Scores Alone Do Not Prove Discovery The Discovery Certification Protocol for Auditing AI Research Agents

Jingjie Ning Shanshan Zhong Xiaochuan Li Ji Zeng

School of Computer Science, Carnegie Mellon University

{jening, szhong2, xiaochu4, jizeng}@cs.cmu.edu

Homepage ↗ PyPI / dcp-audit PyPI / dcp-harness GitHub

Abstract

AI research agents combine prior knowledge, public sources, and experimental feedback to produce useful results. The Discovery Certification Protocol (DCP) turns claims about these results into executable recovery and feedback tests. Gate 1 validates useful improvement on sealed evaluation. Gate 2 gives matched agents the registered starting information and observed Web content while withholding the target research history. Every valid method reaching the numerical target supplies a recovery witness and triggers the Core veto. DCP Core requires adequate controls, zero observed recoveries, and a finite-sample bound on recovery in one fresh registered episode. Optional Gate 3 measures the average effect of truthful feedback relative to a specified neutral policy from a shared checkpoint. DCP Evidence adds this effect after independent null calibration and a registered effect margin. Two controlled audits exercise the complete protocol in SQLite optimization and virtual catalyst control under different models. Each produced zero recoveries in 96 episodes, with an upper bound of 0.0468. Each paired study yielded 30 truthful recoveries and zero neutral recoveries, with passing 60-pair null studies. Additional cases exercise Core, recovered, and audit-incomplete decisions. A deterministic, LLM-free verifier reproduces the decisions from frozen evidence. DCP provides a common evidence language for useful outcomes, alternative routes, and feedback effects across AI research.

1 Introduction

AI research agents choose actions, run experiments, inspect measurements, and revise their proposals. Their outputs include programs, models, data products, and experimental recipes. Recent systems have found useful algorithms and mathematical constructions and have begun to automate broader research workflows (Fawzi et al., 2022; Mankowitz et al., 2023; Romera-Paredes et al., 2024; Novikov et al., 2025; Lu et al., 2026). As these systems become research collaborators, their reports need evidence connecting measured results to the research that produced them.

Consider an agent that runs 87 experiments and reports a program with score x. A sealed test establishes the program's utility. A matched agent may also reach x using initial data, public material, and its own reasoning. Separately, truthful experimental observations may improve the chance of reaching x. These are three measurable questions about one outcome. Implementation and baseline choices shape measured gains (Melis et al., 2018; Musgrave et al., 2020). In automated research, independent implementations of a fixed idea can change its ranking (Ning et al., 2026e). DCP extends this measurement discipline to the information and feedback available to a research agent.

The central design choice is to admit every valid route to the same numerical outcome. A challenger may combine familiar components, transfer a technique across domains, or produce a different implementation. It receives the registered background, initial observations, tools, budget, and captured Web content. The target run's experimental history is withheld. An executable validity check and a score threshold determine recovery, making the rule portable across research domains.

Recovery witnesses and recovery probabilities play complementary roles. A witness establishes an available route within the registered scope and closes the audit as recovered. A positive Core decision requires an adequate audit with

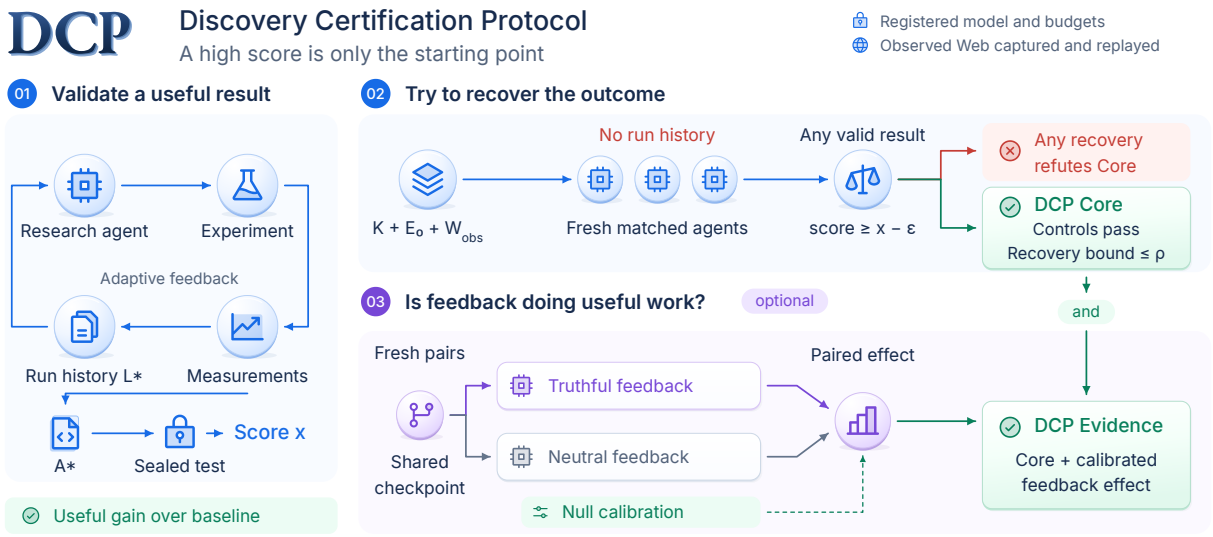

Figure 1: DCP asks three questions about one numerical outcome. Gate 1 validates result A^* on sealed data. Gate 2 gives fresh matched agents K, E_0, and W_{obs} while withholding run history L^*. Any valid recovery at $x - \varepsilon$ refutes Core; certification requires adequate controls and bounded non-recovery. Optional Gate 3 adds Evidence when fresh paired runs show a feedback effect over a null-calibrated neutral channel.

zero recoveries and a bound on a fresh episode's recovery probability. Evidence adds a randomized comparison of truthful and neutral feedback from a shared checkpoint. This separation gives a successful target run, a recoverable outcome, and a beneficial feedback policy distinct evidential meanings.

The paper makes three contributions.

- We define an outcome-level audit that accepts alternative methods through a shared numerical recovery rule and a registered information and resource boundary.
- We combine qualified recovery witnesses, finite-sample recovery bounds, and randomized feedback effects in two explicit decisions, Core and Evidence. Their definitions separate certificate eligibility from statistical and causal conclusions.
- We calibrate the complete protocol across software optimization and virtual experimental control, alongside recovered and incomplete cases. Portable evidence bundles and a shared deterministic verifier make these decisions independently replayable.

2 Related Work

AI research agents and their evaluation. AlphaTensor, AlphaDev, FunSearch, and AlphaEvolve combine search with objective evaluation (Fawzi et al., 2022; Mankowitz et al., 2023; Romera-Paredes et al., 2024; Novikov et al., 2025). The AI Scientist family and AI co-scientist automate broader research and hypothesis development (Lu et al., 2026; Yamada et al., 2025; Gottweis et al., 2026). Specialist-agent studies remove prior-trial history while retaining current-best code and score (Ning et al., 2026a). Molecular and materials agents freeze selected interventions for held-out evaluation (Ning et al., 2026b,c). MLE-bench, MLGym, MLR-Bench, and PaperBench evaluate engineering, open-ended research, and replication (Chan et al., 2025; Nathani et al., 2025; Chen et al., 2025; Starace et al., 2025). DiscoveryWorld measures task success, scientific actions, and explanatory knowledge (Jansen et al., 2024). FIRE-Bench evaluates rediscovery of published scientific insights (Wang et al., 2026), while Bhushan et al. (2026) distinguish psychological novelty, historical novelty, and usefulness. DCP contributes an outcome-recovery test and a separate feedback intervention that can accompany these forms of evaluation.

Iterative feedback and experimental design. ReAct, Tree of Thoughts, Reflexion, and Self-Refine develop tool use, search, reflection, and feedback (Yao et al., 2023b,a; Shinn et al., 2023; Madaan et al., 2023). Controlled revision studies separate additional solving, prompt structure, and draft content (Ning et al., 2026d). SkillLearnBench compares execution-derived and teacher-guided skill refinement (Zhong et al., 2026). DCP tests feedback through randomized pairs (Rubin, 1974), an equivalence region for independent null calibration (Schuirmann, 1987), and preregistered confirmatory choices (Nosek et al., 2018).

Evaluation integrity and process attestation. Broad evaluation, contamination studies, and memorization research examine the influence of prior exposure (Hendrycks et al., 2021; Srivastava et al., 2023; Liang et al., 2023; Golchin and Surdeanu, 2025; Carlini et al., 2021). DeepResearchGym provides stable retrieval from fixed Web corpora (Coelho et al., 2026). Query initialization also affects search success at matched budgets (Murali et al., 2026). SWE-bench emphasizes executable outputs (Jimenez et al., 2024). Proof-of-Learning records training states to attest a training procedure (Jia et al., 2021). DCP couples executable outcomes with registered counterfactual experiments. Provenance establishes the evidence source, and the statistical verifier evaluates the registered claim.

3 Method

3.1 The outcome and its information boundary

DCP organizes an audit around a useful outcome, its recovery under a registered challenger, and the effect of subsequent feedback. Table 1 defines the five objects that specify the outcome and the available information.

Table 1: The objects in a DCP audit

Object	Meaning	Examples
K	Background fixed before the run	Model, task rules, starter code, tools, public methods, known components
E_0	Task observations fixed independently of the run's research actions	Initial measurements, training examples, labels supplied with the task
$L^\star$	Research history created during the target run and used by a later research action	Hypotheses, submitted candidates, experiment results, scores, error analyses, adaptive queries
$A^\star$	Final output selected by a rule fixed before the sealed test	Program, model, policy, data product
P	Machine check for recovering the claimed result	Validity rules, resource limits, and a numerical performance threshold

The E_0 and $L^\star$ boundary follows information provenance. An example present at the start belongs to E_0. A measurement returned because the agent chose an action belongs to $L^\star$. Fixed human guidance enters E_0; guidance chosen after intermediate results enters a human-agent lineage. Core targets adaptive research with at least one new observation consumed by a later research action. A pool of independent candidates followed by final score-based selection is a search-and-selection baseline.

The validity predicate and numerical threshold define recovery across all admissible artifacts. Historical priority is a separate scholarly assessment. The same outcome audit applies to programs, models, data products, and experimental recipes.

3.2 Precommitment and sealed evaluation

Registration precedes the target run and fixes the task, model, information, interface, tools, budgets, baseline, validity rules, selection and stopping procedures, and statistical analysis. It covers the complete production procedure named in the claim. Every started confirmatory audit consumes a preallocated error budget, including recovered and incomplete audits. The registered selection rule freezes $A^\star$, and all candidate generation closes before sealed evaluation. One symmetric evaluator checks the baseline, target, and controls for validity and score.

3.3 Gate 1 establishes useful improvement

Let $\mathrm{Valid}(a)$ denote compliance with the registered artifact constraints. Let b denote the baseline artifact and x the mean sealed score of $A^\star$, on a registered $[0,1]$ scale. For the smallest useful gain $\delta_{\min} > 0$, Gate 1 requires

$$\mathrm{LCB}[\mu(A^\star) - \mu(b)] \geq \delta_{\min}, \tag{1}$$

where μ is mean utility over the registered evaluation population. Sampled populations use a finite-sample confidence interval; exhaustive finite evaluations use an exact mean. Gate 1 also requires valid baseline and target artifacts and a baseline below the recovery region. Confirmed target invalidity or insufficient utility fails the audit. Unresolved checks produce an inconclusive decision.

3.4 Gate 2 tests recovery under a registered challenger

Gate 2 gives a fresh matched agent the same K, complete E_0, model, interface, known components, and registered production opportunities. It withholds $L^\star$ and gives the challenger all Web bytes observed by the target run, denoted W_{obs}. The agent may reason, compile, and perform engineering checks. Responses to its actions follow a frozen policy that supplies non-directional messages and withholds new task scores and scientific measurements. An independent research rerun with fresh truthful experiments evaluates rediscovery under its own lineage.

Every valid method is eligible. For a registered tolerance ε, the recovery rule is

$$P(a) = \mathrm{Valid}(a) \ \wedge \ \mathrm{score}(a) \geq x - \varepsilon. \tag{2}$$

The tolerance expresses a small substantively equivalent score difference. Registration requires $0 \leq \varepsilon < \delta_{\min}$ and $\varepsilon < x - \mathrm{score}(b)$. Repeated paired evaluation and simultaneous confidence intervals handle measurement uncertainty. When the target depends on x, generators receive the metric and selection rule; the verifier computes x after all artifacts are committed.

A qualified recovery witness satisfies both Equation 2 and the registered execution, information, and provenance conditions. It establishes an alternative route and triggers the Core veto, recorded as *recovered* with Core *refuted*. This is a certificate eligibility rule. The probability bound below quantifies recovery in a fresh episode. A rare recovery and a small recovery probability can coexist.

Let Q_B be the registered distribution of complete challenger episodes at budget B. An episode includes its model calls, candidate opportunities, and selection rule. A best-of-k procedure counts as one k-candidate episode. Let $H = 1$ when any admissible candidate in the episode recovers the target, and define $p_B = \Pr_{Q_B}(H = 1)$. With zero recoveries in n independent episodes, the fixed-sample upper bound is

$$p_{\mathrm{upper}} = 1 - \alpha_{\mathrm{recovery}}^{1/n}. \tag{3}$$

Core requires $p_{\mathrm{upper}} \leq \rho$, zero qualified witnesses, and a complete adequate audit. Adequacy checks the opportunity budget, access to the information packet, independent draws, valid-output rates, and positive controls that solve known instances with the necessary information supplied. Candidate score tests have a separate simultaneous error budget. These requirements attach the bound to a specified generation procedure and make weak or incomplete controls inconclusive.

3.5 Gate 3 estimates a feedback-policy effect

Gate 2 measures recovery from the starting information. Gate 3 measures how subsequent feedback changes fresh outcomes from a shared checkpoint c. A rule registered before the run selects c, including its files and available history. Fresh paired branches receive either truthful feedback from their own actions or messages from a specified neutral policy. Both arms share the model, tools, remaining budget, starting state, and output checks.

The neutral policy preserves message timing, schema, and approximate length while withholding information about the correct next action. Later actions may diverge in response to the two policies. Arm assignment, pair count, selection,

and stopping are frozen before the study. Both arms use fresh replicates, which separate an average feedback effect from the selected target run's success.

For registered utility u, the estimand is

$$\Delta_{\text{feedback}} = \mathbb{E}[u(A_{\text{truthful}}) - u(A_{\text{neutral}}) \mid c]\,. \tag{4}$$

A binary utility compares recovery probabilities. This is the total effect of the two registered feedback policies, conditional on c, including their downstream influence on the agent's actions.

Independent null tasks compare the neutral policy with a second non-directional policy on problems whose answers are determined by E_0. Calibration requires their confidence interval to lie inside a registered equivalence band $\pm\delta_{\text{sham}}$. The calibration applies to that family. Evidence additionally requires $\text{LCB}(\Delta_{\text{feedback}}) \geq \delta_{\text{evidence}} + \delta_{\text{sham}}$, where the positive effect threshold and calibration margin are fixed before testing. The target effect and null contrast retain separate intervals. A frozen attrition rule may replace a whole pair after a pre-action infrastructure failure while preserving every started slot.

3.6 Web access

A recording gateway stores each query, response time, and exact model-visible bytes. Gate 2 discloses the complete observed packet at the start, so its recovery bound conditions on W_{obs}. Gate 3 pairs inherit the pages available at their checkpoint and share a frozen environment for later Web requests. A claim about the contribution of Web access uses an additional Web-withholding intervention.

3.7 Threat model and trust boundary

DCP treats the claimant and evidence producer as potentially strategic. An independent audit authority approves the registered question and resource match, holds the sealed test, and anchors the records before execution. Complete ledgers and bound execution records expose omitted attempts, substituted models, early private-score access, and altered evidence. The checker evaluates numerical and record consistency under the accepted registration. Independent provenance verification and registry countersigning complete formal issuance.

Evidence production and decision checking have separate interfaces. Task adapters specify validity, scoring, and feedback; a shared deterministic verifier evaluates the resulting records. The reported audits were produced by task-specific runners. A lightweight reusable harness supports collection and control execution, and the offline verifier recomputes decisions from the resulting bundles. This separation lets independent readers check the same evidence across agent implementations.

3.8 Decisions and their meaning

Core combines useful improvement, qualified control execution, zero observed recoveries, and a bound on p_B. Evidence adds a supported checkpoint-conditional feedback effect and passing null calibration. Failed registration, weak controls, and broken interfaces yield *audit incomplete*. A complete audit whose interval crosses a threshold yields *statistical uncertainty*. The decision record reports these outcomes together with their model, information, budget, and evaluation scope.

4 Experimental Audit Cases

4.1 Study design

We use two complete three-gate audits and three diagnostic cases to calibrate the protocol across controlled information boundaries. Each case has a clear numerical target and a known role in testing the decision rule. Every candidate was committed before sealed scoring, and the baseline, target, and controls shared one evaluator. Each positive case used a

prospectively allocated confirmatory audit slot. The reported audits produce locally registered kernel decisions; formal issuance adds independent provenance verification and countersigning.

The producer configured Claude CLI for the registered model endpoint and checked the model identity reported in the provider response. SQLite-Web used DeepSeek-v4-flash; catalyst optimization used DeepSeek-v4-pro. Target and control sessions matched the registered model, tools, files, policy, and opportunity budget. Gate 1 exhaustively evaluated each finite sealed workload. Gate 2 drew randomized equivalent presentations of the same hidden task from a fixed Q_B, with fresh sessions and committed draw receipts. Gate 3 sampled fresh pairs conditional on a frozen lineage-prefix checkpoint. These are separate sources of evaluation, episode, and feedback uncertainty.

Development cases guided neutral-channel design. The neutral generator, checkpoint rule, effect threshold, equivalence band, and pair distribution were frozen before fresh confirmatory target and null branches were opened.

4.2 SQLite-Web with DeepSeek-v4-flash

The first task models a SQLite event service with 16 query families and one partial-index candidate per family. Exactly four families carry high hidden traffic, and the agent may select four indexes. A valid plan must preserve every query result. Its score is

$$\mathrm{clip}\left(1 - \frac{\text{traffic-weighted VM steps with the plan}}{\text{traffic-weighted VM steps without secondary indexes}}, 0, 1\right).$$

The sealed evaluator counts deterministic SQLite virtual-machine work on a private database and fresh query parameters.

Before its first experiment, the agent read two public SQLite documentation pages through a recording gateway. Its first committed plan scored 0.0369 in development. The harness returned per-family traffic-weighted work, and the final plan selected the four high-traffic families. It scored 0.8855 on the sealed workload, which means an 88.55% reduction from the no-index baseline.

Gate 2 received the public workspace and the exact model-visible bytes from both Web responses. The target plan, score, traffic measurements, and reasoning trace remained sealed from the challengers. Gate 3 began from a checkpoint after the Web snapshot and before the first experiment. Truthful branches received traffic-weighted feedback from their own plans. Neutral branches received the same schema and timing with uniform family weights. Fresh null tasks compared that neutral channel with a second non-informative channel.

4.3 Virtual catalyst optimization with DeepSeek-v4-pro

The second task presents five experimental controls with eight legal levels each, giving $8^5 = 32{,}768$ possible recipes. The controls have familiar laboratory meanings such as temperature, hydrogen pressure, catalyst loading, ligand ratio, and residence time. The environment is virtual, deterministic, and fully sealed.

The agent first commits an anchor recipe. A trusted harness expands that anchor into a 36-well one-factor plate and returns yield, selectivity, and usable-product measurements. The agent then submits one static final recipe. A score of 1 means that all five hidden optimal levels were selected and the recipe achieved optimal behavior over 256 sealed conditions. The fixed baseline scored 0.5990, while the final recipe scored 1.0.

Gate 2 withheld the target plate and all downstream reasoning. Gate 3 started from the frozen anchor checkpoint. Truthful branches received measurements from their own plate, and neutral branches received matched non-directional measurements. A separate 60-pair family whose answer was fixed by its starting information tested whether the neutral channel itself changed performance.

4.4 Calibration cases

The device case has 24 controls with hidden gains in $\{-1, +1\}$. Four gains are given initially. The main agent requests the remaining measurements and predicts 300 sealed single-control and paired-control outputs. The hidden gains create a controlled information deficit, providing a Core calibration positive with a known information boundary.

Table 2: Primary DCP audit outcomes. Scores are normalized and higher is better within each task. Flash and Pro identify the models recorded in the provider responses.

Case	Model	Main, baseline	Recovery line	Gate 2	Decision
SQLite-Web	Flash	0.8855, 0	0.8805	0/96, best 0.6734	Core + Evidence
Virtual catalyst	Pro	1.0000, 0.5990	0.9500	0/96, best 0.8146	Core + Evidence
Device calibration	Flash	1.0000, 0.3667	0.9500	0/80, best 0.7133	Core
Multidimensional knapsack	Flash	0.9349, 0.9140	0.9329	recovered at 0.9363	Dev. refuted

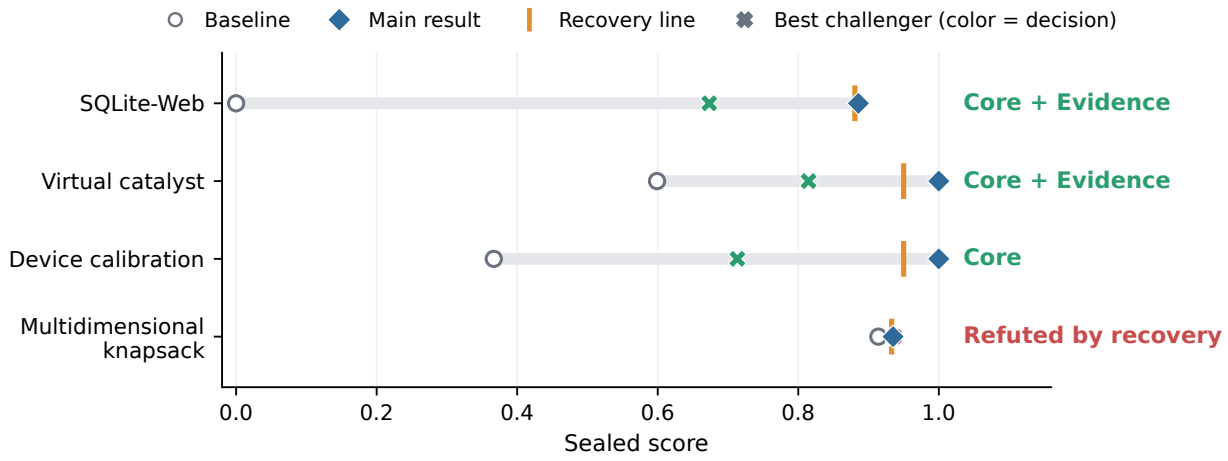

Figure 2: DCP applies the same outcome rule across artifacts. The best no-lineage challenger stays below the registered recovery line in the three Core cases and crosses it in knapsack. The challenger may use any valid method.

The multidimensional knapsack case exercises the recovery rule. Its target improved a fixed baseline, and one matched episode produced two different legal solutions above the recovery line. A low-sample affine case exercises audit incompleteness. Its two positive-control episodes leave the registered adequacy requirement unresolved despite an observed positive feedback contrast.

5 Results

Table 2 reports the primary decisions. Each score comparison is scoped to one task.

5.1 Useful outcomes have different recovery decisions

All four outputs passed Gate 1. SQLite-Web and catalyst each returned $0/96$ recoveries. Their positive controls passed $45/45$, giving a recall lower bound of 0.8889 above the registered 0.8 minimum, while the zero-hit upper bound was $0.0468 < \rho$. Device calibration returned $0/80$ with $60/60$ positive controls and an upper bound of 0.0477. The deterministic verifier returned DCP Core in all three scopes.

Developmental knapsack supplied an alternative route to the target. One matched episode returned legal artifacts scoring 0.9363 and 0.9356, both above the 0.9329 recovery line and the better one above the target run. These qualified witnesses triggered the Core veto through the shared numerical rule. Their evidential role is constructive recovery. The three positive Core audits additionally supply finite-sample probability bounds.

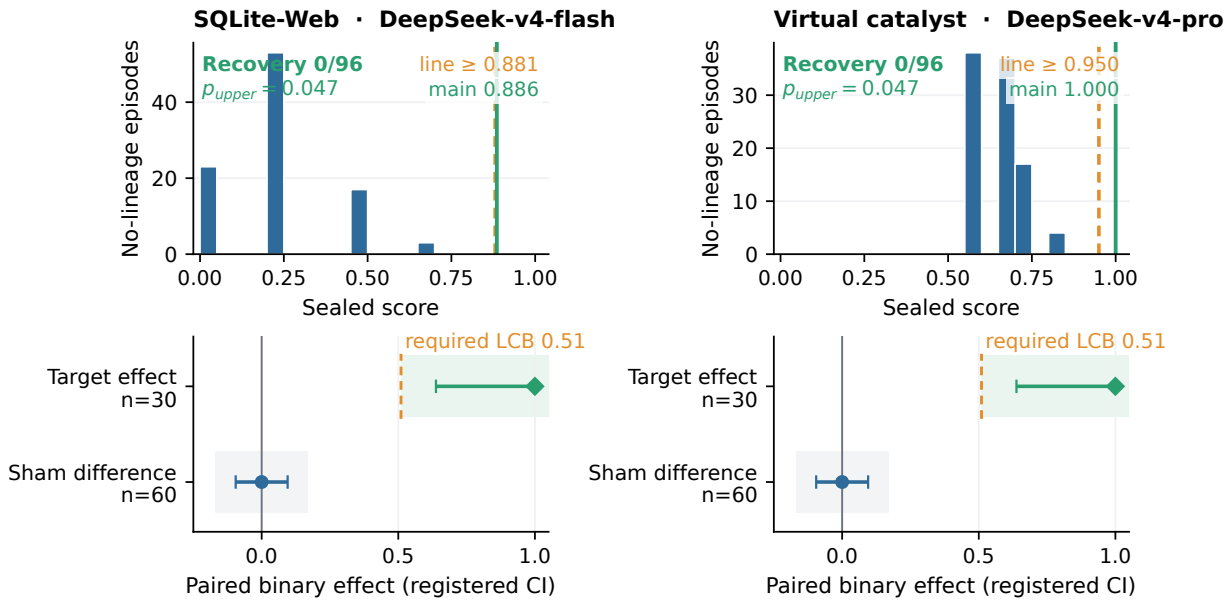

Figure 3: Complete three-gate evidence in two domains and model scopes. The upper row shows all 96 no-lineage scores per audit. The lower row shows the registered 99% intervals for the checkpoint-conditional feedback effect and the separate null calibration. Each column is one audit.

5.2 Two complete audits support a feedback increment

Figure 3 reports the two model-specific audits separately. SQLite-Web challengers received every captured Web byte and reached at most 0.6734, against a recovery target of 0.8805. Catalyst challengers reached at most 0.8146, against 0.95.

Both paired studies yielded $30/30$ truthful recoveries and $0/30$ neutral recoveries. The estimated binary policy effect was 1.0, with a conservative exact paired-binary 99% interval of $[0.6379, 1.0]$. Each separate 60-pair null study gave a contrast of 0 and a 99% interval of $[-0.0950, 0.0950]$, inside the registered ± 0.17 band. The target lower bound exceeded the required total margin $0.34 + 0.17 = 0.51$. Equivalently, the recorded margin calculation was $0.6379 - 0.17 = 0.4679 > 0.34$. The verifier returned Evidence in both scopes.

These Evidence decisions concern the average truthful-versus-neutral policy effect after the registered checkpoint. The records separately report a neutral-recovery upper bound of 0.1619 at 30 episodes. Average feedback benefit and neutral recovery therefore remain distinct reported quantities.

5.3 Controlled cases cover the decision space

Five cases exercise four decision outcomes. SQLite and catalyst support Evidence, device calibration supports Core, knapsack supplies qualified recovery, and affine remains audit incomplete. These controlled cases test the same rule across distinct information and execution conditions. Appendix Figure 4 gives the decision map.

6 Discussion

6.1 An outcome contract admits alternative methods

A component ablation evaluates an intervention within a chosen implementation. DCP evaluates recovery across every admissible implementation offered by the registered challenger. A recombination of known modules, a transfer from another domain, or a different program can all supply the same numerical witness. The knapsack case demonstrates this rule directly. Task adapters contribute the validity and utility definitions; the shared protocol contributes the information boundary and decision procedure. This gives heterogeneous research outputs a common audit interface.

6.2 Reporting a research contribution

A reusable report gives the outcome threshold, model and information scope, complete episode budget, qualified recovery witnesses or probability bound, and any checkpoint-conditional feedback interval. Authors can use this profile to state precisely what their experiments support. Reviewers can inspect alternative routes and replay the numerical decision. Later studies can register a new model or budget and compare recovery under that scope. Core and Evidence summarize these records while preserving the quantities needed for comparison.

Together, these fields turn discovery reporting into a common experimental record that authors can submit, auditors can challenge, and readers can verify.

6.3 Deployment and retrospective recovery

Prospective use starts with an approved registration and sealed evaluator, followed by evidence collection, control execution, offline replay, and independent countersigning. Auditors can stop at the first qualified recovery and reserve fresh paired studies for claims that request feedback evidence. The catalyst and SQLite-Web audits used 435 and 507 recorded sessions and cost 56.40 and 61.17 USD, respectively. Their completed decisions now replay offline. The expensive unit is the registered generation episode; a reader's verification operates on its saved evidence.

Retrospective use reconstructs an outcome target and seeks a qualified recovery witness. An audit with pre-publication-equivalent model and information access can test the original recovery scope. An audit using post-publication knowledge measures present-day recoverability. The record identifies this temporal scope explicitly. Positive Core and Evidence decisions use prospective registration and complete evidence collection.

7 Conclusion

The Discovery Certification Protocol makes evidence about AI research contributions executable. It follows one numerical outcome through sealed utility validation, recovery by matched agents given the starting information, and an optional randomized feedback comparison. Any qualified alternative method can supply a recovery witness. Core requires an adequate audit with zero recoveries and a finite-sample bound for one fresh registered episode. Evidence adds a checkpoint-conditional effect of truthful feedback over a specified neutral policy, supported by independent null calibration and a registered margin. These decisions give utility, alternative routes, and feedback effects separate, checkable meanings.

Two controlled audits exercise the full protocol under distinct models. DeepSeek-v4-flash reduced traffic-weighted SQLite virtual-machine work by 88.55%; DeepSeek-v4-pro reached the optimum in virtual catalyst control. Each audit recorded zero recoveries in 96 episodes and 30 truthful recoveries against zero neutral recoveries, with passing 60-pair null calibration. Device, knapsack, and affine cases exercised Core, recovered, and incomplete decisions. A shared deterministic verifier reproduced the complete decisions from frozen records. The resulting contribution is a portable experimental language that authors can report, auditors can challenge, and readers can verify across research domains.

Reproducibility Statement

Frozen machine-readable bundles and evidence records support the numerical claims. The appendices report task construction, registrations, model and interface contracts, evaluation units, thresholds, statistics, budgets, costs, and identifiers. The release provides the installable `dcp-audit`, task-specific evidence producers, reusable `dcp-harness`, example bundles, and figure-generation scripts. The package and repository links appear on the first page. Decision replay is deterministic and offline. Fresh evidence production uses the model and environment specified in the registration. The harness exposes capture, challenge, paired feedback testing, finalization, and bundle generation as separate operations.

Ethics Statement

AI research agents can accelerate useful discovery while increasing the need for clear and contestable evidence. DCP supports accountable reporting by binding each decision to a registered model, information boundary, budget, probability bound, and replayable record. Certificates should be presented with this scope, and captured Web data should follow applicable privacy, licensing, and security requirements. Deterministic replay gives authors, reviewers, and independent auditors a common record on which to examine a research claim.

AI Use Statement

Generative AI tools supported ideation, protocol and statistical design, adversarial review, experiment design, implementation, testing, evidence inspection, result interpretation, literature work, and manuscript editing. GPT and Codex assisted these activities. Claude interfaces supported review and provided the registered CLI execution layer. DeepSeek-v4-flash and DeepSeek-v4-pro served as the audited agents and matched challengers. Image generation informed early teaser drafts; the publication teaser is native SVG artwork with vector export. Quantitative plots were generated with Matplotlib from frozen evidence. Executable evaluators and statistical code determine scores and decisions. The authors review the scientific claims, check numerical evidence and primary sources, test the code, and take responsibility for the final content and artifacts.

References

Shitanshu Bhushan, Yunxiang Zhang, and Lu Wang. Can LLM agents discover? evaluating creativity on ML engineering tasks. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2608.30047.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, pages 2633–2650, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Madry. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.07095.

Hui Chen, Miao Xiong, Yujie Lu, Wei Han, Ailin Deng, Yufei He, Jiaying Wu, Yibo Li, Yue Liu, and Bryan Hooi. MLR-Bench: Evaluating AI agents on open-ended machine learning research. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025. doi: 10.52202/085713-3945.

João Coelho, Jingjie Ning, Jingyuan He, Kangrui Mao, Abhijay Sai Paladugu, Pranav Setlur, Jiahe Jin, Jamie Callan, João Magalhães, Bruno Martins, and Chenyan Xiong. DeepResearchGym: A free, transparent, and reproducible sandbox for deep research. In *Proceedings of the 2026 International ACM SIGIR Conference on Innovative*

Concepts and Theories in Information Retrieval, pages 34–43. ACM, 2026. doi: 10.1145/3805713.3820409. URL https://doi.org/10.1145/3805713.3820409.

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610:47–53, 2022. doi: 10.1038/s41586-022-05172-4.

Shahriar Golchin and Mihai Surdeanu. Data contamination quiz: A tool to detect and estimate contamination in large language models. *Transactions of the Association for Computational Linguistics*, 13:809–830, 2025. doi: 10.1162/tacl.a.20. URL https://aclanthology.org/2025.tacl-1.37/.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, et al. Accelerating scientific discovery with co-scientist. *Nature*, 655:487–496, 2026. doi: 10.1038/s41586-026-10644-y.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. DiscoveryWorld: A virtual environment for developing and evaluating automated scientific discovery agents. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, pages 10088–10116, 2024. doi: 10.52202/079017-0324. URL https://arxiv.org/abs/2406.06769.

Hengrui Jia, Mohammad Yaghini, Christopher A. Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. Proof-of-learning: Definitions and practice. In *2021 IEEE Symposium on Security and Privacy*, pages 1039–1056, 2021. doi: 10.1109/SP40001.2021.00106.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=iO4LZibEqW.

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.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-2019.

Daniel J. Mankowitz, Andrea Michi, Anton Zhernov, Marco Gelmi, Marco Selvi, Cosmin Paduraru, Edouard Leurent, Shariq Iqbal, Jean-Baptiste Lespiau, Alex Ahern, et al. Faster sorting algorithms discovered using deep reinforcement learning. *Nature*, 618:257–263, 2023. doi: 10.1038/s41586-023-06004-9.

Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ByJHuTgA-.

Sidhaarth Murali, João Coelho, Jingjie Ning, João Magalhães, Bruno Martins, and Chenyan Xiong. Beyond parallel sampling: Diverse query initialization for agentic search. *arXiv preprint arXiv:2606.17209*, 2026. URL https://arxiv.org/abs/2606.17209.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699, 2020. doi: 10.1007/978-3-030-58595-2_41.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Mikhail Plekhanov, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. MLGym: A new framework and benchmark for advancing AI research agents. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ryTr83DxRq.

Jingjie Ning, Xiaochuan Li, Ji Zeng, Hao Kang, and Chenyan Xiong. Auto research with specialist agents develops effective and non-trivial training recipes. *arXiv preprint arXiv:2605.05724*, 2026a. URL https://arxiv.org/abs/2605.05724.

Jingjie Ning, Xiaochuan Li, Ji Zeng, Chenyan Xiong, and Guolin Ke. Closed-loop auto research for molecular property prediction: Discovering and certifying generalizable improvements. *arXiv preprint arXiv:2606.22731*, 2026b. URL https://arxiv.org/abs/2606.22731.

Jingjie Ning, Xiaochuan Li, Shanshan Zhong, Ji Zeng, and Guolin Ke. Auto research for materials: Auditable AI-scientist workflows with held-out transfer. *arXiv preprint arXiv:2607.17100*, 2026c. URL https://arxiv.org/abs/2607.17100.

Jingjie Ning, Xueqi Li, and Chengyu Yu. Revision or re-solving? decomposing second-pass gains in Multi-LLM pipelines. In *Conference on Language Modeling*, 2026d. URL https://arxiv.org/abs/2604.01029.

Jingjie Ning, Shanshan Zhong, Xiaochuan Li, Ji Zeng, and Chenyan Xiong. One run is not an idea: The implementation lottery in automated research. *arXiv preprint arXiv:2607.26587*, 2026e. URL https://arxiv.org/abs/2607.26587.

Brian A. Nosek, Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor. The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11):2600–2606, 2018. doi: 10.1073/pnas.1708274114.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025. URL https://arxiv.org/abs/2506.13131.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024. doi: 10.1038/s41586-023-06924-6.

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974. doi: 10.1037/h0037350.

Donald J. Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6):657–680, 1987. doi: 10.1007/BF01068419.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-0377.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=uyTL5Bvosj.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 56843–56873. PMLR, 2025. URL https://proceedings.mlr.press/v267/starace25a.html.

Zhen Wang, Fan Bai, Zhongyan Luo, Jinyan Su, Kaiser Sun, Xinle Yu, Jieyuan Liu, Kun Zhou, Claire Cardie, Mark Dredze, Zhiting Hu, and Eric P. Xing. FIRE-Bench: Evaluating AI agents on the rediscovery of scientific insights. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.02905.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025. URL https://arxiv.org/abs/2504.08066.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023a. doi: 10.52202/075280-0517.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shanshan Zhong, Yi Lu, Jingjie Ning, Yibing Wan, Lihan Feng, Yuyi Ao, Leonardo F. R. Ribeiro, Markus Dreyer, Sean Ammirati, and Chenyan Xiong. SkillLearnBench: Benchmarking continual learning methods for agent skill generation on real-world tasks. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2604.20087.

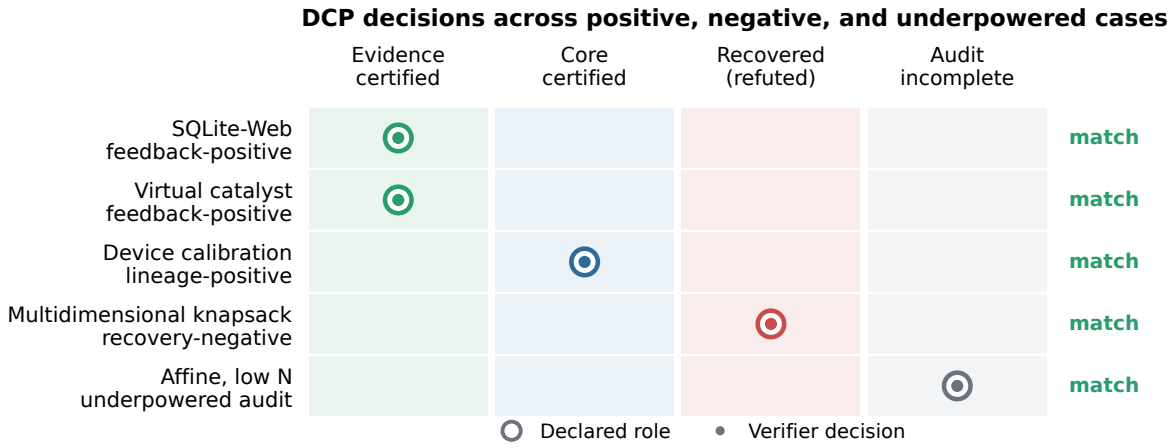

Figure 4: The suite spans two complete Evidence positives, one constructed information-deficit Core positive, one developmental recovery refutation, and one intentionally undersampled audit. Rings show the declared calibration role and filled dots show the deterministic verifier decision.

A Statistical Decision Details

A.1 Main result

For paired private evaluation units, DCP computes differences between $A^\star$ and the baseline. The sampled-population profile uses a bounded empirical-Bernstein interval on the registered $[0, 1]$ score scale. A claim over a fully enumerated finite set uses its exact mean. Score uncertainty, episode recovery uncertainty, and randomized feedback uncertainty each follow their registered sampling unit.

The value x is the registered evaluation mean of $A^\star$, selected before sealed scoring. The baseline must be valid and lie below the recovery region under its adjusted uncertainty check.

A.2 Candidate decisions

Each challenger candidate is scored on the same private units as $A^\star$. Let d_i be its paired score difference from $A^\star$. Candidate i is a confirmed hit when its adjusted lower confidence bound is at least $-\varepsilon$. It is a confirmed miss when its adjusted upper confidence bound is below $-\varepsilon$. Every other candidate is unresolved.

The family of candidate intervals shares a registered error budget. A simple profile assigns Bonferroni corrected error to every possible candidate slot, controlling false recovery across candidates that can change the decision. Equation 3 receives a separate error budget. Its prerequisites are zero qualified hits, resolved candidates, and a complete adequate control ledger.

Both complete audits allocated 0.005 each to main validation, candidate classification, positive-control adequacy, and neutral recovery, and 0.01 each to challenger recovery, the target feedback effect, and null calibration. The allocations total 0.05. Both used $\rho = 0.05$, 45 positive controls, and minimum control recall 0.8. Their feedback threshold was $\delta_{\text{evidence}} = 0.34$ and calibration margin was $\delta_{\text{sham}} = 0.17$.

When $\alpha_{\text{recovery}} = 0.05$, 59 independent misses put the one-sided upper bound near 0.05. With $\alpha_{\text{recovery}} = 0.025$, 72 misses are required. Device calibration allocated 0.02, required at least 77 misses, and registered 80 episodes. Each bound concerns one fresh episode from its registered model, policy, information, and budget distribution.

A.3 Recovery witnesses and repeated search

For independent Bernoulli episode outcomes with recovery probability p_B, the probability of zero recoveries in n episodes is $(1 - p_B)^n$. Thus a fixed-sample zero-hit audit reaches the registered bound when $n \geq \lceil \log(\alpha_{\text{recovery}}) / \log(1 - \rho) \rceil$. For the two complete audits, $\alpha_{\text{recovery}} = 0.01$ and $\rho = 0.05$ require 90 episodes; the registered count was 96. Candidate score classification and the other statistical gates retain their separate error allocations.

A witness establishes an available route, while the binomial bound quantifies the probability that a fresh episode produces any admissible recovery. DCP assigns them different decision roles. A campaign of k independent episodes from the same Q_B has recovery probability $1 - (1 - p_B)^k$. A best-of-k production procedure therefore includes its full search and selection budget in one registered episode. Fresh Gate 3 pairs estimate the checkpoint-conditional feedback effect separately from selection of a successful target.

A.4 Feedback decisions

The paired feedback estimator uses normalized utility differences or paired differences in recovery success, conditional on the frozen checkpoint. The utility and both policies are registered before the pairs start. Let $\delta_{\text{evidence}} > 0$ be the minimum effect and $\delta_{\text{sham}} > 0$ the registered calibration margin. Support requires passing null calibration and

$$\text{LCB}(\Delta_{\text{feedback}}) \geq \delta_{\text{evidence}} + \delta_{\text{sham}}.$$

The effect-below-threshold decision requires

$$\text{UCB}(\Delta_{\text{feedback}}) + \delta_{\text{sham}} < \delta_{\text{evidence}}.$$

Remaining interval positions are inconclusive. The null interval describes equivalence on the registered calibration family. Neutral recovery receives its own candidate checks and probability bound.

B Audit Validity

Valid numerical interpretation requires both branches to implement the registered comparison. DCP records the following evidence.

- The registration is sealed before any private result or confirmatory null case is opened.
- Model identity, interface, tools, information packet, and resource budget match their registered values.
- Every started attempt remains in the ledger, including timeouts, invalid outputs, and infrastructure failures.
- Candidate outputs are committed before private scores are returned.
- The main output, baseline, and controls use one symmetric evaluator.
- Each independent episode identifier corresponds to a committed draw record; its internal candidate opportunities share one statistical unit.
- All Web responses given to the model are recorded byte for byte when Web access is enabled.

A model failure under the correct contract remains a failed attempt. A model, information, or tool-contract violation makes the audit inconclusive. Pre-action infrastructure failures follow the registered attrition rule. Local hashes establish internal consistency; external registration and independent signatures establish the additional provenance required for formal issuance. These records specify the experimental comparison across agent architectures.

C Audit Case Details

C.1 SQLite with captured Web access

The registered assistant payload was DeepSeek-v4-flash. The main score was 0.8855470, the baseline score was 0, and the recovery line was 0.8805470. All 96 matched challenger episodes were valid confirmed misses. Their scores took four values between 0.0368978 and 0.6733847. The zero-hit upper bound was $1 - 0.01^{1/96} = 0.0468381$. All 45 positive controls succeeded, giving a one-sided recall lower bound of 0.8889268. Gate 1 used the exact enumerated finite workload and required a gain of $\delta_{\min} = 0.5$.

The gateway recorded two HTTPS responses from `sqlite.org`. Gate 2 received the normalized model-visible text and content hash for every response. The target feedback test used 30 fresh pairs. Truthful and neutral recovery counts were $30/30$ and $0/30$. The paired exact 99% interval was $[0.6379275, 1]$. In 60 fresh null pairs, both arms succeeded in every pair, giving a difference of 0 and an exact 99% interval of $[-0.0950339, 0.0950339]$.

The machine-readable record is `artifacts/dcp_sqlite_event_web_fullflow_confirmatory_v1_certificate/certificate.json`. Its identifier begins with `b3f0af47968ba2cb`. The self-contained publication Bundle containing the same certificate is `examples/audits/sqlite-web` and its identifier begins with `eff6cf27afa58da5`. All SQLite-Web decisions and Gate 3 numbers in this paper are read from this certificate.

The full recorded run used 507 model sessions and cost 61.17 USD.

C.2 Virtual catalyst optimization

The registered model identity was DeepSeek-v4-pro. The main score was 1.0, the baseline score was 0.5989989, and the recovery line was 0.95. Gate 1 required $\delta_{\min} = 0.35$, with recovery tolerance $\varepsilon = 0.05$. All 96 matched challenger episodes were valid confirmed misses, and the best score was 0.814649. The zero-hit upper bound was $1 - 0.01^{1/96} = 0.0468381$. All 45 positive controls succeeded, giving a one-sided recall lower bound of 0.8889268.

The target feedback study used 30 fresh pairs. Truthful and neutral recovery counts were $30/30$ and $0/30$. The paired exact 99% interval was $[0.6379275, 1]$. In 60 fresh null pairs, both arms had equal outcomes in every pair, giving a difference of 0 and an exact 99% interval of $[-0.0950339, 0.0950339]$.

The public replay bundle is `examples/audits/virtual-catalyst`. Its identifier is `fa17603f2184bb7`. The archived source bundle is `artifacts/dcp_catalyst_anchor_pro_confirmatory_v2/full_bundle`. The confirmatory execution used 419 recorded model sessions and 54.27 USD. The frozen development and confirmatory executions together used 435 sessions and 56.40 USD.

C.3 SQLite with Web disabled

This secondary Core audit predates the Web-enabled case and is retained for direct replay at `examples/audits/sqlite-core`. The main score was 0.9432886, the baseline was 0, and the recovery line was 0.9382886. All 80 challenger episodes were valid confirmed misses. The best challenger scored 0.7947267. The zero-hit upper bound was $1 - 0.02^{1/80} = 0.0477239$. All 45 positive controls succeeded, giving a one-sided recall lower bound of 0.8889268. Gate 1 used the exact enumerated finite workload and required $\delta_{\min} = 0.5$. The evidence bundle identifier is `3690bebc8b3ac882`. The 130 recorded model sessions cost 25.56 USD.

C.4 Device calibration

The public replay Bundle is `examples/audits/device-calibration-core`. The main score was 1.0, the baseline score was $110/300 = 0.3667$, and the recovery line was 0.95. All 80 challenger episodes were valid confirmed misses. The best score was $214/300 = 0.7133$. The zero hit upper bound was $1 - 0.02^{1/80} = 0.0477$. All 60 positive controls succeeded, producing a one sided Clopper Pearson lower bound of 0.9155. Gate 1 used the exact enumerated 300-item set and required $\delta_{\min} = 0.5$. The evidence bundle identifier is `dff7b57b062362e0`. The recorded provider cost was 19.08 USD.

C.5 Multidimensional knapsack

The main score was 0.9348968, the baseline score was 0.9140206, and the recovery line was 0.9328968. Two candidate opportunities inside one challenger episode produced scores 0.9362833 and 0.9355607. These qualified recovery witnesses trigger the Core veto. The record supports a constructive recovery decision. On the deterministic finite sealed set, the main gain was 0.0208762, above the registered $\delta_{\min} = 0.01$.

C.6 Development checks retained as inconclusive

An affine parity run exercised all three gates with two challenger episodes, two target pairs, and two null pairs. The observed main score and feedback contrast were both 1.0, while the positive-control recall lower bound was 0.0707. The verifier returned an incomplete audit at control adequacy. An active automaton run showed a 1.0 main score against 0.5 no-feedback outputs and remained incomplete in challenger adequacy and paired interface records. Earlier single-response induction tests measure E_0-dependence because their labeled examples were available at the start. These developmental records are reported separately from the primary audits.

D Audit Record

Each record contains the claim, information boundary, final output hash, baseline, sealed score data, recovery predicate, challenger contract, complete attempt ledger, control adequacy results, statistical intervals, and decision. An optional feedback record adds the checkpoint, paired branch contracts, neutral channel calibration, and paired estimates.

The reference verifier deterministically recomputes each decision from recorded evidence. Research agents and evidence producers supply records to this separate calculation. Readers can therefore replay the numerical decision offline. The reported bundles use a local content-addressed registry and record `formal_certificate_issued=false`. Their Core and Evidence fields contain the replayed kernel decisions. Independent provenance verification and registry countersigning add formal issuance to an accepted record. Historical priority remains separately recorded as `not_assessed`.