REPLAY THE SCIENTIST: CALIBRATED EVIDENCE AFTER INTERACTIVE LLM HYPOTHESIS SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models increasingly search for scientific claims through interaction. They adapt hypotheses, analyses, repairs, and measurements before reporting a result, but testing only the final hypothesis discards the outcome-dependent path that selected it. We introduce Transcript Replay, a black-box protocol that reruns the discovery procedure under randomized outcomes and ranks the selected score against counterfactual discoveries. Under a declared exchangeable null, the discovery program is the inferential unit and retains its search policy. In controlled studies, final-only rejection reaches 96.3%, while Transcript Replay remains calibrated through adaptive search and acquisition. We evaluated Qwen and Llama on rule search, generated Python features, and external equation discovery. In full-data rule search, Transcript Replay has 3.0% null rejection versus 64.8% for final-only testing and gains 24.0 power points over held-out testing. At fixed full-data scoring, a larger search sample improves power by 11.5 points and planted-mask recovery by 12.2 points. We reuse exact reference scores, reducing required executions by 91.9% in the primary validation batch for repeated queries to a fixed scientist on outcomes related by the specified transformations. The replay p-value evaluates the discovery procedure behind the selected rule or program.

1 INTRODUCTION

Feedback extends a language model from one-shot generation to scientific search. An agent can propose an explanation, execute an analysis, repair invalid code, inspect an anomaly, and choose a new experiment before deciding what to report. This interaction is central to hypothesis generation, equation discovery, automated experimentation, and end-to-end AI research (Zhou et al., 2024; Wang et al., 2024; Shojaee et al., 2025a; Pu et al., 2026; Sen Gupta et al., 2026; Agarwal et al., 2025).

Interaction also changes what counts as evidence. After noticing an association for high frequency, an agent may find that it concentrates at high attack angle, repair an unsupported rule, and stop at a strong two-clause subgroup. The report contains one rule and one nominal p-value, although outcomes determined the rule, its complexity, and the stopping decision. Testing that rule as if it preceded search ignores its selection path. Flexible analysis can therefore create convincing evidence even when each operation is reasonable (Simmons et al., 2011; Silberzahn et al., 2018; Bertran et al., 2026).

Logs record the adaptive path, but inference requires a reference distribution over paths the same scientist would follow under null outcomes. The realized bank does not supply this distribution because feedback can create candidates and change which rows are measured. The alternatives have different requirements: holdout evaluation reserves observations for evidence, whereas exhaustive max-statistic correction requires an enumerable hypothesis universe. We therefore target adaptive discovery with limited data, where Transcript Replay follows a scientist that reaches a small, outcome-dependent part of that universe.

We propose **Transcript Replay** to calibrate the scientist's *discovery procedure* while preserving its policy (Figure 1). The wrapper takes an environment, null randomization, and external randomness. It reruns proposal, execution, feedback, repair, acquisition, stopping, and selection under each randomized outcome vector. We rank the observed score against counterfactual selected scores, allowing hypotheses, acquired rows and transcripts to differ.

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

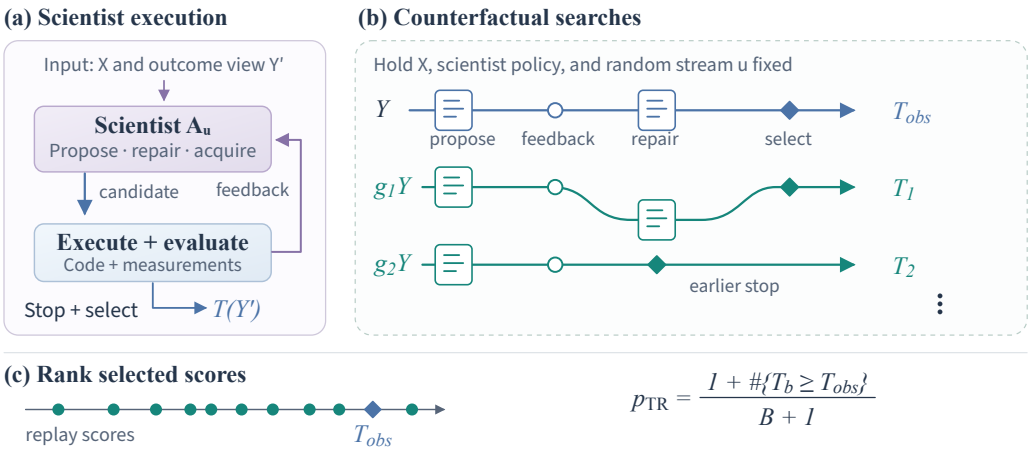

Figure 1: **Calibrating the discovery procedure.** (a) Proposals, execution feedback, repairs, acquisition, and selection define the statistic. (b) Each admissible randomization reruns the program with fixed X and external randomness u, allowing different paths and stopping times. (c) We rank the observed score among replay-selected scores. Paths and score positions are schematic.

We treat the complete outcome-adaptive program as the statistic, calibrating its selected score with a classical rank construction while using all observations for search and inference in the full-data comparisons. Executable counterfactuals support a program-level claim: this scientist selected stronger structure than it typically selects under randomized outcomes. No symbolic characterization of the LLM's selection event is required.

We varied interaction, repair, stopping, data allocation, and acquisition in controlled scientists. Even when final-only rejection reaches 96.3%, Transcript Replay retains calibration and has higher power than the best fixed split across the power grid. Across 8,112 local Qwen rule and Qwen and Llama Python-program executions, independent-query rejection is 3.0–3.6%. The rule comparisons distinguish full-data gains over held-out testing from the 11.5-point improvement due to additional search data at fixed scoring. An external equation-discovery study extends the replay interface to an LLM-SR workflow. Together, these studies connect calibrated program-level evidence with measurable benefits from retaining data for rule search.

Our contributions are:

- We develop an executable replay interface that calibrates evidence from interactive scientific programs while preserving their search policies under a declared exchangeable null.
- We show how incomplete replay distorts evidence and use score-matched comparisons to identify gains in rule detection and planted-mask recovery from additional search data.
- We show how exact reference reuse reduces scientist executions for compatible query batches and give conditions for safe prefix reuse.

2 Related Work

2.1 Interactive Scientific Discovery

Scientific agents use feedback to revise hypothesis banks, executable programs, and natural-language explanations (Zhou et al., 2024; Liu et al., 2025; Wang et al., 2024; Madaan et al., 2023). LLM-SR searches equation programs (Shojaee et al., 2025a), while AutoDiscovery, PiEvo, and LLM-ACES redirect discovery using surprise, anomalous observations, or candidate disagreement (Agarwal et al., 2025; Pu et al., 2026; Abhyankar et al., 2026). ExperiGen and broader AI-scientist systems extend adaptation to experiments and research workflows (Sen Gupta et al., 2026; Lu et al., 2026; Gottweis et al., 2026). Benchmarks evaluate rule induction, data-driven discovery, research ideas, and chemical

hypotheses (Yang et al., 2024; Hua et al., 2025; Majumder et al., 2025; Si et al., 2025; Yang et al., 2025b). We focus on inference that follows the decisions these adaptive scientists make.

2.2 Inference after adaptive search

Adaptive data analysis and reusable holdouts protect data-driven inference (Dwork et al., 2015a;b; Nakkiran & Błasiok, 2018). Conditional selective inference targets selected parameters, with black-box methods estimating selection probabilities (Fithian et al., 2014; Liu et al., 2022). Shiraishi et al. (2025) condition on features and outliers selected by predefined pipelines. Matsukawa et al. (2025) condition on features generated by tree-search feature engineering for linear models. Both solve component-specific selection inequalities to find which points on a response line preserve the observed selection. Selective randomization handles adaptive experiments with imputable selection statistics and design probabilities (Freidling et al., 2026).

Classical permutation tests can treat a complete learner as the statistic (Ojala & Garriga, 2010; Tansey et al., 2022). Rank and resampling guarantees provide our statistical foundation (Phipson & Smyth, 2010; Hemerik & Goeman, 2018; Westfall & Young, 1993). We apply this foundation to an interactive discovery program's global output. Our execution interface specifies which outcome-dependent decisions to rerun and when reference computations allow exact reuse. This program-level target differs from a selection-conditioned coefficient or interval.

2.3 Agent traces, multiverses, and perturbation checks

Many-analyst and multiverse studies reveal variation across defensible analyses (Silberzahn et al., 2018; Steegen et al., 2016; Bertran et al., 2026). Agent benchmarks and trace-based checks expose failures in automated workflows (Chen et al., 2025; Luo et al., 2025; Rewolinski et al., 2026). Fisher-R1 and P-Bench assess method choice and test execution for supplied scientific questions (Miao et al., 2026). POPPER validates a supplied hypothesis through sequential falsification (Huang et al., 2025). False-discovery methods calibrate collections of selected findings, including AI-ranked therapeutic shortlists (Benjamini & Hochberg, 1995; Jin et al., 2026). Our trace specifies a randomized comparison that executes the discovery procedure responsible for selecting the result. Independent holdout evaluation and visible-candidate correction motivate two of our comparators (Sen Gupta et al., 2026; Lin et al., 2025).

3 Transcript Replay

3.1 Interactive discovery as a program

Let X denote candidate descriptions, features, or an environment of available measurements, and let Y denote the associated target outcomes. An interactive scientist A_u has frozen external randomness u. Given (X, Y), it emits an event transcript

$$\tau(Y) = \big(\textsc{propose}, \textsc{compile}, \textsc{feedback}, \textsc{repair}, \textsc{acquire}, \ldots, \textsc{stop}, \textsc{select}\big) \tag{1}$$

and a selected scalar score $T(Y)$. The score may be a subgroup contrast, validation reward, likelihood improvement, or another prespecified quantity. The statistic $T(Y)$ includes selection and scores the claim chosen by running the complete program on Y, not an externally fixed claim. An execution trace records proposed candidates, computed scores, compiler feedback, revisions, acquisition choices, and the stopping decision, whereas Transcript Replay executes the same program with a new outcome view and regenerates the decisions that depend on it.

3.2 A randomization test over scientists

Let $\mathcal{G}$ be a finite group under which the null distribution of Y is invariant conditional on X (*exchangeable null*). The scientist, score, and external randomness are specified independently of the tested outcomes. The program includes any outcome-guided choice among scientists. We draw B transformations $g_1, \ldots, g_B$ independently and uniformly from $\mathcal{G}$. Transcript Replay computes

$$T_b = T(g_b Y) = \mathrm{score}(A_u(X, g_b Y), g_b Y) \tag{2}$$

Table 1: Observed and reference statistics. Selected maxT retains the agent's observed output. Exhaustive search selects its own catalog maximum.

Method	Observed score	Reference statistic
Final-only	Agent-selected output	Same fixed output
Frozen bank	Agent-selected output	Maximum over observed bank
Selected maxT	Agent-selected output	Maximum over full catalog
Exhaustive search	Full-catalog maximum	Maximum over full catalog
Held-out	Half-search output on held-out labels	Held-out randomization
Transcript Replay	Agent-selected output	Output of regenerated search

by rerunning the same discovery procedure. Its one-sided Monte Carlo p-value is

$$p_{\mathrm{TR}} = \frac{1 + \sum_{b=1}^{B} \mathbf{1}\{T_b \geq T_{\mathrm{obs}}\}}{B+1}. \tag{3}$$

The controlled study uses $B = 255$, whereas the direct local study uses 39 reference outcomes per bank. The reference distribution describes this scientist's selected claim strength under the declared null, with no feature-aligned outcome structure but potentially different rules, bank sizes, repairs and stopping times. Mutually reachable outcomes under the declared group form an *outcome orbit*.

The transformation family defines the scientific null: the controlled study uses unrestricted outcome permutation, whereas the direct local study permutes only within the fixed search and held-out strata, keeping each stratum's outcome multiset unchanged. Blocked permutations and conditional randomization can preserve other declared nuisance structure. Transcript Replay regenerates the discovery statistic under the domain's exchangeable randomization.

Proposition 1 (randomization validity). Suppose $Y \mid X$ is invariant under $\mathcal{G}$ and the complete statistic T_u is fixed independently of Y. With the uniform transformation sampling above, the plus-one test in Equation 3 satisfies $\Pr(p_{\mathrm{TR}} \leq \alpha \mid X, u) \leq \alpha$. The probability includes both the null outcomes and the sampled transformations.

Appendix A applies the standard randomization rank argument to this entire discovery procedure. To report the strongest of $A_1, \ldots, A_K$, we use $T(Y) = \max_k T_k(Y)$ and rerun all A_k in every randomized world. The same rule covers outcome-dependent choices of model, prompt, seed, or analysis.

3.3 Competing replay units

Table 1 lists each observed statistic and its reference. In data-use comparisons, *Full Replay* applies Transcript Replay to all available rows, while *Half Replay* applies it to the designated search half. Final-only testing asks how the observed selected rule scores under randomization, fixing that rule and permuting only its score. Frozen-bank correction fixes the observed transcript's evaluated rule set $\mathcal{B}_{\mathrm{obs}}$ and uses $T_{\mathrm{bank}}(gY) = \max_{h \in \mathcal{B}_{\mathrm{obs}}} z(h; gY)$, a max statistic over the realized bank. For a finite catalog, such as the controlled study's 232 rules, *selected maxT* compares the agent-selected observed score against catalog maxima under randomization. *Exhaustive search* instead takes the catalog maximum on both sides. Held-out testing reserves independent evidence, while Transcript Replay reruns the discovery procedure in each randomized world.

We also freeze only the first proposal bank or multiply the final-rule p-value by the visible candidate count, distinguishing fixed candidates and later decisions from count-based correction.

For a repaired rule, final-only testing reuses the observed clauses, whereas Transcript Replay reruns the scientist so that randomized feedback can change the repair and the rule ultimately selected.

3.4 What the rejection means

We interpret rejection as evidence about the discovery procedure: it selected stronger feature-aligned structure than it typically selects under the specified null. We also measure exact recovery of planted mechanisms, separating structure detection from recovery of its generating explanation.

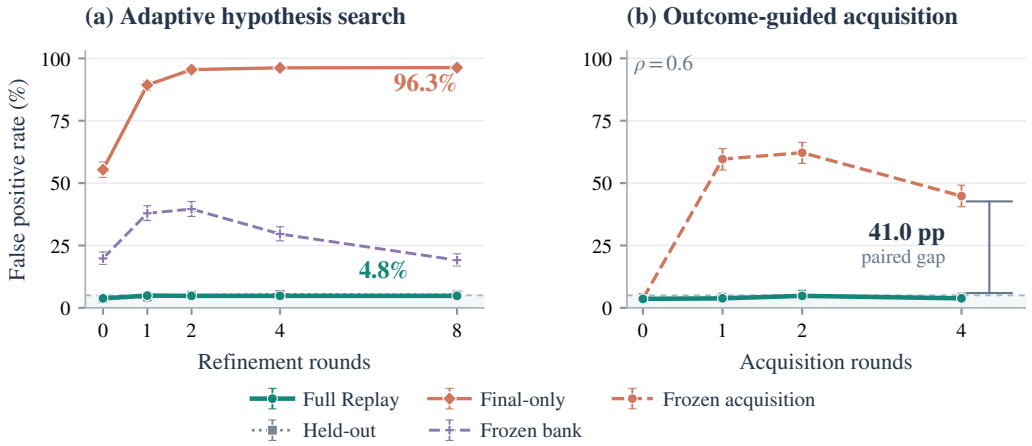

Figure 2: **Calibration through adaptive search and acquisition.** (a) Rejection of independent null worlds as hypothesis feedback deepens. (b) Replaying acquisition at the primary proxy correlation $\rho = 0.6$ closes a 41.0-point rejection gap at four rounds. The dashed horizontal line marks $\alpha = 0.05$. Bars show 95% Wilson intervals. Adaptive initialization and repair still run at zero refinement rounds.

Acquisition replay requires a complete measurement pool or an executable null simulator because each randomized world must regenerate target-related proxies, acquired rows, search, and selection within that environment.

4 Controlled Study

4.1 Controlled scientist and data-generating process

The controlled environment has six binary features and four repeats of each feature vector. Its 232 conjunctions contain one to three literals $X_j = v_j$, $v_j \in \{0, 1\}$. Initialization ranks 12 one-literal candidates and applies compilation and outcome-guided repair before the d feedback-refinement rounds, so adaptive initialization and repair remain even at $d = 0$. The policy ranks candidate pools by outcome scores and ultimately selects the largest standardized subgroup contrast. Reported bank sizes count admitted outputs.

Null outcomes are independent Gaussian noise. Alternative outcomes plant a one-, two-, or three-literal mechanism with term-adjusted signal strength. Each world records all proposals, feedback, repairs, and selection decisions. Repeated feature vectors provide an independent split comparator without changing the hypothesis universe. We evaluate a predefined experimental matrix. Appendix C gives the full data-generating process.

4.2 Experimental design

We varied feedback depth over 0, 1, 2, 4, and 8 rounds in the main family. Three orthogonal families varied initial search width, compiler failure and repair, and outcome-dependent stopping. In the 24,800-world matrix, we compared seven evidence strategies with 255 randomized references per inference. The primary endpoints were null rejection at $\alpha = 0.05$, global power, and exact discovery. We compared methods on paired worlds. Appendix C gives all cell counts, secondary endpoints, and interval procedures.

We compared Full Replay with search-matched Half Replay and splitting in a matched 12,300-world study at 128, 256, and 512 total rows. Half Replay and the split shared the same stratified search half. Full versus Half changes both search and scoring sample sizes. Section 5 adds a score-matched comparison. For sensitivity, we repeated the matched split comparison at 25:75, 50:50, and 75:25 search-to-evidence ratios.

4.3 Calibration under deeper interaction

We observe that Full Replay stays calibrated as interaction deepens, with null rejection of 3.8–4.9% from zero through eight rounds and at most 6.0% across all 14 settings (Figure 2). Final-only rejection rises from 55.4% to 96.3%, while freezing the final bank rejects 19.1–46.0%. With 255 references, visible-bank Bonferroni cannot reject once 13 candidates are admitted, whereas with 4,999 references its rejection is 25.6% at $d = 4$ and 14.9% at $d = 8$. We interpret these higher-resolution results as evidence of selection inflation obscured by the coarser test. Selected maxT and held-out tests remain calibrated. We observe the same separation across width, repair, and stopping and therefore treat the search program as the replay unit (Appendix D).

4.4 Replay separates acquisition validity from policy value

The acquisition study uses a pool of 512 potential measurements. It starts with a balanced 64-row design and adds 16 rows per round. After the current scientist selects a rule, the proxy $Z_i = \rho Y_i + \sqrt{1-\rho^2}\epsilon_i$ prioritizes up to 12 unmeasured rows inside its subgroup, with a fixed exploration order filling the batch. We crossed zero to four rounds with $\rho \in \{0, 0.3, 0.6\}$. Frozen-acquisition replay keeps the measured rows, whereas Full Replay rebuilds Z, acquisition, and search under each permuted target vector.

First, Full Replay rejects 3.4–5.2% of null worlds across the acquisition grid. At $\rho = 0.6$ and four rounds, freezing acquisition rejects 44.8% [40.5, 49.2], versus 3.8% [2.4, 5.9] for Full Replay, a paired gap of 41.0 points [36.8, 45.4]. The $\rho = 0$ control shows no analogous inflation.

Second, uniformly adding 64 rows in matched comparisons raises primary-strength power by 9.6, 16.4, and 28.8 points for one-, two-, and three-clause mechanisms. Third, at fixed row count, subgroup targeting adds 6.4, 18.4, and 26.4 points. The latter two subgroup intervals exclude zero, whereas all six proxy-minus-subgroup estimates are negative across the two strengths, and five intervals exclude zero. We attribute the observed power gains to added rows and subgroup targeting, whereas rerunning acquisition addresses the validity of the evidence produced by that policy. Appendices D.2 and D.3 report the full calibration, policy, and path-stability comparisons.

4.5 Full-data procedures improve detection

All three arms remain calibrated, with null rejection of 4.8–5.6% for Full Replay, 4.8–6.2% for Half Replay, and 3.4–5.8% for the split. Full Replay exceeds both in power in all 27 planted cells. At medium signal, Full-minus-Half gains are 8.75–45.00 points and Full-minus-split gains are 9.25–26.75 points. Both the rows searched and the rows used for scoring change in these full-data comparisons. At this signal level, the search-matched Half-minus-held-out difference ranges from -18.75 to $+0.50$ points. Appendix Figure 7 gives the full decomposition. Across split ratios, Full Replay exceeds the pointwise best 25:75, 50:50, or 75:25 split in all 27 cells, with gains at the primary strength of 9.00–26.75 points (Appendix D.1).

5 Local Scientific Agents

5.1 Tasks and rank-outcome queries

We constructed semi-synthetic queries on fixed 256-row Airfoil and Concrete covariate tables, which have five and eight physical variables, respectively (Brooks et al., 1989; Yeh, 1998). A feature-only split assigns 128 rows to search and 128 to held-out evaluation. The local permutation group preserves these two strata so that Full, Half, and held-out comparisons use exactly matched outcome worlds.

Within each stratum, we map continuous query outcomes by rank onto the fixed empirical multiset of rank-normalized measured targets. This projection commutes with the declared permutations, making every reference an admissible permutation of the particular query being tested. Null queries use independent Gaussian backgrounds, whereas planted queries add standardized mechanisms before rank projection. Rule-search mechanisms contain one to three clauses, at strengths 0.25 and 0.40. Program-search mechanisms use a univariate transform or an interaction, at strengths 0.40 and 0.70. Appendix F specifies the construction and reports all conditions.

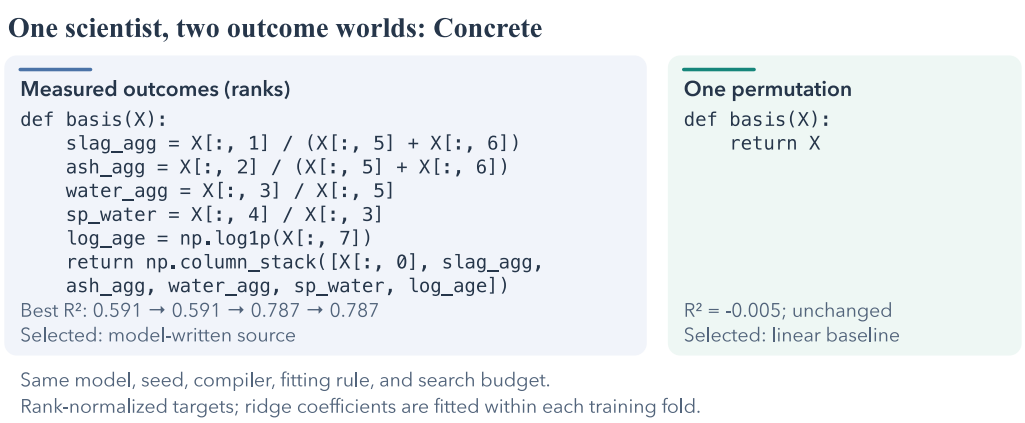

Figure 3: **The executable object that Transcript Replay recalibrates.** Illustrative Qwen3-8B programs on Concrete under measured and permuted outcomes. Full source is line-wrapped, score histories start from the linear baseline, and the common ridge evaluator fits coefficients. Appendix F.4 includes every measured-target case.

5.2 Rule and Python-program scientists

Qwen3-4B and Qwen3-8B (Yang et al., 2025a) search one- to three-clause threshold rules. They receive task metadata and ten outcome-ranked univariate evidence items and then propose four rules per round for three rounds. A compiler enforces the declared thresholds and support constraints, while scores and compiler feedback guide refinement and bounded repair. The statistic is the standardized subgroup contrast used in the controlled study.

Qwen3-8B also writes NumPy feature functions `basis(X)` that compose nonlinear transforms and interactions outside the finite rule catalog. A common evaluator returns four-fold cross-validated R^2. Feature-only transforms use the fixed covariates, and we fit ridge coefficients within training folds. Execution and score feedback guide revision, bounded repair, and stopping. Each search starts from a linear baseline. We used two fixed program seeds and the generation settings in Appendix F.

Llama-3.1-8B-Instruct (Dubey et al., 2024) runs the same Python search on both tasks and data views with the first program seed and its native chat interface. The models share inputs, compiler, scoring, and search budget, but each constructs its own reference distribution.

Figure 3 pairs a measured-target program using ratios and log-age features with a replay that retains the linear baseline. Model-written features are selected in 38.8% of Qwen and 37.3% of Llama reference and query executions. Baseline retention depends on the score. Appendix Table 17 gives admission and fallback counts.

Table 2: Matched data-use contrasts at $\alpha = 0.05$ with 39 references (% or pp), with 95% bank-bootstrap intervals. HF searches half the rows and scores all rows. The Llama parenthesis counts discordant Full versus held-out decisions.

Scientist	Full null rejection	Full − held-out power	Full − HF power
Rule (Qwen3)	3.0 [1.2, 4.9]	24.0 [14.6, 34.7]	11.5 [2.8, 20.5]
Python (Qwen3-8B)	3.6 [1.0, 7.8]	1.6 [0.0, 4.7]	-1.6 [-6.2, 3.1]
Python (Llama-3.1-8B)	3.1 [0.0, 7.3]	0.0 (0 of 32)	-3.1 [-9.4, 0.0]

5.3 Independent references and matched evaluation

Each task supplies six banks of reference outcomes for rule search and four for Python search. Each bank's 39 reference views are independent of its 12 null queries and six or four planted queries, with models, seeds, and data views sharing these outcomes. The 24-cell primary matrix contains 128 sets of reference scores (4,992 executions) and 2,240 query executions. We summarize uncertainty over

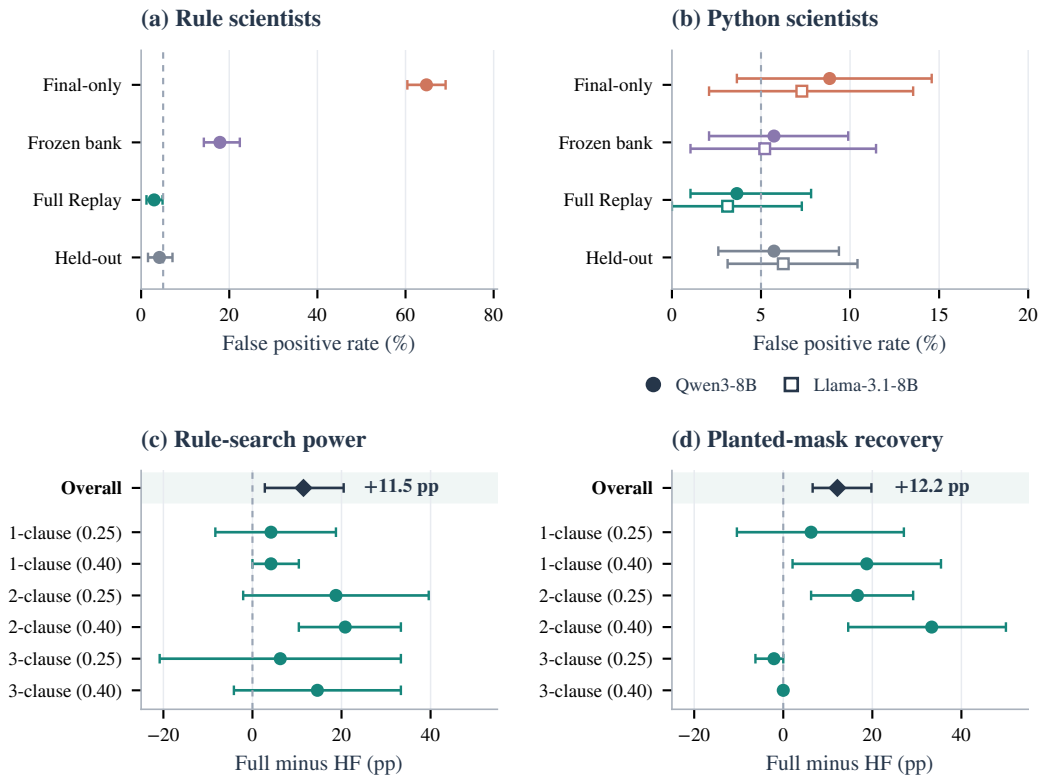

Figure 4: **Calibration and score-matched search gains.** (a, b) Null rejection. Circles and squares in (b) denote Qwen3-8B and Llama-3.1-8B. (c, d) Rule power and planted-mask recovery differences with 256 scoring rows and 256 versus 128 search rows (Full versus HF). Diamonds summarize all conditions. Parentheses give signal strength. Bars show 95% bank-bootstrap intervals. Dashed lines mark 5% (top) and zero (bottom).

12 task-specific outcome banks for rules and eight for programs. Llama adds 16 sets of reference scores and 880 executions using the same eight program outcome banks. We report it separately from the two-seed Qwen study.

Full uses all 256 outcome labels, Half uses the 128 search labels, and held-out testing evaluates Half's choice on the other 128 labels. Frozen-program tests fix the selected feature source but refit coefficients on randomized outcomes. With 39 references, the plus-one test rejects the top two ranks at 5%. Differences within 10^{-10} count conservatively as ties. We form intervals by resampling reference banks within task, retaining matched models, seeds, and queries.

The supplementary HF analysis scores each saved Half-selected output on all 256 rows, including each reference world's own selection, without reranking the bank with full labels. Fixed-linear procedures use the original ridge settings on Full, Half, or held-out data. Exhaustive rule search maximizes the complete catalog in observed and reference worlds. These comparisons were specified after the original results and retain all saved queries (Appendix G).

5.4 Calibration and data-use results

Calibration transfers to local scientists. Table 2 and Figure 4 report independent-query results. In rule search, final-only rejection is 64.8% and frozen-bank rejection is 17.9%, versus 3.0% for Full Replay. The Qwen and Llama Python workflows reject 3.6% and 3.1% of null queries, respectively. Their final-only rates are 8.9% and 7.3%. In both the rule and code interfaces, calibration targets the complete procedure, including repair and baseline selection.

Search data improves rule detection and recovery. At fixed full-data scoring (Figure 4, panels c and d), rule-search power increases by 11.5 [2.8, 20.5] points over HF. *Planted-mask recovery*, exact agreement between selected and planted masks across all rows, rises from 21.5% to 33.7%, a difference of 12.2 [6.6, 19.8] points. We find the same direction for both gains across task, model, and seed margins and when we omit each bank (Appendix G). Across prespecified mechanisms and signal strengths, rule-search power is 75.7% for Full, 64.2% for HF, 41.3% for Half, and 51.7% for held-out testing. The paired Full-minus-held-out gain is 24.0 points [14.6, 34.7]. The Full-minus-held-out point estimate is positive in all six rule conditions.

Exhaustive full-data rule search reaches 84.7% power and 51.4% mask recovery. The full-data linear detector reaches 96.9% program power, versus 95.3% for Qwen and 93.8% for Llama. On the 128-row Half view, program-over-linear gains are 12.5 [3.1, 21.9] and 6.2 [0.0, 12.5] points, respectively. Appendices F and G report every condition and baseline view.

5.5 External equation discovery

We test whether inference follows the complete search procedure that selected a formula. Llama-3.1-8B-Instruct follows LLM-SR on the BPG1 and PO1 synthetic systems from LLM-SRBench (Shojaee et al., 2025b), with two program seeds and 128 candidate slots. Across four reference banks indexed by task and seed, Full Replay rejects 2 of 48 null queries, versus 11 for final-only, 4 for a frozen observed bank, 4 for a label-blind bank, and 3 for held-out testing. All five tests detect the original queries at the available minimum $p = 0.025$. For prediction, Full has lower OOD error in three of four pairs against Half and three against the blind bank, with BPG1 reversals occurring in different seeds. Training-selected polynomials have lower OOD error than Full on BPG1. PySR does so in three of four pairs. The 264-execution study used 61.9 allocated A100-hours (Appendix H).

6 Analysis and Discussion

6.1 A trace becomes an inferential asset

We include the scientist's outcome-dependent operations in the statistic. The acquisition results distinguish path resemblance from invariance: at one round, $\rho = 0$ and $\rho = 0.6$ give similar Jaccard overlap (0.753 and 0.741), but frozen-path rejection is 3.4% and 59.6%. Uniform acquisition is exactly invariant, so its frozen and full statistics coincide and permit exact reuse. Conditional randomization uses imputable selection probabilities, while Transcript Replay uses executable counterfactuals without a symbolic selection model.

6.2 Data efficiency and computation

Without reference reuse, Transcript Replay uses $B + 1$ scientist executions per query, including the observed run. At fixed scoring, we find a gain from more search data for rules, while the linear detector identifies a different regime for programs. In the primary local batch, separately executing 39 references for each of 2,240 queries would require 89,600 scientist executions, whereas sharing compatible banks reduces this to 7,232 (91.9% fewer). Consider a query whose outcomes are permuted within the same strata: with task, scientist, seed and data view fixed, we can map it to the cached reference inputs exactly. A new single-query task still requires 40 executions. We report GPU time separately. Exact input equality also supports prefix reuse (Appendix A.3).

7 Conclusion

Under an exchangeable null, Transcript Replay calibrates the discovery procedure while preserving the scientist's policy. Controlled studies and local Qwen3 and Llama experiments demonstrate the interface for rule, Python-program, and external equation search. In the local rule study, Full Replay exceeds held-out testing by 24.0 power points. Increasing the search sample at fixed full-data scoring adds 11.5 power points and 12.2 points of planted-mask recovery. Compatible inputs allow exact reference reuse across queries on the same outcome orbit. The resulting p-value evaluates the reported rule or program together with the procedure that selected it.

Reproducibility Statement

The appendices describe how we construct tasks and outcomes, implement the scientist interfaces, and set prompt and generation policies. They also give the statistical procedures and resource accounting. We ran all experimental model inference with local Qwen3 or Llama checkpoints. Appendix M summarizes independent verification and compute for each evidence block.

AI Use Statement

In this work, we used generative AI tools for the following tasks: searching for information, identifying relevant literature, generating a portion of the synthetic dataset used in our experiments, simulating peer-review feedback to help refine the manuscript, editing the paper to improve readability. The AI-generated content was inspected, validated, and where necessary corrected by the authors.

References

Nikhil Abhyankar, Sha Li, Sanchit Kabra, Naren Ramakrishnan, Yulia Gel, and Chandan K. Reddy. LLM-ACES: Closed-loop discovery of dynamical systems with LLM-guided adaptive search. *arXiv preprint arXiv:2606.25039*, 2026.

Dhruv Agarwal, Bodhisattwa Prasad Majumder, Reece Adamson, Megha Chakravorty, Satvika Reddy Gavireddy, Aditya Parashar, Harshit Surana, Bhavana Dalvi Mishra, Andrew McCallum, Ashish Sabharwal, and Peter Clark. AutoDiscovery: Open-ended scientific discovery via bayesian surprise. In *Advances in Neural Information Processing Systems*, 2025.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995.

Martin Bertran, Riccardo Fogliato, and Zhiwei Steven Wu. Many AI analysts, one dataset: Navigating the agentic data science multiverse. *Proceedings of the National Academy of Sciences*, 123(29): e2606495123, 2026. doi: 10.1073/pnas.2606495123.

Thomas Brooks, D. Pope, and Michael Marcolini. Airfoil self-noise. UCI Machine Learning Repository, 1989. URL https://doi.org/10.24432/C5VW2C.

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

Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl. *arXiv preprint arXiv:2305.01582*, 2023. doi: 10.48550/arXiv.2305.01582.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783v1.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems*, volume 28, 2015a.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015b. doi: 10.1126/science.aaa9375.

William Fithian, Dennis Sun, and Jonathan Taylor. Optimal inference after model selection. *arXiv preprint arXiv:1410.2597*, 2014.

Tobias Freidling, Qingyuan Zhao, and Zijun Gao. Selective randomization inference for adaptive experiments. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 2026. doi: 10.1093/jrsssb/qkag081.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, et al. Accelerating scientific discovery with Co-Scientist. *Nature*, 655:487–496, 2026. doi: 10.1038/s41586-026-10644-y.

Jesse Hemerik and Jelle Goeman. Exact testing with random permutations. *TEST*, 27(4):811–825, 2018. doi: 10.1007/s11749-017-0571-1.

Wenyue Hua, Tyler Wong, Fei Sun, Liangming Pan, Adam Jardine, and William Yang Wang. InductionBench: LLMs fail in the simplest complexity class. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 26526–26546. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1287.

Kexin Huang, Ying Jin, Ryan Li, Michael Y. Li, Emmanuel Candes, and Jure Leskovec. Automated hypothesis validation with agentic sequential falsifications. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 25372–25437. PMLR, 2025.

Ying Jin, Kexin Huang, Nathaniel Diamant, Kerry R. Buchholz, Steven T. Rutherford, Nicholas Skelton, Tommaso Biancalani, Gabriele Scalia, Jure Leskovec, and Emmanuel J. Candès. TxConformal: Controlling false discoveries in AI-driven therapeutic discovery. *bioRxiv*, 2026. doi: 10.64898/2026.04.27.721076. URL https://doi.org/10.64898/2026.04.27.721076.

Tung-Wei Lin, Runing Yang, Zain ul Abdeen, Alberto Sangiovanni-Vincentelli, Haibo Huang, and Ming Jin. LLMs tackle meta-analysis: Automating scientific hypothesis generation with statistical rigor. In *AI4Research Workshop at AAAI*, 2025. URL https://openreview.net/forum?id=r7Mix4vPNM.

Haokun Liu, Yangqiaoyu Zhou, Mingxuan Li, Chenfei Yuan, and Chenhao Tan. Literature meets data: A synergistic approach to hypothesis generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 245–281. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.12.

Sifan Liu, Jelena Markovic-Voronov, and Jonathan Taylor. Black-box selective inference via bootstrapping. *arXiv preprint arXiv:2203.14504*, 2022.

Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of AI research. *Nature*, 651:914–919, 2026. doi: 10.1038/s41586-026-10265-5.

Ziming Luo, Atoosa Kasirzadeh, and Nihar B. Shah. The more you automate, the less you see: Hidden pitfalls of AI scientist systems. *arXiv preprint arXiv:2509.08713*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

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

Tatsuya Matsukawa, Tomohiro Shiraishi, Shuichi Nishino, Teruyuki Katsuoka, and Ichiro Takeuchi. Statistical test for auto feature engineering by selective inference. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 3214–3222. PMLR, 2025. URL https://proceedings.mlr.press/v258/matsukawa25a.html.

Jiacheng Miao, Jin Mu, Guanhua Chen, and James Zou. Fisher-R1: Training LLM agents for reliable hypothesis testing. *arXiv preprint arXiv:2608.07437*, 2026. URL https://arxiv.org/abs/2608.07437.

Preetum Nakkiran and Jarosław Błasiok. The generic holdout: Preventing false-discoveries in adaptive data science. *arXiv preprint arXiv:1809.05596*, 2018.

Markus Ojala and Gemma C. Garriga. Permutation tests for studying classifier performance. *Journal of Machine Learning Research*, 11(62):1833–1863, 2010.

Belinda Phipson and Gordon K. Smyth. Permutation p-values should never be zero: Calculating exact p-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1):Article 39, 2010. doi: 10.2202/1544-6115.1585.

Yingming Pu, Tao Lin, and Hongyu Chen. Principle-evolvable scientific discovery via uncertainty minimization. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.06448.

Zachary T. Rewolinski, Austin V. Zane, Hao Huang, Chandan Singh, Chenglong Wang, Jianfeng Gao, and Bin Yu. Sanity checks for agentic data science. *arXiv preprint arXiv:2604.11003*, 2026.

Jishu Sen Gupta, Harini SI, Somesh Kumar Singh, Syed Mohamad Tawseeq, Yaman Kumar Singla, David Doermann, Rajiv Ratn Shah, and Balaji Krishnamurthy. Accelerating social science research via agentic hypothesization and experimentation. *arXiv preprint arXiv:2602.07983*, 2026.

Tomohiro Shiraishi, Tatsuya Matsukawa, Shuichi Nishino, and Ichiro Takeuchi. Statistical test for feature selection pipelines by selective inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55283–55302. PMLR, 2025. URL https://proceedings.mlr.press/v267/shiraishi25a.html.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *International Conference on Learning Representations*, 2025a.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 55325–55359. PMLR, 2025b. URL https://proceedings.mlr.press/v267/shojaee25a.html.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *International Conference on Learning Representations*, 2025.

Raphael Silberzahn, Eric L. Uhlmann, Daniel P. Martin, Pasquale Anselmi, Frederik Aust, Eli Awtrey, Stepan Bahnik, Feng Bai, Colin Bannard, Evelina Bonnier, et al. Many analysts, one data set: Making transparent how variations in analytic choices affect results. *Advances in Methods and Practices in Psychological Science*, 1(3):337–356, 2018.

Joseph P. Simmons, Leif D. Nelson, and Uri Simonsohn. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11):1359–1366, 2011.

Sara Steegen, Francis Tuerlinckx, Andrew Gelman, and Wolf Vanpaemel. Increasing transparency through a multiverse analysis. *Perspectives on Psychological Science*, 11(5):702–712, 2016.

Wesley Tansey, Victor Veitch, Haoran Zhang, Raul Rabadan, and David M. Blei. The holdout randomization test for feature selection in black box models. *Journal of Computational and Graphical Statistics*, 31(1):151–162, 2022. doi: 10.1080/10618600.2021.1923520.

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D. Goodman. Hypothesis search: Inductive reasoning with language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.05660.

Peter H. Westfall and S. Stanley Young. *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. Wiley, 1993.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Zonglin Yang, Li Dong, Xinya Du, Hao Cheng, Erik Cambria, Xiaodong Liu, Jianfeng Gao, and Furu Wei. Language models as inductive reasoners. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 209–225. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.eacl-long.13.

Zonglin Yang, Wanhao Liu, Ben Gao, Tong Xie, Yuqiang Li, Wanli Ouyang, Soujanya Poria, Erik Cambria, and Dongzhan Zhou. MOOSE-Chem: Large language models for rediscovering unseen chemistry scientific hypotheses. In *International Conference on Learning Representations*, 2025b.

I-Cheng Yeh. Concrete compressive strength. UCI Machine Learning Repository, 1998. URL `https://doi.org/10.24432/C5PK67`.

Yangqiaoyu Zhou, Haokun Liu, Tejes Srivastava, Hongyuan Mei, and Chenhao Tan. Hypothesis generation with large language models. In *Proceedings of the 1st Workshop on NLP for Science*, pp. 117–139. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.nlp4science-1.10.

A Randomization validity

A.1 Finite-group statement

Let $\mathcal{G}$ be a finite group of transformations containing the identity. Condition on X and on the scientist's outcome-independent external randomness u. Under H_0, $Y \overset{d}{=} gY$ for every $g \in \mathcal{G}$. Define the complete program statistic

$$S_g = \mathrm{score}(A_u(X, gY), gY)\,. \tag{4}$$

For the complete group, use

$$p_{\mathrm{group}}(Y) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \mathbf{1}\{S_g \geq S_{\mathrm{id}}\}.$$

Conditional on its orbit $\mathcal{O}_Y = \{gY : g \in \mathcal{G}\}$, Y is uniform over the distinct orbit elements by null invariance. Every transformed point has this same orbit, and each orbit element is represented by the same number of group elements. Hence at most an α fraction of the orbit has an upper-tail rank at most α, so averaging this orbitwise bound gives $\Pr(p_{\mathrm{group}} \leq \alpha \mid X, u) \leq \alpha$. Counting ties with $\geq$ preserves this bound.

For the Monte Carlo implementation, take $g_0 = \mathrm{id}$ and draw $g_1, \ldots, g_B$ independently and uniformly from $\mathcal{G}$, independently of Y. Conditional on the orbit, $Y, g_1Y, \ldots, g_BY$ are independent uniform draws from that orbit. Because every draw passes through the same fixed map T_u, their scores are exchangeable, and the plus-one rank in Equation 3 is therefore super-uniform. With no ties, attainable rejection probabilities are integer multiples of $1/(B+1)$. Uniform sampling without replacement from $\mathcal{G} \setminus \{\mathrm{id}\}$ is also valid under the corresponding random-transformation argument (Hemerik & Goeman, 2018). Throughout, we distinguish a sampled reference set from the full transformation group.

Reference reuse within an outcome orbit. The same argument permits a shared reference bank. Fix an orbit $\mathcal{O}$ and draw $R_1, \ldots, R_B$ independently and uniformly from $\mathcal{O}$, independently of each tested query. For a query $Y \in \mathcal{O}$, use the cached scores $T_u(R_b)$ in Equation 3. Under the conditional null, $Y, R_1, \ldots, R_B$ are exchangeable, so the cached test inherits the same level guarantee. For every query and reference there exists an admissible transport h_b with $h_bY = R_b$. Evaluating the transported query reproduces the cached input and score exactly. Reuse requires identical outcome multisets within each permutation stratum and the same scientist, features, scoring rule, and randomness. For continuous query outcomes, projection by rank onto a fixed empirical multiset commutes with within-stratum permutations. The shared bank therefore implements the permutation test on the declared rank statistic. Queries sharing a bank are dependent. We retain reference-bank groups in uncertainty summaries and specify query selection independently of the bank. Reuse across unequal outcome multisets does not follow from this result.

A.2 WHY PARTIAL REPLAY CHANGES THE STATISTIC

Write the observed transcript prefix as $H(Y)$. Final-only testing computes $s(H(Y), gY)$ for all g, while complete replay computes $s(H(gY), gY)$. In the first collection, every reference world receives a prefix selected for the unpermuted outcomes, so the collection need not be exchangeable. The same issue applies to an observed final bank. Orbitwise invariance gives a directly checkable sufficient condition for exact partial replay. We measure the effect of freezing outcome-selected prefixes in the controlled study.

An executable sufficient condition is *orbitwise prefix stability*: $H(gY) = H(Y)$ for every $g \in \mathcal{G}$. Under this condition the partial and complete statistics coincide pointwise and therefore have the same rank guarantee. We can verify equality exactly on a finite enumerated randomization group. Agreement on a sampled reference set establishes stability for those transformations, whereas a global certificate requires the complete group. Aggregate path overlap does not certify calibration. The criterion identifies when an expensive outcome-invariant prefix can be cached safely.

A.3 EVENTWISE CACHEABILITY

We extend the prefix condition to an event graph. Write event e_j as a deterministic map of its parent events, stage random variable u_j, and any outcome view available at that stage. A cache set $\mathcal{C}$ replaces the replayed values $e_j(gY)$ with the observed values $e_j(Y)$ for $j \in \mathcal{C}$ and recomputes all remaining events.

Exact eventwise cacheability criterion. For every $g \in \mathcal{G}$, cached replay is pointwise identical to complete replay whenever each $j \in \mathcal{C}$ is orbitwise stable, $e_j(gY) = e_j(Y)$, and every uncached descendant of $\mathcal{C}$ is recomputed from the resulting boundary values using the same outcome view and stage randomness. Since the event graph is acyclic, topological induction propagates this equality through the remaining events. It follows that the downstream event log, selected claim, statistic S_g, and randomization rank are unchanged.

A stronger structural certificate is easier to inspect: the ancestor closure of every cached event must contain neither Y nor an outcome-dependent event, and the cached event must use fixed stage randomness. Static task metadata, a frozen grammar, model weights, and feature-only preprocessing can satisfy this criterion. Scores, outcome-ranked evidence, feedback, repairs prompted by that feedback, stopping decisions, and rule-guided acquisition generally do not. Exact orbitwise agreement provides an executable certificate on a finite group even when a service is too complex for static dependency analysis.

In the original rule-scientist study, we identify a substantial prefix that satisfies this criterion. After chat templating, the system instruction, task description, and fixed clause grid form an identical token prefix before outcome-ranked evidence. Across 12,721 calls, we count 13.5M repeated prefix tokens eligible for exact KV reuse across calls, or 48.7% of 27.7M recorded input tokens. The criterion certifies this input as reusable, while realized latency, GPU-hour, and FLOP gains depend on the serving implementation.

The computation cost follows directly. If a stable prefix costs C_H, the replayed suffix costs C_T per world, and there are B references, complete replay costs

$$C_{\text{full}} = (B + 1)(C_H + C_T), \tag{5}$$

whereas certified prefix caching costs

$$C_{\text{cache}} = C_H + (B + 1)C_T. \tag{6}$$

The saving is BC_H without changing the statistic. The stability study in Appendix E shows why approximate set overlap cannot be substituted for this equality condition.

A.4 RANDOMNESS AND STOCHASTIC SERVICES

We fix program randomness by stage and replay it as common random numbers. An alternative is to redraw u_g independently for every null world, provided (u_g, gY) is exchangeable and the observed run is generated by the same rule. Deterministic decoding is not required. External services must either expose reproducible responses, be recorded as an outcome-independent asset, or be included in the stochastic program being randomized.

Algorithm 1 Full Transcript Replay with stage-aligned common randomness

Require: fixed features X; outcomes Y; null-invariant group $\mathcal{G}$; independent uniform draws $g_1, \ldots, g_B \in \mathcal{G}$ and $g_0 = \mathrm{id}$; scientist A; stage-randomness map u; round and repair caps

for $b = 0, \ldots, B$ **do**
  $Y_b \leftarrow g_b Y$; history $H_b \leftarrow [\,]$; admitted bank $\mathcal{B}_b \leftarrow \varnothing$
  initialize the measured rows and environment state declared by A
  for $r = 0, \ldots, R_{\max} - 1$ **do**
    $Q \leftarrow \textsc{Propose}(X, Y_b, H_b, \mathcal{B}_b, u(r))$
    $(V, E) \leftarrow \textsc{Compile}(Q, \text{grammar}, \mathcal{B}_b)$
    if $V = \varnothing$ and the repair budget remains **then**
      $Q' \leftarrow \textsc{Repair}(Q, E, H_b, \mathcal{B}_b, u(r, \text{repair}))$
      $(V, E') \leftarrow \textsc{Compile}(Q', \text{grammar}, \mathcal{B}_b)$
    evaluate every $h \in V$ on the current Y_b view; append scores, supports, compiler messages, and responses to H_b
    $\mathcal{B}_b \leftarrow \textsc{Admit}(\mathcal{B}_b, V, H_b)$
    if A requests another measurement stage **then**
      update measured rows with $\textsc{Acquire}(X, Y_b, H_b, \mathcal{B}_b, u(r, \text{acquire}))$
    if $\textsc{Stop}(H_b, \mathcal{B}_b)$ **then**
      break
  $h_b \leftarrow \textsc{Select}(H_b, \mathcal{B}_b)$; $S_b \leftarrow \textsc{Score}(h_b, Y_b)$
return $p = (1 + \sum_{b=1}^{B} \mathbf{1}\{S_b \geq S_0\})/(B+1)$, the selected claims $h_{0:B}$, and complete event logs $H_{0:B}$

A.5 Adaptive experiment acquisition

The controlled acquisition extension regenerates a proxy-guided measurement path and then reruns hypothesis search. It follows the executable principle in Proposition 1. Any acquisition map that can change with gY, whether directly through a proxy or indirectly through an outcome-selected hypothesis, is part of A_u. Thus S_g must use the acquired rows produced by $A_u(X, gY)$. Selective randomization methods offer a complementary route when inference can condition on an imputable selection statistic and known adaptive design (Freidling et al., 2026). Full Replay instead requires the acquisition program and counterfactual environment to be executable. It does not require a closed-form description of an LLM's selection event. PiEvo and LLM-ACES illustrate scientific systems in which anomaly or disagreement changes later sampling (Pu et al., 2026; Abhyankar et al., 2026). An environment or surrogate must answer the corresponding counterfactual requests for unconditional Full Replay.

B Replay algorithm and event contract

Algorithm 1 makes the statistical unit explicit. Each transformed world receives a scientist run, not only a rescored final rule. A valid executable trace satisfies three implementation invariants: event indices increase monotonically, every evaluated or selected candidate resolves to a frozen executable identifier, and the selected identifier belongs to the bank admitted in that same replay. Stage-aligned seeds share randomness across corresponding proposal and repair calls without forcing their responses to agree.

Table 3 lists the inputs, outputs, and replay requirement for each event type in the executable trace.

C Controlled protocol

C.1 Rule universe and outcome model

The controlled design contains six binary features. Every element of $\{0,1\}^6$ occurs four times, yielding 256 rows. A literal is $X_j = v_j$ with $v_j \in \{0,1\}$. Conjunctions use one to three distinct features. The inventory has $\sum_{k=1}^{3} \binom{6}{k} 2^k = 232$ rules, including 12 single literals. For mask m_h,

Table 3: Event contract for the experimental workflows.

Event	Inputs	Outputs	Replay requirement
Proposal	task, history, seed	candidate text or JSON	regenerate after outcome feedback
Compile	candidate, grammar	rules and errors	deterministic or recorded
Evaluate	rule, outcome view	score, support	regenerate for every view
Repair	errors, history, seed	revised candidate	regenerate if prompt is outcome dependent
Acquire	rule, proxy, environment	new measurements	rerun if row selection can change with Y
Stop	history, thresholds	continue or stop	reevaluate in every transcript
Select	admitted bank	claim and score	rerun on the replayed bank

group sizes n_1, n_0, and sample standard deviation s_Y, the score is

$$z(h; Y) = \frac{\bar{Y}_{m_h=1} - \bar{Y}_{m_h=0}}{s_Y \sqrt{1/n_1 + 1/n_0}}. \tag{7}$$

Null worlds use iid standard normal noise ϵ_i. Alternatives use $Y_i = \epsilon_i + \lambda \delta_k I_h(X_i)$, with $(\delta_1, \delta_2, \delta_3) = (0.35, 0.55, 0.80)$ and the declared $\lambda \in \{0.70, 1.00, 1.30\}$ grid. The subgroup indicator is not prevalence-normalized.

C.2 Scientist state machine

Initialization ranks the 12 one-literal rules, selects a width-limited proposal set, and applies compilation and repair. Repair may add, replace, or flip a literal, so initialization can already admit two-literal rules before the d subsequent rounds expand the strongest candidates and repeat compilation and repair. The default failure probability 0.15 also applies during initialization. We deduplicate the bank after every round. Outcome-dependent stopping compares the best score with a fixed threshold. The final selector returns the strongest admitted rule with deterministic tie breaking.

The policy consults outcome scores to rank candidate pools. Its reported bank size counts admitted rules, not all scores consulted during screening. At $d = 0$, 485 of the 1,000 null worlds contain an initial repair and 363 select a two-literal rule. Even zero refinement rounds include outcome-adaptive screening, compilation and repair.

C.3 Experimental family matrix

Table 4 lists the controlled families and the number of null and alternative worlds assigned to each.

Table 4: Controlled families. Each row varies one control-flow dimension while holding the remaining defaults fixed.

Family	Varied values	Null worlds per condition	Alternative worlds per condition
Feedback depth	0, 1, 2, 4, 8 rounds	1,000	2,250
Search width	initial bank 2, 4, or 8	500	450
Compiler repair	failure rate 0, 0.15, or 0.30	500	450
Outcome stopping	none, score 2.5, or score 3.0	500	450

C.4 Matched data-use design

We kept the six-feature rule universe, two-round search policy, signal definitions, and 255-reference randomization test fixed. We varied feature-row repeats over 2, 4, and 8 to obtain total sample sizes of 128, 256, and 512. A stratified split assigns exactly half of every repeated feature pattern to search and half to evidence. Three methods share each generated world. Full Replay uses every row. Half Replay searches and replays on the split's search half. The split tests that identical search result on its reserved half. Each sample size contains 500 null worlds and 400 alternatives per mechanism complexity and signal multiplier, yielding 12,300 CPU-only worlds and three paired method contrasts.

For the fixed-ratio sensitivity analysis, we used the same worlds and compared three search-to-evidence allocations: 25:75, 50:50, and 75:25. Each split searches only its designated rows and evaluates the selected rule on its independent evidence rows. We compare Full Replay with the pointwise best fixed ratio as a strong split baseline. The analysis evaluates all three allocations.

C.5 Proxy-guided acquisition design

The acquisition environment contains eight replicates of every six-bit feature pattern, for 512 potential target measurements. One replicate per pattern forms a balanced 64-row initial design. With fixed independent $\epsilon_i \sim \mathcal{N}(0, 1)$, the cheap screen is

$$Z_i = \rho Y_i + \sqrt{1 - \rho^2}\epsilon_i. \tag{8}$$

At each round, the two-round scientist searches the measured target outcomes and selects its strongest rule. The next 16 measurements include up to 12 unmeasured rows in that rule's subgroup, prioritized by Z. A fixed exploration order fills the remaining positions. The final scientist then reruns on all acquired target measurements.

The null family crosses $\rho \in \{0, 0.3, 0.6\}$ with 0, 1, 2, or 4 acquisition rounds over 500 shared worlds. The alternative family fixes $\rho = 0.6$, uses 250 worlds for each of three mechanism complexities and two strengths, and compares 0, 2, and 4 rounds. Every test uses 255 randomized references. The 42,000 method records include final-only, frozen-bank, frozen-acquisition, and Full Replay results. Full Replay recomputes Z from each randomized target vector and reruns acquisition and search. The proxy noise and all program randomness remain fixed.

A matched control extension separates three sources of discovery power. An outcome-invariant uniform policy adds the same number of rows without interim hypothesis search. A subgroup policy uses the current best rule but ranks its unmeasured rows with $\rho = 0$ proxy noise. The original $\rho = 0.6$ policy then adds target-related ranking while holding subgroup allocation fixed. The alternative matrix pairs 250 worlds across all three policies at 0, 2, and 4 rounds for every clause count and strength. A separate 500-world uniform-null family covers 0, 1, 2, and 4 rounds. This yields 15,500 labelled pairs of condition and world and 62,000 method records. Collapsing the identical $R = 0$ policy labels leaves 12,500 operationally distinct pairs of condition and world in this extension.

D Full controlled results

Table 5 reports every null-calibration condition, while Table 6 separates global power, exact discovery, selection exactness, and mask overlap at the medium signal level. Table 7 verifies the same ordering across the two other prespecified signal levels. All rates use the 173,600 records indexed by method and world, and Figures 5 and 6 show all calibration and recovery patterns in these tables.

Table 5: Null rejection rates (%) for every controlled condition. Each setting contains 1,000 null worlds in the depth family and 500 elsewhere.

Family	Setting	Final-only	Bank-Bonf.	Frozen bank	Initial bank	Replay	Selected maxT	Held-out
Feedback depth	$d = 0$	55.4	18.6	19.8	9.3	3.8	0.7	3.6
	$d = 1$	89.3	23.1	37.9	5.8	4.9	3.1	4.3
	$d = 2$	95.5	29.3	39.6	5.1	4.8	4.6	5.1
	$d = 4$	96.2	0.0	29.6	4.7	4.8	4.7	5.4
	$d = 8$	96.3	0.0	19.1	4.7	4.8	4.7	5.2
Search width	$w = 2$	96.6	28.2	46.0	5.6	5.4	4.6	5.0
	$w = 4$	96.2	27.6	40.8	5.0	5.0	4.6	4.6
	$w = 8$	96.2	0.0	32.0	5.0	5.0	4.6	5.2
Compiler repair	$f = 0$	96.6	0.0	32.2	5.8	5.8	5.6	5.4
	$f = 0.15$	97.2	0.0	33.8	5.6	5.6	5.6	5.0
	$f = 0.3$	97.8	0.0	33.8	5.6	5.6	5.6	5.4
Outcome stopping	$s = 2.5$	96.2	29.8	46.0	6.2	6.0	3.0	6.0
	$s = 3$	96.2	14.0	22.4	4.6	4.2	3.6	6.0
	none	96.2	0.0	22.0	5.4	5.6	5.4	6.0

Table 6: Medium-strength planted mechanisms across feedback depth. Entries give global power with exact discovery in parentheses (%).

Depth	Clauses	Replay	Selected maxT	Held-out	Replay selected exact	Replay Jaccard
0	1	55.6 (42.4)	27.6 (22.8)	36.8 (31.2)	65.2	79.7
0	2	56.0 (17.2)	27.6 (14.0)	36.0 (11.6)	19.2	54.7
0	3	35.6 (0.0)	17.2 (0.0)	24.8 (0.0)	0.0	30.2
1	1	41.6 (23.6)	35.2 (21.6)	25.6 (10.4)	41.2	62.6
1	2	68.4 (48.8)	62.8 (46.0)	50.4 (31.2)	64.4	79.3
1	3	65.6 (36.8)	60.4 (35.2)	44.8 (19.2)	40.4	64.5
2	1	36.4 (22.0)	36.0 (21.6)	20.4 (8.0)	37.6	57.6
2	2	68.8 (47.6)	68.0 (47.2)	45.2 (22.8)	61.6	76.8
2	3	76.8 (62.8)	75.2 (61.2)	52.8 (32.8)	72.4	81.2
4	1	36.0 (21.2)	36.0 (21.2)	19.2 (7.6)	36.4	56.4
4	2	68.4 (47.2)	68.0 (46.8)	44.8 (21.6)	60.8	76.0
4	3	76.8 (63.6)	76.4 (63.2)	52.8 (34.4)	75.2	82.7
8	1	36.0 (20.8)	36.0 (20.8)	19.2 (7.6)	36.0	56.0
8	2	68.4 (47.2)	68.0 (46.8)	44.4 (21.6)	60.8	76.0
8	3	76.8 (63.6)	76.4 (63.2)	52.8 (34.4)	75.2	82.7

Table 7: Signal-strength sensitivity at two feedback rounds. Entries give global power with exact discovery in parentheses (%).

Clauses	Strength	Replay	Selected maxT	Held-out
1	0.7	17.2 (6.0)	16.8 (5.6)	11.6 (2.8)
1	1.0	36.4 (22.0)	36.0 (21.6)	20.4 (8.0)
1	1.3	66.8 (49.2)	66.0 (48.8)	43.6 (28.0)
2	0.7	38.0 (17.2)	36.8 (16.4)	20.4 (6.8)
2	1.0	68.8 (47.6)	68.0 (47.2)	45.2 (22.8)
2	1.3	94.4 (80.8)	94.4 (80.8)	73.2 (51.6)
3	0.7	40.4 (23.6)	39.6 (23.2)	26.8 (16.4)
3	1.0	76.8 (62.8)	75.2 (61.2)	52.8 (32.8)
3	1.3	96.4 (90.8)	96.4 (90.8)	83.6 (67.2)

Interaction depth and mechanism size. From zero to two refinement rounds, medium-signal Full Replay power rises from 56.0% to 68.8% for two-clause truths and from 35.6% to 76.8% for three-clause truths. Exact discovery rises from 17.2% to 47.6% and from 0.0% to 62.8%. Over the same change, one-clause exact discovery moves from 42.4% to 22.0%. Reaching more compositional mechanisms also increases the reachable multiplicity that valid evidence must account for.

Resolution of visible-bank correction. With 255 references, its smallest adjusted value is $K/256$, where K is the admitted bank size. Rejection at 5% is impossible for $K \geq 13$. All $d = 4$ and $d = 8$ null worlds meet this condition, as do the width-eight and compiler-repair settings. The maximum observed $d = 8$ bank is 36, below the cap of 40. Table 41 reports the separate 4,999-reference check.

Tables 8 and 9 report the matched-observation calibration and planted-signal power matrices. Figure 7 compares the full-data procedure with the two procedures sharing the search half.

Table 8: Null rejection in the matched-observation study (%). Full Replay uses all rows. Half Replay and split share the same search half. Split tests on the held-out half. Brackets give 95% Wilson intervals over 500 worlds.

Total n	Full Replay	Half Replay	50:50 split
128	4.8 [3.2, 7.0]	6.2 [4.4, 8.7]	5.8 [4.1, 8.2]
256	5.4 [3.7, 7.7]	6.0 [4.2, 8.4]	3.4 [2.1, 5.4]
512	5.6 [3.9, 8.0]	4.8 [3.2, 7.0]	5.0 [3.4, 7.3]

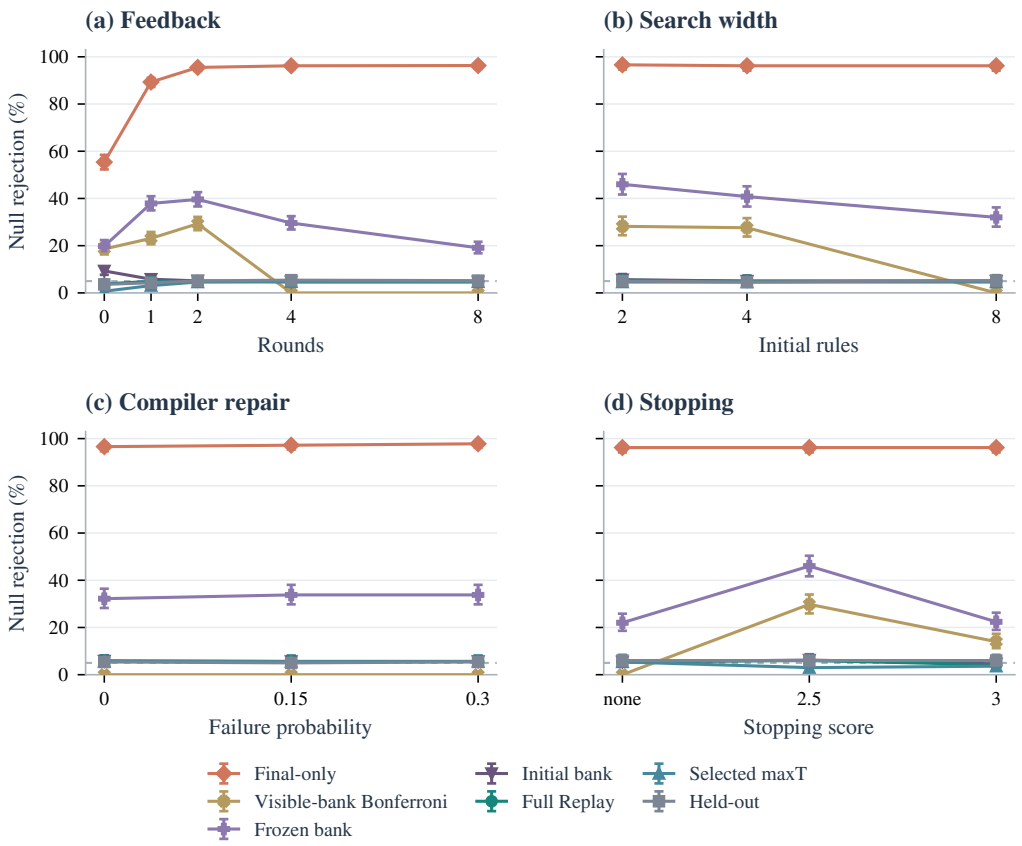

Figure 5: **Calibration across all fixed-table settings.** (a) Null rejection across feedback depth for all replay units and reference strategies. (b, c, d) Search width, compiler failure, and outcome-dependent stopping preserve the gap between complete and partial replay. Points show empirical rates. Bars show 95% Wilson intervals.

D.1 Fixed split-ratio frontier

We supplemented the primary 50:50 comparison with fixed 25:75 and 75:25 search-to-evidence allocations. Full Replay exceeds the best fixed split in all 27 point estimates across the three signal strengths. Table 10 gives the primary-strength cells.

Tables 11, 12, and 13 give the proxy-guided calibration, power, and path diagnostics. Tables 15 and 14 report the matched-policy and outcome-invariant controls.

D.2 Proxy-guided acquisition matrix

The controlled environment contains 512 potential target measurements. We start from a balanced 64-row design and add zero to four 16-row acquisition batches. Only Full Replay recomputes the low-fidelity proxy from each randomized target-outcome vector.

D.3 Matched acquisition-policy controls

We pair the same planted worlds across three policies: outcome-invariant uniform acquisition, current-rule subgroup targeting with an outcome-independent ranking proxy ($\rho = 0$), and the same targeting policy with an outcome-related proxy ($\rho = 0.6$). At $R = 0$, all policy labels share the initial 64-row design. Subsequent rows compare matched acquisition paths.

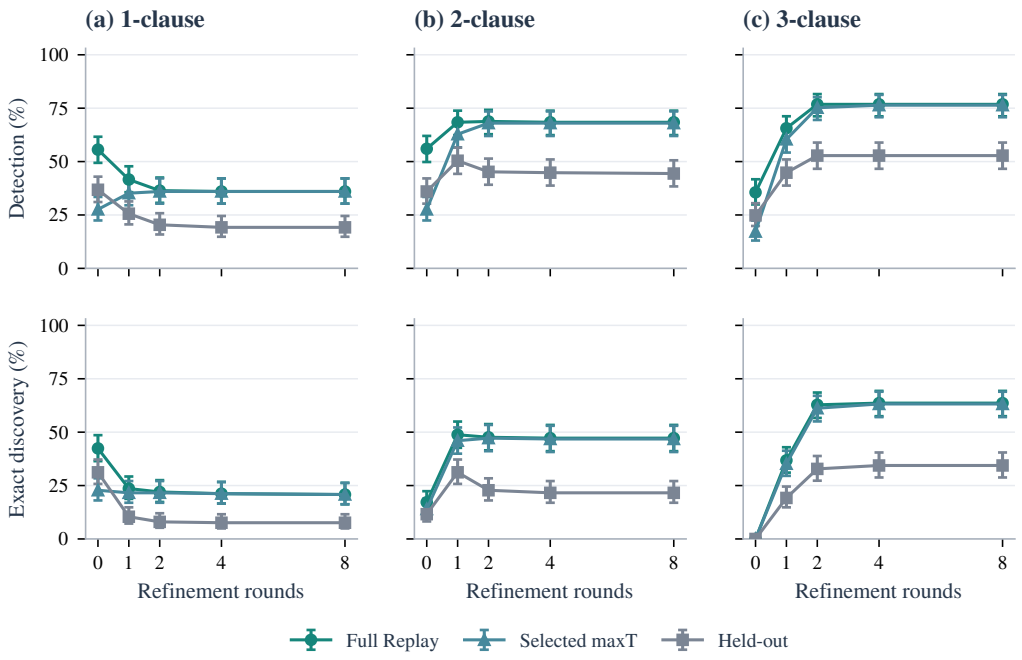

Figure 6: **Interaction depth redistributes discovery ability.** At medium signal, additional feedback improves power and exact recovery for two- and three-clause mechanisms while reallocating search away from one-clause mechanisms. At shallow depth, Transcript Replay is less conservative than selected maxT correction. The two converge as the reachable bank expands. Held-out testing reserves half the rows. Bars show 95% Wilson intervals.

E PATH-STABILITY AND BANK DIAGNOSTICS

We observe two consequences of constructing paths from outcomes (Figure 8). First, overlap among acquired rows is not a calibration surrogate. At one acquisition round, the $\rho = 0$ and $\rho = 0.6$ policies have mean Jaccard overlap of 0.753 and 0.741 between observed and reference paths. Yet freezing those paths rejects 3.4% and 59.6% of 500 null worlds. Increasing the number of rounds lowers overlap for every ρ, but the rejection rate is governed by how target outcomes enter the selection map, not by a universal Jaccard threshold.

Second, enlarging a frozen rule bank does not move its max-test p-value in a single direction. Every one of the 1,134 local pseudo-observed null banks grew beyond round zero, by 5.48 rules on average. The final-bank value was higher in 682 views, lower in 241, and unchanged in 211. Both the observed maximum and randomized reference maxima changed. We therefore assess reuse with the exact eventwise cacheability criterion in Appendix A.3. Safe caching depends on the executable map, not a visual similarity or bank-size heuristic.

F DIRECT LOCAL VALIDATION

F.1 OUTCOME CONSTRUCTION AND PERMUTATION COMPATIBILITY

Let $s(i)$ denote the fixed search or held-out stratum of row i. Let $r_{s,(1)}, \ldots, r_{s,(n_s)}$ be its sorted empirical outcome values. For an untied continuous vector w, define

$$P_s(w)_i = r_{s,\mathrm{rank}_s(w_i)}.$$

This rank projection is equivariant to permutations within each stratum. Null queries project independent Gaussian vectors. For an alternative,

$$w_i = \mathrm{std}_s(\epsilon)_i + \beta\, \mathrm{std}_s(f(X))_i, \qquad Y = P_s(w),$$

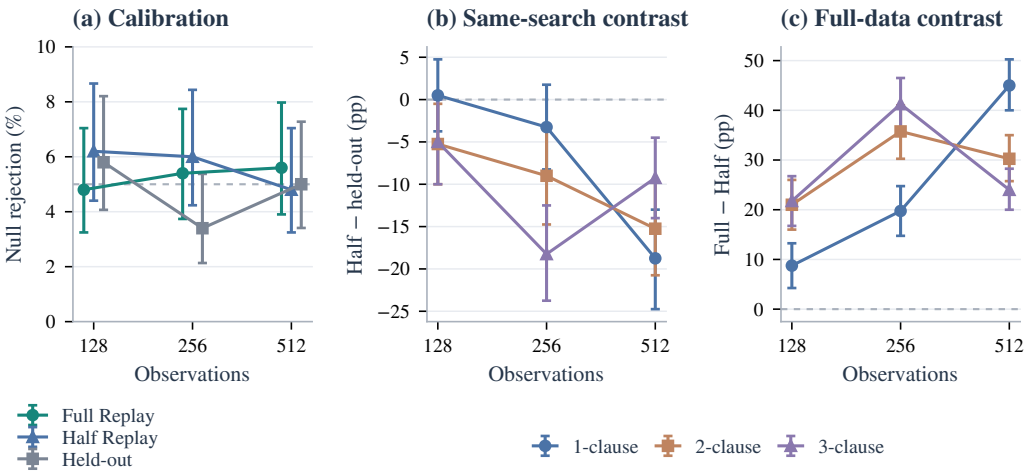

Figure 7: **Matched decomposition of data use.** (a) Full Replay, search-matched Half Replay, and the 50:50 split remain near nominal. (b) Half-minus-held-out power holds the search half fixed. (c) Full-minus-Half changes search and scoring samples together. Null points use 500 worlds. Each alternative point uses 400 matched worlds at medium signal. Bars show 95% Wilson intervals (a) and paired-bootstrap intervals (b, c).

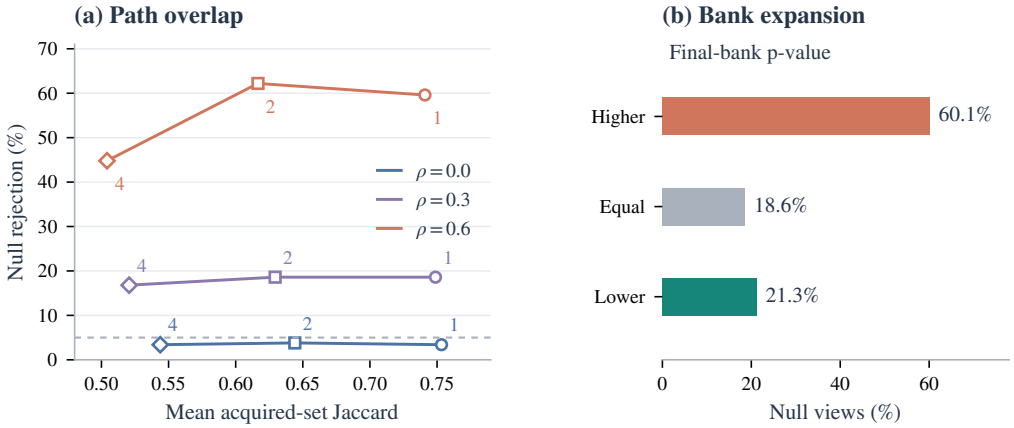

Figure 8: **Path overlap and bank expansion.** (a) Each line fixes the correlation between proxy and outcome and varies acquisition rounds $R \in \{1, 2, 4\}$ (point labels). Mean acquired-set overlap can be nearly identical while frozen-path null rejection differs sharply. (b) Comparing the final observed bank with its round-0 prefix over 1,134 pseudo-observed local null views gives both higher and lower max-test p-values despite nonnegative bank growth. The dashed line marks 5%.

where standardization uses the population standard deviation within the stratum. For threshold rules, $f(X)$ is the rule indicator. For the Python study, it is one of the continuous mechanisms in Table 16. The signal strengths are 0.25 and 0.40 for rule search and 0.40 and 0.70 for program search, before projection. We use these planted mechanisms to define controlled queries on the fixed scientific covariates.

For each reference bank, we draw 39 independent permutations of the empirical values within the same strata. Distinct random streams generate the reference bank, null queries, and each query defined by mechanism and signal strength. The corresponding arms share their outcome vectors

Table 9: Planted-signal power across all conditions in the three-arm matched-observation study (%). Each row contains 400 matched worlds. Differences are paired. The released analysis includes their 95% intervals.

Total n	Clauses	Strength	Full	Half	Split	Full−Split	Half−Split	Full−Half
128	1	0.7	8.0	4.8	7.2	0.75	-2.50	3.25
128	1	1.0	20.0	11.2	10.8	9.25	0.50	8.75
128	1	1.3	28.0	11.0	18.8	9.25	-7.75	17.00
128	2	0.7	16.0	9.0	7.8	8.25	1.25	7.00
128	2	1.0	35.2	14.2	19.5	15.75	-5.25	21.00
128	2	1.3	57.8	24.2	33.5	24.25	-9.25	33.50
128	3	0.7	19.0	8.5	10.5	8.50	-2.00	10.50
128	3	1.0	38.8	17.0	22.0	16.75	-5.00	21.75
128	3	1.3	69.0	24.2	43.8	25.25	-19.50	44.75
256	1	0.7	15.2	10.2	12.5	2.75	-2.25	5.00
256	1	1.0	37.5	17.8	21.0	16.50	-3.25	19.75
256	1	1.3	60.2	27.5	38.2	22.00	-10.75	32.75
256	2	0.7	37.0	17.0	22.5	14.50	-5.50	20.00
256	2	1.0	69.2	33.5	42.5	26.75	-9.00	35.75
256	2	1.3	97.0	60.5	76.8	20.25	-16.25	36.50
256	3	0.7	39.5	17.2	23.5	16.00	-6.25	22.25
256	3	1.0	78.2	37.0	55.2	23.00	-18.25	41.25
256	3	1.3	98.2	67.2	84.2	14.00	-17.00	31.00
512	1	0.7	37.2	17.2	21.8	15.50	-4.50	20.00
512	1	1.0	79.0	34.0	52.8	26.25	-18.75	45.00
512	1	1.3	96.2	64.2	75.5	20.75	-11.25	32.00
512	2	0.7	70.0	32.5	48.2	21.75	-15.75	37.50
512	2	1.0	97.5	67.2	82.5	15.00	-15.25	30.25
512	2	1.3	100.0	94.0	97.8	2.25	-3.75	6.00
512	3	0.7	79.2	37.5	52.5	26.75	-15.00	41.75
512	3	1.0	99.5	75.5	84.8	14.75	-9.25	24.00
512	3	1.3	100.0	97.5	98.8	1.25	-1.25	2.50

Table 10: Fixed split-ratio sensitivity at the primary controlled signal strength. We compare Full Replay with the pointwise best of the fixed 25:75, 50:50, and 75:25 allocations. Every row favors Full Replay.

Total n	Clauses	Full	Best split	Search-to-evidence	Difference (pp)
128	1	20.00	11.00	25:75	+9.00
128	2	35.25	19.50	50:50	+15.75
128	3	38.75	22.00	50:50	+16.75
256	1	37.50	21.00	50:50	+16.50
256	2	69.25	42.50	50:50	+26.75
256	3	78.25	55.25	50:50	+23.00
512	1	79.00	52.75	50:50	+26.25
512	2	97.50	82.50	50:50	+15.00
512	3	99.50	84.75	50:50	+14.75

across models, seeds, and data views. Because every query and reference has identical sorted values in each stratum, sorting identifies an explicit permutation that transports one to the other. The implementation checks this compatibility before reuse.

There are 144 distinct numeric null-query vectors and 72 numeric planted-query vectors across the two tasks and six banks. The Python study has 96 null and 32 planted vectors across two tasks and four banks. Replications across models, program seeds, and data views expand these to 2,240 query executions. The other 4,992 executions construct the independent reference banks. These counts distinguish generated query data from repeated scientist executions on the same data.

Table 11: Null calibration across all proxy-guided acquisition settings (%). The 12 reported $\rho \times R$ rows represent 10 operationally distinct acquisition settings. The $R = 0$ design is shared across ρ. Each entry gives a rejection rate and 95% Wilson interval over 500 shared null worlds. All tests use 255 randomized references. R is acquisition depth and n is the number of measured rows.

ρ	R	n	Final-only	Frozen bank	Frozen acquisition	Full Replay
0.0	0	64	97.6 [95.9, 98.6]	44.0 [39.7, 48.4]	3.6 [2.3, 5.6]	3.6 [2.3, 5.6]
0.0	1	80	92.4 [89.7, 94.4]	34.6 [30.6, 38.9]	3.4 [2.1, 5.4]	4.0 [2.6, 6.1]
0.0	2	96	90.6 [87.7, 92.9]	30.8 [26.9, 35.0]	3.8 [2.4, 5.9]	4.8 [3.2, 7.0]
0.0	4	128	90.4 [87.5, 92.7]	31.6 [27.7, 35.8]	3.4 [2.1, 5.4]	4.6 [3.1, 6.8]
0.3	0	64	97.6 [95.9, 98.6]	44.0 [39.7, 48.4]	3.6 [2.3, 5.6]	3.6 [2.3, 5.6]
0.3	1	80	99.0 [97.7, 99.6]	70.4 [66.3, 74.2]	18.6 [15.4, 22.2]	3.4 [2.1, 5.4]
0.3	2	96	98.2 [96.6, 99.1]	68.0 [63.8, 71.9]	18.6 [15.4, 22.2]	3.4 [2.1, 5.4]
0.3	4	128	97.4 [95.6, 98.5]	64.8 [60.5, 68.9]	16.8 [13.8, 20.3]	5.2 [3.6, 7.5]
0.6	0	64	97.6 [95.9, 98.6]	44.0 [39.7, 48.4]	3.6 [2.3, 5.6]	3.6 [2.3, 5.6]
0.6	1	80	100.0 [99.2, 100.0]	95.8 [93.7, 97.2]	59.6 [55.2, 63.8]	3.8 [2.4, 5.9]
0.6	2	96	99.8 [98.9, 100.0]	95.6 [93.4, 97.1]	62.2 [57.9, 66.3]	4.8 [3.2, 7.0]
0.6	4	128	98.6 [97.1, 99.3]	79.0 [75.2, 82.3]	44.8 [40.5, 49.2]	3.8 [2.4, 5.9]

Table 12: Full Replay power across all planted-signal conditions under the $\rho = 0.6$ proxy (%). We report power at every frozen acquisition depth. The paired difference Δ compares four rounds with zero. We show exact discovery and mask overlap at four rounds.

Clauses	Strength	Power $R = 0$	Power $R = 2$	Power $R = 4$	Δ [95% CI]	Exact	Jaccard
1	1.0	8.8	14.4	17.2	+8.4 [+3.2, +13.6]	7.2	39.2
1	1.3	14.4	22.4	28.8	+14.4 [+8.4, +20.4]	12.0	46.5
2	1.0	14.8	32.8	37.2	+22.4 [+16.4, +28.4]	21.2	54.1
2	1.3	19.2	44.8	57.6	+38.4 [+31.6, +44.8]	34.4	62.5
3	1.0	10.0	38.4	38.8	+28.8 [+22.8, +34.8]	19.6	54.2
3	1.3	31.2	68.0	75.6	+44.4 [+37.2, +51.2]	60.8	78.4

Table 13: Acquisition-path diagnostics under the null. Exact gives the mean fraction of randomized references that reproduce the observed acquired-row set. Jaccard measures set overlap.

ρ	Rounds	Exact path (%)	Mean Jaccard
0.0	0	100.0	1.000
0.0	1	0.4	0.753
0.0	2	0.0	0.644
0.0	4	0.0	0.544
0.3	0	100.0	1.000
0.3	1	0.0	0.749
0.3	2	0.0	0.629
0.3	4	0.0	0.521
0.6	0	100.0	1.000
0.6	1	0.0	0.741
0.6	2	0.0	0.617
0.6	4	0.0	0.504

For the Llama extension, we reused the same input vectors for the first program seed and separately executed 624 references and 256 queries. This adds 880 scientist executions, with eight task-specific banks for uncertainty estimation. The combined studies contain 8,112 executions on 28 cells covering models, tasks, seeds, and data views.

Table 14: Outcome-invariant uniform acquisition null control. Frozen-acquisition replay and Full Replay are pointwise identical because the acquired-row path does not depend on randomized outcomes. Entries are rejection rates with 95% Wilson intervals over 500 worlds.

Rounds	Target rows	Frozen acquisition	Full Replay
0	64	3.6 [2.3, 5.6]	3.6 [2.3, 5.6]
1	80	4.2 [2.8, 6.3]	4.2 [2.8, 6.3]
2	96	6.2 [4.4, 8.7]	6.2 [4.4, 8.7]
4	128	6.2 [4.4, 8.7]	6.2 [4.4, 8.7]

Table 15: Full Replay power under matched acquisition policies (%). $R = 0$ is the shared 64-row design. All $R = 4$ policies add 64 target rows. Differences are paired percentage points with 95% bootstrap intervals over 250 shared worlds.

A. Detection rates (%)

Clauses	Strength	Shared $R = 0$	Uniform	Subgroup	Proxy
1	1.0	8.8	18.4	24.8	17.2
1	1.3	14.4	32.4	30.8	28.8
2	1.0	14.8	31.2	49.6	37.2
2	1.3	19.2	53.2	74.8	57.6
3	1.0	10.0	38.8	65.2	38.8
3	1.3	31.2	71.6	92.8	75.6

B. Paired differences (pp) [95% CI]

Clauses	Strength	Subgroup − uniform	Proxy − subgroup
1	1.0	+6.4 [+0.0, +12.8]	-7.6 [-13.2, -2.0]
1	1.3	-1.6 [-8.4, +5.2]	-2.0 [-8.0, +4.0]
2	1.0	+18.4 [+10.8, +25.6]	-12.4 [-19.2, -5.6]
2	1.3	+21.6 [+14.0, +28.8]	-17.2 [-23.6, -11.2]
3	1.0	+26.4 [+18.8, +34.0]	-26.4 [-33.2, -19.6]
3	1.3	+21.2 [+15.2, +27.2]	-17.2 [-22.0, -12.4]

Table 16: Mechanisms fixed before the direct local validation. Rule quantiles are computed from the fixed covariate catalog.

Task	Family	Mechanism
Airfoil	1 clause	frequency $\geq q_{75}$
	2 clauses	frequency $\geq q_{50}$ and attack angle $\geq q_{50}$
	3 clauses	the two-clause rule and chord length $\leq q_{50}$
	Transform	$\log(1 + \text{frequency})$
	Interaction	$\log(1 + \text{frequency}) \times \text{attack angle}$
Concrete	1 clause	age $\geq q_{50}$
	2 clauses	cement $\geq q_{50}$ and age $\geq q_{50}$
	3 clauses	the two-clause rule and water $\leq q_{50}$
	Transform	$\log(1 + \text{age})$
	Interaction	$\log(1 + \text{age}) \times \text{cement}/\max(\text{water}, 10^{-6})$

F.2 Python feature programs and coefficient fitting

The agent returns a JSON object containing a Python function `basis(X)`, a short rationale, and a stop flag. The function may use assignments, array indexing, arithmetic, comparisons, and declared NumPy transforms, including logarithms, powers, trigonometric functions, reductions, and column stacking. A syntax validator admits this bounded numerical language. The language supports an unenumerated space of feature programs with a fixed execution interface and at most 12 returned

columns. We normalize array reductions such as `X.min(axis=0)` to their NumPy function equivalents.

Feature-only transformations use all rows of the fixed covariate table. We then select the evaluation view and its four folds. Fold identifiers cycle by row index and are independent of outcomes. Within each training fold, we center and scale the features, then fit ridge regression with penalty 1 and an unpenalized intercept. The statistic is the out-of-fold R^2 over the full evaluation view. This fitting procedure is identical for all generated candidates. A linear original-column model is present in every bank.

Each scientist has three rounds and at most one repair after a rejected program. We return execution errors as feedback. The model can stop after the second round. Its final program maximizes the evaluated score, and a canonical source identifier breaks ties. We retain the complete bank and generated responses. The frozen-program comparator keeps the selected feature source and refits numerical coefficients under each reference outcome. The frozen-bank comparator additionally maximizes over all source programs admitted in the query run. For held-out evaluation, the Half scientist's selected program is fitted on its 128 search outcomes. We compare its predictions against permutations of the 128 held-out outcomes.

Model interfaces. Both Python scientists use temperature 0.6, top-p 0.9, top-k 40, and 512 new tokens per call. Qwen uses program seeds 2026090711 and 2026090712. Llama uses the first seed. Llama-3.1-8B-Instruct uses its native chat template and a fixed reminder of the JSON format and Python function signature. The reminder supplies the original-column baseline as a syntax example. It adds no task-specific transform. We insert native special tokens once and omit the Qwen-specific non-thinking suffix. The response source passes unchanged to the common compiler. These interface settings, together with the common generation, repair, and stopping budgets, define the Llama scientist used for both queries and references.

Table 17 describes the realized program workflows after the frozen inference. It includes rejected attempts and the fixed linear baseline in the full execution inventory.

Table 17: Descriptive execution characteristics of the Python workflows, including every reference and query run. A model program is a newly admitted feature function. Rejected attempts include parsing, validation, and execution failures.

Scientist	Runs	At least one model program (%)	Model source selected (%)	Rejected attempts (%)
Qwen3-8B	1760	50.9	38.8	73.2
Llama-3.1-8B	880	56.5	37.3	71.8

Selected-source composition. Table 18 decomposes the Full null queries by their selected source. Every baseline-selected query has reference worlds that select a model-written program, so baseline retention varies across the sampled orbit. These groups describe observed selection outcomes, while the full query set remains the calibration target.

Table 18: Full-data null-query rejections grouped by the source selected on the query. Rejection cells give counts (percentages). The groups are defined after search.

Scientist	Selected source	Queries	Final-only	Frozen bank	Full Replay
Qwen3-8B	Linear baseline	119	8 (6.7)	5 (4.2)	3 (2.5)
Qwen3-8B	Model program	73	9 (12.3)	6 (8.2)	4 (5.5)
Llama-3.1-8B	Linear baseline	65	4 (6.2)	3 (4.6)	2 (3.1)
Llama-3.1-8B	Model program	31	3 (9.7)	2 (6.5)	1 (3.2)

F.3 Analysis, uncertainty, and real-target illustrations

The primary calibration endpoint is rejection at $\alpha = 0.05$ among null queries independent of their reference bank. Additional CDF points are 0.025, 0.10, and 0.20. The primary detection comparison

averages paired Full-minus-held-out rejection differences equally over the prespecified combinations of mechanism and signal strength. Full-minus-Half compares the joint change in search and scoring samples. Full-minus-HF fixes full-data scoring (Appendix G). Numerical differences within 10^{-10} count against rejection.

We drew reference banks within each task in ten thousand bootstrap resamples, retaining all matched observations across queries, models, seeds, and data views. There are 12 task-specific banks in the numeric study and eight in the Python study. These banks are the independent resampling units within the fixed task, model and seed grid. Queries sharing a bank use the same reference-score tail. Independent numerical checks are summarized in Appendix M.

Score ties and reference sampling. The Full null queries comprise 576 rule queries, 192 Qwen program queries, and 96 Llama program queries. None ties a reference score within 10^{-10}. Uniform tie ordering therefore retains the 17, 7, and 3 rejections, respectively. With 39 references, 5% is an attainable marginal level over reference and query draws. The empirical rates describe the realized bank grid. A post hoc role-label check reallocates the 51 saved null executions within each outcome bank into 39 references and 12 queries, keeping paired scientists aligned. Its central 95% ranges are 2.60–7.99%, 1.04–9.90%, and 1.04–10.42% across 20,000 reallocations. Bank bootstrap resamples the observed task-bank units to estimate uncertainty in mean rejection, whereas role reallocation holds each 51-score pool fixed and varies which scores serve as references. The resulting ranges describe reference-role variation, while bank-bootstrap intervals quantify uncertainty across the observed task-bank units.

We analyze original measured-target queries separately from the null and planted matrices. For a fixed scientist, their source scores use pooled compatible banks with 234 numeric or 156 program references. Outer selection maximizes over the aligned combinations of model and program seed separately within each task, scientist type, and data view. Checks in a fresh process transport a prespecified null query exactly to a stored reference and compare the resulting execution. Real-target illustrations are not counted as independent validation trials.

Tables 19 and 22 give null-query rejection rates. Tables 20 and 23 retain every planted condition. Tables 21 and 24 show the task, model, and seed breakdowns.

Table 19: Rule program: null-query p-value CDFs (%). The interval at $\alpha = 0.05$ is the 95% bank-bootstrap interval.

View	Method	$\alpha = 0.025$	$\alpha = 0.05$ [95% CI]	$\alpha = 0.10$	$\alpha = 0.20$
Full	Final-only	43.2	64.8 [60.4, 69.1]	80.9	94.3
Full	Frozen bank	9.4	17.9 [14.2, 22.4]	36.5	58.2
Full	Selected maxT	0.0	0.0 [0.0, 0.0]	0.3	0.9
Full	Replay	1.0	3.0 [1.2, 4.9]	6.8	20.8
Half	Final-only	46.5	63.4 [58.9, 67.4]	83.9	94.4
Half	Frozen bank	12.3	23.4 [19.1, 27.8]	37.3	58.7
Half	Selected maxT	0.3	0.3 [0.0, 1.0]	1.7	3.0
Half	Held-out	1.2	4.2 [1.6, 7.1]	8.7	17.4
Half	Replay	2.8	5.7 [3.1, 8.9]	8.5	17.9

Table 20: Rule program: detection rates and paired differences for every prespecified planted condition.

Mechanism	Strength	Full	Half	Held-out	Full − held-out [95% interval]
1-clause	0.25	79.2	35.4	58.3	20.8 [6.2, 37.5]
1-clause	0.40	100.0	75.0	83.3	16.7 [2.1, 33.3]
2-clause	0.25	62.5	27.1	37.5	25.0 [-2.1, 50.0]
2-clause	0.40	95.8	52.1	58.3	37.5 [22.9, 52.1]
3-clause	0.25	47.9	16.7	33.3	14.6 [-8.3, 39.6]
3-clause	0.40	68.8	41.7	39.6	29.2 [10.4, 47.9]

Table 25 reports Llama null-query CDFs. Table 26 gives every planted condition, and Table 27 separates the two tasks. Full and held-out rejection decisions agree for all 32 Llama pairs. Resampling

Table 21: Rule program: task, model, and seed breakdown. Entries give rejection percentages. Power averages all declared planted conditions. Seed labels follow the configured order.

Model	Task	Seed	Full null	Full power	Half power	Held-out power
Qwen3-4B	Airfoil	1	2.8	58.3	16.7	44.4
Qwen3-4B	Airfoil	2	4.2	63.9	33.3	41.7
Qwen3-4B	Concrete	1	1.4	69.4	47.2	50.0
Qwen3-4B	Concrete	2	4.2	77.8	58.3	66.7
Qwen3-8B	Airfoil	1	1.4	83.3	38.9	41.7
Qwen3-8B	Airfoil	2	4.2	77.8	38.9	38.9
Qwen3-8B	Concrete	1	2.8	83.3	50.0	66.7
Qwen3-8B	Concrete	2	2.8	91.7	47.2	63.9

Table 22: Python program: null-query p-value CDFs (%). The interval at $\alpha = 0.05$ is the 95% bank-bootstrap interval.

View	Method	$\alpha = 0.025$	$\alpha = 0.05$ [95% CI]	$\alpha = 0.10$	$\alpha = 0.20$
Full	Final-only	2.6	8.9 [3.6, 14.6]	17.2	25.5
Full	Frozen bank	1.0	5.7 [2.1, 9.9]	11.5	22.4
Full	Replay	1.6	3.6 [1.0, 7.8]	10.9	19.8
Half	Final-only	3.1	4.2 [1.0, 7.3]	7.3	18.2
Half	Frozen bank	1.6	3.6 [1.0, 7.3]	4.7	12.5
Half	Held-out	3.6	5.7 [2.6, 9.4]	11.5	16.7
Half	Replay	1.6	2.6 [0.5, 5.2]	5.7	13.0

Table 23: Python program: detection rates and paired differences for every prespecified planted condition.

Mechanism	Strength	Full	Half	Held-out	Full − held-out [95% interval]
Interaction	0.40	100.0	87.5	93.8	6.2 [0.0, 18.8]
Interaction	0.70	100.0	93.8	100.0	0.0 [0.0, 0.0]
Transform	0.40	81.2	75.0	81.2	0.0 [0.0, 0.0]
Transform	0.70	100.0	93.8	100.0	0.0 [0.0, 0.0]

Table 24: Python program: task, model, and seed breakdown. Entries give rejection percentages. Power averages all declared planted conditions. Seed labels follow the configured order.

Model	Task	Seed	Full null	Full power	Half power	Held-out power
Qwen3-8B	Airfoil	1	6.2	100.0	100.0	100.0
Qwen3-8B	Airfoil	2	0.0	100.0	87.5	100.0
Qwen3-8B	Concrete	1	2.1	93.8	93.8	93.8
Qwen3-8B	Concrete	2	6.2	87.5	68.8	81.2

these observed zero differences gives a zero-width bootstrap interval. It describes the sample and does not establish population equivalence.

Table 25: Llama-3.1-8B Python scientist: null-query p-value CDFs (%). The interval at $\alpha = 0.05$ is a 95% bank-bootstrap interval.

View	Method	$\alpha = 0.025$	$\alpha = 0.05$ [95% CI]	$\alpha = 0.10$	$\alpha = 0.20$
Full	Final-only	2.1	7.3 [2.1, 13.5]	11.5	22.9
Full	Frozen bank	0.0	5.2 [1.0, 11.5]	9.4	14.6
Full	Replay	0.0	3.1 [0.0, 7.3]	12.5	18.8
Half	Final-only	3.1	4.2 [1.0, 8.3]	6.2	14.6
Half	Frozen bank	1.0	3.1 [0.0, 6.2]	6.2	12.5
Half	Held-out	3.1	6.2 [3.1, 10.4]	12.5	18.8
Half	Replay	0.0	2.1 [0.0, 6.2]	6.2	11.5

Table 26: Llama-3.1-8B detection rates and paired power differences for all four declared combinations of mechanism and signal strength.

Mechanism	Strength	Full (%)	Half (%)	Held-out (%)
Transform	0.40	75.0	50.0	75.0
Transform	0.70	100.0	87.5	100.0
Interaction	0.40	100.0	87.5	100.0
Interaction	0.70	100.0	100.0	100.0

Mechanism	Strength	Full − held-out [95% CI]	Full − Half [95% CI]
Transform	0.40	0.0 [0.0, 0.0]	25.0 [-25.0, 75.0]
Transform	0.70	0.0 [0.0, 0.0]	12.5 [0.0, 37.5]
Interaction	0.40	0.0 [0.0, 0.0]	12.5 [0.0, 37.5]
Interaction	0.70	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]

Table 27: Task-level Llama-3.1-8B results for the fixed program seed. Power averages the four declared planted conditions.

Task	Full null (%)	Full power (%)	Half power (%)	Held-out power (%)
Airfoil	2.1	100.0	93.8	100.0
Concrete	4.2	87.5	68.8	87.5

F.4 An executable measured-target example

The following feature program was selected on Concrete by Qwen3-8B, seed 1, using all 256 rows. Its four-fold score is $R^2 = 0.787$ and its compatible-reference Replay value is $p = 0.0064$. The numerical evaluator fits ridge coefficients to these features. We explicitly chose this case to illustrate one selected program. The tables below include all measured-target cases.

```
def basis(X):
    slag_agg = X[:, 1] / (X[:, 5] + X[:, 6])
    ash_agg = X[:, 2] / (X[:, 5] + X[:, 6])
    water_agg = X[:, 3] / X[:, 5]
    sp_water = X[:, 4] / X[:, 3]
    log_age = np.log1p(X[:, 7])
    return np.column_stack([X[:, 0], slag_agg, ash_agg, water_agg, sp_water, log_age])
```

Tables 28 and 29 report all measured-target selections. Table 30 gives the aligned outer selectors. Table 31 gives the separate fresh-process checks.

G Supplementary matched-procedure comparisons

We specified these comparisons after the original results. They use all saved queries and references, with the same task catalogs and score definitions.

Table 28: Rule scientist on original rank-normalized targets. T is the selected standardized subgroup contrast. Each p-value uses 234 compatible references.

Model	Task	Seed	Half T	Half p	Full T	Full p
Qwen3-4B	Airfoil	1	5.579	0.0043	7.971	0.0043
Qwen3-4B	Airfoil	2	3.968	0.0043	7.786	0.0043
Qwen3-4B	Concrete	1	5.857	0.0043	8.013	0.0043
Qwen3-4B	Concrete	2	5.929	0.0043	7.856	0.0043
Qwen3-8B	Airfoil	1	-0.227	0.9915	1.491	0.7277
Qwen3-8B	Airfoil	2	1.119	0.8936	1.491	0.7106
Qwen3-8B	Concrete	1	5.929	0.0043	9.139	0.0043
Qwen3-8B	Concrete	2	5.929	0.0043	9.139	0.0043

Table 29: Python scientist on original rank-normalized targets. T is the selected cross-validated R^2. Each p-value uses 156 compatible references.

Model	Task	Seed	Half T	Half p	Full T	Full p
Qwen3-8B	Airfoil	1	0.535	0.0064	0.514	0.0064
Qwen3-8B	Airfoil	2	0.535	0.0064	0.514	0.0064
Qwen3-8B	Concrete	1	0.683	0.0064	0.787	0.0064
Qwen3-8B	Concrete	2	0.637	0.0064	0.591	0.0064

Table 30: Measured-target outer selection. Each maximization ranges over aligned combinations of model and program seed within one task, scientist type, and data view.

Scientist	Task	View	Variants	Maximum T	Replay p
Rule	Airfoil	Full	4	7.971	0.0043
Rule	Airfoil	Half	4	5.579	0.0043
Rule	Concrete	Full	4	9.139	0.0043
Rule	Concrete	Half	4	5.929	0.0043
Python	Airfoil	Full	2	0.514	0.0064
Python	Airfoil	Half	2	0.535	0.0064
Python	Concrete	Full	2	0.787	0.0064
Python	Concrete	Half	2	0.683	0.0064

Table 31: Fresh-process transport checks. The comparison uses the complete canonical JSON transcript and the original selected-statistic difference.

Model	Scientist	Transcript match	Score difference
Qwen3-4B	Rule	Yes	0
Qwen3-8B	Rule	Yes	0
Qwen3-8B	Python	Yes	0

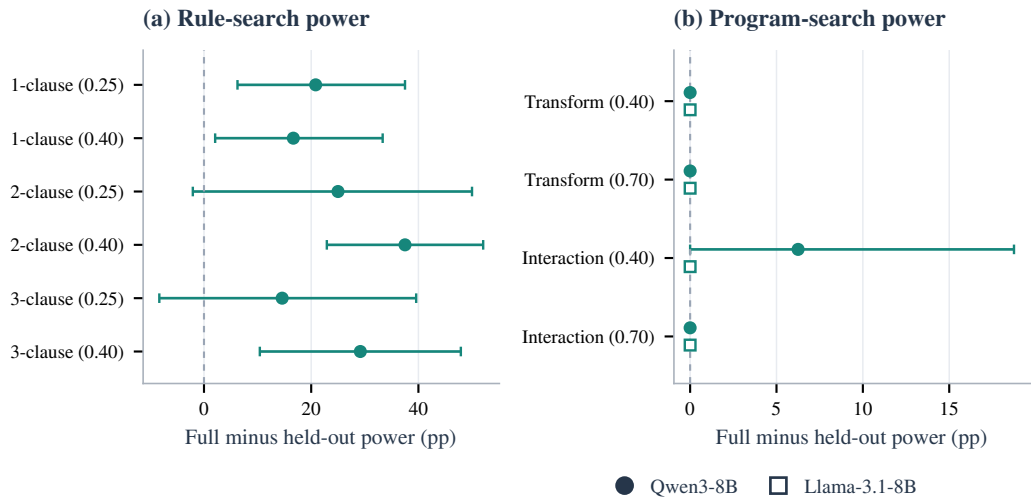

Figure 9: **Full-data versus held-out detection.** Paired power differences for all rule and program conditions, with signal strength in parentheses and 95% bank-bootstrap intervals. Circles and squares in (b) denote Qwen3-8B and Llama-3.1-8B. Dashed lines mark zero.

Score-matched search. HF is the fixed procedure $T_{\mathrm{HF}}(Y) = s_{\mathrm{full}}(A_{\mathrm{half}}(Y_{\mathrm{search}}), Y)$. Each reference scores its own Half-selected output on its complete outcome vector. For programs, we evaluate the selected feature function on the fixed covariate table and fit ridge coefficients within the full-data folds. We never reselect the Half bank using full labels.

Simple detectors and exhaustive discovery. The linear procedure uses the fixed original columns and the common ridge penalty, with coefficients refitted in each world. We evaluate this procedure on every query, not only those on which the LLM retained it, and its held-out version fits on the search half and scores the other half. Numeric exhaustive search selects the maximum over the full fixed rule catalog in the observed and reference worlds. The original selected-maxT comparator retains the LLM-selected observed score. We break ties between equal maxima by inventory order.

Endpoints and uncertainty. Table 34 gives null rejection and power. Table 35 gives all overall paired power contrasts. Tables 36, 37, and 38 preserve every planted condition. Table 39 reports numeric selected-mask recovery and Jaccard overlap. These compare the selected row mask with the declared planted mask, including observationally identical masks with different source indices. We resample independent task-bank units within task and retain paired scientists, seeds, queries and views. The supplementary bootstrap uses 10,000 draws and a fixed seed. The original null intervals are unchanged.

Panels c and d of Figure 4 show score-matched detection and recovery across all six rule conditions. Mask-recovery gains concentrate in one- and two-clause conditions. Figure 9 gives all Full-minus-held-out power contrasts summarized in Table 2. The machine-readable supplement also includes every predefined contrast and structural metric.

Robustness of the score-matched gains. Table 32 separates task, model, and program-seed margins. Both core differences have positive marginal estimates, with subgroup intervals reflecting the smaller samples. Omitting each task-bank unit preserves positive detection and exact-mask differences. Across the individual units, detection differences are positive in 10 of 12 banks, exact-mask differences are positive in 11 and tied in one, and Jaccard differences are positive in all 12. These summaries retain paired queries within the original task-bank hierarchy.

Table 33 varies the stored reference subset. Within each fixed task, scientist, and seed, the available pool contains 234 rule references or 156 program references. Twenty fixed random orders define nested subsets in which Full and HF use the same world identifiers, while queries remain outside

the reference pool. Rule power differences remain positive through $B = 199$. The Qwen program difference reaches zero at $B = 119$, while the Llama difference stays negative. Overlapping subsets provide descriptive sensitivity ranges. The original independent-bank intervals remain the uncertainty summary for the experiment.

Table 32: Full minus HF rule contrasts (pp). Brackets in the first six rows give 95% task-bank bootstrap intervals. The last row gives the range of point estimates after omitting each of the 12 task-bank units.

Subset	Detection difference	Exact-mask difference
Airfoil	14.6 [-1.4, 31.2]	17.4 [7.6, 31.2]
Concrete	8.3 [0.7, 15.3]	6.9 [2.8, 12.5]
Qwen3-4B	4.9 [-6.9, 17.4]	7.6 [0.0, 16.0]
Qwen3-8B	18.1 [3.5, 31.9]	16.7 [8.3, 25.7]
First program seed	9.7 [1.4, 20.1]	15.3 [8.3, 22.9]
Second program seed	13.2 [3.5, 21.5]	9.0 [0.7, 18.1]
Leave-one-bank-out range	[7.9, 14.6]	[8.9, 13.5]

Table 33: Stored-reference sensitivity across 20 fixed nested subset orders. Null columns give mean rejection rates (%). The last column gives the mean Full minus HF power difference (pp) and its range across subsets.

Workflow	B	Full null	HF null	Power difference [range]
Rules	39	3.6	4.2	12.0 [6.9, 18.8]
	79	3.2	3.8	11.6 [8.7, 14.9]
	119	3.3	4.0	11.1 [9.0, 13.9]
	159	3.4	4.1	10.9 [8.7, 13.5]
	199	3.5	4.3	10.8 [9.0, 12.5]
Python: Qwen	39	3.6	3.2	-0.5 [-1.6, 1.6]
	79	3.4	3.8	-0.4 [-1.6, 0.0]
	119	3.8	5.1	0.0 [0.0, 0.0]
Python: Llama	39	3.4	2.7	-2.8 [-3.1, 0.0]
	79	2.5	2.3	-3.1 [-3.1, -3.1]
	119	3.0	2.6	-3.1 [-3.1, -3.1]

Controlled reanalysis. Table 40 uses all 6,100 unique original controlled worlds, with the original 255 references and an independently selected exhaustive maximum. Table 41 changes only the reference count for fixed-statistic diagnostics on the 1,000 depth-family null worlds. Its first 255 draws are the original reference draws. This high-reference analysis measures fixed-statistic resolution on the original null queries. Appendix M summarizes its verification.

Execution-time scope. Table 42 summarizes time measured inside saved scientist runs. For M compatible queries and 39 references, independent reference generation uses $40M$ executions, whereas shared references use $39 + M$. A cold single query therefore needs 40 executions in either implementation. For $M = 4$ and $M = 16$, the shared counts are 43 and 55, compared with 160 and 640. These counts measure workload. Model loading, scheduling and serving determine the corresponding end-to-end time.

H External symbolic-regression study

H.1 Tasks, scientist and data allocation

We used BPG1 (biological population growth) and PO1 (physical oscillation) from the LSR-Synth component of LLM-SRBench (Shojaee et al., 2025b). We selected these two synthetic-system cases from their metadata before formal generation. Each has two input variables and 4,000 training rows. We selected 256 training rows and fixed a 128-row search half and 128-row held-out half. We assessed predictions on the author's separate 500-row ID and 500-row OOD sets. Their labels are used only for evaluation, entering neither search nor model selection.

Table 34: Local comparator endpoints: null rejection and power (%), with 95% task-stratified bank-bootstrap intervals. Shared linear and exhaustive outputs are paired with each scientist.

Workflow	Procedure	Null rejection	Power
Rule (Qwen3)	Full	3.0 [1.2, 4.9]	75.7 [70.1, 82.3]
	HF	4.9 [2.3, 7.6]	64.2 [59.7, 69.1]
	Half	5.7 [3.1, 8.9]	41.3 [33.0, 49.3]
	Held-out	4.2 [1.6, 7.1]	51.7 [45.5, 58.0]
	Exhaustive (full)	3.5 [0.7, 6.2]	84.7 [76.4, 93.1]
	Exhaustive (half)	4.2 [1.4, 6.9]	50.0 [38.9, 61.1]
	Selected maxT	0.0 [0.0, 0.0]	49.7 [42.4, 57.6]
Python (Qwen3-8B)	Full	3.6 [1.0, 7.8]	95.3 [90.6, 100.0]
	HF	5.2 [1.0, 9.4]	96.9 [93.8, 100.0]
	Half	2.6 [0.5, 5.2]	87.5 [79.7, 95.3]
	Held-out	5.7 [2.6, 9.4]	93.8 [89.1, 98.4]
	Linear (full)	5.2 [1.0, 11.5]	96.9 [90.6, 100.0]
	Linear (half)	2.1 [0.0, 4.2]	75.0 [65.6, 84.4]
	Linear (held-out)	7.3 [3.1, 11.5]	93.8 [87.5, 100.0]
Python (Llama-3.1-8B)	Full	3.1 [0.0, 7.3]	93.8 [87.5, 100.0]
	HF	3.1 [0.0, 6.2]	96.9 [90.6, 100.0]
	Half	2.1 [0.0, 6.2]	81.2 [71.9, 93.8]
	Held-out	6.2 [3.1, 10.4]	93.8 [87.5, 100.0]
	Linear (full)	5.2 [1.0, 11.5]	96.9 [90.6, 100.0]
	Linear (half)	2.1 [0.0, 4.2]	75.0 [65.6, 84.4]
	Linear (held-out)	7.3 [3.1, 11.5]	93.8 [87.5, 100.0]

Table 35: Overall paired power contrasts (pp), with 95% bank-bootstrap intervals. The last column counts positive and negative paired rejection differences. We report workflows separately.

Workflow	Contrast	Difference [95% CI]	Signs
Rule (Qwen3)	Full − Half (original)	34.4 [25.0, 45.1]	111, 12
	Full − HF	11.5 [2.8, 20.5]	54, 21
	HF − Half	22.9 [16.0, 30.9]	75, 9
	Exhaustive (full) − Full	9.0 [-1.0, 19.4]	49, 23
	Exhaustive (half) − Half	8.7 [-0.3, 18.1]	46, 21
Python (Qwen3-8B)	Full − Half (original)	7.8 [-1.6, 17.2]	7, 2
	Full − HF	-1.6 [-6.2, 3.1]	1, 2
	HF − Half	9.4 [-1.6, 18.8]	7, 1
	Full − Linear (full)	-1.6 [-4.7, 0.0]	0, 1
	Half − Linear (half)	12.5 [3.1, 21.9]	10, 2
	Held-out − Linear (held-out)	0.0 [-4.7, 4.7]	1, 1
Python (Llama-3.1-8B)	Full − Half (original)	12.5 [-3.1, 25.0]	5, 1
	Full − HF	-3.1 [-9.4, 0.0]	0, 1
	HF − Half	15.6 [6.2, 25.0]	5, 0
	Full − Linear (full)	-3.1 [-9.4, 0.0]	0, 1
	Half − Linear (half)	6.2 [0.0, 12.5]	3, 1
	Held-out − Linear (held-out)	0.0 [0.0, 0.0]	0, 0

Table 36: Matched data-use contrasts by planted condition (pp), with 95% bank-bootstrap intervals.

Scientist and mechanism	Strength	Full − HF	HF − Half
Qwen 1-clause	0.25	4.2 [-8.3, 18.8]	39.6 [18.8, 62.5]
Qwen 1-clause	0.40	4.2 [0.0, 10.4]	20.8 [6.2, 33.3]
Qwen 2-clause	0.25	18.8 [-2.1, 39.6]	16.7 [4.2, 33.3]
Qwen 2-clause	0.40	20.8 [10.4, 33.3]	22.9 [6.2, 39.6]
Qwen 3-clause	0.25	6.2 [-20.8, 33.3]	25.0 [8.3, 43.8]
Qwen 3-clause	0.40	14.6 [-4.2, 33.3]	12.5 [-10.4, 35.4]

Table 37: Matched data-use contrasts by planted condition (pp), with 95% bank-bootstrap intervals.

Scientist and mechanism	Strength	Full − HF	HF − Half
Qwen Interaction	0.40	6.2 [0.0, 18.8]	6.2 [-12.5, 25.0]
Qwen Interaction	0.70	0.0 [0.0, 0.0]	6.2 [0.0, 18.8]
Qwen Transform	0.40	-12.5 [-25.0, 0.0]	18.8 [0.0, 43.8]
Qwen Transform	0.70	0.0 [0.0, 0.0]	6.2 [0.0, 18.8]
Llama Interaction	0.40	0.0 [0.0, 0.0]	12.5 [0.0, 37.5]
Llama Interaction	0.70	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]
Llama Transform	0.40	-12.5 [-37.5, 0.0]	37.5 [12.5, 75.0]
Llama Transform	0.70	0.0 [0.0, 0.0]	12.5 [0.0, 37.5]

Table 38: Program-scientist minus fixed-linear power by condition (pp) for all data views.

Scientist and mechanism	Strength	Full	Half	Held-out
Qwen Interaction	0.40	0.0 [0.0, 0.0]	12.5 [-6.2, 37.5]	-6.2 [-18.8, 0.0]
Qwen Interaction	0.70	0.0 [0.0, 0.0]	6.2 [0.0, 18.8]	0.0 [0.0, 0.0]
Qwen Transform	0.40	-6.2 [-18.8, 0.0]	25.0 [0.0, 43.8]	6.2 [0.0, 18.8]
Qwen Transform	0.70	0.0 [0.0, 0.0]	6.2 [0.0, 18.8]	0.0 [0.0, 0.0]
Llama Interaction	0.40	0.0 [0.0, 0.0]	12.5 [0.0, 37.5]	0.0 [0.0, 0.0]
Llama Interaction	0.70	0.0 [0.0, 0.0]	12.5 [0.0, 37.5]	0.0 [0.0, 0.0]
Llama Transform	0.40	-12.5 [-37.5, 0.0]	0.0 [-25.0, 25.0]	0.0 [0.0, 0.0]
Llama Transform	0.70	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]

Table 39: Local rule recovery before and after inferential rejection. Exactness means equality of the selected and planted row masks on the full covariate table. HF and Half select the same mask by construction.

Procedure	Exact mask (%)	Jaccard	Rejected and exact (%)
Full	33.7	0.617	33.0
HF	21.5	0.501	21.5
Half	21.5	0.501	16.0
Held-out	21.5	0.501	20.8
Exhaustive (full)	51.4	0.796	47.2
Exhaustive (half)	22.2	0.566	13.9

Table 40: True exhaustive global tests on all 6,100 unique controlled worlds, with 255 references. Each world selects its own catalog maximum. Values give rejection percentages [95% Wilson intervals]. Exact discovery also requires selection of the planted rule.

Family	Truth	Strength	Worlds	Rejection [95% CI]	Exact
Repair	1-clause	1.00	150	38.7 [31.2, 46.7]	17.3
Repair	2-clause	1.00	150	72.7 [65.0, 79.2]	52.7
Repair	3-clause	1.00	150	84.0 [77.3, 89.0]	71.3
Repair	Null	0.00	500	5.8 [4.1, 8.2]	
Depth	1-clause	0.70	250	17.6 [13.4, 22.8]	6.0
Depth	1-clause	1.00	250	36.0 [30.3, 42.1]	20.8
Depth	1-clause	1.30	250	66.0 [59.9, 71.6]	48.4
Depth	2-clause	0.70	250	38.0 [32.2, 44.2]	16.8
Depth	2-clause	1.00	250	68.0 [62.0, 73.5]	46.8
Depth	2-clause	1.30	250	94.8 [91.3, 96.9]	81.6
Depth	3-clause	0.70	250	41.6 [35.7, 47.8]	25.2
Depth	3-clause	1.00	250	76.8 [71.2, 81.6]	64.4
Depth	3-clause	1.30	250	98.4 [96.0, 99.4]	95.6
Depth	Null	0.00	1000	4.8 [3.6, 6.3]	
Stopping	1-clause	1.00	150	35.3 [28.1, 43.3]	18.0
Stopping	2-clause	1.00	150	80.0 [72.9, 85.6]	55.3
Stopping	3-clause	1.00	150	76.7 [69.3, 82.7]	64.0
Stopping	Null	0.00	500	5.4 [3.7, 7.7]	
Width	1-clause	1.00	150	36.7 [29.4, 44.6]	16.7
Width	2-clause	1.00	150	74.7 [67.2, 81.0]	50.7
Width	3-clause	1.00	150	76.7 [69.3, 82.7]	64.7
Width	Null	0.00	500	4.6 [3.1, 6.8]	

Table 41: Monte Carlo resolution diagnostic on the same 1,000 depth-family null worlds (%). Paired columns use B=255 and B=4,999. Only reference counts change. The original Full Replay runs retain B=255.

d	Final-only		Visible Bonf.		Selected maxT	
	255	4,999	255	4,999	255	4,999
0	55.4	59.8	18.6	20.6	0.7	0.8
1	89.3	92.3	23.1	36.0	3.1	3.2
2	95.5	96.8	29.3	35.6	4.6	4.8
4	96.2	97.8	0.0	25.6	4.7	4.9
8	96.3	97.9	0.0	14.9	4.7	4.9

Table 42: Mean scientist-execution time (s), pooling tasks, seeds and views within each workflow and excluding model loading and job overhead.

Workflow	References	Queries	Ref. mean (s)	Query mean (s)
Rule Qwen3-4B	1872	864	33.1	32.4
Rule Qwen3-8B	1872	864	21.5	21.4
Python Llama-3.1-8B	624	256	29.9	28.8
Python Qwen3-8B	1248	512	29.7	28.8

The scientist follows the LLM-SR proposal, numerical evaluation and experience-buffer workflow (Shojaee et al., 2025a). Llama-3.1-8B-Instruct generates four candidates per batch with two experience islands and a maximum of 128 new candidates. We set temperature to 0.8 and top-p to 0.95. We disabled top-k and set input and output limits to 8,192 and 1,024 tokens. We retained the upstream exact-zero-training-error stopping rule and its fixed initial equation. Seeds 1 and 2 in the tables denote 2026091011 and 2026091023.

A bounded numerical interpreter accepts straight-line equations and explicit NumPy operations. For each candidate, we fit ten coefficients with SciPy BFGS on its permitted training rows, initializing them to one. As in the configured workflow, we retain finite fits even if the optimizer's success flag is false. Invalid or nonfinite candidates receive the worst score. The statistic is negative training mean squared error. Search returns the candidate with the largest value.

Full uses all 256 rows. Half searches and fits coefficients on its 128 search rows. Its fitted predictor supplies the held-out test. A label-blind bank uses the same model and candidate budget, but each generation receives metadata and the initial template without responses, scores or feedback. We fit and select its candidates separately in each outcome world.

H.2 Reference worlds and evidence strategies

Each combination of task and seed has 39 reference worlds, 12 null queries from an independent random stream, and one original-response query. We uniformly permute outcomes within the fixed search and held-out halves. Full Replay regenerates search and selection in every reference world. Final-only fixes the query-selected source and refits coefficients on each reference. The frozen observed bank retains every query proposal and the initial template, then refits and reselects on each reference. Membership does not depend on successful fitting in the query world. The label-blind bank is fixed before fitting, so all of its numerical selection is repeated under randomization.

Held-out testing fixes the Half-selected source and coefficients. It evaluates that predictor on the permuted held-out responses. Prediction failure gives the worst statistic in every reference and in the query. For every method,

$$p = \frac{1 + \sum_{b=1}^{39} \mathbf{1}\{T_b \geq T_{\text{query}}\}}{40}.$$

We count ties conservatively and reject at $p \leq 0.05$. We reuse the four reference banks within their respective units. The 48 null queries therefore share four reference tails. The table gives unit-level counts and their sum.

The matrix contains 208 Full executions, 52 Half query executions, and four label-blind banks. Table 43 reports all null counts. On the four original-response queries, all five strategies give $p = 0.025$, reaching the available minimum and hence the reference resolution limit.

Table 43: Rejected null queries at $\alpha = 0.05$. Each row for a task and seed contains 12 queries and 39 reference worlds. Columns distinguish the replayed or fixed search object.

Task	Seed	Full	Final-only	Frozen bank	Blind bank	Held-out
BPG1	1	1	4	2	1	1
BPG1	2	0	0	0	2	0
PO1	1	0	2	0	1	0
PO1	2	1	5	2	0	2
Total		2	11	4	4	3

H.3 Prediction quality and alternative searchers

We evaluated saved original-query predictors with $\text{NMSE} = \text{MSE}/\text{Var}(Y)$ on each independent evaluation split. Tables 44 and 45 report every task, seed, data view and baseline. We distinguish the statistical evidence for a selected search policy from the prediction quality of alternative policies.

Full improves OOD NMSE over Half in three pairs and over the blind bank in three pairs, although the reversals occur in different BPG1 seeds. The first blind bank achieves 4.188×10^{-9} OOD NMSE,

Table 44: Original-query prediction error of the Llama equation scientists. We evaluate NMSE on the author's separate 500-row ID and 500-row OOD sets. Lower is better.

Task	Seed	Search	ID NMSE	OOD NMSE
BPG1	1	Full	5.004×10^{-6}	0.007241
BPG1	1	Half	0.0005096	2.209
BPG1	1	Label-blind bank	1.889×10^{-9}	4.188×10^{-9}
BPG1	2	Full	1.063×10^{-5}	0.06681
BPG1	2	Half	4.39×10^{-11}	1.784×10^{-8}
BPG1	2	Label-blind bank	6.723×10^{-5}	0.0904
PO1	1	Full	0.001523	0.002736
PO1	1	Half	0.002372	0.006363
PO1	1	Label-blind bank	0.02478	0.117
PO1	2	Full	0.002339	0.005237
PO1	2	Half	0.002398	0.006631
PO1	2	Label-blind bank	0.01151	0.03011

and the second Half search achieves 1.784×10^{-8}. These large differences accompany the pairwise counts. Both Full seeds outperform both corresponding alternatives on PO1.

For the polynomial baselines, we standardize inputs and expand all terms through degree one, two or three. We then standardize the expansion and fit linear or ridge coefficients. Five-fold training cross-validation chooses regularization from $\{0, 10^{-6}, 10^{-4}, 0.01, 1, 100\}$ within each degree. A further training-CV selector chooses the degree. All degrees remain in the table. Folds use seed 2026091004. We fit preprocessing within each fold.

PySR 1.5.9 (Cranmer, 2023) uses the two program seeds with deterministic serial search, 15 populations of 27 expressions, at most 10^6 evaluations, maximum expression size 30, and a 10,000-iteration ceiling. Binary operators are addition, subtraction, multiplication and division. Unary operators are sine, cosine, exponential, logarithm and square root. We use 64-bit precision and its `best` model-selection setting. Each Full or Half fit uses only that view's training rows.

Training-selected polynomials give lower OOD error than Full on BPG1. PySR gives lower OOD error in three of four Full-data pairs, including both PO1 seeds. We assessed calibration for the chosen discovery policy with the five evidence strategies in Table 43, while the CPU baselines show the prediction quality obtainable with alternative search policies.

H.4 Implementation and compute

The model runtime uses PyTorch 2.9.1, Transformers 5.3.0, NumPy 2.3.5 and SciPy 1.17.1. The baseline environment uses PySR 1.5.9, JuliaCall 0.9.26, NumPy 2.5.3, SciPy 1.18.1 and scikit-learn 1.9.0. We cache numerical fits by the complete input array and equation source. Canonicalization removes documentation and annotations without algebraic simplification.

Table 46 reports the full allocated GPU and CPU-only cost, including model loading and interrupted attempts. Model generation is entirely local. Appendix M summarizes numerical verification, recovery and cross-study resource accounting.

I Scientific task construction

I.1 Task catalogs and executable language

We use Airfoil and Concrete for direct validation (Appendix F) and fixed-background analysis (Appendix L). Nanohelix appears in the latter. All three catalogs contain 256 rows. A split balanced by feature hash assigns 128 rows to search and 128 to held-out evaluation. Row admission and splitting are label blind. Table 47 lists the searchable language, an enumerated catalog of one- to three-clause threshold rules.

For each feature, the builder computes empirical q_{25}, q_{50}, and q_{75} thresholds on the frozen 256-row feature table and creates both $\leq$ and $\geq$ literals. It deduplicates tied thresholds. A legal rule is a

Table 45: All CPU prediction baselines. CV polynomial selects the degree using training folds. Polynomial rows are deterministic for the fixed data and folds. PySR uses the two program seeds.

Task	Data	Estimator	Seed	ID NMSE	OOD NMSE
BPG1	Full	Ridge degree 1		0.003149	0.5026
BPG1	Full	Ridge degree 2		2.589×10^{-6}	0.03257
BPG1	Full	Ridge degree 3		2.042×10^{-10}	0.0002999
BPG1	Full	CV polynomial		2.042×10^{-10}	0.0002999
BPG1	Full	PySR	1	1.824×10^{-5}	0.07796
BPG1	Full	PySR	2	3.993×10^{-6}	0.01242
BPG1	Half	Ridge degree 1		0.003152	0.7349
BPG1	Half	Ridge degree 2		2.759×10^{-6}	0.03318
BPG1	Half	Ridge degree 3		3.493×10^{-10}	0.0002963
BPG1	Half	CV polynomial		3.493×10^{-10}	0.0002963
BPG1	Half	PySR	1	5.94×10^{-6}	0.1037
BPG1	Half	PySR	2	5.925×10^{-6}	0.0006414
PO1	Full	Ridge degree 1		0.0498	0.2131
PO1	Full	Ridge degree 2		0.04861	0.2447
PO1	Full	Ridge degree 3		0.02471	0.3338
PO1	Full	CV polynomial		0.02471	0.3338
PO1	Full	PySR	1	3.514×10^{-5}	0.0001326
PO1	Full	PySR	2	5×10^{-6}	1.293×10^{-5}
PO1	Half	Ridge degree 1		0.0502	0.21
PO1	Half	Ridge degree 2		0.04878	0.2443
PO1	Half	Ridge degree 3		0.0249	0.1977
PO1	Half	CV polynomial		0.0249	0.1977
PO1	Half	PySR	1	3.382×10^{-6}	1.125×10^{-5}
PO1	Half	PySR	2	2.366×10^{-5}	3.274×10^{-5}

Table 46: Allocated resources for the external study. GPU hours include loading, the planned initial reference, both timeout allocations and their resumed work. CPU columns count CPU-only jobs.

Computation	A100-hours	CPU core-hours
Search and proposal banks	61.9	
Input preparation and fixtures		0.0458
Polynomial and PySR baselines		0.134
Reference scoring and collection		19.1
Total	61.9	19.2

Table 47: Frozen local task and rule-space inventory. Rule counts are shown for one-, two-, and three-clause conjunctions after support filtering.

Task	Frozen features	Rows	1 clause	2 clauses	3 clauses	Total rules (support range)
Nanohelix	pitch, fiber radius, turns, helix radius, wavelength, axis code, direction	256	38	752	6,400	7,190 (16–192)
Airfoil	frequency, attack angle, chord, free-stream velocity, displacement thickness	256	29	343	1,678	2,050 (16–228)
Concrete	cement, slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, age	256	41	803	7,089	7,933 (16–218)

conjunction of one to three literals on distinct features. Both the selected subgroup and its complement must contain at least 16 rows. These checks explain why the theoretical Cartesian product is larger than the admitted counts in Table 47. Canonical identifiers order literals by the frozen feature order, so semantically identical proposals compile to one bank entry.

I.2 NANOHELIX

We used the public PiEvo Nanohelix task as a source of scientific rules. Before search, we materialized all 256 rows, feature values, outcome ranks, quartile thresholds, and 7,190 support-valid rules. The experiment studies adaptive hypothesis search over a fixed measurement table. The features describe helix geometry and illumination. The target is the released surrogate chirality g-factor. A scrambled Sobol design with a fixed seed supplied the 256 feature rows. We then stored the surrogate outcome and all rule masks in the frozen catalog.

I.3 AIRFOIL AND CONCRETE

Airfoil Self-Noise contains NASA wind-tunnel measurements with five input variables and scaled sound pressure as outcome. Concrete Compressive Strength contains eight mixture and age variables with laboratory strength as outcome. We deterministically chose 256 rows from each source using feature-only logic. We assigned 128 rows each to search and held-out sets without reading outcomes. We derived rule thresholds from the frozen feature table and recorded them in the task catalog. For Airfoil, the rule language can combine frequency, attack angle, chord, flow velocity, and displacement thickness. For Concrete, it can combine the seven mixture quantities and curing age. Frozen measured targets are rank normalized before constructing the outcome views for each experiment. Standardized subgroup contrasts are computed on the declared search rows, so the same scoring rule applies across the physical units.

I.4 ADDITIVE MECHANISMS FOR THE EMPIRICAL-BACKGROUND STUDY

For each task, one-, two-, and three-clause planted rules are selected from the frozen universe before model inference. Let I_h be a rule indicator with prevalence q. We add

$$\beta \frac{I_h - q}{\sqrt{q(1-q)}} \tag{9}$$

to a permuted empirical outcome background, with $\beta \in \{0.25, 0.40\}$. Search and held-out splits receive independent background permutations. This construction places a known executable rule on an empirical outcome background.

J PROMPTS, COMPILER, AND GENERATION CONTRACT

The threshold-rule scientist uses one prompt program across tasks, models, seeds, and outcome views. Angle-bracketed fields in Listings 1 to 3 are replaced deterministically by canonical JSON. The clause grid comes from the fixed task catalog. We recompute evidence and feedback in each replay. Listings 1 to 3 give the natural-language instruction sequence.

J.1 INITIAL PROPOSAL TEMPLATE

```
SYSTEM
You are an empirical scientific hypothesis searcher. Propose concise,
executable subgroup principles from the evidence supplied. Return one
JSON object only, with no markdown or hidden reasoning.

USER
Scientific task: <TASK_DESCRIPTION>

Goal: find a subgroup with an unusually HIGH rank-normalized outcome.
Scores are standardized selected-minus-complement contrasts on the
search rows; larger is better.

Frozen clause grid (copy feature, operator, and quantile strings exactly):
<JSON_CLAUSE_GRID_SORTED_BY_FEATURE>
```

```
Strongest one-clause evidence in this outcome view:
<TOP_10_JSON_ITEMS_WITH_RULE_SCORE_SUPPORT>

Propose exactly 4 distinct hypotheses. Each hypothesis uses 1--3
clauses on distinct features. Use exactly this schema:
{"hypotheses":[{"clauses":[["feature",">=","q75"]],
"rationale":"brief scientific interpretation"}],"stop":false}

Do not copy the example unless it is supported. Return JSON now. /no_think
```

Listing 1: Exact system and initial-user template. Dynamic fields are canonical JSON serializations.

We sort the evidence list by decreasing standardized contrast and break ties by canonical rule identifier. Each entry contains the rendered rule, score to four decimal places, and support. The frozen grid lists both operators and the available q_{25}, q_{50}, and q_{75} thresholds for every task feature.

J.2 Feedback and repair templates

```
Round <ROUND> compiler/evaluation feedback:
valid-rule leaderboard: <TOP_8_ADMITTED_RULES_WITH_SCORE_SUPPORT>
compiler messages: <JSON_ERROR_LIST>
already evaluated canonical IDs: <JSON_ID_LIST>

Propose exactly 4 NEW rules that refine or challenge the current
principles. Use the same JSON schema and frozen clause grid. Set
`stop` true only if further refinement is unlikely to improve the
best rule. Return JSON only. /no_think
```

Listing 2: Exact later-round feedback template.

```
The compiler admitted no rule. Repair the JSON using only the frozen
schema. Every repaired hypothesis must be NEW: do not repeat any
canonical ID already evaluated, and change at least one clause when
repairing a duplicate. Already evaluated IDs: <JSON_ID_LIST>.
Errors: <JSON_ERROR_LIST>
```

Listing 3: Exact bounded-repair instruction appended after a zero-admission response.

The compiler extracts the first JSON object containing a `hypotheses` list, accepts one to three distinct-feature clauses, and normalizes quantiles to q_{25}, q_{50}, or q_{75}. It rejects unknown features or operators, repeated features, unsupported masks, and duplicate canonical identifiers. If bounded repair still admits nothing, we admit the best unseen one-clause item from the replay-specific initial evidence as a declared fallback. Responses enter the pipeline through automatic compilation, bounded repair, and this fallback.

Table 48 lists the threshold-rule search, repair, sampling, randomness, and model settings.

Table 48: Threshold-rule scientist: search and generation settings.

Component	Setting
Search	3 rounds; 4 proposals per round; top-10 initial evidence; top-8 feedback; bank cap 12; model stop allowed after round 2
Repair	At most one repair call after a zero-admission response
Sampling	512 new tokens; temperature 0.6; top-p 0.9; top-k 40
Randomness	One program seed per scientist; corresponding stages share a derived generation seed across observed and replay worlds
Models	Local Qwen3-4B and Qwen3-8B

Each program seed deterministically determines a stage-specific generation seed. Corresponding proposal and repair stages therefore share common random numbers across observed and replay worlds.

K END-TO-END COUNTERFACTUAL REPLAY CASE

Table 49 pairs a representative Qwen3-4B Concrete run under observed outcomes with a run under null outcomes. We select the first source and null transcripts in frozen file order that share a three-round successful-repair event skeleton. This selection is independent of scores and p-values. We abbreviate cement as C, water as W, age as A, superplasticizer as SP, fine aggregate as FA, fly ash as Fly, and blast-furnace slag as BFS. Scores are standardized subgroup contrasts.

Table 49: Event-by-event comparison for one observed scientist and one counterfactual null scientist. "Fail" combines support failures, other compiler errors, and duplicate identifiers.

Stage	Observed outcomes	Null world `null-01`
Top evidence	$\mathrm{A} \geq q_{50}$; $\mathrm{A} \geq q_{25}$; $\mathrm{C} \geq q_{75}$; $\mathrm{W} \leq q_{25}$; followed by six more outcome-ranked one-clause rules	$\mathrm{Fly} \leq q_{75}$; $\mathrm{SP} \leq q_{50}$; $\mathrm{SP} \leq q_{25}$; $\mathrm{FA} \geq q_{50}$; followed by six more outcome-ranked one-clause rules
Round 0	Four proposed, three admitted: $\mathrm{C} \geq q_{75} \wedge \mathrm{SP} \geq q_{75} \wedge \mathrm{A} \geq q_{50}$ (5.05); $\mathrm{C} \geq q_{50} \wedge \mathrm{W} \leq q_{25} \wedge \mathrm{A} \geq q_{50}$ (5.86); $\mathrm{C} \geq q_{50} \wedge \mathrm{FA} \leq q_{50} \wedge \mathrm{A} \geq q_{50}$ (4.01). One fails.	Four proposed, two admitted: $\mathrm{W} \leq q_{50} \wedge \mathrm{FA} \geq q_{50} \wedge \mathrm{A} \leq q_{25}$ (-0.20); $\mathrm{C} \leq q_{50} \wedge \mathrm{SP} \leq q_{50} \wedge \mathrm{FA} \geq q_{50}$ (2.63). Two fail.
Round 1	Feedback on the three admitted rules yields two new rules: $\mathrm{C} \geq q_{50} \wedge \mathrm{SP} \geq q_{50} \wedge \mathrm{A} \geq q_{50}$ (5.19) and $\mathrm{C} \geq q_{50} \wedge \mathrm{BFS} \leq q_{50} \wedge \mathrm{A} \geq q_{50}$ (2.58). Two fail.	Feedback on the two admitted rules yields three new rules with scores 1.78, 2.33, and 1.74; the strongest uses $\mathrm{BFS} \leq q_{25} \wedge \mathrm{SP} \leq q_{25} \wedge \mathrm{coarse} \leq q_{75}$. One fails.
Round 2 and repair	The first response contributes no new rule (two support failures, two duplicates), triggering repair. Two repaired rules enter with scores 4.94 and 4.64; the first is $\mathrm{C} \geq q_{50} \wedge \mathrm{SP} \geq q_{75} \wedge \mathrm{A} \geq q_{50}$.	The first response contributes no new rule (one support failure, three duplicates), triggering repair. Two repaired rules enter with scores 1.09 and 2.02; the second is $\mathrm{C} \leq q_{50} \wedge \mathrm{SP} \leq q_{75} \wedge \mathrm{FA} \geq q_{75}$.
Selection	Retain the round-0 $\mathrm{C} \geq q_{50} \wedge \mathrm{W} \leq q_{25} \wedge \mathrm{A} \geq q_{50}$ rule, score 5.86; seven rules evaluated.	Retain the round-0 $\mathrm{C} \leq q_{50} \wedge \mathrm{SP} \leq q_{50} \wedge \mathrm{FA} \geq q_{50}$ rule, score 2.63; seven rules evaluated.

The two rationales are both superficially plausible. The observed agent links high cement, low water, and older age to hydration and reduced porosity. The null agent links moderate cement, high fine aggregate, and low superplasticizer to workability and strength. Although the two runs have identical stage seeds, they follow different scientific paths, with different ranked evidence, compiler outcomes, feedback, and repairs.

A final-only reference would carry the observed rule on cement, water, and age into `null-01`. Full Replay instead produces and scores the null-specific rule on cement, superplasticizer, and fine aggregate. Within the finite 63-world null orbit, treating `null-01` as pseudo-observed gives $p = 1/63$ for the final-only statistic and $p = 15/63$ for Full Replay. The pair shows the difference between the estimands. One procedure asks how a source-selected rule scores under shuffled outcomes. The other asks what the complete scientist would select there.

L FIXED-BACKGROUND SENSITIVITY AND RANK DIAGNOSTICS

In this additional study, we used Qwen3-4B and Qwen3-8B, three program seeds, and the Airfoil, Concrete, and Nanohelix tables. The 128-row and 256-row matrices contain 4,032 transcripts. We compared queries with additive signals against a shared bank of empirical-background references. This construction differs from the per-query permutation orbit used for direct validation in Appendix F. We give all sensitivity results below.

Treating each of 63 reference views as pseudo-observed against the other 62 gives the self-ranking rate $3/63 = 4.76\%$ within each reference block. This is an algebraic rank-consistency check. The observed final bank contains every round-0 rule, but its max-test p-value is higher in 682 pseudo-observed views, lower in 241, and tied in 211. Adding rules changes both the observed threshold and the reference maxima. The value is therefore not monotone in bank size. Source rules agree across all seeds in four of six cells defined by model and task, compared with 11.1–41.3% agreement under null views, and 15 of 18 source Replay p-values reach the minimum $1/64$. Table 50 reports the sampled-reference diagnostics. Table 51 gives planted-signal power and exact discovery. Table 52 measures executable recovery, and Table 53 lists every real-outcome selection.

Table 50: Sampled-reference diagnostics. Panel A reports rejection across 1,134 pseudo-observed views and admitted-bank sizes. Full Replay is a self-ranking consistency check. Panel B reports the comparator rates by model and task.

Panel A: overall replay boundary

Method	Rejection (%)	Mean admitted rules
Final-only	76.37	1.00
Initial bank	34.22	3.71
Frozen bank	29.63	9.18
Full Replay rank check	4.76	9.18
Held-out	4.06	—
Selected maxT	0.26	—

Panel B: original comparators by model and task

Method	4B Nano	4B Airfoil	4B Concrete	8B Nano	8B Airfoil	8B Concrete
Final-only	79.9	80.4	76.2	75.1	68.8	77.8
Frozen bank	33.9	24.3	35.4	32.8	28.0	23.3
Held-out	1.6	4.2	4.2	2.1	5.8	6.3
Selected maxT	0.0	0.0	0.0	0.0	0.0	1.6

Table 51: Additive-signal queries with empirical-background references. Entries give rejection with exact planted-rule discovery in parentheses (%) for this fixed-background construction.

Clauses	Strength	Final-only	Frozen bank	Replay	Selected maxT	Held-out
1	0.25	93.1 (26.4)	74.3 (26.4)	44.4 (18.1)	19.4 (12.5)	45.8 (20.8)
1	0.40	94.4 (49.3)	92.4 (49.3)	77.8 (45.1)	60.4 (34.7)	79.2 (47.9)
2	0.25	79.9 (8.3)	59.0 (6.9)	27.1 (4.9)	16.0 (3.5)	22.9 (4.9)
2	0.40	86.1 (7.6)	79.9 (7.6)	52.8 (7.6)	34.0 (7.6)	56.2 (7.6)
3	0.25	84.0 (0.0)	57.6 (0.0)	20.8 (0.0)	9.0 (0.0)	11.8 (0.0)
3	0.40	84.0 (0.0)	74.3 (0.0)	45.1 (0.0)	22.2 (0.0)	43.8 (0.0)

Table 52: Executable recovery by the local scientists before inferential correction. Exact rates compare the selected rule with the planted rule. F1 and Jaccard compare literals and row masks.

Clauses	Strength	Initial exact	Final exact	Literal F1	Mask Jaccard	Score gain
1	0.25	37.5	26.4	53.7	58.2	0.17
1	0.40	53.5	49.3	75.7	74.8	0.15
2	0.25	3.5	8.3	25.3	39.7	0.19
2	0.40	1.4	7.6	37.6	45.5	0.46
3	0.25	0.0	0.0	19.9	25.1	0.18
3	0.40	0.0	0.0	22.7	32.2	0.33

Table 53: All real-outcome selections. Seeds are shortened to their final two digits. Replay, Selected maxT, and Held-out columns give Monte Carlo p-values. Of these, 15 of 18 Replay values reach the attainable minimum $1/64$.

Task	Model	Seed	Selected executable rule	Search score	p_{TR}	p_{univ}	p_{holdout}
Nanohelix	Qwen3-4B	11	pitch≥q50	6.63	0.016	0.016	0.016
Nanohelix	Qwen3-4B	12	pitch≥q50	6.63	0.016	0.016	0.016
Nanohelix	Qwen3-4B	13	pitch≥q50	6.63	0.016	0.016	0.016
Nanohelix	Qwen3-8B	11	pitch≥q75	4.57	0.016	0.016	0.016
Nanohelix	Qwen3-8B	12	pitch≥q75	4.57	0.016	0.016	0.016
Nanohelix	Qwen3-8B	13	pitch≥q75	4.57	0.016	0.016	0.016
Airfoil	Qwen3-4B	11	frequency≤q50 ∧ attack_angle≤q75	4.88	0.016	0.016	0.016
Airfoil	Qwen3-4B	12	frequency≤q50	5.21	0.016	0.016	0.016
Airfoil	Qwen3-4B	13	frequency≤q50 ∧ attack_angle≤q75	4.88	0.016	0.016	0.016
Airfoil	Qwen3-8B	11	chord_length≥q25 ∧ free_stream_velocity≥q75	-0.79	1.000	1.000	0.328
Airfoil	Qwen3-8B	12	attack_angle≥q50 ∧ free_stream_velocity≥q75	0.15	0.984	1.000	0.047
Airfoil	Qwen3-8B	13	free_stream_velocity≥q75 ∧ displacement_thickness≥q75	-0.82	1.000	1.000	0.531
Concrete	Qwen3-4B	11	cement≥q50 ∧ water≤q25 ∧ age≥q50	5.86	0.016	0.016	0.016
Concrete	Qwen3-4B	12	cement≥q50 ∧ water≤q25 ∧ age≥q50	5.86	0.016	0.016	0.016
Concrete	Qwen3-4B	13	cement≥q50 ∧ water≤q25 ∧ age≥q50	5.86	0.016	0.016	0.016
Concrete	Qwen3-8B	11	age≥q50	5.93	0.016	0.016	0.016
Concrete	Qwen3-8B	12	age≥q50	5.93	0.016	0.016	0.016
Concrete	Qwen3-8B	13	age≥q50	5.93	0.016	0.016	0.016

L.1 Outer-loop scientist selection

For each task, the outer selector compares the six scientists already executed for the model and program-seed combinations and reports the largest selected score. Full outer-loop replay repeats this maximization in every null view. A partial alternative chooses the scientist with the largest score in the pseudo-observed view. It then fixes that scientist when scoring the remaining views. Table 54 reports the aligned comparisons.

Table 54: Outer-selection diagnostics over six scientists per task. Full outer Replay is a self-ranking check on the 63 aligned null views. Winner changes are relative to the scientist selected on measured outcomes.

Task	Full outer Replay (rank check)	Frozen winner	Winner changes
Airfoil	3/63 (4.76%)	6/63 (9.52%)	48/63
Concrete	3/63 (4.76%)	11/63 (17.46%)	54/63
Nanohelix	3/63 (4.76%)	4/63 (6.35%)	39/63

The winning scientist differs from the measured-outcome winner in 48, 54, and 39 null views for Airfoil, Concrete, and Nanohelix, respectively. Full outer Replay repeats the selection in each world. The replay unit thus includes the outcome-dependent choice of scientist.

L.2 Full-data empirical-background experiment

Table 55 compares additive-signal queries against shared empirical-background references, as defined in Appendix L. We pair the 256-row and 128-row query worlds by model, task, and seed.

Table 55: Fixed-background rejection rates for 256-row search, 128-row search, and a split with 128 search rows and 128 evidence rows (%). Each cell has 144 worlds matched by model, task, and seed. Brackets are 95% bootstrap intervals clustered by combinations of task and world for percentage-point differences.

Clauses	Strength	Full	Half	Split	Full−Half	Full−Split
1	0.25	65.97	44.44	45.83	+21.53 [6.25, 38.19]	+20.14 [6.94, 33.33]
1	0.40	88.89	77.78	79.17	+11.11 [2.08, 21.53]	+9.72 [1.39, 18.75]
2	0.25	40.97	27.08	22.92	+13.89 [-2.08, 29.86]	+18.06 [4.17, 33.33]
2	0.40	77.78	52.78	56.25	+25.00 [10.42, 40.30]	+21.53 [7.64, 36.11]
3	0.25	37.50	20.83	11.81	+16.67 [4.86, 29.17]	+25.69 [13.89, 37.50]
3	0.40	71.53	45.14	43.75	+26.39 [11.81, 41.67]	+27.78 [11.81, 43.06]

M Reproducibility and resource accounting

Table 56 summarizes the controlled and fixed-background computations. Controlled matrices are CPU-only. Qwen3 and Llama inference uses the reported GPU allocations.

Table 56: Output and compute inventory for the controlled and fixed-background studies.

Evidence block	Output inventory	Compute
Fixed-table controls	24,800 worlds; 173,600 records indexed by method and world	CPU only
Matched-data efficiency	12,300 worlds; 36,900 method records	CPU only
Fixed-background local study	4,032 transcripts; 12,721 model calls	33.6 A100-h
Fixed-background transport checks	6 canonical transcripts; 42 model calls	1.02 A100-h
Outer scientist selection	189 aligned null views; 6 scientists per task	CPU postprocess
Proxy acquisition	10,500 labelled worlds; 42,000 method records	3.27 CPU-h
Acquisition controls	15,500 labelled worlds; 62,000 method records	4.43 CPU-h

Independent implementations reproduce the selected statistics, query ranks, and paired contrasts. We checked depth-zero initialization, selection across six scientists, uniform-acquisition equivalence

and fresh-process replay (Table 31). External checks reproduce all 260 query-method ranks and 72 ID/OOD metrics from saved programs and coefficients, including native PySR equations and independently reconstructed polynomial predictions. Detailed verification records accompany the analysis outputs.

These blocks use 34.7 allocated A100-hours. The acquisition experiments record 7.7 CPU execution-hours. We report execution time separately from the direct local validation's allocated CPU core-hours.

Table 57: Allocated GPU cost of the direct local validation, including development.

Computation	GPU allocations	A100-hours
Qwen rule and program validation	24	57.2
Llama program validation	4	7.48
Measured targets and transport	2	0.292
Qwen development	3	0.393
Llama development	2	0.312
Total	35	65.7

The direct validation, measured-target illustrations, and transport checks contain 8,139 executions and 30,171 model calls, using 51.6M input and 6.57M output tokens. Original cluster analysis uses 0.14 allocated core-hours. GPU cost includes model loading, development, and transcript runtimes (Table 57). The fixed-background and direct local studies total 100 allocated A100-hours. Hosted-model API cost is zero.

External allocations add 61.9 A100-hours and 19.2 CPU-only core-hours (Table 46), including all attempts and 0.44 A100-hours for two resumed executions. Completed outputs were retained. We separately record GPU-associated host CPU use of 248 core-hours. Preliminary interface and public-data development adds 2.91 A100-hours. All model inference is local.