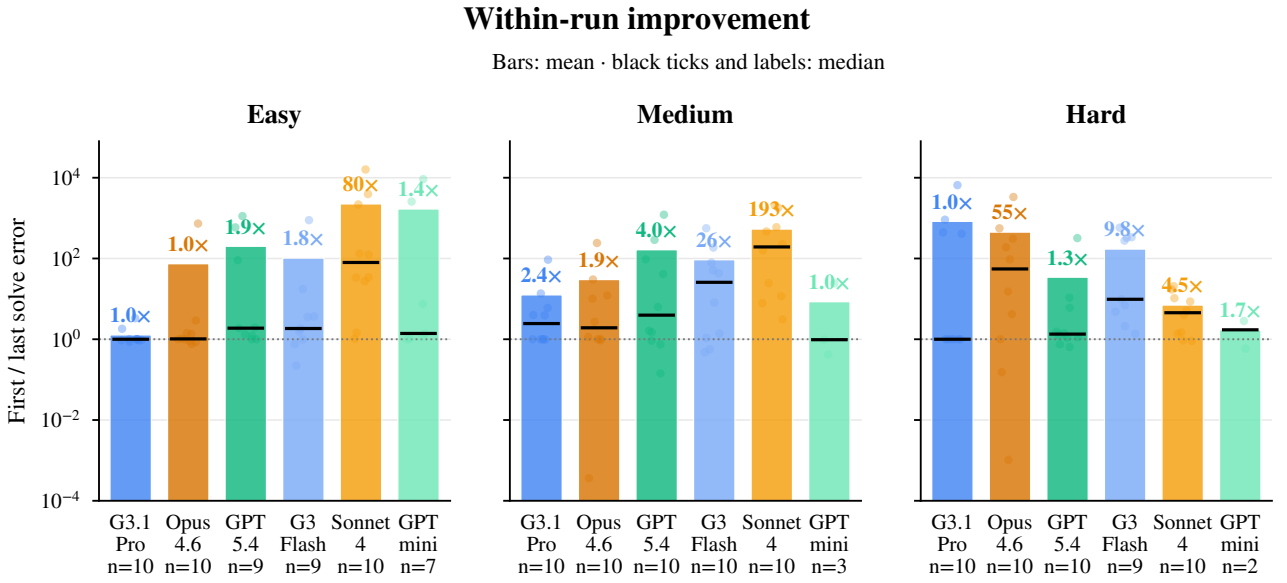


Figure 4. Error reduction between the first and last recorded solves. Bars show means, black ticks and labels show medians, and faint dots show individual runs. The logarithmic scale accommodates large differences. Sample sizes below the model names count repeated-solve runs; the 21 single-solve runs have no ratio.

A narrow family can therefore be effective. Entropy describes the form of the chosen expressions; the rank and conditioning of the resulting measurements determine what they reveal about the equation.

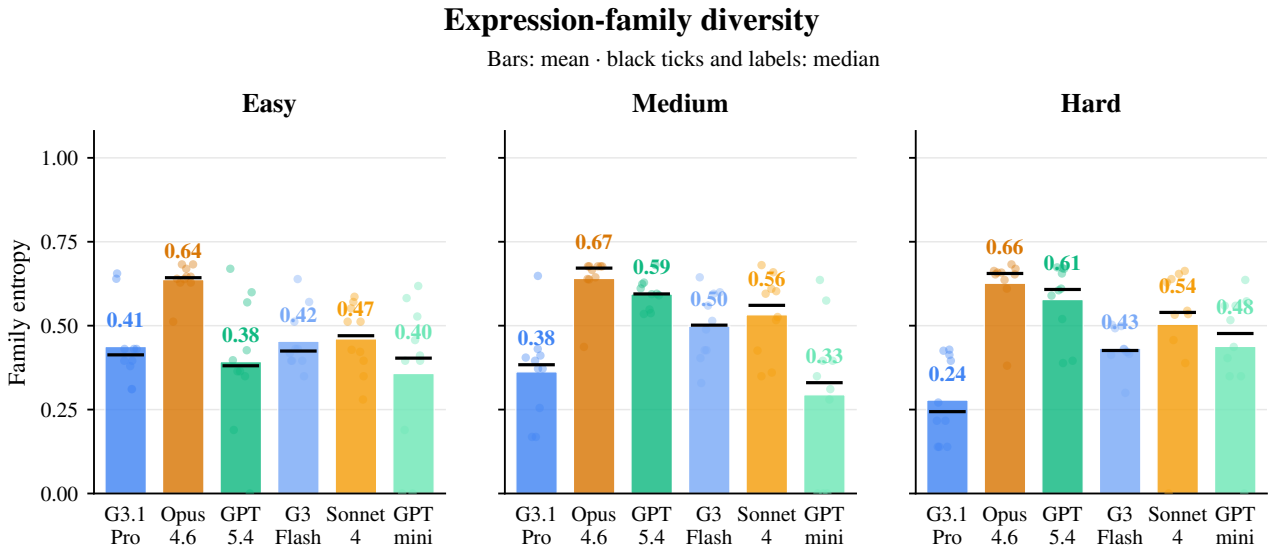


Figure 5. Diversity of test-function families. Bars show mean normalized entropy, black ticks and labels show medians, and faint dots show the ten individual runs in each cell. Higher entropy means a broader mix of expression categories, not necessarily more informative measurements.

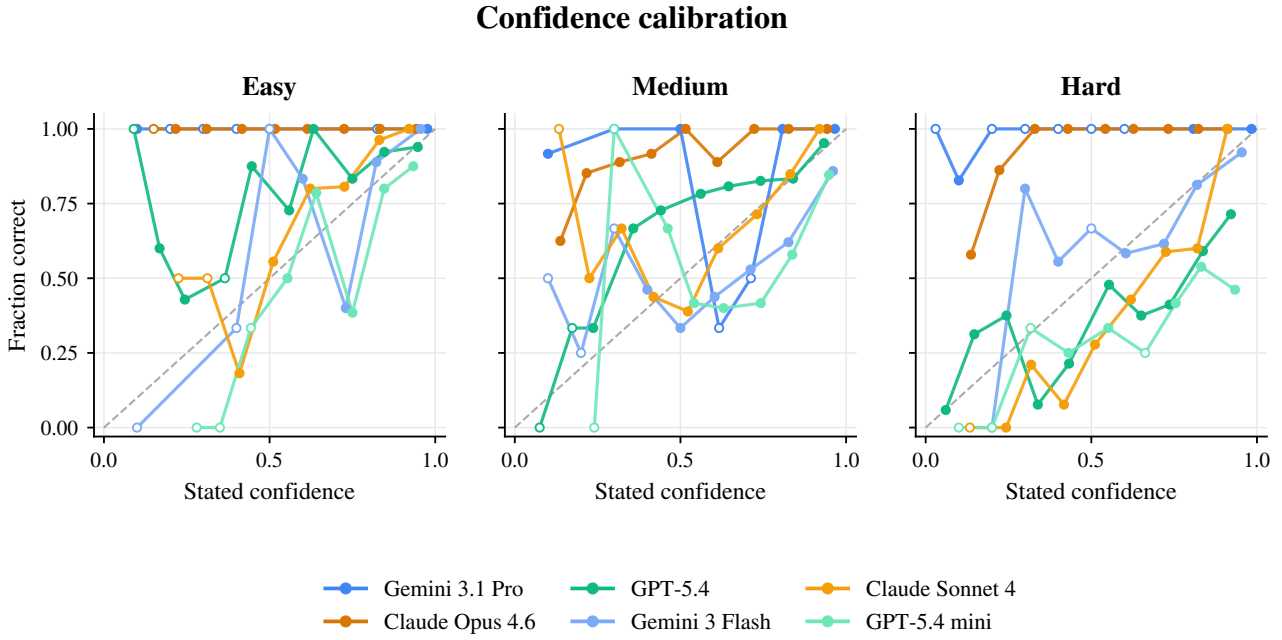


Figure 6. Confidence calibration. Each point compares mean stated confidence with observed coefficient correctness within a confidence bin; the diagonal indicates agreement. Lines connect populated bins as visual guides; hollow points contain fewer than five coefficient observations. Curves pool reports for a, b, c and weight reports equally; Appendix D gives the matching rule and complementary run-balanced summaries.

4.5. Confidence Calibration

Figure 6 compares stated confidence with the accuracy of the estimates available at the time. Points above the diagonal indicate that coefficients are correct more often than confidence suggests; points below it indicate the reverse. These curves concern coefficient correctness $E_q < 0.1$, rather than final-run success $E < 0.01$.

Table 3. Fixed sine on 100 fresh seeds per setting. All planned tasks remain in the denominator. Parentheses give Wilson 95% intervals for the minimum-budget rate; zero-source counts are descriptive.

Setting	$4N$ successes	$6N$ successes	$f = 0$
Easy	92/100 (0.85, 0.96)	92/100	7
Medium	91/100 (0.84, 0.95)	85/100	3
Hard	98/100 (0.93, 0.99)	93/100	0

The complementary run-balanced summaries show that Gemini 3.1 Pro and Claude Opus 4.6 are underconfident on average in all three settings. On hard tasks, mean confidence falls below coefficient correctness by 28.8 and 42.6 percentage points, respectively; GPT-5.4 mini is 32.2 points above. Reporting frequency and stage differ across models, so these are descriptions of the observed sessions. Appendix D gives Brier scores, signed gaps, and support details.

5. Numerical Controls and Interpretation

The task's mathematical structure gives a way to investigate the behavioral patterns. A fixed sine policy, $\varphi_k(x) = \sin(k\pi x)$, selects measurements without feedback. On the 30 reconstructed archived tasks it achieves 30/30 success at the first permitted solve, using 8, 12, or 16 queries depending on the setting. Boundary-Legendre tests achieve 29/30 at full budget. Thus a single well-chosen family can be effective; broad expression diversity and repeated revision are not prerequisites for solving these instances.

We also evaluate the sine policy on 100 fresh seeds per setting. At the minimum budget it succeeds on 92/100 easy, 91/100 medium, and 98/100 hard tasks. All planned tasks remain in the denominator, including ten with zero forcing and zero response, for which the operator coefficients cannot be recovered. At full budget the counts are 92/100, 85/100, and 93/100: adding rows can worsen coefficient recovery in the finite-precision system. These are post-design validation results for the numerical policy, not fresh model runs.

Two mathematical checks help explain these results. First, distinct operators can sometimes generate exactly the same response and source. We construct an admissible polynomial family with this property; it is an example, not a classification of the 30 archived tasks. Second, small singular values can amplify the forward solver's and quadrature's consistency errors. A small residual then need not imply an accurate coefficient vector. The observation model, proof, reconstruction checks, and full controls appear in Appendices A–B.

These controls establish the appropriate interpretation of the model study. The environment exposes choices in an investigation and their consequences, while the current task set does not demonstrate an advantage from adaptive language-model measurement selection. Whether adaptation helps on fresh tasks under matched information budgets remains an experimental question.

6. Related Work

Interactive scientific-agent environments such as DiscoveryWorld, BoxingGym, and SciGym study sequential observation and experimentation (Jansen et al., 2024; Gandhi et al., 2025; Duan et al., 2025). Our setting contributes a small inverse problem in which measurement matrices, coefficient truth, and numerical sensitivity can be examined directly. The cognitive framework of Burnell et al. (2026) motivates the original behavioral questions, while calibration analysis distinguishes stated confidence from empirical correctness (Guo et al., 2017).

LLM-SR and LLM-SRBench address scientific equation discovery (Shojaee et al., 2025a;b); here the operator family is fixed and a numerical tool performs coefficient fitting. Weak SINDy also uses integral formulations for identification (Messenger & Bortz, 2021). Experimental design provides a reference for selecting informative observations (MacKay, 1992; Alexanderian, 2021). Identifiability results depend on the observation operator (Egger et al., 2014; Browning et al., 2024; Salmaniw & Browning, 2025); we therefore analyze the particular fixed-solution observation model used here.

7. Discussion and Conclusion

The benchmark records an investigation as well as an answer. Tool calls show how models allocate activity; paired error observations connect revision to endpoint accuracy; confidence reports can be checked against the estimates available at the time. These records allow distinct questions to be asked of runs with equal success counts, without treating the resulting metrics as interchangeable measures of cognition.

The numerical controls delimit that interpretation. A fixed measurement design already solves the reconstructed archived tasks, and fresh tasks expose both unidentifiable cases and numerical failures. The present evidence covers 30 shared archived tasks, bundled difficulty settings, and one retained trajectory per model–task pair. Incomplete attempt histories and source provenance limit claims about provider reliability and exact historical reconstruction.

A prospective study could compare fixed, randomized, and adaptive policies under one budget accounting for every truth-dependent observation. Retaining all attempts, exact

prompts, model settings, and internal numerical matrices would support closer replay. Requesting confidence at common stages would make monitoring comparisons more interpretable. Additional source or boundary conditions could test whether agents choose observations that resolve ambiguity. The current environment and records provide a concrete starting point for these experiments: scientific reasoning studied through observable measurement choices, feedback, and revision.

Impact Statement

This work develops a controlled environment for studying scientific-agent behavior. Its contribution is methodological: connecting observable decisions with mathematical outcomes and numerical controls. The results concern this specific task and protocol and do not establish readiness for autonomous scientific deployment.

Reproducibility

The benchmark repository is https://github.com/Hydrodynamical/PDE_Learning. Reference implementation and analysis sources are pinned to commit 9eb8b1d4393f. The accompanying source package includes the plotting scripts and saved summaries used for this manuscript. Appendix C specifies data provenance, settings, and reconstruction limits. All reported offline controls and figure analyses require no new model calls.

References

Alexanderian, A. Optimal experimental design for infinite-dimensional bayesian inverse problems governed by PDEs: A review. *Inverse Problems*, 37(4):043001, 2021. doi: 10.1088/1361-6420/abe10c. URL https://doi.org/10.1088/1361-6420/abe10c.

Browning, A. P., Taşcă, M., Falcó, C., and Baker, R. E. Structural identifiability analysis of linear reaction–advection–diffusion processes in mathematical biology. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 480(2286):20230911, 2024. doi: 10.1098/rspa.2023.0911. URL https://doi.org/10.1098/rspa.2023.0911.

Burnell, R., Yamamori, Y., Firat, O., Olszewska, K., Hughes-Fitt, S., Kelly, O., Galatzer-Levy, I. R., Morris, M. R., Dafoe, A., Snyder, A. M., Goodman, N. D., Botvinick, M., and Legg, S. Measuring progress toward AGI: A cognitive framework. *arXiv preprint arXiv:2605.28405*, 2026. doi: 10.48550/arXiv.2605.28405. URL https://arxiv.org/abs/2605.28405.

Duan, H., Lu, S., Harrigan, C. F., Desai, N., Lu, J., Koziarski, M., Cotta, L., and Maddison, C. J. Measuring scientific capabilities of language models with a systems biology dry lab. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3733. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/a23760ba51036d530a2656c3835f826c-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Egger, H., Pietschmann, J.-F., and Schlottbom, M. Simultaneous identification of diffusion and absorption coefficients in a quasilinear elliptic problem. *Inverse Problems*, 30(3):035009, 2014. doi: 10.1088/0266-5611/30/3/035009. URL https://doi.org/10.1088/0266-5611/30/3/035009.

Gandhi, K., Li, M. Y., Goodyear, L., Bhatia, A., Li, L., Bhaskar, A., Zaman, M., and Goodman, N. D. BoxingGym: Benchmarking progress in automated experimental design and model discovery. *arXiv preprint arXiv:2501.01540*, 2025. doi: 10.48550/arXiv.2501.01540. URL https://arxiv.org/abs/2501.01540. Version 2, 14 October 2025.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Jansen, P., Côté, M.-A., Khot, T., Bransom, E., Mishra, B. D., Majumder, B. P., Tafjord, O., and Clark, P. DiscoveryWorld: A virtual environment for developing and evaluating automated scientific discovery agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=cDYqckEt6d. Datasets and Benchmarks Track.

MacKay, D. J. C. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992. doi: 10.1162/neco.1992.4.4.590. URL https://doi.org/10.1162/neco.1992.4.4.590.

Messenger, D. A. and Bortz, D. M. Weak SINDy for partial differential equations. *Journal of Computational Physics*, 443:110525, 2021. doi: 10.1016/j.jcp.2021.110525. URL https://doi.org/10.1016/j.jcp.2021.110525.

Miller, J. K. Can language models learn a PDE? Kaggle Measuring Progress Toward AGI competition writeup, April 2026. URL https://www.kaggle.com/competitions/kaggle-measuring-agi/writeups/can-language-models-learn-a-pde

#3442067. Published 16 April 2026; released under CC0.

Salmaniw, Y. and Browning, A. P. Structural identifiability of linear-in-parameter parabolic PDEs through auxiliary elliptic operators. *Journal of Mathematical Biology*, 91: 4, 2025. doi: 10.1007/s00285-025-02225-w. URL https://link.springer.com/article/10.1007/s00285-025-02225-w.

Shojaee, P., Meidani, K., Gupta, S., Barati Farimani, A., and Reddy, C. K. LLM-SR: Scientific equation discovery via programming with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=m2nmp8P5in.

Shojaee, P., Nguyen, N.-H., Meidani, K., Barati Farimani, A., Doan, K. D., and Reddy, C. K. LLM-SRBench: A new benchmark for scientific equation discovery with large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55325–55359. PMLR, 2025b. URL https://proceedings.mlr.press/v267/shojaee25a.html.

A. Mathematical Model and Identifiability

A.1. Observation Matrices and Coefficient Recovery

Equations (1)–(2) define the forward problem and the observed weak identity. The coefficient basis uses ordinary shifted Legendre polynomials on $[0,1]$, for example $P_0(x) = 1$ and $P_1(x) = 2x - 1$. They are orthogonal but not orthonormal. At the continuum level, take smooth coefficients with positive diffusion, an observed solution $u \in H_0^1(0,1)$, and tests $\varphi_k \in H_0^1(0,1)$. The zero endpoint values remove the boundary term in integration by parts, so the weak measurements require only first derivatives of u.

Each test supplies one operator constraint. Define

$$D_{kj} = \int P_j u' \varphi_k' \,\mathrm{d}x, \qquad B_{kj} = \int P_j u' \varphi_k \,\mathrm{d}x,$$
$$C_{kj} = \int P_j u \varphi_k \,\mathrm{d}x, \qquad F_{kj} = \int P_j \varphi_k \,\mathrm{d}x, \tag{4}$$

with integrals over $(0,1)$. After K queries, let $A = [D \mid B \mid C] \in \mathbb{R}^{K \times 3N}$, $\theta = (\theta_a^\top, \theta_b^\top, \theta_c^\top)^\top$, and $r_k = \int f \varphi_k \,\mathrm{d}x$. The tool separately fits

$$\widehat{\theta} \in \arg\min_{\theta} \|A\theta - r\|_2, \qquad \widehat{\theta}_f \in \arg\min_{\theta_f} \|F\theta_f - r\|_2. \tag{5}$$

For a nonunique fit, it selects the minimum Euclidean norm solution under its numerical singular-value cutoff. Both fits use the measured vector r, so operator recovery does not require a preceding source estimate. Source moments provide information in addition to the observed solution.

The tool constructs each matrix row internally. The agent receives the source moment after a query, then coefficient estimates and selected diagnostics when it requests a solve. The full joint matrices and spectra in the analysis are reconstructed from archived expressions. The forward solution is computed by finite differences, rather than expanded in the chosen test functions. The protocol requires $K \geq 4N$ before a solve, although the operator fit has $3N$ unknowns; this rule does not guarantee rank or uniqueness.

A.2. Reference Prompt and Numerical Implementation

The reference implementation supplies the equation, weak identity, and parameter count, but does not name the Legendre basis. Table 1 lists the available operations. A response may contain several free commands and at most one stored query; submission cannot share a response with a query. The reference rules explicitly encourage using the free tools:

> `COMPUTE: solve` is free and unlimited—call it after every few queries to track how your estimates evolve. This is good experimental practice. `COMPUTE: check` is free—use it to verify your estimates against new test functions.

The enhanced initial message asks the model to explain its choices and adds:

> Note: in elliptic inverse problems of this type, the three coefficient functions $a(x), b(x), c(x)$ are not equally identifiable from weak-form data. The difficulty of recovering each coefficient depends on the magnitude of its contribution to the weak-form equations relative to the others—and this varies with the solution $u(x)$ and the test functions you choose.

The confidence instruction is:

> After each `COMPUTE: solve`, report your confidence (0–100%) that each coefficient vector (a, b, c, f) is within 0.1 of the true value.

It prescribes a reporting format and one sentence about uncertainty, but no coordinate norm. Our correctness event $E_q < 0.1$ is the analysis's explicit operationalization. These excerpts describe the pinned reference implementation; the logs do not establish the exact full system prompt and source revision for every historical provider call. The accompanying `reference_prompt_components.txt` supplies the base prompts for all three settings, protocol, confidence instruction, and enhanced initial message, extracted from the available source. Optional runtime overrides and turn reminders are identified as outside that reference extract.

The expression parser accepts polynomial, trigonometric, exponential, and localized forms subject to sampled-value and endpoint checks. Numerical acceptance does not certify membership in the continuum test space; predefined controls use smooth functions. There is no common amplitude normalization. Free consistency checks expose source moments without adding rows to the fitted system, while solution evaluations and term-integral diagnostics are also free.

Table 4. Additional generator and discretization parameters for reconstructed archived tasks. Basis dimensions and interaction budgets appear in Table 2. Sparsity is the generator's zeroing parameter.

	Easy	Medium	Hard
Generator sparsity	.3	.2	.1
Generator scale	.5	1.0	1.5
Diffusion parameter $a_{\min}$	1.0	.5	.3
Solution grid points	2001	2001	4001

The generator checks diffusion positivity on 500 spatial points; $a_{\min}$ controls the constant diffusion coefficient, not a certified pointwise lower bound. Reaction can have either sign, so diffusion positivity alone does not guarantee a unique continuum forward solution. The evaluation takes the computed solution as data. Dimension, scale, sparsity, and resolution vary together, so the difficulty labels describe bundled settings.

The reference implementation uses second-order finite differences for the solution, a fourth-order interior derivative with boundary stencils, and composite Simpson quadrature on the solution grid. Each stored matrix entry and right-hand-side value is rounded to 12 decimal places; the fit is unweighted least squares. The model sees coefficients rounded to eight decimals, while final scoring uses the internally retained unrounded estimate. Symbolic test functions are differentiated with a fourth-order centered stencil and step 10^{-6}, independently of solution-grid spacing. These choices affect recovery when singular values are small.

A.3. Coefficient Error, Uniqueness, and Stability

Equation (3) defines the coefficient-coordinate error. For each field, $\|\widehat{q} - q\|_{L^\infty(0,1)} \le NE_q$, since $|P_j| \le 1$. Thus a small coefficient error controls field error, but the same numerical threshold for these two quantities is not equivalent.

Three questions should be separated. *Structural identifiability* asks whether different coefficient triples can produce the same u, f, even with all admissible weak measurements. *Measurement selection* asks whether a finite set of tests distinguishes triples in the chosen coefficient space. *Numerical stability* asks how small data and discretization errors affect the recovered coefficients. Full column rank of an exact moment matrix formed from the continuum solution would certify uniqueness in the operator coefficient space. Full rank after numerical approximation and rounding need not do so. Separate full-rank blocks D, B, C do not exclude dependencies between blocks.

To see the role of numerical error, write A_h, r_h for the assembled system and θ_* for the true operator coefficients. Its consistency defect is $e_h = A_h\theta_* - r_h$. The forward solve, derivative approximation, quadrature, and rounding need not preserve the weak identity exactly, so this defect can be nonzero even at the truth. If A_h has full column rank, exact least squares for that assembled system gives

$$\begin{aligned} \widehat{\theta} - \theta_* &= -A_h^\dagger e_h, \\ \|\widehat{\theta} - \theta_*\|_2 &\le \frac{\|e_h\|_2}{\sigma_{\min}(A_h)}. \end{aligned} \tag{6}$$

Here $A_h^\dagger$ is the pseudoinverse. The bound concerns the operator fit; the score also includes the separately fitted source. A small fitted weak residual $\|A_h\widehat{\theta} - r_h\|_2$ need not imply a small coefficient error: a small singular value can amplify a small consistency defect. The forward finite-difference residual, fitted weak residual, and coefficient score therefore answer different questions.

Let $\mathcal{P}_d$ denote polynomials of degree at most d. The following example shows an ambiguity that persists even before discretization.

Proposition 1 (An admissible nonidentifiable family)**.** *Let $u(x) = x(1-x)$, $N \ge 2$, and $T(\delta a, \delta b, \delta c) = -(\delta a u')' + \delta b u' + \delta c u$ with domain $\mathcal{P}^3_{N-1}$. Then T has rank $N+2$ and nullity $2N-2$. Every element of its kernel leaves all continuum weak-form measurements unchanged at fixed u, f.*

Even arbitrarily many exact weak measurements cannot distinguish operators separated by these kernel directions. Appendix A.4 gives the proof and examples with positive diffusion and nonnegative reaction. This quadratic- solution family is an admissible counterexample, not a classification of the 30 sampled instances. There is also a positive-probability zero-excitation event in the generator: each forcing coefficient is zeroed independently with probability s, giving $\Pr(f \equiv 0) = s^N$, or 0.09, 0.008, and 0.0001 in the three settings. When $f \equiv 0$, under homogeneous boundary data the native forward solve returns $u = 0$ when its matrix is nonsingular. Then every operator design row vanishes, irrespective of the test function, so no measurement policy can identify a, b, c from these observations. The nonzero-source construction in Proposition 1 is a separate ambiguity mechanism.

A.4. Structural Nonuniqueness: Proof and Examples

Proof of Proposition 1. Let $\mathcal{P}_d$ denote polynomials of degree at most d. For $u = x(1-x)$, all three terms in T lie in $\mathcal{P}_{N+1}$, so the rank is at most $N+2$. The reaction component contains $u\mathcal{P}_{N-1}$. Taking constant δa gives a nonzero constant because $-u'' = 2$, and taking constant δb gives $u' = 1 - 2x$. These span $\mathcal{P}_1$. Polynomial division by the quadratic u gives $\mathcal{P}_{N+1} = u\mathcal{P}_{N-1} \oplus \mathcal{P}_1$, proving surjectivity and rank $N+2$. The domain has dimension $3N$, so rank–nullity gives $\dim\ker T = 2N - 2$. Multiplication by any admissible test function and integration by parts shows that a kernel element changes no weak measurement. □

For $N = 2$, $u = x(1-x)$ and $f = 3$, the operators

$$a = \tfrac{3}{2} + \varepsilon, \qquad b = \varepsilon(4x - 2), \qquad c = -8\varepsilon$$

all satisfy the same equation, since $2\varepsilon + \varepsilon(4x-2)(1-2x) - 8\varepsilon x(1-x) = 0$. For example $\varepsilon = -1/8$ preserves positive diffusion and gives nonnegative reaction. For $N \geq 3$, the base choice $a = 1, b = 0, c = 1, f = 2 + u$ belongs to the class. Sufficiently small perturbations in every kernel direction preserve positive diffusion and reaction, giving a relatively open $(2N-2)$-dimensional family in the affine kernel of admissible coefficient tuples with identical observations. Neither construction is asserted to occur among the 30 sampled tasks. The audit script also verifies ranks over exact rational arithmetic for $N = 2, \ldots, 8$; these checks supplement the proof rather than replace it.

B. Numerical Controls and Reconstruction

B.1. Numerical Controls and Replay

For the controls, we regenerate all 30 tasks from their recorded seeds, verify exact equality with saved ground-truth coefficient vectors, and evaluate non-language-model query policies under the same accepted-query limits and supplied query/solve interface. The policies use no hidden coefficients, solution values, measurement responses, or diagnostics to select expressions. Fixed sine uses the Dirichlet Laplacian eigenfunctions $\varphi_k = \sin(k\pi x)$, $k = 1, \ldots, K_{\max}$, where $K_{\max} = 6N$ is the stored-query budget. Boundary Legendre multiplies a polynomial test by the endpoint-vanishing factor $x(1-x)$, giving $x(1-x)P_{k-1}(x)$. The implementation uses factored expressions for numerical stability. Random mixed draws uniformly from five admissible families, using ten independently seeded query lists per setting, reused across the ten PDE seeds. Its fixed parameter ranges and exact expressions are released in the experiment output. All controls solve at every query count from $4N$ through $6N$; additional solves do not alter their predefined query lists. These controls test whether simple nonadaptive policies suffice on these instances; they do not reproduce the models' free-action or stopping decisions.

For the numerical audit, we replay the archived query expressions, compare the returned source moments with the logs, and reconstruct joint matrix spectra. We report both raw and unit-column-norm spectra, with explicit relative singular value cutoffs rather than a binary rank label alone. We additionally multiply each reconstructed row and its source moment by a common amplitude before 12-decimal rounding, leaving the query sets fixed and refitting all stored rows. This audit scores 8-decimal diagnostic estimates, unlike unrounded primary final scores. In exact arithmetic this common scaling leaves least squares unchanged; its effect here tests the absolute quantization rule. It is a post hoc numerical sensitivity analysis, rather than an optimized query policy or a model rerun.

We also refine the forward and quadrature grids together by factors two and four. Original-resolution primary controls use the native dense solver. Refinements use an independently assembled banded solver for the same finite- difference equations; we validate its solution and score sensitivity against native solves on all 30 tasks. Refinement changes the numerical environment, rather than adding independent tasks or measuring observation-noise robustness.

Table 5. New offline controls at the original resolution and full stored-query budget. Deterministic policies have ten tasks per difficulty. Random rows summarize ten query-list replicates on the same ten tasks; the denominator 100 is not 100 independent PDEs.

Policy	Setting	Successes	Median E
Fixed sine	Easy	10/10	9.5e-05
Fixed sine	Medium	10/10	0.0013
Fixed sine	Hard	10/10	0.00057
Boundary Legendre	Easy	10/10	3.4e-05
Boundary Legendre	Medium	9/10	0.0014
Boundary Legendre	Hard	10/10	0.00057
Random mixed	Easy	100/100	6.4e-05
Random mixed	Medium	92/100	0.0016
Random mixed	Hard	85/100	0.00097

To test specificity to the archived seeds, we additionally evaluate the fixed sine policy on 100 fresh seeds (1001–1100) in each difficulty, with the original generator and grid settings. Every planned task is retained, including zero forcing and numerical failures; there are no retries or filters. This is validation after design selection, not a preregistered comparison or a new language-model experiment. We report outcomes on all 100 tasks per setting, with Wilson intervals and descriptive zero/nonzero-source breakdowns. Reused seed labels across difficulties are not pooled into independent sampling units.

B.2. Simple Nonadaptive Queries Suffice on These Tasks

The fixed sine policy reaches $E < 0.01$ on all ten tasks in each difficulty, already at the minimum $4N$ stored queries (8, 12, and 16). It retains 30/30 success at the full $6N$ budget. Boundary Legendre succeeds on 29/30 at full budget. Random mixed succeeds on 100/100 easy, 92/100 medium, and 85/100 hard policy–task combinations (Table 5); these reuse ten PDE instances per setting and are not 100 independent tasks. Across random query-list replicates, hard success ranges from 6/10 to 10/10. Figure 7 shows recovery at every permitted solve count.

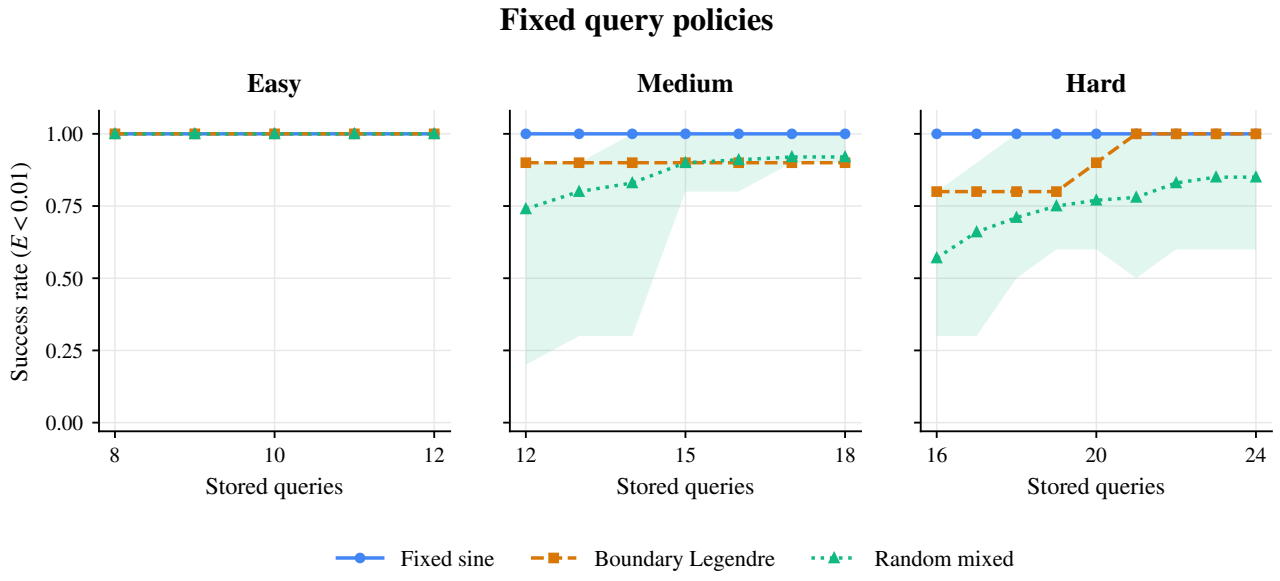


Figure 7. New controls on the 30 reconstructed tasks at the original grid resolution. Fixed sine and boundary Legendre use one predefined query list per setting. Random mixed averages ten independently seeded lists, each reused on the same ten tasks; shading is the minimum–maximum success fraction across lists, not a confidence interval. No query policy uses feedback or hidden truth.

These controls show that the archived tasks can be solved by a simple nonadaptive design with the supplied numerical tools. They do not show a statistically established advantage over the best archived model, nor isolate reasoning from parsing and stopping behavior. Because exact historical internal matrices were not retained, we compare newly reconstructed controls

with archived outcomes transparently rather than claiming a prospective provider comparison. The evidence does not support a claim that language-model adaptation is necessary for recovery on this task set.

Grid refinement improves numerical errors: for fixed sine at full budget, median E changes from 9.47×10^{-5} to 2.18×10^{-6} on easy, 1.32×10^{-3} to 2.82×10^{-5} on medium, and 5.65×10^{-4} to 1.56×10^{-5} on hard when the grid is refined fourfold. At twofold refinement, random mixed reaches 100/100 hard successes and boundary Legendre reaches 10/10 medium successes. The original-grid native and banded solves agree on both success thresholds at every checkpoint across all 360 paired policy–task runs. Thus the banded implementation does not change these threshold comparisons, although individual numerical errors differ.

B.3. Fresh Seeds Expose the Limits of the Archive Result

Table 3 validates the fixed sine policy on all 300 fresh tasks. At the minimum query budget, success is 92/100 easy, 91/100 medium, and 98/100 hard. At full budget, it is 92/100, 85/100, and 93/100, respectively. All 300 executions complete without errors; no task is retried or excluded. Seven easy and three medium tasks have zero forcing, exactly zero solutions, and joint operator rank zero at both budgets. All ten fail, as predicted by the zero-excitation argument. Excluding them only for a descriptive breakdown, full-budget recovery is 92/93 easy, 85/97 medium, and 93/100 hard.

Thus the archived 30/30 result does not establish perfect generator-wide recovery. Nonzero-source failures also remain, and increasing the stored-query budget can reduce the number of coefficient-threshold successes in the same fixed policy. More rows do not guarantee lower coefficient error in an inconsistent finite-precision system. This validation evaluates one selected numerical policy; it does not determine whether adaptive policies improve recovery on fresh tasks. It does establish why a benchmark needs both simple controls and a stated treatment of unidentifiable instances.

B.4. What Numerical Reconstruction Can Establish

All 30 ground-truth coefficient vectors regenerate exactly, and the reconstructed reference implementation reproduces all 2,874 archived query source moments exactly at the stored precision. Source-moment agreement does not test the historical forward solution or unretained operator matrix: the source moment depends only on f, the test function, and quadrature. It does not establish exact historical solve replay: across 687 solves, the median maximum deviation from the logged 8-decimal coefficient vectors is 2.55×10^{-5}, and the largest is 3.50. The archive lacks historical internal matrices and sufficient environment provenance to localize every discrepancy. New controls and spectral diagnostics must therefore be identified as reconstructions.

All 180 reconstructed final stored-query matrices have full column rank at NumPy's default floating-point tolerance. After unit-column scaling, however, the median $\sigma_{\min}/\sigma_{\max}$ is 2.26×10^{-7}, 5.53×10^{-10}, and 1.80×10^{-10} for easy, medium, and hard runs. At a declared relative cutoff of 10^{-8}, 13/60, 42/60, and 37/60 matrices are rank deficient. These are diagnostics of limited numerical resolution, not continuum nonuniqueness certificates for the sampled instances.

Common-amplitude scaling changes the reconstruction's $E < 0.01$ counts (Figure 8). At amplitudes $10^{-3}, 1, 10^{3}$, the counts are respectively 24/50/53 of 60 easy runs, 11/36/48 medium runs, and 19/28/33 hard runs. No additional test functions are selected: the exact-arithmetic measurements differ only by a common nonzero factor, while their relative quantization precision changes. The changed counts show that absolute rounding and ill conditioning can alter a success event even when the underlying span of measurements is unchanged. They do not establish that larger amplitude universally improves recovery.

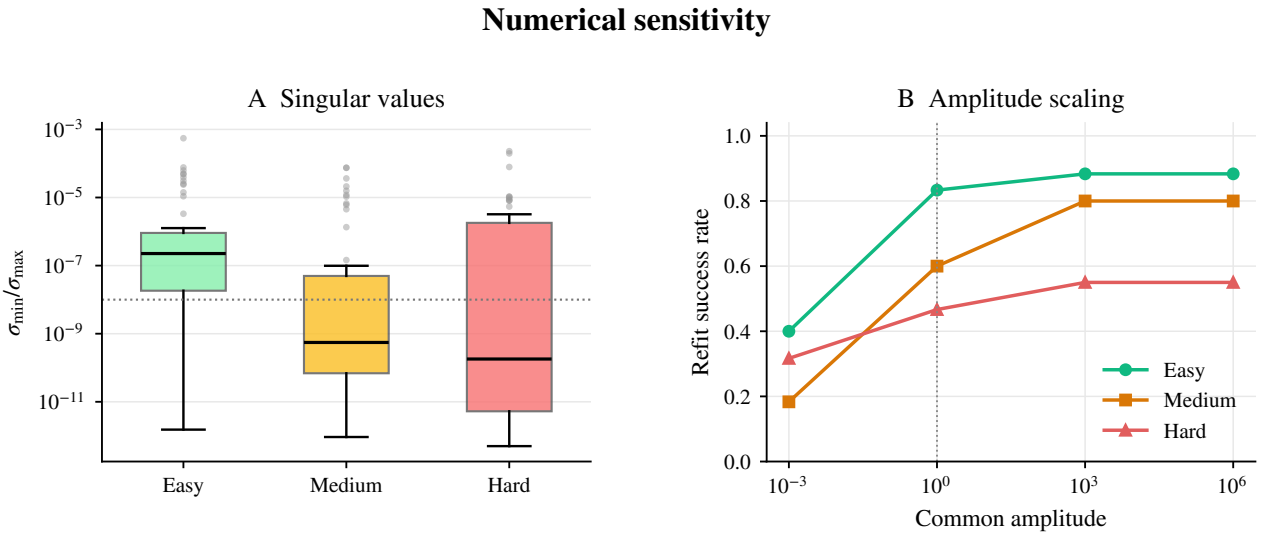


Figure 8. Numerical audit of reconstructed archived measurements. Left: unit-column-norm smallest-to-largest singular-value ratios across 60 query designs per setting. Boxes span the interquartile range, central lines mark medians, and whiskers reach the most extreme values within 1.5 interquartile ranges; points beyond the whiskers are outliers. The dotted line is a diagnostic cutoff. Right: strict-threshold success after common row/RHS scaling before absolute 12-decimal rounding, using all stored queries and fixed archived expression choices. These post hoc solves use reconstructed matrices and 8-decimal diagnostic estimates, and include later stored queries when a run stopped solving before its final query; they are not archived final submissions.

C. Data, Implementation, and Statistical Details

C.1. Data Provenance

The historical aggregate analysis uses *Elliptic Differential Equation LLM Reasoning*, version 1, released on Kaggle on 16 April 2026. The `paper_artifacts/data_manifest.json` file records its dataset URL, ZIP checksum, expected members, provider IDs, seeds, and logged settings.

The run-level analyses in this manuscript use locally retained JSON records, provided under `data/recorded_runs/`. Their individual sizes and SHA-256 hashes are recorded in `data/local_run_manifest.json`. Validation confirms 180 scored runs, the shared-task design, finite scores, and agreement with the saved sweep summary. Regenerating the complete aggregate summary reproduces the saved values at relative and absolute tolerance 10^{-12}; all 179 supported per-run confidence records also match. Parsed command sequences agree with the recorded actions. These checks establish agreement of the analyzed records and summaries; the historical ZIP bytes were not revalidated from this local copy. The detailed checks are retained in `paper_artifacts/review_run_analysis.json`.

Table 6. Exact provider IDs in the April 2026 archive. Display names describe that archived experiment, rather than current provider offerings.

Display name	Recorded model ID
Gemini 3.1 Pro	`gemini-3.1-pro-preview`
Claude Opus 4.6	`claude-opus-4-6`
GPT-5.4	`gpt-5.4`
Gemini 3 Flash	`gemini-3-flash-preview`
Claude Sonnet 4	`claude-sonnet-4-20250514`
GPT-5.4 mini	`gpt-5.4-mini`

OpenAI records specify `reasoning_effort=medium`; the other providers have no recorded override. Temperature and exact per-run source revisions are absent. *PDE Library*, version 9, is a separately published source snapshot that postdates

some runs. Exact coefficient regeneration and source-moment agreement constrain the reconstruction, while the solve discrepancies in Appendix B.4 preclude a claim of exact historical replay.

The scored archive contains the `standard` condition and no prompt ablation. It does not preserve all failed attempts or retries preceding the retained runs. Separate preliminary pilot tables have no recovered source logs and are excluded from this analysis. The numerical reference environment uses Python 3.12 and pinned packages in `requirements.txt`; derived JSON outputs retain per-task values, definitions, and numerical cutoffs.

C.2. Task-Level Comparisons and Confidence

The comparison unit is the shared PDE seed within each difficulty, not an individual coefficient, solve, or confidence report. Each model has ten tasks per setting. We report Wilson 95% intervals for marginal success proportions and paired task resampling for success differences (20,000 draws; seed 20261020). Exploratory pairwise tests condition on discordant tasks, with Holm correction across the 15 model pairs within each difficulty. There are no repeated model draws from which to estimate within-task response variance. Empirical paired-bootstrap intervals can degenerate when all ten observed task differences agree. These intervals describe uncertainty under task resampling assumptions and do not resolve selection into the scored archive.

A confidence report states the model's probability that a coefficient block is accurate. We pair it only with a solve returned on a strictly earlier turn. Its correctness event is $E_q < 0.1$, separately for $q \in \{a, b, c\}$, which differs from final-run success $E < 0.01$. All reports and coefficient blocks within a run are kept together when resampling; run-balanced scores avoid giving a model more weight merely because it reports more often. Action counts, first-to-last solve improvement, and expression-family entropy are descriptive diagnostics. They do not independently validate planning, learning, or information gain.

Table 7. Strict-threshold archived successes with Wilson 95% intervals. Each cell uses ten scored tasks; intervals cannot account for missing attempts or model-sampling variability.

Model	Easy	Medium	Hard
Gemini 3.1 Pro	10/10 [0.72, 1.00]	9/10 [0.60, 0.98]	10/10 [0.72, 1.00]
Claude Opus 4.6	10/10 [0.72, 1.00]	9/10 [0.60, 0.98]	8/10 [0.49, 0.94]
GPT-5.4	7/10 [0.40, 0.89]	3/10 [0.11, 0.60]	1/10 [0.02, 0.40]
Gemini 3 Flash	8/10 [0.49, 0.94]	7/10 [0.40, 0.89]	7/10 [0.40, 0.89]
Claude Sonnet 4	8/10 [0.49, 0.94]	6/10 [0.31, 0.83]	0/10 [0.00, 0.28]
GPT-5.4 mini	7/10 [0.40, 0.89]	4/10 [0.17, 0.69]	2/10 [0.06, 0.51]

Table 8. Exploratory exact paired success tests, Holm-adjusted within the 15 model pairs of each difficulty. All 45 comparisons are shown. Values are conditional on the retained scored tasks; no adjustment is made across difficulties.

First model	Second model	Easy	Medium	Hard
Gemini 3.1 Pro	Claude Opus 4.6	1.0000	1.0000	1.0000
Gemini 3.1 Pro	GPT-5.4	1.0000	0.9141	0.0547
Gemini 3.1 Pro	Gemini 3 Flash	1.0000	1.0000	1.0000
Gemini 3.1 Pro	Claude Sonnet 4	1.0000	1.0000	0.0293
Gemini 3.1 Pro	GPT-5.4 mini	1.0000	0.8750	0.1016
Claude Opus 4.6	GPT-5.4	1.0000	0.4688	0.1719
Claude Opus 4.6	Gemini 3 Flash	1.0000	1.0000	1.0000
Claude Opus 4.6	Claude Sonnet 4	1.0000	1.0000	0.1016
Claude Opus 4.6	GPT-5.4 mini	1.0000	1.0000	0.2812
GPT-5.4	Gemini 3 Flash	1.0000	1.0000	0.5000
GPT-5.4	Claude Sonnet 4	1.0000	1.0000	1.0000
GPT-5.4	GPT-5.4 mini	1.0000	1.0000	1.0000
Gemini 3 Flash	Claude Sonnet 4	1.0000	1.0000	0.1719
Gemini 3 Flash	GPT-5.4 mini	1.0000	1.0000	0.5000
Claude Sonnet 4	GPT-5.4 mini	1.0000	1.0000	1.0000

D. Behavioral Figures and Their Interpretation

The main figures summarize the same six models across easy, medium, and hard tasks. Compact plot labels abbreviate Gemini as G, Claude Opus as Opus, and Claude Sonnet as Sonnet; GPT mini denotes GPT-5.4 mini. Stacked tool-use bars use means, so their heights are mean combined counts of these three action types. Improvement and entropy bars show means alongside median ticks and individual observations.

All 159 repeated-solve ratios are available; the 21 remaining runs contain a single solve and have no plotted ratio. The previous summary-file convention assigned one to these cases, but those values are not reused as observed revisions. Means and medians condition on the eligible repeated-solve runs, whose counts appear below the model names. A large reduction can reflect a poor initial estimate or the ordinary benefit of additional measurements.

Query-family entropy is $H = -\sum_{i=1}^{5} p_i \log_2 p_i / \log_2 5$, with $0 \log 0 = 0$. The five categories are polynomial, trigonometric, exponential, localized, and other, assigned by a heuristic string classifier. The archived taxonomy is recorded at the final solve; later queries in six runs are absent from it. A high entropy does not imply linearly independent or useful measurements.

The confidence plots use disjoint bins $[l, h)$, with 1 included in the last bin, and show mean confidence against empirical correctness. Reports are matched only to strictly earlier solves, since a solve requested in a report's own response has not yet been returned. The archive contains 733 confidence reports: 707 have a preceding solve and 26 are excluded. Causal matching changes the solve paired to 71 reports relative to the legacy query-count rule; 827 observations lie on interior bin boundaries that the old plots could count twice. The retained reports contribute 2,121 dependent observations for a, b, c, excluding f. Individual coefficient reports are not treated as independent tasks.

For run i with m_i retained coefficient observations, the run-balanced Brier score is

$$\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \sum_{r=1}^{m_i} (p_{ir} - y_{ir})^2.$$

We also report the signed gap obtained by replacing the square with $p_{ir} - y_{ir}$; positive values indicate average overconfidence. This gap is calibration in the large, not expected calibration error or a full calibration criterion. Intervals resample whole supported runs with 20,000 draws. There are 179 supported runs: Gemini Flash hard has nine and all other cells ten. Run balancing does not align reporting stages, query counts, or base accuracies. Thus these scores cannot compare intrinsic metacognitive ability without a matched reporting schedule and correctness distribution.

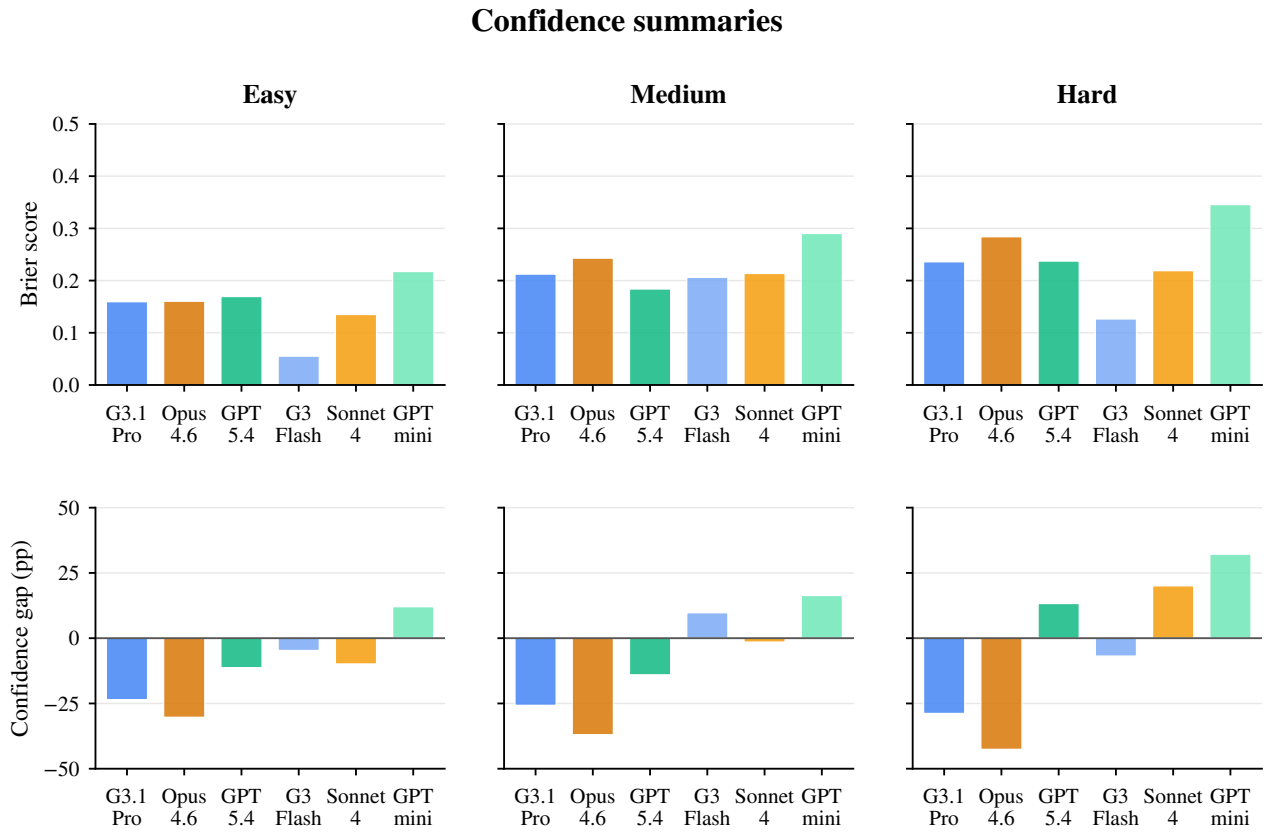


Figure 9. Run-balanced confidence summaries. Brier scores combine confidence and correctness; lower is better. Signed confidence gaps are in percentage points: negative values indicate average underconfidence and positive values overconfidence. Each model-setting cell has ten supported runs except Gemini 3 Flash hard, which has nine. Whole-run bootstrap intervals are retained in the released statistical summaries.