CBR-VERIFY: BUDGETED EVIDENCE REVISION FOR REUSABLE SCIENTIFIC AGENT WORKFLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific agents must revise the evidence supporting their conclusions when measurements, assumptions or evaluation protocols change. CBR-VERIFY formulates this decision as budgeted claim completion. Executable predicates define decisions, version-bound obligations define required evidence, and closure selection allocates computation between archive validation and conditional repair. The completion objective captures complementary prerequisites and shared acquisition costs; bounded seed enumeration ranks ready actions by covering density. In the previously reported 12-task GPT comparison, full CBR improves observed utility over Plan–Verify by 9.52 percentage points while using 7.85% fewer reported tokens and 12.13% more scientific credits. In the reported evaluation on 61 thermal runs, full CBR gains 5.71 utility points over a simple dependency-aware scheduler at the middle budget and saves 18.82% of credits at equal utility at the wider budget. In the author-checked numerical summary, full CBR gains 15.48 utility points over immediate-completion greedy with 4.36 fewer credits and gains 5.12 points over singleton selection. The corrected summary passes aggregate accounting checks; original records remain unavailable for independent reaggregation. Fixed-evidence feedback summaries examine reporting under a common interface. The interface separates evidence validity, predicate decisions and resource allocation across reusable workflows.

1 INTRODUCTION

An agent that revises a timestamp, calibration, or train–test split must also revise the conclusions that depend on it. Repeating every computation wastes resources; retaining a result requires current evidence for its interpretation. Our motivating thermal archive contains 61 runs, 60 wall-temperature channels, and 26,140 timestamped records. Its irregular sampling makes the time-selection contract part of the scientific result, rather than incidental bookkeeping. The archive combines repeated runs with spatially organized multichannel measurements, enabling controlled revisions on unchanged observations (Appendix A).

Time selection and evaluation splits define the computational question: nearest-sample diagnostics differ from exact-time interpolation, and row-random prediction differs from group transfer. CBR-VERIFY connects each report to this evaluation contract, using current claim evidence as the unit of both maintenance and action selection.

CBR-VERIFY couples three decisions: which claims merit the remaining budget, which archived artifacts can be reused, and which conclusions their current values support. Figure 1 shows the resulting lifecycle. Sources and assumptions enter a typed ledger; probes establish archive validity; repairs acquire missing evidence; predicates produce `supported`, `refuted`, or `insufficient`. The interface changes domain computations while preserving this control loop and the distinct roles of numerical results and evidence certificates.

The technical contribution is a *claim-completion action-selection layer*. A claim can require many complementary prerequisites, while several claims can share one expensive acquisition. We price their prerequisite unions and rank ready actions by the weight of claims the union completes. This criterion drives deterministic or LLM control. We derive complementary utility, shared-cost density, pair sufficiency conditions, and certificate locality, and evaluate policy quality, component behavior, and real-data contracts.

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.

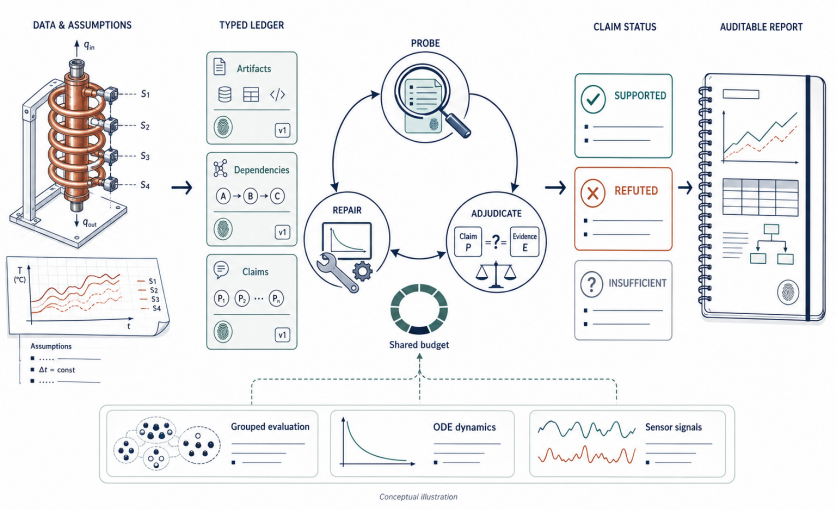

Figure 1: Reusable evidence-revision interface. Domain computations supply typed artifacts; a shared budget funds validation and repair; executable predicates produce claim status. The thermal apparatus is a conceptual motif.

2 Related Work and Technical Position

Scientific agents integrate experimentation and writing (Lu et al., 2024; Yamada et al., 2025); Curie structures experimental control and verification (Kon et al., 2025). ReAct and Reflexion supply reasoning–action and feedback-memory policies (Yao et al., 2023; Shinn et al., 2023). EviGraph already uses a typed evidence graph as operational state and repairs affected subgraphs; XScientist provides portable research records with claim anchors and re-execution hooks (Ren et al., 2026; Luo, 2026). Claim-aware observability explicitly records artifact lineage, evidence bindings and verification (Yin et al., 2026). CBR focuses on selecting weighted claim closures under a scientific budget with alternative validation and repair prices. Our policy comparisons use common-interface adaptations.

Truth maintenance supplies justification and retraction (Doyle, 1979; de Kleer, 1986). Workflow provenance, W3C PROV, and PROV-AGENT represent derivations and execution histories (Davidson & Freire, 2008; Moreau & Missier, 2013; Souza et al., 2025); DVC versions data and pipelines (DVC, 2026). Incremental computation and build systems already provide change propagation, dynamic dependencies, and reuse (Acar et al., 2006; Hammer et al., 2014; Mokhov et al., 2018). CBR adds a specific operational decision on top of this substrate: choose which *scientific claim closures* to acquire within a budget, with validation and replacement as alternative actions and executable predicates as admission conditions. A trace describes a derivation; our weighted completion objective determines which next action earns evidence for a report.

The fixed-cost subproblem is set-union knapsack (Goldschmidt et al., 1994). Complete-evidence rewards are supermodular, with increasing returns from complementary prerequisites; this identifies the appropriate closure unit for planning, distinct from objectives satisfying adaptive submodularity (Golovin & Krause, 2011). ScienceAgentBench and AstaBench motivate executable scientific assessment (Chen et al., 2025; Bragg et al., 2026), while POPPER supplies statistical hypothesis validation (Huang et al., 2025). Our evaluation separately records current evidence binding, numerical-reference agreement, and acquisition resources across scientific revisions.

3 CBR-VERIFY

Evidence contracts and decisions. Let $G = (V, E)$ be a declared dependency DAG with sources S and parents $\mathrm{pa}(v)$. Claim $c = (\phi_c, R_c, s_c, w_c)$ registers a Boolean predicate, required artifacts, scope metadata, and nonnegative weight; checked scope conditions are encoded in ϕ_c. The standalone

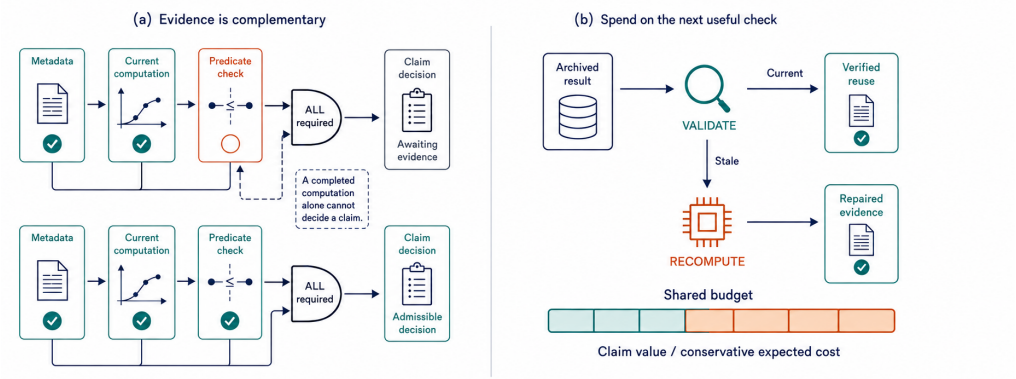

Figure 2: Evidence completion and conditional acquisition. A claim requires its joint obligations (a); validation selects reuse or repair (b). Shared acquisitions are charged once. A completed check can support or refute a claim.

ledger hashes canonical JSON signatures:

$$\sigma_v = (v, \tau_v, \nu_v, x_v, \{u \mapsto \bar{b}_{vu}\}_{u \in \mathrm{pa}(v)}), \qquad b_v = H(\sigma_v), \quad H = \mathrm{SHA256}. \tag{1}$$

Here τ_v, ν_v, x_v denote type, revision, and value; $\bar{b}_{vu}$ records the execution-time parent binding. With current-state marker $\chi_t(v)$,

$$V_t(v) = \chi_t(v)\mathbf{1}\{b_v = H(\sigma_v(t))\} \prod_{u \in \mathrm{pa}(v)} \mathbf{1}\{\bar{b}_{vu} = b_u(t)\} V_t(u), \qquad A_t(c) = \prod_{v \in R_c} V_t(v). \tag{2}$$

An explicit unresolved candidate dependency sets $A_t(c) = 0$ for affected claims. The checker evaluates the registered predicate on current evidence:

$$d_t(c) = \begin{cases} \texttt{supported}, & A_t(c) = 1 \ \wedge \ \phi_c(x_t(R_c)) = \text{true}, \\ \texttt{refuted}, & A_t(c) = 1 \ \wedge \ \phi_c(x_t(R_c)) = \text{false}, \\ \texttt{insufficient}, & \text{otherwise}. \end{cases} \tag{3}$$

Binding identifies the evidence used; the predicate determines the decision under its scope. Evaluation separately tests agreement with a numerical reference.

Proposition 1 (Certificate locality). For the standalone ledger with a complete declared DAG, unchanged deterministic tools, initially valid certificates, and collision-free comparisons, revising $D \subseteq S$ invalidates only $D^+ \setminus D$, where $D^+ = D \cup \mathrm{Desc}_G(D)$, and retracts claims with $R_c \cap D^+ \neq \varnothing$. Certificates outside D^+ remain valid: topological induction gives unchanged parents outside D^+ for every such node. Appendix F gives the proof.

Claim completion as the planning objective. Let O_c be fixed unpaid obligations, T completed obligations, and $q_o \geq 0$ their prices. Completion and single-obligation marginal gain are

$$F(T) = \sum_c w_c \mathbf{1}\{O_c \subseteq T\}, \qquad \Delta_e F(T) = \sum_{c : e \in O_c} w_c \mathbf{1}\{O_c \setminus \{e\} \subseteq T\} \quad (e \notin T). \tag{4}$$

Each marginal indicator is nondecreasing in T, proving monotonicity and supermodularity. A sole claim with $|O_c| > 1$ has zero initial single-action gain but complete-closure value w_c. Planning one seed claim therefore already spans arbitrarily many prerequisites. The fixed-cost problem is

$$\max_{z,y} \sum_c w_c y_c \quad \text{s.t.} \quad \sum_o q_o z_o \leq B, \quad y_c \leq z_o \ (o \in O_c), \quad z_o, y_c \in \{0,1\}. \tag{5}$$

This set-union knapsack objective values resolving a question, whether its answer supports or refutes the claim. Evidence-correct evaluation additionally requires the submitted answer to match the reference. Figure 2(a) illustrates the complementary evidence needed for adjudication.

Conditional acquisition and shared costs. The interface exposes `inspect`, `validate`, `recompute`, and `check`. With validation price v_a, repair price r_a, and configured stale probability p_a, a conclusive probe gives the planning model

$$\mathbb{E}_p C_a^{\text{probe}} = v_a + p_a r_a, \qquad S_a = (1 - p_a) r_a - v_a; \qquad \text{probe iff } S_a > 0. \tag{6}$$

The configured prior p_a becomes zero when inspected parent hashes match; a known mismatch triggers repair. Verified, unverified, stale, and known unavailable artifacts have respective prices

$$\ell_t(a) = 0, \quad \min\{r_a, v_a + p_a r_a\}, \quad r_a, \quad +\infty. \tag{7}$$

For seed claims K, let $I_t(K)$ collect uninspected sources potentially required by repair, $Q_t(K)$ remaining acquisitions, and k_c check prices:

$$\overline{C}_t(K) = \sum_{s \in I_t(K)} i_s + \sum_{a \in Q_t(K)} \ell_t(a) + \sum_{c \in K} k_c, \qquad \mathbb{E}_p C_t(K) \le \overline{C}_t(K). \tag{8}$$

For fixed acquisition plans, charging every possible inspection bounds its expected inspection cost under the configured marginal model; independence is unnecessary. The subscript p identifies the planning distribution, not an empirical calibration claim. Shared obligation identifiers are priced once. For two closures of costs C_1, C_2, overlap h, and no extra covered claims,

$$\rho_{12} = \frac{w_1 + w_2}{C_1 + C_2 - h}. \tag{9}$$

The overlap measures reusable work, not correlation between answers. Two unit-weight claims sharing an eight-credit input and requiring separate one-credit repairs score $2/10$, versus $1/9$ individually.

Closure ranking and optional pair expansion. Singleton seeds provide the base candidate family; adding pairs considers joint acquisitions. Enumerate $1 \le |K| \le 2$ with $0 < \overline{C}_t(K) \le B_t$. Let $U_t(K)$ denote visited nodes and paid obligations, and $\mathcal{C}_t(U)$ all unresolved claims covered by that union:

$$\rho_t(K) = \frac{\sum_{c \in \mathcal{C}_t(U_t(K))} w_c}{\overline{C}_t(K)}, \qquad s_t(u) = \max_{K:\, u \in \text{Ready}_t(K)} \rho_t(K). \tag{10}$$

One seed can cover several claims. Eligible ready zero-cost actions take priority; these are checks in the reported interfaces. Other actions follow decreasing s_t, then increasing actual price and identifiers. The deterministic selector replans after each action. The LLM receives refreshed proposals at each model call and can issue up to three sequential actions before its next observation.

Closure advisor Input: public state o_t, budget B_t, seed limit k.
1. Construct ready actions and unpaid claim-closure acquisition plans.
2. If an eligible ready zero-cost action exists, return such actions in interface order.
3. Enumerate seed unions $1 \le |K| \le k$; retain $0 < \overline{C}_t(K) \le B_t$.
4. Include every covered claim; assign each ready action its maximum $\rho_t(K)$.
5. Rank by score, actual price, tool and target; return at most six proposals.
6. Execute a selected proposal through the actual-cost gate; refresh the state.

Proposition 2 (Closure-density completeness). Fix the public state, acquisition plans, and ready map with no eligible ready zero-cost action. For the nonempty family $\mathcal{K}_2$ of feasible singleton/pair closures with a ready action,

$$\max_u s_t(u) = \max_{K \in \mathcal{K}_2} \rho_t(K). \tag{11}$$

If every feasible closure has the same visited-node and positive-cost obligation union as a singleton or pair, this maximum covers all feasible closures: equal unions preserve coverage, cost and density. This is a fixed-state density result. The $n + \binom{n}{2}$ candidate count concerns seed closures, not tool steps or the number of decisions covered. Appendix H gives complexity and dynamic-scope arguments.

Execution interfaces and reusable adapters. Adapters register source contracts, tools, prerequisites, prices and predicates. The *checker gate* produces Eq. 3. The *actual-cost gate* admits only $0 \le q_t \le B_t$, giving $B_{t+1} = B_t - q_t \ge 0$ and $\sum_{j<t} q_j \le B_0$, including model overrides. The *final-report interface* accepts labels separately: deterministic control copies checked labels or abstains; LLM submissions receive independent scoring. Archives persist across rounds. The standalone ledger

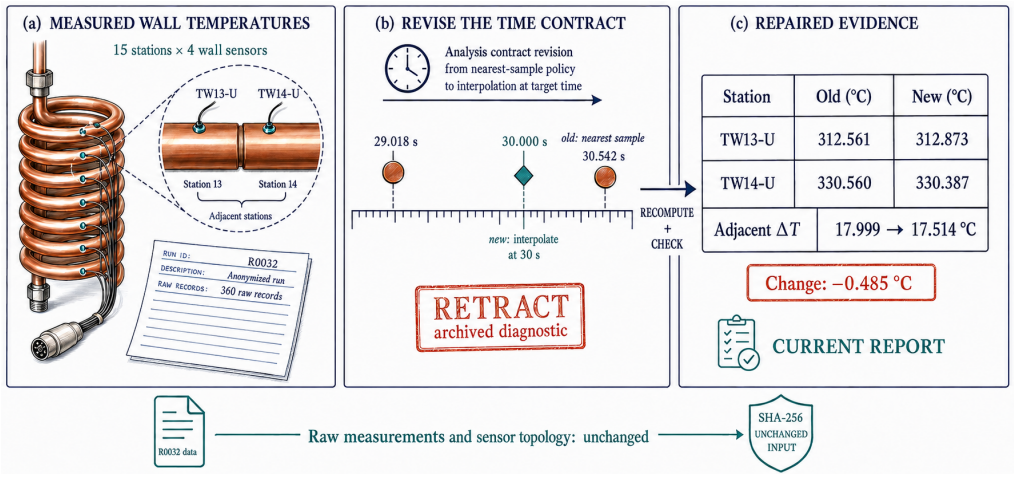

Figure 3: Recorded thermal revision with an executable claim. R0032 satisfies the predeclared 18°C diagnostic threshold under both nearest-time selection (17.999°C) and interpolation (17.514°C). The current round detects the stale binding and establishes a checked `supported` decision for eight scientific credits. "Retract" annotates the obsolete binding's state. The earlier value is a historical numerical reference; the apparatus is schematic.

suspends decisions for explicit candidate dependencies and invalidates affected artifacts when an edge is confirmed. Numerical, stress and thermal adapters expose verified states through their own binding digests and contract transitions; the protocol scores outcomes against separate numerical references. The recursive certificate specification and locality proof concern the standalone ledger.

Each proposal exposes its action, actual price, planned closure price and covered claim identifiers, connecting workflow state to the weighted decision-completion objective.

A concrete scientific revision. Figure 3 instantiates the predicate $\max_i[\max_d T_{i+1,d}(30\,\mathrm{s}) - \max_d T_{i,d}(30\,\mathrm{s})] \le 18°\mathrm{C}$ on recorded run R0032, with axial station i and wall-sensor direction d. The 18°C value is a predeclared diagnostic protocol threshold. Nearest-time selection at 30.542 s gives 17.999°C; interpolation at 30 s gives 17.514°C. Both satisfy the predicate. The current trajectory validates the archive, detects the stale time contract, inspects four parent contracts, recomputes the diagnostic, and obtains `supported` with current evidence. It spends $1+4+3=8$ credits; checking is free. The unchanged verdict and changed binding demonstrate two distinct outputs of evidence revision.

4 Experimental Design

This section describes the previously reported protocols; the independent thermal follow-up is identified separately in the results. Original records for Table 2 have not been recovered for independent reaggregation; author-checked corrections now pass the aggregate consistency checks in Appendix E.1.

Numerical tasks and observable information. Scientific Revision contains 12 development and 36 test tasks across grouped prediction, damped dynamics, and irregular spectra, with disjoint generation seeds. Each task has three rounds and five weighted claims, with unchanged, single, cascading or unavailable-reference revisions. Public observations expose parents, bindings, prices, metrics and predicates; future revisions and reference decisions remain hidden. The earlier audit reports agreement on 656 evaluable predicates; 704 numerical comparisons differ by at most 5.25×10^{-9}.

Fixed paired controller comparison. The new GPT batch compares full CBR, singleton-only, no-sharing planning prices, no advisor, and a Plan–Verify role adaptation on the same 12 test tasks and task-generation seeds. The minimum-seed task in each family–revision stratum defines the subset: one three-round trajectory per policy/task, 60 in total. Each round provides 11 scientific credits, six model-call slots and at most three sequential actions per call; the last slot is reserved

for submission. These limits, initial states, observation interfaces and objectives are shared across policies. Advisor variants change proposal rules; no-advisor removes proposals. The provider returns `gpt-5.6-luna`: temperature zero, reasoning effort `none`, maximum 700 output tokens, and no provider generation seed. Each call includes current state, last tool feedback and at most 450 characters of model memory.

Sources, prompts, ceilings, contrasts and shuffled execution order were frozen after eight development trajectories. The supplement describes resource/schema amendments and their excluded runs; historical cohorts remain separate.

Documented selector protocol. The five selectors are full CBR, singleton, no-sharing, immediate-completion greedy, and full-subset enumeration (Appendix E). Table 2 reports 36 test tasks, 11 credits/round and $p = .10$, with a $p = 0$ contrast. A separately reported sweep covers 48 development/test tasks, five selectors and seven budget/prior settings: 1,680 trajectories. Its $p = .5$ local-density audits and paired statistics are not transferred to Table 2.

Endpoints and paired inference. For reference decision d^*_{irc}, submission $\widehat{d}_{irc}$ and decisive labels $\mathcal{D} = \{\texttt{supported}, \texttt{refuted}\}$, the specified task scores are

$$U_i = \frac{\sum_{r,c} w_c \mathbf{1}\{\widehat{d}_{irc} = d^*_{irc} \in \mathcal{D}\} A_{ir}(c)}{\sum_{r,c} w_c}, \qquad A_i = \frac{1}{15} \sum_{r,c} \mathbf{1}\{\widehat{d}_{irc} = d^*_{irc}\}. \tag{12}$$

Accuracy A measures reference agreement; utility U additionally requires current evidence for a decisive answer. F counts decisive outputs with wrong labels or insufficient evidence, and D counts all decisive outputs. The specified numerical/GPT scorer credits a valid answer with or without a literal check call. Credits and tokens are per-task resources. Tokens include reported usage from retries; unreported failed-request usage remains unknown and reserves its configured cost upper bound in the separate fee estimate.

The reported GPT controller and fixed-state feedback analyses use whole task trajectories. Their previously reported paired differences use 95% percentile intervals from 10,000 family-stratified whole-task bootstrap draws. The three primary GPT contrasts compare full CBR with Plan–Verify, singleton and no-sharing on U, A, credits and tokens; two-sided sign-flip tests use Holm adjustment across these 12 comparisons. No-advisor contrasts are descriptive. The standalone supplement preserves paired statistics. Intervals describe task variation for this generator and family mix.

Real-data policy evaluation. The 61 thermal runs undergo five predeclared three-round sequences: unchanged, time/window, calibration, channel availability, and combined revisions. Full CBR, singleton, simple dependency-aware scheduling and direct recomputation share five diagnostic claims and budgets 7, 11 and 15. Both controls reserve missing inspections plus full repair cost in fixed axial/bulk/directional order and prioritize free checks. After parent inspection, the simple control validates matching bindings and repairs mismatches; direct recomputation always repairs. This gives 3,660 budgeted trajectories and 305 separate unbudgeted references. The diagnostic protocol includes Fig. 3's $18°$C threshold. The earlier audit reports 5,368 numerical checks differing by at most 1.14×10^{-13}. Means average scenarios within runs, runs within day/condition components, then the three components equally. The reported counts imply 59,475 claim outcomes; current access to their action records is unverified. Runtime and credits remain distinct.

5 Results

5.1 Paired model-guided controllers

Table 1 reports all 60 trajectories. Relative to Plan–Verify, full CBR improves paired utility by 9.52 points (95% interval $[5.16, 13.89]$); accuracy improves by 10.56 points $[5.00, 16.11]$. Utility wins/ties/losses are 7/5/0. Full CBR produces 125 decisive outputs with zero false admissions, versus 107 and one for Plan–Verify. These outcomes connect closure proposals to the controller's ability to complete and report current evidence under matched interaction limits.

Figure 4 separates decision quality from interaction cost. Relative to Plan–Verify, full CBR uses 7.85% fewer reported tokens $[1.45, 15.05]$ and 15.33 rather than 18 successful calls per task, while

Table 1: Paired GPT controllers (12 tasks; one trajectory per cell). U/A: means $\pm$ task SD. F/D: totals over 180 claim-rounds per policy. Reported tokens, scientific credits and successful calls: per-task means. Unknown token usage: Singleton (4 requests), No advisor (2 requests).

Policy	U (%)	A (%)	F/D	Tokens (k)	Credits	Calls
Full CBR	69.84 ± 24.22	78.33 ± 16.61	0/125	37.90	22.33	15.33
Plan–Verify	60.32 ± 26.58	67.78 ± 21.62	1/107	41.13	19.92	18.00
Singleton-only	69.44 ± 23.46	77.22 ± 14.62	0/123	38.01	21.25	15.42
No-sharing	65.08 ± 26.03	72.78 ± 20.78	0/115	38.30	20.92	15.50
No advisor	63.10 ± 27.78	71.11 ± 20.17	5/117	34.47	23.50	14.58

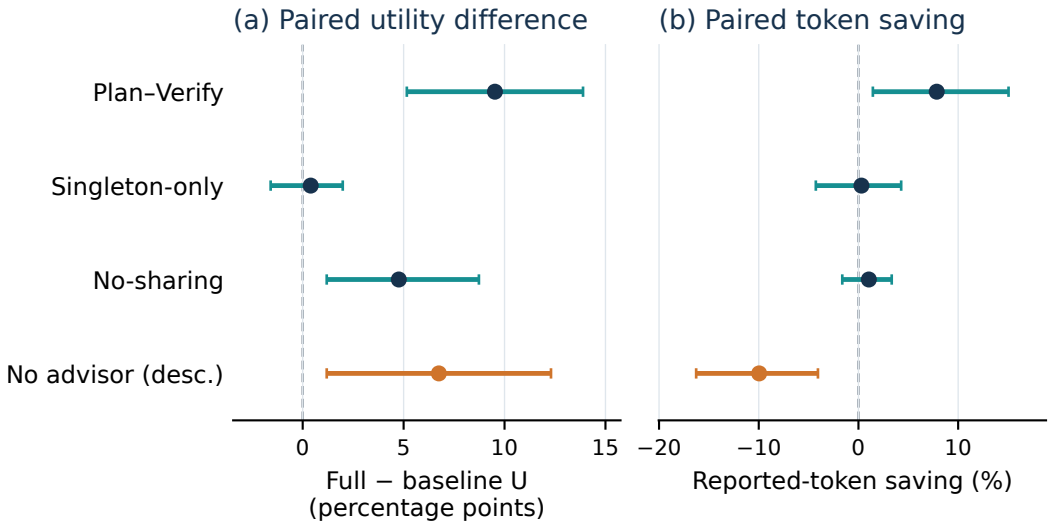

Figure 4: Paired controller effects: utility difference (left) and reported-token savings (right), both positive in favor of full CBR. Negative savings mean higher reported use. Bars: marginal 95% task-bootstrap intervals; orange: descriptive contrast. Singleton/no-advisor totals exclude unknown tokens from four/two failed requests.

spending 2.42 more scientific credits $[1.08, 3.92]$. The controller allocates more work to scientific tools while using fewer model interactions; the two resource axes are not interchangeable. Against no-sharing, utility increases by 4.76 points $[1.19, 8.73]$. Against singleton, the difference is .40 points $[-1.59, 1.98]$, giving no clear pair-expansion benefit in this GPT cohort. The 12 prespecified tests have Holm-adjusted $p \geq .0938$; the reported improvements are observed paired effects, not family-wise significant superiority. No-advisor has the lowest reported-token count but has five false admissions and $U = 63.10\%$. This descriptive comparison places the cost of structured proposals alongside their evidence-completion outcome.

5.2 Deterministic-selector comparison

Table 2 reports the author-checked selector summary. The corrected entries are compatible with the stated accuracy, decisive counts and integer-cost protocol (Appendix E.1). Original execution records remain unavailable for independent reaggregation; contrasts use the displayed means.

Completion and shared planning prices. At $p = .10$, full CBR's reported utility exceeds immediate-completion greedy by 15.48 percentage points, with accuracy 16.11 points higher and 4.36 fewer credits. Relative to singleton and no-sharing, its utility is 5.12 and 6.96 points higher while credits are 2.39 and 1.11 lower. These configuration-specific aggregates favor pair expansion and shared pricing; paired uncertainty and task-level causes cannot be recovered from the means. Full-subset has 1.22 points lower utility and uses 3.14 more credits than full CBR.

At $p = 0$, full CBR's utility is 4.76 points higher and credits 0.67 lower than singleton/no-sharing. No confidence intervals or task win counts are inferred from these aggregates.

Table 2: Author-checked deterministic-selector summary: 36 test tasks, 11 credits/round, default $p = .10$; F/D over 540 claim-rounds and credits per three-round task. Corrected entries pass aggregate count and integer-cost compatibility checks (Appendix E.1); original execution records have not been independently reaggregated.

Selector	U (%)	A (%)	F/D	Credits
Full CBR	81.35	90.00	0/438	21.22
Singleton-only	76.23	82.78	0/399	23.61
No-sharing planning prices	74.39	79.81	0/383	22.33
Immediate-completion greedy	65.87	73.89	0/351	25.58
Full-subset enumeration	80.13	87.04	0/422	24.36
Documented $p = 0$ contrast, same budget and tasks				
Full CBR	78.57	87.22	0/423	20.50
Singleton / no-sharing	73.81	82.78	0/399	21.17

Table 3: Feedback on 36 fixed states (12 tasks). U/A: means $\pm$ task SD; F/D and label corrections/harms: totals over 180 claim-rounds per arm. Reported tokens: per-task logical costs including the shared provisional response, thus nonadditive across arms. Unknown token usage: No feedback (2 requests).

Feedback arm	U (%)	A (%)	F/D	Tokens (k)	Corr./harm
Check feedback	81.35 ± 20.25	90.00 ± 7.25	0/146	10.40	9/0
No feedback	80.56 ± 20.55	86.11 ± 11.18	7/151	9.79	2/0
Reflection	80.56 ± 20.55	86.11 ± 11.18	7/151	9.82	2/0

Relation to the method. Closure planning values complementary prerequisites with zero immediate single-action completion. A larger seed family can improve fixed-state density without improving final utility; Appendix H gives the theoretical limits. The separate $p = .5$ local audit (Appendix D) retains its own configuration. The corrected summary motivates a matched, execution-level comparison with stronger closure-aware controllers.

5.3 Checking with matched interaction slots

We freeze 36 evidence states from deterministic full acquisition on the same 12 tasks, removing generation-time checked labels and private references from model observations. Each state receives one provisional response, shared verbatim across three final-response arms: actual checker feedback, no additional feedback, and general reflection. All arms have two logical calls and the same output cap; no new acquisitions are permitted. The design records 36 shared initial and 108 final responses. It isolates feedback at fixed evidence and interaction slots; Table 3 also reports the different input-token costs.

Actual feedback corrects nine initial labels without harming any initially correct label; either control corrects two. It yields $A = 90.00\%$ versus 86.11%, a paired difference of 3.89 points $[.56, 9.44]$, and zero rather than seven false admissions. Its seven additional corrections comprise five justified abstentions and two corrected decisive labels. The utility difference is .79 points $[0, 1.59]$; all four primary quality tests have Holm-adjusted $p = 1.00$. Thus this cohort records improved reporting from explicit feedback at fixed evidence, with a small utility effect and no family-wise significant quality contrast. Reflection and no-feedback have identical endpoints. Checking communicates the executable predicate's current verdict; it does not acquire missing scientific evidence.

5.4 Complete policies on recorded thermal data

Figure 5 preserves the reported budgeted decisions on the same observations and revision sequences. At 11 credits per round, full CBR achieves $U = 71.43\%$ and $A = 72.00\%$, against 65.71% and 64.00% for the simple dependency-aware scheduler, using 20.8 versus 22.6 credits per three-round task. At budget 15, both achieve $U = 94.29\%$ and $A = 94.67\%$; CBR uses 27.6 versus 34.0 credits, saving 18.82% of registered scientific work. Direct recomputation reaches $U = 85.71\%$ at 39.0 credits. These retained cells report zero false admissions under checked reporting.

Budget adequacy and independent follow-up. In a subsequently conducted independent thermal experiment, full CBR achieves 47% utility at the low budget of 7 credits per round. This result is sep-

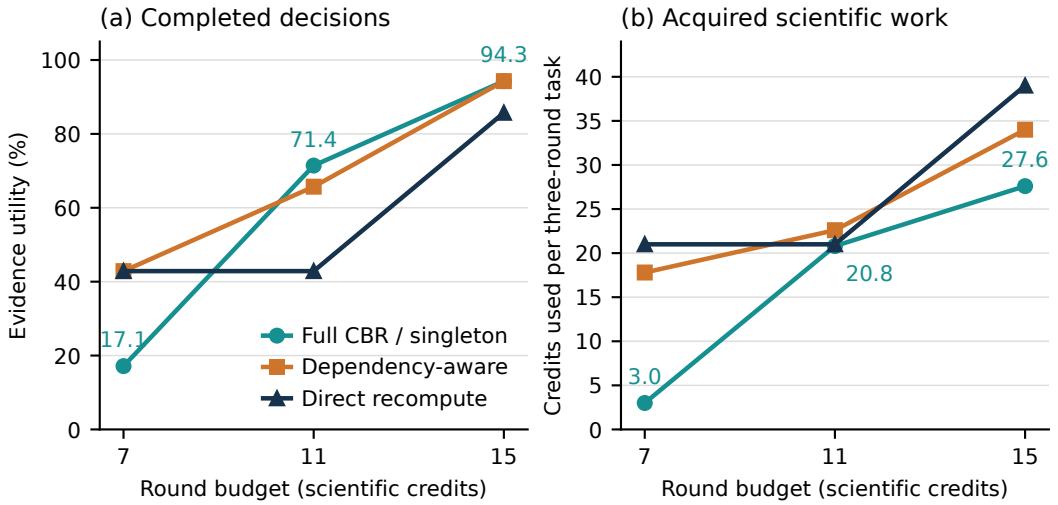

Figure 5: Earlier recorded-thermal cohort: 61 runs, five revision sequences and three budgets, with three day/condition components weighted equally. The displayed full/singleton means coincide; this does not assert identical trajectories. The later independent follow-up is reported in the text, not pooled into these curves. Repeated scenarios are not independent runs.

arate from Tables 1–3 and the earlier thermal aggregates in Fig. 5. The follow-up reports differences in 43 of 200 paired test trajectories, with full judged more consistent with the validation outcome. Matching aggregate means do not establish trajectory equivalence. These subset observations indicate behavioral differences; without paired scores and a specified adjudication rule, they do not quantify 43 utility wins (Appendix E.2).

Concrete decisions and transferable contracts. The time-selection example in Fig. 3 changes the evidence binding while preserving the verdict. Calibration exercises a different mechanism: for a common offset δ,

$$\bar{T}^{\,\delta}(t) = \bar{T}(t) + \delta, \qquad [T_{i+1}(t) + \delta] - [T_i(t) + \delta] = T_{i+1}(t) - T_i(t). \tag{13}$$

Mean temperature changes while adjacent differences remain invariant. The same control interface accommodates these distinct contracts. Previously reported gas-sensor and concrete-data audits examine split revisions separately (Vergara, 2012; Yeh, 1998; Kapoor & Narayanan, 2023).

Credits measure declared work, while runtime includes controller overhead: at budget 15, full and simple trajectories average 8.91 and 5.20 milliseconds. The reported decomposition separates scheduler and tool time so other workflows can substitute measured operation prices.

6 Conclusion

CBR-VERIFY couples claim-completion value, shared costs and conditional repair. Its analysis explains complementary prerequisites and the limits of local-density and expected-cost decisions. Reported GPT and thermal summaries describe configuration-dependent tradeoffs. The corrected selector aggregates are internally compatible and favor pair expansion and shared pricing at the reported setting; execution-level verification and stronger closure-aware comparisons remain necessary. Planned release covers recoverable code, protocols and an authorized thermal subset; new replications require distinct records.

AI USE STATEMENT

AI assistants supported literature retrieval, method design, implementation, analysis, review, writing, and illustrations. The prior reports describe DeepSeek and GPT trajectories. Conceptual images received author edits; quantitative plots preserve reported summaries. Missing records were not reconstructed. Human authors are responsible for verification, interpretation, and submission.

REPRODUCIBILITY STATEMENT

Appendix A documents the data structure and functions; Appendix B separates numerical and scientific validity. Complete derivations state their assumptions. Appendix E.1 documents Table 2's resolved aggregate checks and remaining record-availability limits. Other cohorts preserve previously reported summaries and uncertainty, without revalidation here. Appendix E distinguishes available manuscript sources from planned release materials. Recovery or separately identified new runs are needed for an auditable execution package.

REFERENCES

Umut A. Acar, Guy E. Blelloch, and Robert Harper. Adaptive functional programming. *ACM Transactions on Programming Languages and Systems*, 28(6):990–1034, 2006. doi: 10.1145/1186632.1186634.

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi, Sergey Feldman, Dany Haddad, Jena D. Hwang, Peter Jansen, Varsha Kishore, Bodhisattwa Prasad Majumder, Aakanksha Naik, Sigal Rahamimov, Kyle Richardson, Amanpreet Singh, Harshit Surana, Aryeh Tiktinsky, Rosni Vasu, Guy Wiener, Chloe Anastasiades, Stefan Candra, Jason Dunkelberger, Dan Emery, Rob Evans, Malachi Hamada, Regan Huff, Rodney Kinney, Matt Latzke, Jaron Lochner, Ruben Lozano-Aguilera, Cecile Nguyen, Smita Rao, Amber Tanaka, Brooke Vlahos, Peter Clark, Doug Downey, Yoav Goldberg, Ashish Sabharwal, and Daniel S. Weld. AstaBench: Rigorous benchmarking of AI agents with a scientific research suite. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.21652v2.

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. URL https://arxiv.org/abs/2410.05080v3.

Susan B. Davidson and Juliana Freire. Provenance and scientific workflows: Challenges and opportunities. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 1345–1350. ACM, 2008. doi: 10.1145/1376616.1376772.

Johan de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28(2):127–162, 1986. doi: 10.1016/0004-3702(86)90080-9.

Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12(3):231–272, 1979. doi: 10.1016/0004-3702(79)90008-0.

DVC. Pipelines, 2026. URL https://doc.dvc.org/user-guide/pipelines. Official documentation, accessed 16 September 2026.

Olivier Goldschmidt, David Nehme, and Gang Yu. Note: On the set-union knapsack problem. *Naval Research Logistics*, 41(6):833–842, 1994. doi: 10.1002/1520-6750(199410)41:6⟨833::AID-NAV3220410611⟩3.0.CO;2-Q.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011. doi: 10.1613/jair.3278.

Matthew A. Hammer, Khoo Yit Phang, Michael Hicks, and Jeffrey S. Foster. Adapton: Composable, demand-driven incremental computation. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pp. 156–166. ACM, 2014. doi: 10.1145/2594291.2594324.

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 of *Proceedings of Machine Learning Research*, pp. 25372–25437, 2025. URL https://proceedings.mlr.press/v267/huang25n.html.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

Patrick Tser Jern Kon, Jiachen Liu, Qiuyi Ding, Yiming Qiu, Zhenning Yang, Yibo Huang, Jayanth Srinivasa, Myungjin Lee, Mosharaf Chowdhury, and Ang Chen. Curie: Toward rigorous and automated scientific experimentation with AI agents, 2025. URL https://arxiv.org/abs/2502.16069.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery, 2024. URL https://arxiv.org/abs/2408.06292.

Jixiang Luo. XScientist: A git-like research protocol for long-running autonomous scientific discovery, 2026. URL https://arxiv.org/abs/2607.12301.

Andrey Mokhov, Neil Mitchell, and Simon Peyton Jones. Build systems à la carte. *Proceedings of the ACM on Programming Languages*, 2(ICFP), 2018. URL https://simon.peytonjones.org/assets/pdfs/build-systems-original.pdf.

Luc Moreau and Paolo Missier. PROV-DM: The PROV data model. W3C Recommendation, 30 April 2013, 2013. URL https://www.w3.org/TR/prov-dm/.

Zhenjiang Ren, Ruiji Li, Xujing Zhang, Ziliang Pang, Shuo Ren, and Jiajun Zhang. EviGraph: Evidence-guided autonomous research agents, 2026. URL https://arxiv.org/abs/2608.04738.

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*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Renan Souza, Amal Gueroudji, Stephen DeWitt, Daniel Rosendo, Tirthankar Ghosal, Robert Ross, Prasanna Balaprakash, and Rafael Ferreira da Silva. PROV-AGENT: Unified provenance for tracking AI agent interactions in agentic workflows. In *IEEE 21st International Conference on e-Science*, 2025. URL https://arxiv.org/abs/2508.02866.

Alexander Vergara. Gas sensor array drift dataset. UCI Machine Learning Repository, 2012. URL https://doi.org/10.24432/C5RP6W.

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, 2025. URL https://arxiv.org/abs/2504.08066.

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*, 2023. URL https://arxiv.org/abs/2210.03629v3.

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

Xiangyu Yin, Ming Du, Michael H. Prince, and Mathew J. Cherukara. Artifact-centered claim-aware observability for autonomous scientific agents, 2026. URL https://arxiv.org/abs/2608.18312.