PAY ONCE, CERTIFY EVERY ROUND: ANYTIME-VALID VERDICTS FOR SELF-IMPROVING PROGRAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

A program that improves itself repeats one round. It proposes a batch of candidates, edited copies of its own code, scores them on a fixed set of tasks, and keeps at most one. Each score costs a model call on one task, so the cost of a round is the number of program–task cells it reads. This paper asks how few cells a round can read, together with the decision rule that makes that small sample enough. We present BranchCert, a certificate that returns one of two verdicts for the round: ACCEPT for a named candidate that beats the parent, the program the loop is running, or NONE BETTER when no candidate does. It returns UNDECIDED when the budget runs out first. The certificate reads every candidate and the shared parent along one random order of the tasks. The unread tasks then form a uniformly random sample of the table, which lets the certificate stop as soon as the comparison is settled and bound its error at whatever time it stops. A second form spends the paid history of the loop. Earlier rounds scored other programs on the same tasks, those scores predict how the current batch will fare, and the certificate measures the residual the prediction leaves. On recorded rounds of a running self-improvement loop, on public evaluation tables, and on a paired re-execution of a recorded round, the certificate reaches the verdict of the complete table after a fraction of the reads. It agrees with the complete table in every run on the recorded loop, errs in at most 0.5% of runs on the public tables, and lowers its mean cost as the loop accumulates paid history.

1 INTRODUCTION

A self-improving program proposes a batch of candidate edits, evaluates them against its current version (the parent), and keeps at most one (Duan et al., 2026; Zhang et al., 2026a; Fan & Huang, 2026; Agrawal et al., 2026). Evaluating every candidate on every task can require many model calls. We ask a direct question: can the loop decide whether to accept an edit without completing all of those evaluations?

We study rounds with a fixed, finite task set and a binary score for each program–task pair. A candidate must meet the required improvement on every target. The rule must either identify a qualifying candidate, conclude that none qualifies, or report insufficient evidence. A small sample can give the wrong answer, especially when the rule chooses when to stop. Confidence sequences control error under early stopping (Howard et al., 2021; Waudby-Smith & Ramdas, 2024; 2020); here that control must cover the whole batch. Two sources of savings are available: candidates share the same parent, and earlier rounds have already paid to evaluate programs on the same tasks.

We present BranchCert (Figure 1), a rule that uses both. Its exact form bounds every unread score by its possible values. Its statistical form, BranchCert-Σ, reads candidates along a shared random order of tasks and stops when confidence bounds establish either verdict. Each parent score is paid for once and reused across candidates. BranchCert-Σ+H also uses earlier scores to predict current paired differences, then applies the bounds to the prediction residuals. Predictions and valid ranges are fixed before sampling; inaccurate predictions can affect cost without invalidating the error guarantee. The guarantee concerns the frozen score table, with a prespecified error budget that can be shared across rounds. Budget exhaustion returns UNDECIDED.

We evaluate these rules on public model-evaluation tables, recorded self-improvement chains, and tables from live tool-use rounds. On the chains, BranchCert-Σ+H reads a mean 19.4% of the table

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.

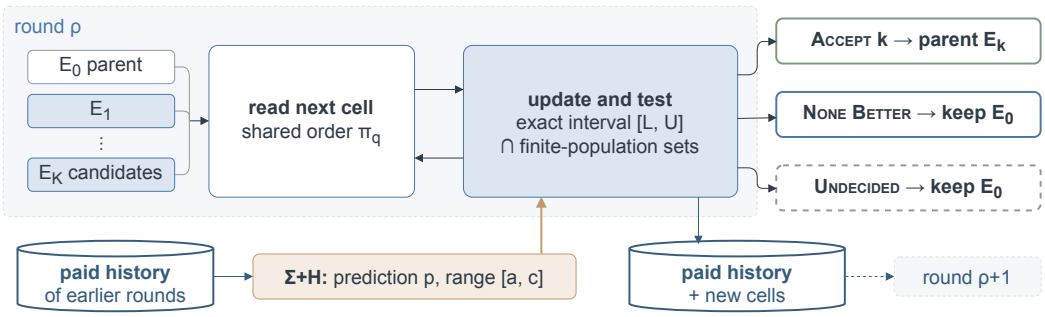

Figure 1: One round of BranchCert with paid-history reuse.

and observes no wrong verdict. Its mean saving over BranchCert-Σ reaches 22.0% in rounds five to eight. Public-table error rates remain below the specified level. Fresh-task evaluations measure whether the accepted edits generalize, and a live re-scoring study checks whether a recorded verdict persists when scores change. These measurements complement the finite-table guarantee; they do not extend it to arbitrary future tasks or stochastic re-evaluations.

2 Related work

Update gates for self-improving systems. A self-improving loop keeps whatever its gate admits, so the gate decides how the loop evolves (Duan et al., 2026; Zelikman et al., 2024; Robeyns et al., 2025; Fan & Huang, 2026). Existing gates judge one candidate against a retained parent (the incumbent) and return it or reject it: fresh paired tests (Ma & Wu, 2026), Two-Gate selection with a retention test (Cai et al., 2026), sequential tests with a per-candidate false-commit guarantee (Sengupta, 2026), gates assembled from quality checks, reversible safety artifacts and critic verdicts (Jiang et al., 2026; Mao et al., 2026), and one frozen pairwise validator (Liu et al., 2026). The Statistical Gödel Machine admits a recursive self-modification when an e-value certifies superiority, and names LLM agent loops as outside its setting (Wu et al., 2025). A whole batch here faces one shared parent row, the verdict that no candidate qualifies enters the decision, and the paid history of the loop reduces the reads, so one certificate decides the whole round.

Anytime-valid evaluation and control variates. Confidence sequences built by betting (Howard et al., 2021; Ramdas et al., 2023) give bounds valid at every stopping time, and they rest on the conditional law of a random permutation when the sample is drawn without replacement (Waudby-Smith & Ramdas, 2020; Serfling, 1974). ALPHA audits one assertion about one population mean with without-replacement betting martingales (Stark, 2023). The 2026 evaluators estimate one model's score on a fixed question set: CELEUS runs an e-process with surrogate predictions (Zhou et al., 2026b), FAQ runs finite-population active inference at a fixed budget (Wu et al., 2026), and AV-AIVAT subtracts a learned control variate inside a confidence sequence over paired game outcomes (Li et al., 2026). Each is instantiated for one stream under one rule for choosing the next cell. BranchCert-Σ runs these tests for K candidates and Q targets jointly on one shared order, and BranchCert-Σ+H builds its control variate from the paid history of the loop.

Racing and threshold evaluation. Racing and best-arm identification select high-performing candidates (Kalyanakrishnan et al., 2012; Karnin et al., 2013). Feasibility-constrained identification can also report that no feasible arm exists (Cai & Kandasamy, 2026). Shared-control comparisons go back to Dunnett (1955), and stochastic function evaluation studies which costly observations suffice to determine a function's value (Charikar et al., 2000; Deshpande et al., 2016; Ghuge et al., 2026). Our setting combines set-level decisions on a finite paired table with shared parent reads and paid-history reuse. Table 1 lists the nearest 2026 methods, and Appendix C covers subset evaluation, bandit identification and prediction-powered inference.

Table 1: The nearest 2026 neighbors. Reused information names the side information a method reuses; a dash marks a property outside the method's setting.

Method	Decision object	Sampling	Reused information
PACE (Shawn, 2026)	one candidate	i.i.d. tasks	—
DISCERN (Balachandran, 2026)	risk difference	i.i.d. inputs	—
Slice-wise (Zhang et al., 2026b)	one upgrade	i.i.d. inputs	—
CELEUS (Zhou et al., 2026b)	one model score	adaptive	surrogate
FAQ (Wu et al., 2026)	one model score	finite, active	factor model
AV-AIVAT (Li et al., 2026)	two agents	i.i.d. games	value model
Best-model ID (Tolochinsky et al., 2026)	best model	finite, bandit	low-rank fit
BranchCert-Σ+H (ours)	candidate set	shared order	paid history

3 SET-LEVEL CERTIFICATES ON A FINITE TABLE

A *round* fixes a parent program E_0, candidates $E_1, \dots, E_K$ written by a proposer, and Q targets. Target q owns N_q tasks and a requirement b_q on the mean paired score difference; a required target has $b_q > 0$, and a control target has $b_q \le 0$ and asks for non-inferiority. Running candidate k on task i of target q reveals a score $y_{kqi} \in \{0, 1\}$, and running the parent reveals z_{qi}. The complete score table x collects all these scores, and the decision reads each task through the difference of the two scores. The total paired difference, the total requirement it has to clear, and the qualifying set are

$$D_{kq} = \sum_{i=1}^{N_q} \big(y_{kqi} - z_{qi}\big), \qquad T_q = N_q b_q, \qquad \mathcal{S}(x) = \{k : \; D_{kq} \ge T_q \;\; \forall q\}. \tag{1}$$

A *cell* is one program on one task, one *rollout* is one complete execution of a program on a task, and the score table assigns the score of a rollout to its cell. Reading a cell means paying for its score, and every rule here is charged one unit per distinct cell it reads. The loop reads cells one at a time and chooses each read after seeing the earlier ones; a *history* h lists the cells read in the round and their scores, and a budget B caps the reads. The round returns one of three outcomes: the verdict ACCEPT of a named k when a candidate clears the requirement on every target, the verdict NONE BETTER when no candidate in the batch does, and UNDECIDED when the budget runs out first, in which case the parent stays. Write $\mathcal{X}(h)$ for the tables that agree with h on every read cell and keep every unread score in $[0, 1]$. An unread candidate score can add at most one to its own total and an unread parent score can subtract at most one from every candidate total, so with $O_{kq}(h)$ and $O_q(h)$ the read candidate and parent tasks,

$$L_{kq}(h) = R_{kq}(h) - \big(N_q - |O_q(h)|\big) \le D_{kq} \le R_{kq}(h) + \big(N_q - |O_{kq}(h)|\big) = U_{kq}(h), \tag{2}$$

where $R_{kq}(h) = \sum_{i \in O_{kq}(h)} y_{kqi} - \sum_{i \in O_q(h)} z_{qi}$. Reading the parent on one task moves the bound of every candidate at once, raising L_{kq} by $1 - z_{qi}$ and lowering U_{kq} by z_{qi} for all K candidates, so one parent row serves every comparison on its target.

Theorem 1 (Exact set-level certificates)**.** *For every history h and candidate k, $k \in \mathcal{S}(x')$ for all $x' \in \mathcal{X}(h)$ exactly when $L_{kq}(h) \ge T_q$ for every q, and $\mathcal{S}(x') = \varnothing$ for all $x' \in \mathcal{X}(h)$ exactly when every k has a target with $U_{kq}(h) < T_q$. Both equivalences hold at every step of any adaptive reading rule, and the complete table yields one of the two verdicts.*

The bounds of Equation (2) are attained by completions that set every unread candidate score to zero and every unread parent score to one, and the completions achieving the two ends differ freely across targets, which gives necessity (Appendix A). Validity rests on the observed history alone, so any schedule and any stopping rule keep it. Reading a cell twice adds nothing to either bound.

Proposition 1 (Distinct-cell cost)**.** *If a protocol draws task indices of a target with N tasks independently of the table and reads the candidate and the parent on each draw, every decision it computes is a function of the paired outcomes on the distinct tasks drawn, so g draws resolve at most $\min\{g, N\}$ comparisons at a cost of at most $2\min\{g, N\}$ distinct cells.*

We therefore count cost in distinct cells, the unit every later comparison uses. On the live rounds of Section 5.4, the fixed-batch paired test draws 256 required and 512 control pairs and spends 1,536

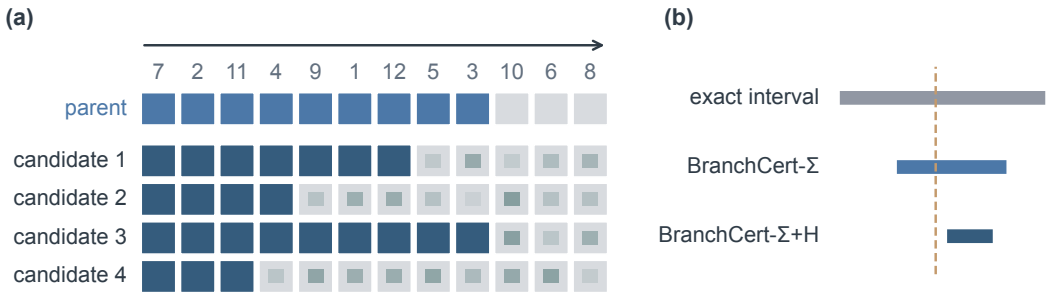

Figure 2: The certificate on one target. (a) All rows read the tasks in one shared random order; unread candidate cells carry a prediction from paid history. (b) Intervals of one candidate at equal reads; the dashed line is the requirement.

rollouts, which the proposition charges as 40 distinct cells covering 20 distinct comparisons. A set-level round is charged in the same unit. The parent row is paid once for all K candidates, so a round with K candidates on a target of N_q tasks holds $(K+1)N_q$ distinct cells, and every saving reported later is a share of that total.

4 BranchCert: certificates that stop early and reuse paid evaluations

4.1 Shared random order and lazy parent reads

This section builds the certificate a round runs. It returns the verdict of the round at a stopping time the loop chooses, charges one unit per distinct cell read, and rests on the exact interval of Section 3 and the guarantee of Theorem 2. The exact interval fills every unread cell with its worst value, so the exact certificate closes after most of a row has been read. BranchCert-Σ draws, for each target q, an order π_q uniform over the $N_q!$ permutations of its tasks, independently across targets and before any cell of the round is read, and reads every row of that target along it. Candidate k reads a prefix of π_q; the parent cell of a task is read the first time any candidate reaches the task, and once a candidate has read half of a target the rest of the parent row is read, which tightens the exact intervals of all K candidates (Figure 2). The cost of a round is the number of distinct cells it reads, the unit Proposition 1 fixes. Given its prefix, the unread tasks of a row are in uniformly random order among the remaining tasks of the target, which is the sampling law of a finite population drawn without replacement (Serfling, 1974; Waudby-Smith & Ramdas, 2020). We call the exact certificate of Theorem 1, run with the acquisition planner of Appendix B.2, BranchCert-E.

4.2 Residual betting on the paid history

Earlier rounds of a loop have already paid for cells on the same tasks, the *paid history* of the loop, and BranchCert-Σ leaves them unused. Before the orders of a round are drawn, BranchCert-Σ+H fixes for every candidate, target and task a prediction p_i of the paired difference $d_i = y_{kqi} - z_{qi}$ and a range $[a_i, c_i]$ that contains it. Both come from the paid history. Let n_i be the number of earlier programs read on task i, s_i the sum of their scores, and $\bar{s}$ the mean score over all paid cells of the target. When the current parent was read on task i in an earlier round,

$$p_i = \frac{s_i + \kappa \bar{s}}{n_i + \kappa} - z_{qi}, \qquad [a_i, c_i] = [-z_{qi},\, 1 - z_{qi}], \tag{3}$$

the difficulty of the task smoothed toward $\bar{s}$ with a pseudo-count $\kappa > 0$, minus the known parent score; a task without a paid parent cell keeps $p_i = 0$ and $[a_i, c_i] = [-1, 1]$. Setting every $p_i = 0$ and every range to $[-1, 1]$ recovers BranchCert-Σ. The evaluator caches scores by program and task, so a paid parent score equals the score a new read returns.

After t reads the row has observed $S_t = \sum_{j \le t} d_{\pi(j)}$ and $P_t = \sum_{j \le t} p_{\pi(j)}$, and $P = \sum_i p_i$ is known. Under the hypothesis $D_{kq} = m$, the next task is uniform over the unread ones, so the residual

$e_t = d_{\pi(t)} - p_{\pi(t)}$ has the known conditional mean $\nu_t(m) = \big((m - S_{t-1}) - (P - P_{t-1})\big)/(N_q - t + 1)$. Two mixtures of betting processes test m from each side,

$$M_t^{\pm}(m) = \sum_{\vartheta \in \Theta} w_\vartheta \prod_{j \le t} \big(1 \pm \lambda_j^{\pm}(m)\,(e_j - \nu_j(m))\big), \tag{4}$$

with fixed weights $w_\vartheta \ge 0$ summing to one over a finite grid Θ, and predictable fractions $\lambda_j^{\pm}(m)$ that scale the grid point ϑ by the range of the unread residuals, so every factor stays nonnegative. M^+ grows when the true total exceeds m and M^- when it falls short. The lower set $\mathcal{C}_t^+$ keeps the totals m with $M_s^+(m) < 2K/\alpha$ for all $s \le t$, the upper set $\mathcal{C}_t^-$ keeps those with $M_s^-(m) < 2Q/\alpha$, and each set is intersected with the exact interval $[L_{kq}, U_{kq}]$, with fallback to that interval when the intersection is empty. The round returns ACCEPT for k when every lower set of k lies at or above T_q, excludes k when some upper set of k lies below T_q, and returns NONE BETTER once every candidate is excluded. The residual e_t keeps the mean identity exact for every prediction. An accurate prediction can shrink the residual spread, allowing wealth at a false total to grow faster and the sets to close earlier, and the saving varies from run to run.

Theorem 2 (Finite-population certificates with paid history)**.** *Fix a round with a binary table. Fix predictions, ranges containing every paired difference, and mixture weights before the orders are drawn, and draw each order uniformly given the table and this side information. Then (i) for each candidate k and target q, $M_t^{\pm}(D_{kq})$ are nonnegative martingales starting at one in the row's own filtration, so D_{kq} leaves the lower sets at some time with probability at most $\alpha/(2K)$ and the upper sets with probability at most $\alpha/(2Q)$; (ii) under any schedule across rows and any stopping rule on the joint history, a wrong* ACCEPT *and a wrong* NONE BETTER *each have probability at most $\alpha/2$; (iii) on the same cells the certificate decides whenever the exact certificate of Theorem 1 decides, and a full read returns the exact totals; (iv) when rounds $\rho = 1, 2, \ldots$ of a chain each fix their side information from the paid history of all earlier rounds, rounds at levels α_ρ with $\sum_\rho \alpha_\rho \le \alpha$ err in any round with probability at most α.*

The proof (Appendix A.4) needs three classical facts. The conditional law of a random permutation read as a prefix makes $\nu_t(D_{kq})$ the exact conditional mean of e_t, so each factor has conditional mean one and the products are martingales; Ville's inequality bounds the chance that a nonnegative martingale starting at one ever crosses $2K/\alpha$ by $\alpha/(2K)$ (Howard et al., 2021). The main difficulty is part (ii): a global schedule stops each row at a random time that depends on every other row, and this time lies outside the stopping times of the row's own filtration in general. The path lemma handles it.

Lemma 1 (Path lemma)**.** *Let $(\mathcal{C}_t)_{t \le N}$ be random sets and ξ a fixed value with $\Pr[\exists t \le N : \xi \notin \mathcal{C}_t] \le \delta$. Then $\Pr[\xi \notin \mathcal{C}_\tau] \le \delta$ for every random index $\tau \in \{0, \ldots, N\}$, from any filtration.*

Appendix A.5 proves the lemma by a pathwise event inclusion, so the bound holds for every random index τ. A wrong ACCEPT then needs the lower set of a failing pair (k, q^*) to exclude D_{kq^*}, and a wrong NONE BETTER needs an upper set of a fixed qualifying candidate to exclude its total, so union bounds over K candidates and Q targets give $\alpha/2$ each. Part (iii) follows from $\mathcal{C}_t^{\pm} \subseteq [L_{kq}, U_{kq}]$, and part (iv) from conditioning on the paid history. A *chain* is a sequence of rounds in which each accepted candidate becomes the next parent, and ρ is the position of a round in its chain, counting from one. A chain assigns its rounds levels α_ρ that decrease with ρ and sum to at most α, so one level α covers the whole chain.

4.3 CONDITIONS THAT THE PAID HISTORY MUST MEET

Theorem 2 asks five things of how the paid history enters a round, all met here (Appendix E.3). Predictions, ranges and mixture weights are computed from earlier rounds before the orders are drawn, and each order is drawn uniformly given the table and this side information. Every range contains the true paired difference, since the evaluator caches by program and task, a recorded parent score equals the current one, and a task without a paid parent value keeps the range $[-1, 1]$. Bets and residual ranges use the frozen predictions and the row's own prefix, so parent cells read during the round enter the exact interval and leave the bets unchanged. Scores are binary, so the integer grid of hypothesized totals contains the true total; the total requirement T_q may be fractional. The levels of the rounds of a chain sum to at most α.

5 EXPERIMENTS

We ask three questions. How much of a table does a set-level certificate read against current anytime-valid evaluators, and at what error (Section 5.1)? Does the paid history of a running loop make its later rounds cheaper (Section 5.2)? Which parts of the mechanism carry the saving (Section 5.3)? Every rule reads the same complete tables under one set-level wrapper, is charged one unit per distinct cell, and spends one level per round. A candidate is accepted when its rule certifies every target, the round returns NONE BETTER when every candidate is excluded, and a rule that exhausts its budget returns UNDECIDED. A verdict is counted wrong when it disagrees with the complete table (Appendix D.6), and the conditions of Section 4.3 pass every behavior check (Appendix E.3).

5.1 PUBLIC TABLES: HELM LITE

HELM Lite (Liang et al., 2023) publishes per-instance scores of 95 models on six benchmark families. A round takes one family as the required target and another as the control target, a parent at one of four ability quantiles, and ten candidate models (Appendix D.3). The comparison crosses six fixed (required, control) family pairs with the four parent quantiles, with nearest-ability candidates, over sixteen reading orders each. The 2026 evaluators (Zhou et al., 2026b; Hsu & Shekhar, 2026; Wu et al., 2026) receive the same low-rank factor model here, and each keeps its native querying rule and its own interval inside the same wrapper.

Cost against error. Table 4 of Appendix E.1 draws one boundary. The two BranchCert certificates control their error, decide every run and read at most 38.2% of a table on the average round. The other rules that control their error and decide every run, the without-replacement sequences and CELEUS, read from 42.9% to 61.9%. Every compared rule outside the ablation block that reads less errs above α or leaves most runs undecided, so a loop that commits to each verdict forfeits those savings.

Wrapper against sequence. The set-level wrapper carries most of the saving. Inside the eager-parent wrapper of Appendix D.6 every without-replacement sequence, BranchCert-Σ included, reads close to the exact certificate. Moving BranchCert-Σ onto its own wrapper lowers its share read to 38.2%. One parent row serves all candidates, and each verdict spends its level on one side of the comparison.

Level sweep. From $\alpha = 0.01$ to 0.2, both certificates decide every run and keep their wrong-verdict rates at or below $\alpha/2$, as Theorem 2 predicts. BEACON-style stopping and successive halving exceed $\alpha/2$ at every level (Figure 6, Appendix D.6). Slice-wise certification decides 22% of runs with no observed wrong verdict, and its fallback to the parent counts as UNDECIDED. On these rounds the paid history is unusual: the earlier rounds scored other models, so a predictor built from them describes how hard the tasks are and says little about the models being compared. BranchCert-Σ+H reads almost exactly what BranchCert-Σ reads, and we report BranchCert-Σ as the HELM method.

5.2 SELF-IMPROVEMENT CHAINS

Qwen3.5-9B (Qwen Team, 2026) rewrites its own system prompt. Each round it proposes eight candidates from the current parent, and every program is scored on required MATH tasks and control MMLU tasks at fixed requirements b_q (Appendix D.4). An accepted candidate becomes the next parent, and NONE BETTER sends the loop back to propose. We record the complete table of every round, over 6 chains of 48 rounds, and replay each certificate on it under 8 reading orders, every method starting from an empty paid history and charged for each cell on first read.

Figure 3 shows the per-round cost of the three certificates. The exact certificate reads the most in all eight rounds, BranchCert-Σ reads a little over half of that, and BranchCert-Σ+H reads less than BranchCert-Σ from the second round onward. The saving is exactly zero in the first round, where no paid history exists and the two certificates coincide, and its mean stays positive once the loop accumulates paid history. The per-chain lines show the spread across chains, and 41 of the 48 rounds end in NONE BETTER, an outcome outside the answer set of two-model gates.

Reuse of paid history. BranchCert-Σ+H reads a mean 19.4% of each table, less than BranchCert-Σ and the exact certificate in every chain, with no observed wrong verdict (Table 3), and its mean saving

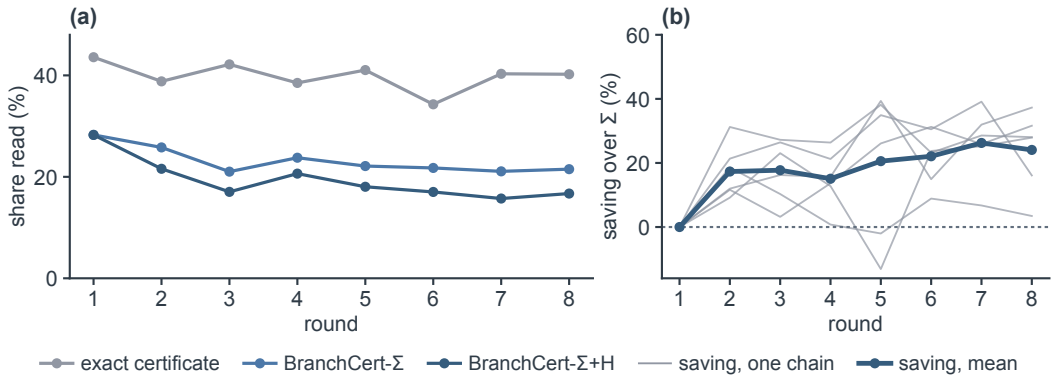

Figure 3: Recorded Qwen3.5-9B chains. (a) Share read of each 27,000-cell table per round by the exact certificate and the two BranchCert certificates, averaged over the six chains. (b) Saving of BranchCert-Σ+H over BranchCert-Σ per round.

over BranchCert-Σ grows from zero in the first round to 22.0% over rounds five to eight. Each round adds its cells to the paid history that predicts the next, and the saving persists when both methods start from the cells BranchCert-Σ read earlier, with BranchCert-Σ+H reading 19.1%. Per run the saving ranges from -99.7% to 63.7%, and 42 of the 336 runs with paid history read more than BranchCert-Σ.

Why the saving varies. The saving falls as the share of paired-difference variance the paid history leaves unexplained rises, with Spearman correlation -0.64 over the 42 rounds with paid history: a residual with less variance lets the wealth at a false total grow faster, so the sets close after fewer reads. Chain 6 has the least predictable paid history, and it carries the smallest saving.

Agreement with the complete table. No run of either certificate disagrees with the complete table, so early stopping left every observed verdict unchanged. The exact interval, on the same cells, decides 1% of chain runs, so the confidence sequences decide the verdict.

5.3 Removals of the mechanism

Finite-population correction. Among the changes to the betting process in Table 5 (Appendix E.2), the conditional mean $\nu_t(m)$ carries the largest effect: betting against the with-replacement mean of the unread residuals raises the share read of BranchCert-Σ+H from 19.4% to 27.0% on the chains and from 37.4% to 41.5% on HELM. The correction is also the step of the proof this removal breaks, since the with-replacement mean ignores that the hypothesis and the prefix together fix the total of the unread cells.

The predictor and the shared row. With zero prediction and the history ranges kept, the chains read the share of BranchCert-Σ, so the saving enters entirely through the residual bet. Charging each parent cell once per candidate, the accounting a per-candidate gate would use, raises the share read of BranchCert-Σ+H to 87.6% on HELM and to 27.0% on the chains.

The conditions of the theorem. A fixed reading order breaks a condition of Theorem 2, and running at level 10α relaxes the level. The two variants produce wrong verdicts in 20.8% and 2.3% of decided runs, and BranchCert-Σ errs in 0.5%.

5.4 Live rounds and out-of-sample gains

The chains are replayed from recorded tables, and Granite-4.2-8B (Granite Team, IBM, 2026) supplies a live test. It proposes candidates for a tool-use program, and each round scores the parent and up to twelve candidates on required and control tasks of ToolFailBench (Soni, 2026) held out from candidate construction. On the twelve-candidate round, Granite R3, BranchCert-Σ certifies None Better at 106 cells against 174 of 260 for the parent-first order, and every verdict of BranchCert-Σ

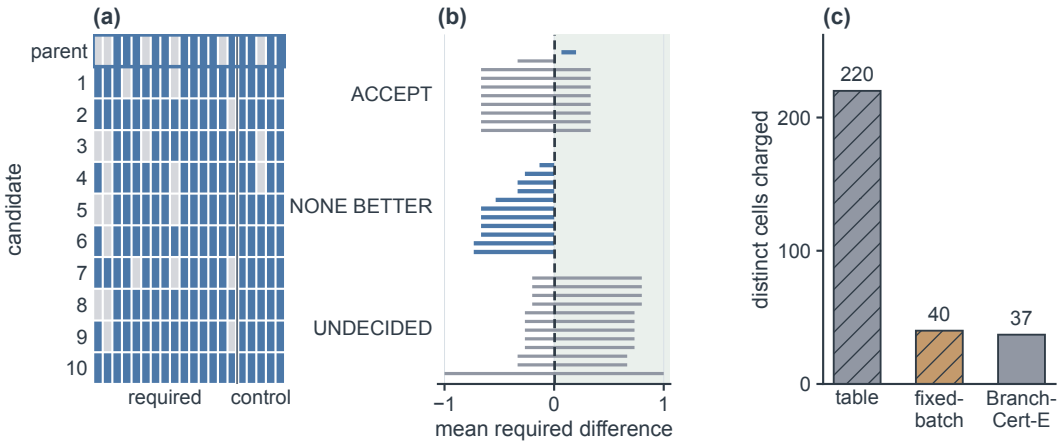

Figure 4: Live evidence. (a) The matched Qwen3.5 table, one shared parent row above ten candidates. (b) Candidate intervals at the stopping time on the Granite rounds, grouped by outcome; the shaded side of the dashed line qualifies. (c) Distinct cells each method is charged.

on the other six tables agrees with the complete table (Table 7). The fixed-batch paired test covers 40 distinct cells per round and returns no per-cell decision.

All three outcomes on live tables. Figure 4 shows every outcome a round can return on tables a running loop produced. On the matched Qwen3.5 table, a shared parent row above ten candidates over 15 required and five control tasks, BranchCert-E accepts a named candidate after 37 cells, the parent row and part of one candidate row. Granite R4, with ten candidates, ends in ACCEPT with the certified interval above the requirement. Granite R3 ends in NONE BETTER once every candidate has a target whose upper bound falls short, and the candidate-first order on that same round spends its budget and returns UNDECIDED. The fixed-batch paired test covers one candidate and reaches 40 cells on the Granite rounds with no decision once each cell is charged once, and on the matched Qwen3.5 table the set-level certificate settles all ten candidates with 37 cells, fewer than that single comparison.

A live paired re-execution. Replay reuses recorded scores, so we also run both certificates on fresh executions. Both re-execute the second round of chain 1, a 27,000-cell table of one parent and eight candidate system prompts, generating every response again under the frozen decoding settings of the round. BranchCert-Σ re-executes 11,136 cells and reaches its verdict after reading 9,595 of them, and BranchCert-Σ+H re-executes 8,860 and reaches its verdict after 7,856; the other re-executed cells were prefetched and left unread. The fresh scores differ from the recorded ones on 740 and 501 of the re-executed cells, disagreement rates of 6.6% and 5.7% from nondeterministic decoding at temperature zero. Both certificates keep their verdicts under the fresh scores: each returns ACCEPT of the same candidate as the complete recorded table, and the accepted candidate clears the requirement under the fresh scores as well.

Gains that outlive the round. A certificate that admits a candidate has to leave the loop better off. We measure gains on fresh tasks, since the round selected the candidate on its own tasks. For each of the six chains we compare the prompt the loop started from against the prompt it held after round eight, on 500 fresh MATH tasks and 500 fresh MMLU tasks drawn outside every partition of the loop, the 300-task held-out split of Table 3 included, in the same paired arms and under the same scoring function. The final prompt gains 4.9 points of MATH accuracy on average over the six chains, with 5 of the 6 chains gaining and the largest gain, 13.8 points, on chain 6. MMLU moves by 0.4 points. On the held-out split the mean MATH accuracy moves from 80.6% to 83.9%. The gains sit on the targeted tasks and the control tasks hold their accuracy, consistent with the admitted candidates generalizing to fresh tasks.

6 CONCLUSION

A round of a self-improvement loop decides a whole batch of candidates against one shared parent on a finite table, and both of its verdicts have exact conditions in the bounds of the cells the round has not read. One shared random order turns those bounds into confidence sequences that hold at every stopping time, and betting on the residuals of the paid history lowers the mean cost of later rounds on the recorded chains, most where the paid history describes the tasks well. The candidates the certificates admit come with gains that outlast the round that admitted them, 4.9 points of MATH accuracy on fresh tasks with control accuracy within 0.4 points. Two limits bound what we can claim. The saving depends on how much of the paired differences of a round the paid history predicts. The evidence comes from loops that improve one program against a fixed task set, so applying a certificate to a round drawn from a stream of different tasks remains open (Ma & Wu, 2026). The guarantees cover verdicts on finite tables and, under the stated history conditions, across rounds; performance on fresh tasks rests on the held-out measurements.

AI USE STATEMENT

Generative AI and agent orchestration assisted research direction, hypothesis and experiment design, implementation, result interpretation, figure construction, manuscript drafting, editing, and mathematical proofs. The authors checked the code, executable identities, numerical claims, citations, and visual encodings against the submitted artifacts and take responsibility for the complete submission.

ETHICS STATEMENT

The experiments score language models on public benchmarks and on tool-use tasks in simulated environments, with no human subjects and no personal data. A certificate that admits candidates faster can speed up self-improving systems, and the control target and the level α give the loop a stated check against regressions.

REPRODUCIBILITY STATEMENT

Section 5 states the rounds, task sets, budgets, and comparators. Appendix A proves the main-text results, Appendix B gives the rollout and planner guarantees with their complexity, and Appendix D records task construction, decoding settings, sampling rules, and cost accounting. Every table and figure is generated from compact machine-readable result records by scripts released with the code; code and result records are released with the camera-ready version.

REFERENCES

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, pp. 8479–8565, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0e9e708b6f48e14fd0ac29e167413f76-Abstract-Conference.html.

Anastasios N. Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I. Jordan, and Tijana Zrnic. Prediction-Powered Inference. *Science*, 382(6671):669–674, 2023. doi: 10.1126/science.adi6000. URL https://doi.org/10.1126/science.adi6000.

Ruicheng Ao, Hongyu Chen, Siyang Gao, Hanwei Li, and David Simchi-Levi. Best Arm Identification with LLM Judges and Limited Human. *arXiv preprint arXiv:2601.21471*, 2026. URL https://arxiv.org/abs/2601.21471.

Kianoosh Ashouritaklimi, Valentin Kilian, Daolang Huang, Tom Rainforth, and François Caron. Prediction-Powered Active Testing. *arXiv preprint arXiv:2607.08347*, 2026. URL https://arxiv.org/abs/2607.08347.

Vishnu Bindu Balachandran. Pay only for disagreement: Certified no-regression verdicts for model updates with matching label-complexity bounds. *arXiv preprint arXiv:2609.17560*, 2026. URL https://arxiv.org/abs/2609.17560.

Gabrielle Berrada, Jannik Kossen, Freddie Bickford Smith, Muhammed Razzak, Yarin Gal, and Thomas Rainforth. Scaling Up Active Testing to Large Language Models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 173670–173697, 2025. doi: 10.52202/085713-5777. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/fdb11be1acf5e3724737dd585e590146-Abstract-Conference.html.

Qianshu Cai, Yonggang Zhang, Jun Nie, Maohao Ran, Huajiang Zheng, Jun Song, Xinmei Tian, Yike Guo, and Wei Xue. Safe harness self-evolution: A theoretical analysis of feasibility and limits. *arXiv preprint arXiv:2609.08175*, 2026. URL https://arxiv.org/abs/2609.08175.

Ting Cai and Kirthevasan Kandasamy. Constrained best arm identification with tests for feasibility. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(24):19808–19816, 2026. doi: 10.1609/aaai.v40i24.39063. URL https://ojs.aaai.org/index.php/AAAI/article/view/39063.

Moses Charikar, Ronald Fagin, Venkatesan Guruswami, Jon Kleinberg, Prabhakar Raghavan, and Amit Sahai. Query strategies for priced information (extended abstract). In *ACM Symposium on Theory of Computing*, pp. 582–591, 2000. doi: 10.1145/335305.335382. URL https://doi.org/10.1145/335305.335382.

Yizhe Chi, Wenyi Li, Deyao Hong, Xiaoqiu Wang, Mingju Gao, Kaisen Yang, Bingxiang He, Youjie Zheng, Calvin Xiao, and Qinhuai Na. AI4AI-Bench: Benchmarking LLM agents in algorithmic design for recursive self-improvement. *arXiv preprint arXiv:2608.20318*, 2026. URL https://arxiv.org/abs/2608.20318.

Eugenio Clerico, Tobias Wegel, Iskander Azangulov, and Patrick Rebeschini. Time-Sensitive Anytime-Valid Testing. *arXiv preprint arXiv:2605.06521*, 2026. URL https://arxiv.org/abs/2605.06521.

Charlie Cowen-Breen, Alekh Agarwal, Stephen Bates, William W. Cohen, Jacob Eisenstein, Amir Globerson, and Adam Fisch. Multiple-Prediction-Powered Inference. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2603.27414.

Devleena Das, Rajeev Patwari, Vikram Kumar Bukka, Nithin Kumar Guggilla, Elliott Delaye, and Ashish Sirasao. MINCE: Shrinking LLM Evaluation Datasets via Few-Model Monte Carlo Calibration. *arXiv preprint arXiv:2606.22826*, 2026. URL https://arxiv.org/abs/2606.22826.

Rianne de Heide. Where Does the Union Bound Go? Best-Arm Identification and Strong FWER Control. *arXiv preprint arXiv:2608.19903*, 2026. URL https://arxiv.org/abs/2608.19903.

Amol Deshpande, Lisa Hellerstein, and Devorah Kletenik. Approximation algorithms for stochastic submodular set cover with applications to boolean function evaluation and min-knapsack. *ACM Transactions on Algorithms*, 12(3):1–28, 2016. doi: 10.1145/2876506.

Abhishek Divekar. Statistically Reliable LLM-Based Ranking Evaluation via Prediction-Powered Inference. *arXiv preprint arXiv:2606.05308*, 2026. URL https://arxiv.org/abs/2606.05308.

Yi Duan, Ying Liu, Zirui Tang, et al. The Last AI Built by Humans: Toward Genuine Recursive Self-Improvement. *arXiv preprint arXiv:2609.11873*, 2026. URL https://arxiv.org/abs/2609.11873.

Charles W. Dunnett. A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association*, 50(272):1096–1121, 1955. doi: 10.1080/01621459.1955.10501294.

Tianyu Fan and Chao Huang. HELIX: Model–Harness Co-evolution for Recursive Self-Improvement. *arXiv preprint arXiv:2608.13951*, 2026. URL https://arxiv.org/abs/2608.13951.

Chen Feng, Minghe Shen, Ananth Balashankar, Carsten Gerner-Beuerle, and Miguel R. D. Rodrigues. Noisy but Valid: Robust Statistical Evaluation of LLMs with Imperfect Judges. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2601.20913.

Aurélien Garivier and Emilie Kaufmann. Optimal Best Arm Identification with Fixed Confidence. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pp. 998–1027. PMLR, 2016. URL https://proceedings.mlr.press/v49/garivier16a.html.

Rohan Ghuge, Anupam Gupta, and Viswanath Nagarajan. Nonadaptive stochastic score classification and explainable half-space evaluation. *Operations Research*, 73(4):2204–2222, 2025. doi: 10.1287/opre.2023.0431. URL https://doi.org/10.1287/opre.2023.0431.

Rohan Ghuge, Jai Moondra, and Mohit Singh. Stochastic function certification with correlations. *arXiv preprint arXiv:2604.02611*, 2026. URL https://arxiv.org/abs/2604.02611.

Granite Team, IBM. Granite-4.2-8B. Model card, 2026. URL https://huggingface.co/ibm-granite/granite-4.2-8b.

Eliya Habba, Itay Itzhak, Asaf Yehudai, Yotam Perlitz, Elron Bandel, Michal Shmueli-Scheuer, Leshem Choshen, and Gabriel Stanovsky. Growing Pains: Extensible and Efficient LLM Benchmarking Via Fixed Parameter Calibration. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2604.12843.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021. doi: 10.1214/20-AOS1991. URL https://doi.org/10.1214/20-AOS1991.

Chia-Yu Hsu and Shubhanshu Shekhar. Efficient sequential evaluation of large language models. *arXiv preprint arXiv:2607.17409*, 2026. URL https://arxiv.org/abs/2607.17409.

Zhongzhan Huang, Junxin Li, Guoming Ling, Yupei Lin, Shanshan Zhong, and Hefeng Wu. Zipbench: Low-Cost Framework for Compressing Comprehensive Benchmarks of Large Language Models. *arXiv preprint arXiv:2609.12475*, 2026. URL https://arxiv.org/abs/2609.12475.

Alex Iacob, Andrej Jovanović, William F. Shen, Daniel Burkhardt, Meghdad Kurmanji, Nurbek Tastan, Lorenzo Sani, Niccolò Alberto Elia Venanzi, Ambroise Odonnat, Zeyu Cao, Bill Marino, Xinchi Qiu, and Nicholas D. Lane. The Red Queen Gödel Machine: Co-Evolving Agents and Their Evaluators. *arXiv preprint arXiv:2606.26294*, 2026. URL https://arxiv.org/abs/2606.26294.

Wen Jiang, Mingmin Chu, Yimeng Tian, Qianxin Zhang, Haofei Yang, Rui Yang, Yang Liu, Tao Lv, and Fangming Li. HarnessEvolve: Learning from reference trajectories for reliable agent self-evolution. *arXiv preprint arXiv:2609.00829*, 2026. URL https://arxiv.org/abs/2609.00829.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. PAC subset selection in stochastic multi-armed bandits. In *International Conference on Machine Learning*, 2012. URL https://icml.cc/2012/papers/359.pdf.

Hideaki Kano, Junya Honda, Kentaro Sakamaki, Kentaro Matsuura, Atsuyoshi Nakamura, and Masashi Sugiyama. Good Arm Identification via Bandit Feedback. *Machine Learning*, 108 (5):721–745, 2019. doi: 10.1007/s10994-019-05784-4. URL https://doi.org/10.1007/s10994-019-05784-4.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1238–1246. PMLR, 2013. URL https://proceedings.mlr.press/v28/karnin13.html.

Valentin Kilian, Stefano Cortinovis, and François Caron. Asymptotically Log-Optimal Bayes-Assisted Confidence Sequences for Bounded Means. *arXiv preprint arXiv:2605.07964*, 2026. URL https://arxiv.org/abs/2605.07964.

Michael Krumdick, Adam Wiemerslage, Seth Ebner, Charles Lovering, and Chris Tanner. Cost-Efficient Estimation of General Abilities Across Benchmarks. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2604.01418.

Boning Li, Yu Chen, and Longbo Huang. AV-AIVAT: 74x Cheaper Agent Evaluation with Certified Anytime-Valid Stopping in Imperfect-Information Games. *arXiv preprint arXiv:2608.06362*, 2026. URL https://arxiv.org/abs/2608.06362.

Yitao Li. Who Drifted: the System or the Judge? Anytime-Valid Attribution in LLM Evaluation Pipelines. *arXiv preprint arXiv:2606.15474*, 2026. URL https://arxiv.org/abs/2606.15474.

Zitian Li and Wang Chi Cheung. Closing the Gap on the Sample Complexity of 1-Identification. *arXiv preprint arXiv:2601.15620*, 2026. URL https://arxiv.org/abs/2601.15620.

Percy Liang, Rishi Bommasani, Tony Lee, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=iO4LZibEqW.

Minghao Liu, Yu Wang, Jiayun Wang, and Wei Wei. Reward-free evolving agents via pairwise validator. *arXiv preprint arXiv:2607.14408*, 2026. URL https://arxiv.org/abs/2607.14408.

Qinzhen Ma and Ruihai Wu. When validation stops learning: Auditing update admission for continual embodied agents. *arXiv preprint arXiv:2609.10873*, 2026. URL https://arxiv.org/abs/2609.10873.

Zaid Mahboob, Yujia Chen, and Bowen Weng. Betting for Sim-to-Real Performance Evaluation. *arXiv preprint arXiv:2604.24018*, 2026. URL https://arxiv.org/abs/2604.24018.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with Fewer Examples. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326. PMLR, 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html.

Alexandre Cristovão Maiorano. Automated Self-Testing as a Quality Gate: Evidence-Driven Release Management for LLM Applications. *arXiv preprint arXiv:2603.15676*, 2026. URL https://arxiv.org/abs/2603.15676.

Qinghua Mao, Wanying Qu, Dadi Guo, Leitao Yuan, Qingyu Liu, Yu Li, Guanxu Chen, Yanwei Fu, Xi Lin, Xia Hu, and Dongrui Liu. SafeEvolve: Harness-policy co-evolution from agent experience for safety alignment. *arXiv preprint arXiv:2609.02786*, 2026. URL https://arxiv.org/abs/2609.02786.

Diego Martinez-Taboada and Aaditya Ramdas. Gaussian-Efficient Testing by Betting on the Mean of Bounded Data. *arXiv preprint arXiv:2608.21694*, 2026. URL https://arxiv.org/abs/2608.21694.

Michael Nguyen, Quoc Nguyen, and Paul Vuong. Recursive Self-Evolving Agents via Held-Out Selection. *arXiv preprint arXiv:2606.28374*, 2026. URL https://arxiv.org/abs/2606.28374.

Sangwoo Park, Matteo Zecchin, and Osvaldo Simeone. Adaptive Prediction-Powered AutoEval with Reliability and Efficiency Guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.18659.

Toby D. Pilditch. Knowing When to Stop: Bayesian Optimal Stopping for LLM Evaluations. *arXiv preprint arXiv:2608.14425*, 2026. URL https://arxiv.org/abs/2608.14425.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026. URL https://qwen.ai/blog?id=qwen3.5.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-Theoretic Statistics and Safe Anytime-Valid Inference. *Statistical Science*, 38(4):576–601, 2023. doi: 10.1214/23-STS894. URL https://doi.org/10.1214/23-STS894.

Anton Razzhigaev, Andrei Gritsaev, Andrei Kaznacheev, Nikita Dragunov, Roman Yampolskiy, and Andrei Kuznetsov. Ouroboros: A Self-Developing Frontier Coding Agent with Reviewed Core Evolution. *arXiv preprint arXiv:2608.08311*, 2026. URL https://arxiv.org/abs/2608.08311.

Maxime Robeyns, Martin Szummer, and Laurence Aitchison. A self-improving coding agent. In *ICLR 2025 Workshop on Scaling Self-Improving Foundation Models*, 2025. URL https://openreview.net/forum?id=rShJCyLsOr.

Ricardo J. Sandoval, Ian Waudby-Smith, and Michael I. Jordan. Multi-Armed Sequential Hypothesis Testing by Betting. *arXiv preprint arXiv:2603.17925*, 2026. URL https://arxiv.org/abs/2603.17925.

Biswa Sengupta. Self-Evolving Agents with Anytime-Valid Certificates. *arXiv preprint arXiv:2607.00871*, 2026. URL https://arxiv.org/abs/2607.00871.

R. J. Serfling. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, 2(1):39–48, 1974. doi: 10.1214/aos/1176342611.

Zayx Shawn. PACE: Anytime-valid acceptance tests for self-evolving agents. *arXiv preprint arXiv:2606.08106*, 2026. URL https://arxiv.org/abs/2606.08106.

Shubhanshu Shekhar, Ziyu Xu, Zachary C. Lipton, Pierre J. Liang, and Aaditya Ramdas. Risk-limiting Financial Audits via Weighted Sampling without Replacement. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 1932–1941. PMLR, 2023. URL https://proceedings.mlr.press/v216/shekhar23a.html.

Harsh Soni. ToolFailBench: Diagnosing tool-use failures in LLM agents. In *Second Workshop on Agents in the Wild: Safety, Security, and Beyond*, 2026. URL https://arxiv.org/abs/2607.04686.

Philip B. Stark. ALPHA: Audit that learns from previously hand-audited ballots. *The Annals of Applied Statistics*, 17(1):641–679, 2023. doi: 10.1214/22-AOAS1646. URL https://doi.org/10.1214/22-AOAS1646.

Ege Onur Taga, Samet Oymak, and Shubhanshu Shekhar. Learning to Bet for Horizon-Aware Anytime-Valid Testing. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2603.19551.

Yaniv Tenzer, Elad Tolochinsky, and Yaniv Romano. Semi-Supervised Hypothesis Testing by Betting on Predictions. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2605.28533.

Elad Tolochinsky, Yaniv Tenzer, and Yaniv Romano. Valid Best-Model Identification for LLM Evaluation via Low-Rank Factorization. *arXiv preprint arXiv:2605.10405*, 2026. URL https://arxiv.org/abs/2605.10405.

Vladimir Vovk and Ruodu Wang. E-Values: Calibration, Combination and Applications. *The Annals of Statistics*, 49(3), 2021. doi: 10.1214/20-AOS2020. URL https://doi.org/10.1214/20-AOS2020.

Guangya Wan, Zixin Stephen Xu, Sasa Zorc, Manel Baucells, Mengxuan Hu, Hao Wang, and Sheng Li. BEACON: Bayesian optimal stopping for efficient LLM sampling. *arXiv preprint arXiv:2510.15945*, 2025. URL https://arxiv.org/abs/2510.15945.

Ian Waudby-Smith and Aaditya Ramdas. Confidence sequences for sampling without replacement. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e96c7de8f6390b1e6c71556e4e0a4959-Abstract.html.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(1):1–27, 2024. doi: 10.1093/jrsssb/qkad009. URL https://doi.org/10.1093/jrsssb/qkad009.

Skyler Wu, Yash Nair, and Emmanuel J. Candès. Efficient evaluation of LLM performance with statistical guarantees. *arXiv preprint arXiv:2601.20251*, 2026. URL https://arxiv.org/abs/2601.20251.

Xuening Wu, Shenqin Yin, Yanlan Kang, Xinhang Zhang, Qianya Xu, Zeping Chen, and Wenqiang Zhang. SGM: A statistical Gödel machine for risk-controlled recursive self-modification. *arXiv preprint arXiv:2510.10232*, 2025. URL https://arxiv.org/abs/2510.10232.

Ruiqing Yue, Yu Cui, Zhuoyu Sun, Sicheng Pan, Xianhong Xue, Tingyu Li, Ting Li, Wenzhuo Zhu, Yi Chen, Yifei Liu, Baohan Huang, Zhe Cui, Haibin Zhang, and Cong Zuo. Ecdysis: Efficient and effective training of runtime harnesses for llm agents. *arXiv preprint arXiv:2609.11677*, 2026. URL https://arxiv.org/abs/2609.11677.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-taught optimizer (STOP): Recursively self-improving code generation. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2310.02304.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Gödel Machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/aa5f5e6eb6f613ec412f1d948dfa21a5-Abstract-Conference.html.

Shengwei Zhang, Tao Wu, and Fei Qian. Certifying model upgrades with slice-wise non-regression and incumbent fallback. *arXiv preprint arXiv:2609.13714*, 2026b. URL https://arxiv.org/abs/2609.13714.

Congjie Zheng, Chuanyi Xue, Bin Liang, Jun Yang, and Changshui Zhang. SEAGym: An Evaluation Environment for Self-Evolving LLM Agents. *arXiv preprint arXiv:2606.17546*, 2026. URL https://arxiv.org/abs/2606.17546.

Siyu Zhou, Patrick Vossler, Venkatesh Sivaraman, Yifan Mai, and Jean Feng. Adaptive Auditing of AI Systems with Anytime-Valid Guarantees. *arXiv preprint arXiv:2605.07002*, 2026a. URL https://arxiv.org/abs/2605.07002.

Zhijian Zhou, Zesheng Ye, Zhaorun Chen, Bo Li, and Feng Liu. CELEUS: Certifiable and Efficient LLM Evaluation via E-Processes. *arXiv preprint arXiv:2606.20820*, 2026b. URL https://arxiv.org/abs/2606.20820.

Tijana Zrnic and Emmanuel J. Candès. Active Statistical Inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62993–63010. PMLR, 2024. URL https://proceedings.mlr.press/v235/zrnic24a.html.

A PROOFS

This appendix proves the results of the main text in the order they appear there. Throughout, a table x' is *consistent* with history h when it agrees with h on every read cell and keeps every unread score in $[0, 1]$. Table 2 collects the symbols.

A.1 NOTATION

Table 2: Symbols of the paper and where each is defined.

Symbol	Meaning	Defined in
$E_0, E_1, \dots, E_K$	the parent program and the K candidates of one round	Section 3
Q, N_q, b_q, T_q	targets, tasks per target, mean requirement, total requirement	Section 3
y_{kqi}, z_{qi}	score of candidate k and of the parent on task i of target q	Section 3
x, D_{kq}	the complete score table and the total paired difference on it	Equation (1)
$\mathcal{S}(x)$	the candidates meeting every requirement on table x	Equation (1)
h, $\mathcal{X}(h)$	the cells read in the round with their scores, and the tables consistent with h	Section 3
$O_{kq}(h)$, $O_q(h)$, $R_{kq}(h)$	read candidate tasks, read parent tasks, and their paired sum	Equation (2)
$L_{kq}(h)$, $U_{kq}(h)$	the exact interval on D_{kq} given h	Equation (2)
B, β	the cell budget of a round and the cells that remain	Section 3, Appendix B.2
α, ρ, α_ρ	level, position of a round in its chain, level of round ρ	Theorem 2
π_q	the shared uniformly random order of the tasks of target q	Section 4.1
d_i, p_i, $[a_i, c_i]$	paired difference on task i, its prediction, and its range	Equation (3)
n_i, s_i, $\bar{s}$, κ	earlier programs read on task i, their score sum, mean paid score, pseudo-count	Section 4.2
S_t, P_t, P	prefix sums of differences and predictions, and the total prediction	Section 4.2
m, e_t, $\nu_t(m)$	hypothesized total, residual, and its conditional mean under $D_{kq} = m$	Section 4.2
$M_t^{\pm}(m)$, $\lambda_j^{\pm}(m)$	betting mixtures testing m from each side and their betting fractions	Equation (4)
Θ, ϑ, w_ϑ	betting grid, grid point, and mixture weight	Equation (4)
$\mathcal{C}_t^+$, $\mathcal{C}_t^-$	lower and upper confidence sets of hypothesized totals	Section 4.2
ξ, τ	a fixed value and a random index in the path lemma	Lemma 1
g	task draws of a protocol that samples with replacement	Proposition 1
r, η	rollouts per cell, and the flip probability of one rollout	Appendix B.1
θ_c, γ	success probability of one rollout of cell c, and the rollout gap of a cell set	Proposition 2
$\Delta_k(h)$	shortfall of candidate k, the favorable cells it needs	Appendix B.2
φ, $\mathcal{P}_k(\beta)$, $F_q(\varphi)$	read plan, plans of cost at most β, favorable units of φ on target q	Appendix B.2
$W_\beta(h)$	completion probability, the best plan value within budget β	Equation (5)
$V_\beta(h)$	largest acceptance probability over score laws and policies	Appendix B.2
$\tilde{V}_\beta(h)$, $V_\beta^\star(h)$	acceptance probability of one-step lookahead, and the exact optimum	Theorem 3
$N_{\max}$	the longest row, $\max_q N_q$	Appendix B.2

A.2 PROOF OF THEOREM 1

Theorem 1 (Exact set-level certificates)**.** *For every history h and candidate k, $k \in \mathcal{S}(x')$ for all $x' \in \mathcal{X}(h)$ exactly when $L_{kq}(h) \geq T_q$ for every q, and $\mathcal{S}(x') = \varnothing$ for all $x' \in \mathcal{X}(h)$ exactly when every k has a target with $U_{kq}(h) < T_q$. Both equivalences hold at every step of any adaptive reading rule, and the complete table yields one of the two verdicts.*

Proof. Soundness. For any consistent x', each unread candidate score is at least 0 and each unread parent score is at most 1, so $D_{kq}(x') \geq L_{kq}(h)$, and symmetrically $D_{kq}(x') \leq U_{kq}(h)$. If $L_{kq}(h) \geq T_q$ for every q, then $k \in \mathcal{S}(x')$ for every consistent x'. If every candidate has a target with $U_{kq}(h) < T_q$, then no candidate belongs to $\mathcal{S}(x')$ for any consistent x'.

Necessity. Let x^- assign 0 to every unread candidate score and 1 to every unread parent score. It is consistent and attains $D_{kq}(x^-) = L_{kq}(h)$ for every k, q. If $k \in \mathcal{S}(x')$ for all consistent x', then $L_{kq}(h) = D_{kq}(x^-) \geq T_q$ for every q. Let x^+ assign 1 to unread candidate scores and 0 to unread

parent scores. It is consistent and attains every upper bound. If $\mathcal{S}(x') = \varnothing$ for all consistent x', then $\mathcal{S}(x^+) = \varnothing$, so every candidate k has a target q with $U_{kq}(h) = D_{kq}(x^+) < T_q$.

Adaptivity and completeness. Both equivalences quantify over the history alone, so they hold for the history produced by any score-dependent rule at any step. Once every cell is read, $L_{kq}(h) = U_{kq}(h) = D_{kq}(x)$ for the complete table x, and either some candidate clears every target, which is ACCEPT, or every candidate misses some target, which is NONE BETTER. □

A.3 PROOF OF PROPOSITION 1

Proposition 1 (Distinct-cell cost)**.** *If a protocol draws task indices of a target with N tasks independently of the table and reads the candidate and the parent on each draw, every decision it computes is a function of the paired outcomes on the distinct tasks drawn, so g draws resolve at most $\min\{g, N\}$ comparisons at a cost of at most $2\min\{g, N\}$ distinct cells.*

Lemma 2 (A repeated read moves no bound)**.** *Let h be a history containing cell c with score s, and let h' extend h by a further read of c returning s. Then $L_{kq}(h') = L_{kq}(h)$ and $U_{kq}(h') = U_{kq}(h)$ for every candidate k and target q.*

Proof. Equation (2) sums the observed scores of O_{kq} and O_q and charges the unobserved remainder through $N_q - |O_q|$ and $N_q - |O_{kq}|$. The sets $O_{kq}(h')$ and $O_q(h')$ index tasks, so re-reading a task already in them leaves both sets and both sums unchanged, and the observed score agrees with the recorded one by hypothesis. □

Proof of Proposition 1. Write $I_1, \ldots, I_g$ for the drawn indices and d_i for the paired outcome of task i on the target. The protocol observes $(I_j, d_{I_j})_{j \le g}$. Let $J = \{I_1, \ldots, I_g\}$ be the set of distinct indices drawn. Every observed pair (I_j, d_{I_j}) is determined by J, the draw sequence, and $(d_i)_{i \in J}$, so the protocol's observation is a function of the draw sequence and $(d_i)_{i \in J}$. The draw rule is independent of the table by hypothesis, so conditioning on the draw sequence leaves the table's contribution equal to $(d_i)_{i \in J}$, which proves the first claim. Since $J \subseteq \{1, \ldots, N\}$ and $|J| \le g$, the count of distinct paired comparisons is at most $\min\{g, N\}$. Each such comparison reads the candidate cell and the parent cell of one task, and Lemma 2 makes every further read of either cell redundant, so the distinct cells read number at most $2|J| \le 2N$. □

A.4 PROOF OF THEOREM 2

Theorem 2 (Finite-population certificates with paid history)**.** *Fix a round with a binary table. Fix predictions, ranges containing every paired difference, and mixture weights before the orders are drawn, and draw each order uniformly given the table and this side information. Then (i) for each candidate k and target q, $M_t^{\pm}(D_{kq})$ are nonnegative martingales starting at one in the row's own filtration, so D_{kq} leaves the lower sets at some time with probability at most $\alpha/(2K)$ and the upper sets with probability at most $\alpha/(2Q)$; (ii) under any schedule across rows and any stopping rule on the joint history, a wrong* ACCEPT *and a wrong* NONE BETTER *each have probability at most $\alpha/2$; (iii) on the same cells the certificate decides whenever the exact certificate of Theorem 1 decides, and a full read returns the exact totals; (iv) when rounds $\rho = 1, 2, \ldots$ of a chain each fix their side information from the paid history of all earlier rounds, rounds at levels α_ρ with $\sum_\rho \alpha_\rho \le \alpha$ err in any round with probability at most α.*

The proof uses three classical facts: the conditional law of a uniformly random permutation read as a prefix (sampling without replacement from a finite population), a control variate fixed before sampling, and Ville's inequality for nonnegative martingales. A last step, the path lemma of Appendix A.5, carries the time-uniform statement to the random reading time at which a global schedule stops.

Proof of Theorem 2. Fix candidate k and target q, and write $N = N_q$, $D = D_{kq}$ and $d_i = y_{kqi} - z_{qi}$. Before the orders are drawn the round fixes a prediction p_i of each d_i, a range $[a_i, c_i] \ni d_i$ for each task, a grid of betting fractions and nonnegative mixture weights summing to one. BranchCert-Σ is the case $p_i = 0$, $a_i = -1$, $c_i = 1$. Write $P = \sum_i p_i$, which is known. Row k reads a prefix of the uniformly random order π of target q; let $\mathcal{F}_t$ be generated by the side information and the first t entries of π with their differences.

Conditional mean of the residual. Given $\mathcal{F}_{t-1}$, the order π is uniform over the permutations consistent with the prefix, so $\pi(t)$ is uniform over the $N-t+1$ unread tasks. Their differences sum to $D-S_{t-1}$ and their predictions to $P-P_{t-1}$, where S_{t-1} and P_{t-1} are the prefix sums. Under the hypothesis $D=m$ the residual $e_t = d_{\pi(t)} - p_{\pi(t)}$ therefore has the known conditional mean $\nu_t(m) = \big((m-S_{t-1}) - (P-P_{t-1})\big)/(N-t+1)$. This is the finite-population control variate: the prediction leaves the mean identity exact, and a prediction close to d shrinks the spread of e_t around $\nu_t(m)$.

Martingales. The residual lies in $[a_{\pi(t)} - p_{\pi(t)},\, c_{\pi(t)} - p_{\pi(t)}]$, and the extremes $\underline{e}_t$ and $\overline{e}_t$ of these ranges over the unread tasks are $\mathcal{F}_{t-1}$-measurable. Each betting factor $1+\lambda_t(e_t-\nu_t(m))$ with $0 \le \lambda_t \le 1/(\nu_t(m)-\underline{e}_t)$ for the upper process, and $1-\lambda_t(e_t-\nu_t(m))$ with $0 \le \lambda_t \le 1/(\overline{e}_t-\nu_t(m))$ for the lower one, is nonnegative and has conditional mean one at $m=D$, since λ_t is predictable; a fraction whose bound is not positive is set to zero. The product of such factors is a nonnegative martingale starting at one, and so is the average over the grid with the fixed mixture weights. Every range contains the true difference, so $\nu_t(D)$ lies inside the range of the unread residuals and the admissible fractions are well defined. Ville's inequality (Howard et al., 2021) gives $\Pr[\exists t \le N : M_t^+(D) \ge 2K/\alpha] \le \alpha/(2K)$ and $\Pr[\exists t \le N : M_t^-(D) \ge 2Q/\alpha] \le \alpha/(2Q)$. The exact interval $[L_{kq}, U_{kq}]$ always contains D, so intersecting with it, with fallback to it when the intersection is empty, keeps D in the set on every path where the betting set keeps it. Lazily read parent values enter the exact interval and leave the martingale untouched. This proves (i) in the row's own filtration.

Set-level verdicts. A non-qualifying k has a target q^* with $D_{kq^*} < T_{q^*}$, and ACCEPT for k requires the lower set of (k,q^*) to lie at or above T_{q^*}, which excludes D_{kq^*}. The row's reading time at the global stop is a random index, so Lemma 1 and (i) bound this event by $\alpha/(2K)$, and a union bound over the K candidates gives $\alpha/2$ for a wrong ACCEPT. A wrong NONE BETTER excludes a qualifying candidate k^*, fixed by the table, which needs an upper set of k^* to exclude its true total on one of Q targets; the union bound over targets gives $\alpha/2$. The argument uses no property of the schedule across rows or of the stopping rule, which proves (ii).

Exact nesting. Since $\mathcal{C}_t^{\pm} \subseteq [L_{kq}, U_{kq}]$, every exact certificate of Theorem 1 on the cells read is a certificate for the confidence sets, and after a full read with the parent row complete $[L_{kq}, U_{kq}] = \{D\}$, so both sets equal $\{D\}$, which proves (iii).

Adaptive rounds. When round ρ fixes its predictions, ranges and weights from the paid history of all earlier rounds before drawing its orders, the argument above holds conditionally on that history, so round ρ errs with conditional probability at most α_ρ. The union bound over rounds gives total error at most $\sum_\rho \alpha_\rho \le \alpha$, which proves (iv). The chains decide round ρ at level $\alpha_\rho = 0.05/(\zeta(2)\rho^2)$ with $\zeta(2) = \sum_{\rho \ge 1} \rho^{-2}$, and these levels sum to 0.05. □

A.5 PROOF OF LEMMA 1

Lemma 1 (Path lemma)**.** *Let $(\mathcal{C}_t)_{t \le N}$ be random sets and ξ a fixed value with $\Pr[\exists t \le N : \xi \notin \mathcal{C}_t] \le \delta$. Then $\Pr[\xi \notin \mathcal{C}_\tau] \le \delta$ for every random index $\tau \in \{0, \dots, N\}$, from any filtration.*

Proof. If $\xi \notin \mathcal{C}_\tau$, then $\xi \notin \mathcal{C}_t$ at the index $t = \tau \le N$, so $\{\xi \notin \mathcal{C}_\tau\} \subseteq \{\exists t \le N : \xi \notin \mathcal{C}_t\}$ and $\Pr[\xi \notin \mathcal{C}_\tau] \le \delta$. The inclusion holds pathwise, so the bound holds for every random index τ in any filtration. □

B REPEATED ROLLOUTS AND ACQUISITION ORDER

B.1 REPEATED ROLLOUTS PER CELL

A rollout executes a program on a task once. BranchCert scores a cell by the majority score over r rollouts, with r odd and declared with the round, and records that score as the entry of the cell in the scored table. The repeat count multiplies the cost of every cell by r, and Theorem 1 applies unchanged to the table those majority scores populate.

Independent rollouts with true minority probability $\eta < 1/2$ reduce misrecording by majority vote. The largest observed minority fraction is $2/11$, a sample statistic of the audited cells, and

Appendix E.6 reports the repeated-rollout observations for each evaluator. The next proposition bounds the chance that an r-rollout majority table changes a verdict.

Proposition 2 (Repeated rollouts bound the chance of a wrong verdict)**.** *Let each rollout of cell c return one with probability θ_c, independently across rollouts, and let the true score of the cell be $\mathbf{1}\{\theta_c > 1/2\}$. Let a verdict depend on the true scores of a set A of cells and write $\gamma = \min_{c\in A}|\theta_c - \frac{1}{2}|$ for their rollout gap. For odd r, the r-rollout majority table reproduces every true score in A, and therefore the verdict, with probability at least $1 - |A|e^{-2r\gamma^2}$. The single-rollout table reproduces them with probability at least $1 - \sum_{c\in A}\min\{\theta_c, 1-\theta_c\}$. In particular, at a common flip probability $\eta < \frac{1}{2}$ per rollout, one rollout errs on a cell with probability η, the r-rollout majority errs on a cell with probability at most $e^{-2r(1/2-\eta)^2}$, and the chance of a wrong verdict is at most $|A|\, e^{-2r(1/2-\eta)^2}$.*

Lemma 3 (Majority error of one cell)**.** *Let $X_1, \ldots, X_r$ be independent with $\Pr[X_j = 1] = \theta$ and r odd, and let $M = \mathbf{1}\{\frac{1}{r}\sum_j X_j > \frac{1}{2}\}$. Then $\Pr[M \neq \mathbf{1}\{\theta > 1/2\}] \leq e^{-2r(\theta-1/2)^2}$.*

Proof. Take $\theta > 1/2$ and set $t = \theta - \frac{1}{2} > 0$. The majority errs exactly when $\frac{1}{r}\sum_j X_j < \frac{1}{2}$, which is $\frac{1}{r}\sum_j X_j - \theta < -t$; odd r excludes the boundary. Hoeffding's inequality for independent variables in $[0,1]$ bounds that probability by e^{-2rt^2}. For $\theta < 1/2$ apply the same argument to $1 - X_j$. For $\theta = 1/2$ the bound reads one and holds trivially. □

Proof of Proposition 2. Each cell of A satisfies $|\theta_c - \frac{1}{2}| \geq \gamma$, so Lemma 3 bounds its majority error by $e^{-2r\gamma^2}$. Write $v_c = \mathbf{1}\{\theta_c > 1/2\}$ for the true score of cell c and $\hat{v}_c$ for the score a majority of its r rollouts returns. A union bound over A gives $\Pr[\exists c \in A :\ \hat{v}_c \neq v_c] \leq |A|e^{-2r\gamma^2}$. On the complement every true score in A is reproduced, and the verdict depends on the true scores of A, so the verdict is reproduced as well. For a single rollout, $\Pr[X_1 \neq \mathbf{1}\{\theta_c > 1/2\}] = \min\{\theta_c, 1-\theta_c\}$, and a union bound over A gives the second claim. Solving $|A|e^{-2r\gamma^2} \leq \delta$ for r yields $r \geq \gamma^{-2}\ln(|A|/\delta)/2$.

For the stated specialization, let every cell carry the same flip probability $\eta < \frac{1}{2}$, meaning a rollout returns the true score of the cell with probability $1 - \eta$. Then $\theta_c \in \{\eta, 1-\eta\}$, $\min\{\theta_c, 1-\theta_c\} = \eta$, and $\gamma = \frac{1}{2} - \eta$. Lemma 3 bounds the majority error of one cell by $e^{-2r(1/2-\eta)^2}$, one rollout errs on a cell with probability exactly η, and the union bound over A bounds the chance of a wrong verdict by $|A|\, e^{-2r(1/2-\eta)^2}$. □

Under these assumptions, $r \geq \gamma^{-2}\ln(|A|/\delta)/2$ rollouts per cell hold the chance of a wrong verdict below δ at r times the cost. For a sensitivity calculation, stipulate a true flip probability $\eta = 2/11$ at every cell and independent rollouts. Then $\gamma = 7/22$, and $40\, e^{-2r(7/22)^2} \leq 0.05$ needs $r \geq 33.01$, first met by the odd count 35. If errors are also independent across cells, the exact binomial majority calculation puts the error of the 40-cell table below 0.05 at 17 rollouts per cell (Figure 10). The observed two minority rollouts out of eleven motivate this illustrative value of η, and both repeat counts hold under the stated assumptions.

B.2 ACQUISITION ORDER

Call $\Delta_k(h) = \sum_q \max\{0, \lceil T_q - L_{kq}(h)\rceil\}$ the *shortfall* of candidate k, the favorable cells it needs across targets in the most generous case. One cell read supplies at most one favorable unit, so a candidate whose shortfall exceeds the remaining budget β leaves acceptance planning and stays in the set-level exclusion test (Proposition 3).

Proposition 3 (A lower bound on certificate cost)**.** *Let $V_\beta(h)$ be the largest probability, under any law on the unread scores, of reaching* ACCEPT *within β further cell reads. If $\Delta_k(h) > \beta$ for every candidate k, then $V_\beta(h) = 0$.*

Proof. Fix candidate k. Revealing a candidate score $y_{kqi} \in \{0,1\}$ raises L_{kq} by $y_{kqi} \leq 1$ and leaves every other target of k unchanged. Revealing a parent score $z_{qi} \in \{0,1\}$ raises L_{kq} by $1 - z_{qi} \leq 1$ on target q alone. So one cell read lowers $\Delta_k(h)$ by at most one, and ACCEPT for k needs $\Delta_k = 0$. If $\Delta_k(h) > \beta$ for every k, no sequence of β cell reads reaches ACCEPT, so every policy has success probability zero and $V_\beta(h) = 0$. □

Algorithm 1 implements the exact certificate as an acquisition loop. It checks both certificates before each cell read, updates every affected bound after each cell read, and charges each parent cell once for all K candidates. Decision validity comes from Theorem 1, so every acquisition order leaves both certificates sound, and BranchCert-E ranks the next cell by a completion probability under a frozen product prior on the unread scores.

Each row of the table, meaning one candidate on one target or the parent on one target, is read in a fixed order, and a frozen product prior gives every unread score a probability of returning a favorable unit. A *read plan* φ for candidate k picks, for each target q, how many more candidate cells and how many more parent cells to read, with total cost at most the remaining budget β. Under the prior, the favorable units a plan collects on target q follow a Poisson-binomial law, and the plan clears target q when that count reaches the shortfall $\lceil T_q - L_{kq}(h) \rceil$. Write $\mathcal{P}_k(\beta)$ for the read plans for candidate k of cost at most β and $F_q(\varphi)$ for the favorable units plan φ collects on target q. The cells a plan assigns to different targets of a fixed candidate are disjoint, so the per-target probabilities multiply. Maximizing the product over plans, and then over candidates, gives the *completion probability*

$$W_\beta(h) = \max_k \max_{\varphi \in \mathcal{P}_k(\beta)} \prod_q \Pr\big[F_q(\varphi) \;\geq\; \lceil T_q - L_{kq}(h) \rceil\big]. \tag{5}$$

A budget knapsack over the Q targets computes it in $O(KQ(N_{\max} + B^3))$ arithmetic operations (Proposition 4), where $N_{\max} = \max_q N_q$ is the longest row and B is the budget. BranchCert-E reads the first cell of the maximizing plan, observes its score, and recomputes from the new history. The plan ranks the next cell, and the schedule recomputes the plan after every read.

Theorem 3 (Completion probability under adaptive replanning)**.** *For binary scores under a fixed product prior, the replanning schedule reaches* ACCEPT *with probability at least $W_\beta(h)$ from every undecided history h. The one-step lookahead schedule that reads the legal cell a maximizing $\mathbb{E}[W_{\beta-1}(h, a, X_a)]$ reaches* ACCEPT *with probability $\tilde{V}_\beta(h)$, and $W_\beta(h) \leq \tilde{V}_\beta(h) \leq V^\star_\beta(h)$, where $V^\star_\beta$ is the exact finite-horizon optimum over all adaptive plans.*

Proof. Lower bound. If $W_\beta(h) = 0$, nonnegativity proves the claim. At an undecided history with $W_\beta(h) > 0$, let $\varphi^\star$ be a read plan attaining $W_\beta(h)$ for candidate k, and let a be its first cell. Conditioning $\varphi^\star$ on the observed value $X_a \in \{0, 1\}$ yields a plan $\varphi^\star_{X_a}$ for the same candidate with the same remaining cells, cost at most $\beta - 1$, and shortfalls updated by the observation. This residual is feasible for the successor planner, so $W_{\beta-1}(h, a, X_a) \geq \Pr[\varphi^\star \text{ clears every target} \mid X_a]$. Taking expectations over X_a under the product prior gives $\mathbb{E}[W_{\beta-1}(h, a, X_a)] \geq W_\beta(h)$. Induction on the budget makes the success probability of the replanning schedule from h at least $W_\beta(h)$. At an accepted history both values equal one. At a NONE BETTER history both equal zero. At zero budget, $W_0(h) \in \{0, 1\}$ equals the terminal reward.

Sandwich. Nonnegativity handles $W_\beta(h) = 0$. For $W_\beta(h) > 0$, let a_g maximize $\mathbb{E}[W_{\beta-1}(h, a, X_a)]$ over legal cells. By the argument above, $\mathbb{E}[W_{\beta-1}(h, a_g, X_a)] \geq \mathbb{E}[W_{\beta-1}(h, a, X_a)] \geq W_\beta(h)$ for the plan head a. Induction on β gives $\tilde{V}_\beta(h) = \mathbb{E}[\tilde{V}_{\beta-1}(h, a_g, X_{a_g})] \geq \mathbb{E}[W_{\beta-1}(h, a_g, X_{a_g})] \geq W_\beta(h)$. The lookahead schedule is itself a feasible adaptive plan, so $\tilde{V}_\beta(h) \leq V^\star_\beta(h)$. □

Proposition 4 (Cost of the completion probability)**.** *The completion probability $W_\beta(h)$ of Equation* (5) *takes $O(KQ(N_{\max} + B^3))$ arithmetic operations.*

Proof. For a fixed candidate and target, the Poisson-binomial law of favorable units over a prefix of length ℓ follows from the law over length $\ell - 1$ by one convolution with a Bernoulli variable. One sweep along the row therefore produces every prefix law, and the remaining budget truncates each to at most B entries. Scanning a row costs $O(N_{\max})$ and constructing its budget-truncated prefix laws costs $O(B^2)$. For one candidate and target, enumerating candidate and parent prefix lengths and evaluating their combined success tail costs $O(B^3)$. The budget knapsack combines the Q target option tables in $O(QB^2)$ operations. Summing over candidates gives $O(KQ(N_{\max} + B^3))$ arithmetic operations; the bit complexity of exact rational arithmetic is additional. □

Appendix E.7 measures where these schedules separate on exact finite models. On every live table of Section 5, BranchCert-E and the fixed parent-first order acquire the same cells, so the planner and the parent-first order spend equal cells on real data.

C Extended related work

Self-improving agents choose their next parent by reviewed commits (Razzhigaev et al., 2026), by training a runtime harness that separates model failures from harness failures (Yue et al., 2026), by selecting on held-out tasks across generations (Nguyen et al., 2026), or by co-evolving the evaluator with the agent (Iacob et al., 2026). These approaches develop the proposal process; the certificate here controls the admission error and adds the verdict that no candidate of the round qualifies. Evaluation environments for such loops record training, validation, replay and cost for every harness change (Zheng et al., 2026), measure whether an agent can design training algorithms (Chi et al., 2026), and gate releases on several quality dimensions with promote, hold and rollback decisions (Maiorano, 2026); they define what a round measures, and the certificate decides when that measurement may stop.

Subset evaluation estimates a model's benchmark score from a small curated or sampled set, calibrated on many earlier models (Maia Polo et al., 2024; Das et al., 2026; Huang et al., 2026), through fixed item-response parameters (Habba et al., 2026), or through latent abilities shared across benchmarks (Krumdick et al., 2026). These methods focus on estimating benchmark performance. The certificate reads adaptively, decides a whole candidate set at a stated level, and returns the exact totals after a full read.

Prediction-powered inference corrects model predictions with a small labeled sample (Angelopoulos et al., 2023; Cowen-Breen et al., 2026; Divekar, 2026; Park et al., 2025); active inference and active testing choose which points to label from a surrogate (Zrnic & Candès, 2024; Berrada et al., 2025; Ashouritaklimi et al., 2026); betting on predictions uses unlabeled data to sharpen sequential tests (Tenzer et al., 2026); and judge-based certification corrects for judge error rates or attributes drift between a system and its judge (Feng et al., 2026; Li, 2026). BranchCert-Σ+H uses the loop's own paid history as a control variate on a finite table read without replacement, with the prediction frozen before the reading order is drawn.

Testing by betting supplies the confidence sequences used here (Ramdas et al., 2023; Vovk & Wang, 2021). Recent work sharpens betting intervals for bounded means (Martinez-Taboada & Ramdas, 2026; Kilian et al., 2026), adapts the bet to a deadline or a time preference (Taga et al., 2026; Clerico et al., 2026), tests a global null over several arms (Sandoval et al., 2026), and explains where the union bound enters best-arm identification (de Heide, 2026). Weighted sampling without replacement gives confidence sequences for financial audits (Shekhar et al., 2023), and betting also audits AI systems adaptively (Zhou et al., 2026a), transfers simulated robot performance to real trials (Mahboob et al., 2026), and stops two-agent game evaluations with a variance-reducing control variate (Li et al., 2026). Our certificate allocates its error level over a candidate set with two one-sided verdicts on a finite table.

Bandit identification is the closest decision problem. Good-arm identification and 1-identification return an arm above a threshold or report that none exists (Kano et al., 2019; Li & Cheung, 2026); best-arm identification has matching lower bounds (Garivier & Kaufmann, 2016) and uses LLM judges with selective human audits (Ao et al., 2026); valid best-model identification (Best-model ID) completes a partially observed score matrix by low-rank factorization (Tolochinsky et al., 2026); and Bayesian optimal stopping sets per-item sampling for LLM evaluations (Pilditch, 2026). A candidate here qualifies through a conjunction of requirements on several targets, its differences share one parent row, and its tasks are read without replacement.

D Experimental setup

D.1 Live rounds and tasks

Tasks come from ToolFailBench (Soni, 2026), 1,000 tool-use tasks over five domains (finance, medical, legal, cybersecurity, real estate) and four target failure modes. Three failure modes form the required target, and the fourth, on which the parent already succeeds, forms the control target. A hash-ranked stratified split assigns 400 tasks to candidate construction, 300 to admission and 300 to a hidden audit, and every round draws its 15 required and five control tasks from the admission split with its own seed. The evaluation used here scores a task one when the trajectory reaches the declared success condition. Qualification requires mean requirements of $b_q = 1/225$ on the required

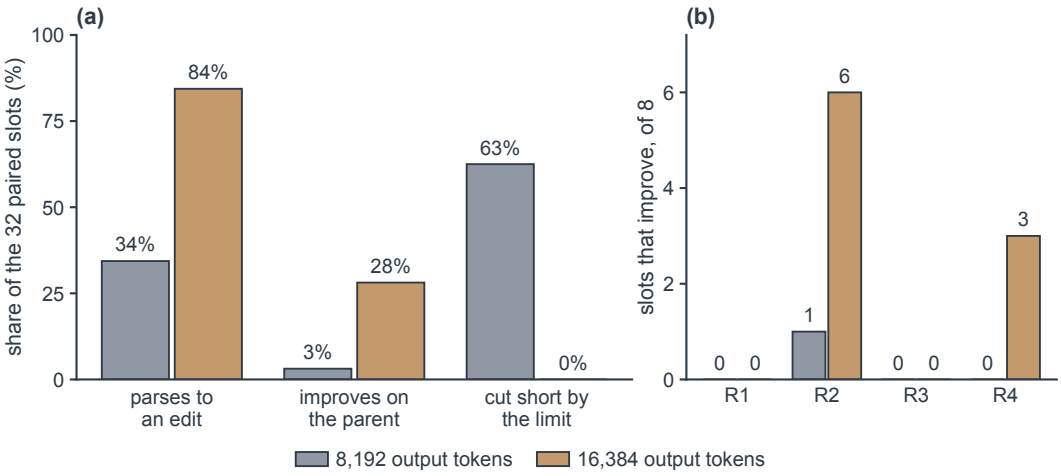

Figure 5: Proposal supply at two output budgets. (a) The 32 slots, from a parsed response to an improving candidate. (b) The same slots per round.

target and $b_q = -1/75$ on the control target, so the total requirements $T_q = 1/15$ and $-1/15$ are fractional, and a round with K candidates holds $20(K + 1)$ cells.

Granite-4.2-8B (Granite Team, IBM, 2026) proposes and evaluates four rounds, Granite R1 to R4, with 9, 7, 12 and 10 candidates, of which 0, 7, 0 and 3 qualify. Granite R3 runs live at a budget of 220 cells of 260, and Granite R4 at 80 of 220. Qwen3.5-9B (Qwen Team, 2026) re-scores the ten candidates of Granite R4 and their parent on the same tasks, the matched Qwen3.5 table of 220 cells on which all ten qualify, at a budget of 80. Two further complete tables, rounds 6 and 8 of a later Granite run with one and two candidates and no qualifying candidate, enter Table 7 as Granite R6 and R8. Task rollouts decode at temperature zero with top-p 1.0 and 2,048 output tokens, and these runs execute each cell once ($r = 1$). The sampling baselines use 32 reading orders and report the median cost; BranchCert-E and the fixed orders are deterministic and run once, and BranchCert-Σ runs 16 reading orders and reports the median.

D.2 Candidate supply and the proposal budget

Construction evaluates the parent on 75 tasks and generates eight proposal slots per round under two prompting conditions, decoded at temperature 0.8 and top-p 1.0. Every slot is decoded twice with the same prompt and seed, at 8,192 and at 16,384 output tokens, and a response enters the candidate set when it parses to a program that differs from the parent. Across the 32 slots, 11 responses are valid at the smaller budget and 27 at the larger, and one and nine of them qualify, which gives the 38 candidates and ten qualifying candidates of the four rounds (Figure 5). The length limit cuts 63% of responses at the smaller budget and none at the larger, nine slots start improving, one stops, and 22 keep their status. Prompts, seeds, tasks and requirements stay fixed across the two budgets. Two of the four rounds contain no qualifying candidate, the case the None Better verdict serves.

D.3 HELM Lite rounds

The rounds of Section 5.1 use the per-instance scores that HELM Lite publishes for every model it evaluates, collected from its public release. A task scores one when the family's primary metric (final-number exact match on GSM8K, chain-of-thought equivalence on MATH, quasi-exact match on MedQA and LegalBench, exact match on MMLU and OpenBookQA) equals one. The (required, control) target pairs are GSM8K–MMLU, MATH–OpenBookQA, MedQA–LegalBench, MMLU–MedQA, LegalBench–GSM8K and OpenBookQA–MATH. The parent sits at the 20th, 40th, 60th or 80th percentile of required accuracy among the models scored on both families, and the ten candidates are its nearest neighbors in required accuracy, a uniform random draw, or an ability-stratified draw. Each rule runs on the same table with its own reading orders, and every verdict is scored against the complete table.

Table 3: Six Qwen3.5-9B chains of eight rounds. Shares read and savings (%) average rounds and 8 reading orders, with standard errors in parentheses. MATH first and last give the accuracy (%) of the initial and final prompt on each chain's 300-task held-out MATH split, averaged over chains in the All row.

Chain	Accepted	Exact certificate	BranchCert-Σ	BranchCert-Σ+H	Saving	MATH first	MATH last
1	2	65.8 (0.1)	22.3 (0.4)	18.3 (0.3)	18.2 (1.2)	78.3	79.7
2	0	37.9 (0.1)	23.9 (0.4)	18.2 (0.3)	23.7 (1.5)	78.7	78.7
3	1	36.2 (0.1)	23.5 (0.5)	17.9 (0.3)	23.7 (1.5)	82.0	83.3
4	1	33.0 (0.0)	18.7 (0.8)	15.5 (1.0)	17.6 (3.5)	81.7	83.7
5	2	33.9 (0.1)	22.7 (0.3)	20.0 (0.4)	11.9 (2.1)	82.0	84.7
6	1	32.4 (0.1)	27.8 (0.7)	26.3 (0.4)	5.2 (2.4)	81.0	93.7
All	7	39.9 (0.0)	23.2 (0.2)	19.4 (0.1)	16.4 (0.7)	80.6	83.9

D.4 Self-improvement chains

Qwen3.5-9B rewrites its own system prompt over six chains of eight rounds, with 1,500 required MATH tasks and 1,500 control MMLU tasks per round and 27,000 cells per table. The mean requirements are $b_q = 0.02$ on MATH and $b_q = -0.02$ on MMLU, and round ρ of a chain runs at level $\alpha_\rho = 0.05/(\zeta(2)\rho^2)$. Table 3 reports each chain, and the All row pools the chains.

D.5 Algorithms

Algorithm 1 is the exact certificate with acquisition planning, and Algorithm 2 is BranchCert-Σ with the lazy parent row.

Algorithm 1 BranchCert-E for one round

Require: parent E_0, candidates $E_{1:K}$, targets $(N_q, T_q)_{q \le Q}$, budget B, repeats r, product prior
$h \leftarrow \varnothing, \quad \beta \leftarrow B$
loop
  if $\exists k : \forall q,\ L_{kq}(h) \ge T_q$ **then**
    return ACCEPT with candidate k
  end if
  if $\forall k\ \exists q : U_{kq}(h) < T_q$ **then**
    return NONE BETTER
  end if
  if $\beta < r$ **then**
    return UNDECIDED
  end if
  exclude every k with $\Delta_k(h) > \lfloor \beta / r \rfloor$ from acceptance planning
  compute $W_{\lfloor \beta/r \rfloor}(h)$ and a maximizing read plan
  if the plan contains a cell **then**
    $c \leftarrow$ its first cell
  else
    $c \leftarrow$ the next unscored cell in the fixed row-prefix order
  end if
  execute c r times, append its majority score to h, and set $\beta \leftarrow \beta - r$
end loop

D.6 Baselines

Every baseline runs on the same tables, reading orders, level and distinct-cell accounting as BranchCert-Σ (Section 4.1).

Shared-cell rules on the live tables. On the Granite rounds and the matched Qwen3.5 table we run the non-adaptive halfspace order (Ghuge et al., 2025) at unit weights, successive halving (Karnin et al., 2013), racing against a shared parent (Kalyanakrishnan et al., 2012; Dunnett, 1955), a

Algorithm 2 BranchCert-Σ for one round

Require: parent E_0, candidates $E_{1:K}$, targets $(N_q, T_q)_{q \le Q}$, level α, grid Θ, budget B
draw one uniform ordering π_q of the tasks of each target; set every $M^{\pm} \leftarrow 1$
loop
  for each candidate k not excluded **do**
    if $\forall q : \min \mathcal{C}^{+}_{kq} \ge T_q$ **then**
      return ACCEPT with candidate k
    end if
    if $\exists q : \max \mathcal{C}^{-}_{kq} < T_q$ **then**
      exclude k
    end if
  end for
  if every candidate is excluded **then**
    return NONE BETTER
  end if
  if the budget is spent **then**
    return UNDECIDED
  end if
  $k \leftarrow$ the first candidate not excluded with an uncertified target; $q \leftarrow$ its target with the smallest $(\max \mathcal{C}_{kq} - T_q)/N_q$
  if k has read at least $N_q/2$ tasks of q and the parent row of q is incomplete **then**
    read the rest of the parent row of q and tighten every $[L_{k'q}, U_{k'q}]$; **continue**
  end if
  $i \leftarrow$ the next task of π_q in the row of k; read z_{qi} if unread, then read y_{kqi}
  update $M^{\pm}_{kq}(m)$ for every m by Equation (4), drop every m whose wealth reaches $2K/\alpha$ from $\mathcal{C}^{+}_{kq}$ or $2Q/\alpha$ from $\mathcal{C}^{-}_{kq}$, and intersect both with $[L_{kq}, U_{kq}]$ (unread tasks with a known parent contribute $[-z_{qi}, 1 - z_{qi}]$, the others $[-1, 1]$), falling back to $[L_{kq}, U_{kq}]$ when a set empties
end loop

betting screen inspired by Sengupta (2026), fixed parent-first and candidate-first orders, and score-and-keep evaluation (Zhang et al., 2026a; Robeyns et al., 2025) of the required target. Successive halving recommends its surviving leader, the Hoeffding race uses differences in $[-1, 1]$ with time and candidate error spending, and the betting screen uses wealth $\prod_t (1 + 0.5 d_t)$, with d_t the t-th drawn paired difference, and threshold $1/0.1$ for a zero-mean null on each candidate in turn.

Fixed-batch paired test. The fixed-batch paired test of Ma & Wu (2026), a paired-binomial test, draws 256 required and 512 control pairs uniformly with replacement, executes the candidate and the parent with a shared reset for every pair, and decides from the positive and negative disagreement counts at spending level $\delta = 0.1$, with $\alpha = 0.0608$ at the first attempt and 0.0152 at the second. Its candidate ranks first on the complete table, a choice that favors the test. The 1,536-rollout batch is charged as 40 distinct cells at a duplication factor of 38.4, and its deferral is the UNDECIDED outcome.

Two-Gate adaptation. The Two-Gate rule (Cai et al., 2026) runs per candidate on the full required target with the parent cached and an even risk split. Its source configuration stores 256 tasks and 64 failure tasks. At the 15 required tasks of a round its gain bound is non-positive, so the rule rejects every candidate, and we report its full-batch cost with no budgeted decision.

Certified update gates. Neither paper releases code, so both gates are reproduced from their text and run on the same tables, reading orders, level $\alpha = 0.05$ and distinct-cell accounting as BranchCert-Σ. DISCERN (Balachandran, 2026) certifies one candidate against a retained parent (its incumbent) through the disagreement identity $\Delta = \mathbb{E}[D\,A]$ with a predictable-plug-in empirical-Bernstein confidence sequence. Reading a disagreement indicator here reads both cells, so its label-sampling tier labels every drawn task; each target's stream draws tasks with replacement, each distinct cell is charged once, its safe and regression verdicts are shifted to our integer requirements, the level is split as $\alpha/(KQ)$ following its multi-slice rule, candidates are audited in order, the first candidate certified on both targets is accepted, and a stream stops at the 60,000 draws of the paper's longest horizon. Slice-wise certification (Zhang et al., 2026b) runs a one-sided Hoeffding non-inferiority test per slice with an intersection-union rule across slices and a Bonferroni split over a frozen candidate

library, falling back to the parent when certification fails. We record this fallback as UNDECIDED, since it certifies no exclusion of every candidate. Each target is one slice; a tenth of each target's shuffled tasks serves as its development split, which freezes the library and sets the per-slice sample size by its planning rule with power 0.8; the certification sample is read from the remaining tasks, the bound is tightened by the finite-population correction and the exact interval, and among certified candidates the one with the largest lower slack is accepted. The development share of a tenth was the most favorable to the baseline on these tables.

Without-replacement betting sequences. The sequence block of Table 4 runs published without-replacement sequences inside one eager-parent Bonferroni wrapper, which reads the parent row before the round and gives each candidate–target pair a two-sided level $\alpha/(KQ)$. Every sequence shares the certificate driver of BranchCert-Σ: the parent cache, first-read charging, the grid of hypothesized totals and the exact-interval fallback, with paired differences mapped to $[0, 1]$ by $d \mapsto (d + 1)/2$. Hedged-capital betting (Waudby-Smith & Ramdas, 2024) uses predictable plug-in bets truncated at $1/2$, and the empirical-Bernstein sequence (Waudby-Smith & Ramdas, 2020) unions its two one-sided bounds. ALPHA (Stark, 2023) runs the shrink-trunc test with shrinkage parameter 100, with its prior mean set from the frozen history prediction or from the paper's adaptive rule. Control-variate betting (Shekhar et al., 2023) uses ApproxKelly bets with uniform weights and the frozen history predictor as control variate. BranchCert-Σ and BranchCert-Σ+H also run inside this wrapper, so the spread within the block prices the sequence, and the gap to the first rows of Table 4 prices the one-sided set-level wrapper and the lazy parent row.

Bandit and subset rules. The bandit block of Table 4 runs these rules under the same set-level wrapper, level and distinct-cell accounting. Good-arm identification, run with the HDoC index and with LUCB confidence bounds, and 1-identification certify whether some candidate clears a threshold; each candidate is an arm, each pull reads one task of the candidate and its parent, and a run that reaches its pull limit returns UNDECIDED. tinyBenchmarks and MINCE estimate each candidate's score from a fixed task subset and accept the best estimate above the requirement, so they decide every run with no error control. Good-arm identification with LUCB bounds reads 94.9% and decides 54% of runs, and MINCE on a fixed subset reads 13.6% with 59.1% wrong verdicts. The certified rules have zero wrong verdicts and read most of each table; the subset estimators read the least and exceed the level α.

E ADDITIONAL RESULTS

E.1 HELM LITE COMPARISON

Table 4 lists every rule on the 24 rounds at $\alpha = 0.05$. Figure 6 repeats the comparison at five levels, and Figure 7 places each rule by its share read and wrong verdicts. On the 216-round difficulty sweep the saving of BranchCert-Σ over the exact certificate is largest when the candidates are the parent's nearest neighbours at the largest mean requirement (Figure 8).

E.2 MECHANISM ABLATIONS

Table 5 changes each mechanism of Theorem 2 on the chains and on the HELM rounds, averaging over individual runs, and Section 5.3 reads the same numbers. The paired-sequence variant of Table 4 uses paired-only bounds and charges each parent cell once, and the parent-charging row of Table 5 changes the accounting alone. The relaxed variant of Table 4 multiplies α by ten.

E.3 BEHAVIOR CHECKS OF THE PAID HISTORY

The chains run Equation (3) with the pseudo-count $\kappa = 2$, and Algorithm 2 completes a parent row once a candidate has read half of its target. We check the conditions of Section 4.3 on 15 scenarios, and all 15 hold. Masking checks show that flipping any cell earlier rounds have not read, or any cell of the current and later rounds, leaves the predictions and the certificate run unchanged, and that the same corruption changes the predictions once the mask is dropped; on HELM the predictor reads the scores of the other models. Charging checks show that an empty cache reproduces the run without a cache and that rerunning a round on its own paid cells charges nothing and returns the same

Table 4: The 24 HELM Lite (Liang et al., 2023) rounds at $\alpha = 0.05$. Share read is the within-round median share, averaged over rounds; Decides is the share of runs that decide; Wrong verdicts is the share of decided runs whose verdict disagrees with the complete table (%). A dash marks a rule with no decided run. * Uses other models' scores.

Rule	Share read (%)	Decides (%)	Wrong verdicts (%)
BRANCHCERT SET-LEVEL WRAPPER			
BranchCert-Σ+H (ours)*	37.8	100	0.3
BranchCert-Σ (ours)	38.2	100	0.5
BRANCHCERT-Σ MECHANISM ABLATIONS			
no lazy parent read	41.5	100	0.5
two-sided levels	40.3	100	0.0
unscaled bets	38.4	100	0.3
exact interval, Σ reading order	44.6	100	0.0
with-replacement mean	42.2	100	0.0
paired-sequence variant	62.1	100	0.0
fixed reading order	27.7	100	20.8
level 10α	32.8	100	2.3
WITHOUT-REPLACEMENT SEQUENCES			
Exact certificate	44.7	100	0.0
WoR hedged capital (Waudby-Smith & Ramdas, 2024)	44.4	100	0.0
WoR empirical-Bernstein (Waudby-Smith & Ramdas, 2020)	44.7	100	0.0
ALPHA shrink-trunc, history prior (Stark, 2023)	44.4	100	0.0
ALPHA shrink-trunc, adaptive prior (Stark, 2023)	43.6	100	0.0
control-variate WoR betting* (Shekhar et al., 2023)	44.1	100	0.0
BranchCert-Σ under the same wrapper, parent read first	43.1	100	0.0
BranchCert-Σ+H under the same wrapper, parent read first*	42.9	100	0.0
PER-CANDIDATE GATES			
DISCERN (Balachandran, 2026)	69.3	40	0.0
Slice-wise certification (Zhang et al., 2026b)	28.8	22	0.0
ANYTIME-VALID EVALUATORS			
CELEUS* (Zhou et al., 2026b)	61.9	100	0.0
Active confidence sequences* (Hsu & Shekhar, 2026)	46.8	9	0.0
AV-AIVAT asymptotic CS* (Li et al., 2026)	60.3	57	0.0
AV-AIVAT empirical-Bernstein CS* (Li et al., 2026)	71.5	51	0.0
Best-model ID (PULSE)* (Tolochinsky et al., 2026)	56.5	58	0.0
FAQ* (Wu et al., 2026)	30.6	5	0.0
BANDIT RULES			
HDoC good-arm (Kano et al., 2019)	93.8	54	0.0
1-identification (Li & Cheung, 2026)	100.0	16	0.0
tinyBenchmarks* (Maia Polo et al., 2024)	13.6	100	65.1
MINCE sized to requirement* (Das et al., 2026)	67.0	100	19.8
SCREENING AND RACING RULES			
Successive halving (Karnin et al., 2013)	32.6	100	75.5
Hoeffding race (Kalyanakrishnan et al., 2012)	44.2	0	—
PACE-style paired gate (Shawn, 2026)	77.7	4	0.0
STOPPING RULES WITHOUT ERROR CONTROL			
BEACON-style stopping (Wan et al., 2025)	16.8	63	39.3

verdict. Verdict checks cover a candidate that gains on the required target and fails the control target, which stays unaccepted in every check, an empty qualifying set, two qualifying candidates, and early acceptance of a dominant candidate. Over every reading order of seven-task tables at the qualification boundary, the share of wrong verdicts stays at or below $\alpha/2$ for BranchCert-Σ and BranchCert-Σ+H, with informed and misleading predictions. A declared range that misses a paired difference raises an error, an empty intersection falls back to the exact interval, a full read returns the exact totals, zero betting fractions reproduce the exact certificate, and on the cells BranchCert-Σ reads the exact certificate decides no earlier than BranchCert-Σ.

E.4 LIVE TABLES AT FOUR BUDGETS

The four Granite rounds and the matched Qwen3.5 table each have a complete score table. Table 6 replays every shared-cell rule on each table at one quarter, one half, three quarters and all of its cells. BranchCert-E and the parent-first order acquire the same cells at every budget on all five tables, so the shared parent row accounts for the cost of BranchCert-E. It reaches NONE BETTER over twelve

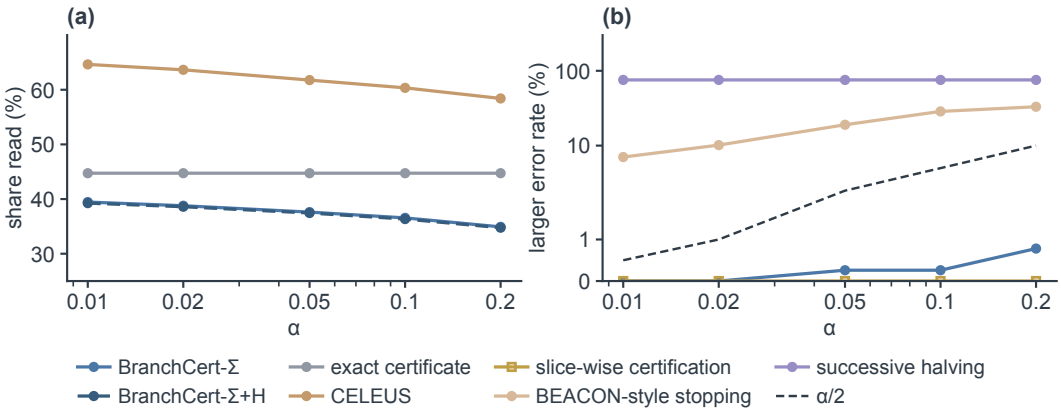

Figure 6: The 24 HELM Lite rounds at five levels. (a) Share read. (b) The larger of the two wrong-verdict rates over all runs, with the $\alpha/2$ line.

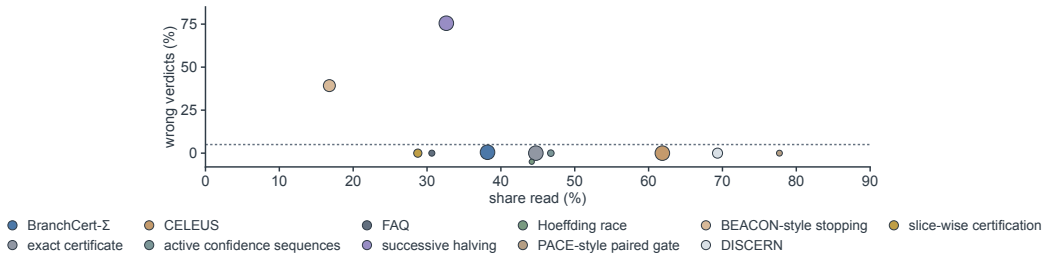

Figure 7: Share read against wrong verdicts on the 24 HELM Lite rounds. Marker area grows with the decision rate, and the dotted line marks $\alpha = 5\%$, the bound on the two wrong-verdict rates combined.

candidates at 174 cells of 260 and accepts a named candidate over ten at 53 of 220, the live costs. On the two rounds with no qualifying candidate it returns NONE BETTER at 154 of 200 and 174 of 260 cells, the halfspace and candidate-first orders spend the whole table, the betting screen decides at most 9% of its reading orders with every verdict wrong, the Hoeffding race decides none, and successive halving names a leader that fails the requirements in every reading order. On the matched Qwen3.5 table at the live budget of 80 cells, the betting screen reads 25 cells and decides 94% of its reading orders, successive halving 48, the halfspace order 38, the Hoeffding race 78 with no decision, and score-and-keep 165, past the budget. These costs come from a separate replay at the live budget of 80 cells, between the quarter and half budgets of Table 6. The sampling rules draw task indices independently of the scores, so on the two tables of the same shape, Granite R4 and the matched Qwen3.5 table, the Hoeffding race, which eliminates no candidate, reads the same cells, and successive halving reads nearly the same number. Table 7 runs BranchCert-Σ on all seven complete live tables.

E.5 DEGRADED CONDITIONS

We replay the certificate under three degraded conditions on the matched Qwen3.5 table and on Granite R3. Withholding one cell makes its read return a typed failure; the run then ends with that failure identity and the certificate reached so far, and no candidate is accepted on that history. Withholding a cell outside the requests of the schedule leaves the verdict and the cost unchanged. A budget below the certificate's own cost ends in UNDECIDED at budget exhaustion. A fixed acquisition order that ignores the row-prefix rule leaves both certificates valid: every verdict such an order reaches agrees with the complete table, and the three uniformly random orders accept on the matched Qwen3.5 table at 150 to 165 cells and reach NONE BETTER on Granite R3 at 237 to 252, where BranchCert-E needs 37 and 174. A fixed list reads every cell in its listed order, including cells of rows the certificate has already decided, so the candidates-before-parent list spends more than the

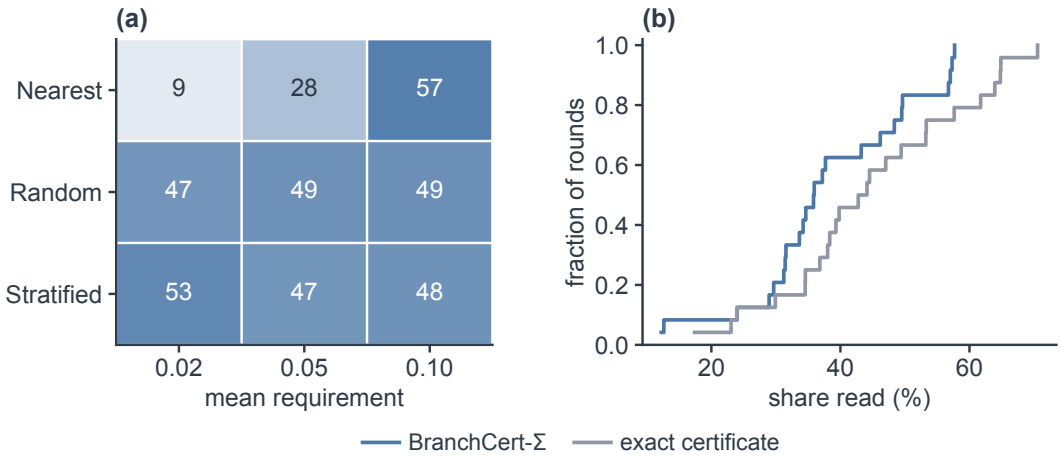

Figure 8: Saving of BranchCert-Σ over the exact certificate. (a) Median saving (%) on the 216-round difficulty sweep. (b) Share read of each table on the 24 HELM Lite rounds.

Table 5: Mechanism ablations on the chains and the HELM Lite rounds: run-mean share read and decision rate (%). Table 4 reports within-round medians.

Variant	Chains read	Chains decide	HELM read	HELM decide
BranchCert-Σ+H	19.4	100	37.4	100
zero prediction, history ranges	23.2	100	37.6	100
without finite-population correction	27.0	100	41.5	100
parent cells charged per candidate	27.0	100	87.6	100
BranchCert-Σ	23.2	100	37.6	100
parent cells charged per candidate	30.5	100	87.4	100
exact interval with the Σ reading policy	39.9	100	44.5	100
exact interval on the Σ trace	23.2	1	37.6	10

candidate-first order of Table 6, which skips those cells (206 against 204 on the matched Qwen3.5 table and 259 against 254 on Granite R3).

E.6 Repeated rollouts of the fixed-batch paired test

An audit executes each of the 40 cells of the fixed-batch paired test on every Granite round many times at temperature zero. The four rounds contain 11 unsettled cells, cells that return both scores across rollouts, and in all 11 the first rollout disagrees with the majority of the cell (Figure 9). Under independent rollouts with a minority share of at most 0.18 that pattern has probability below 10^{-8}, so the rollouts of these tests are dependent and Proposition 2 falls outside its scope for them. A fresh re-execution of the test on Granite R1 executes its cells 11 to 59 times and returns the same four unsettled cells. Five re-executions of the matched Qwen3.5 table and of the complete tables of Granite R3 and R4 move none of their cells, and the certificate returns the same verdict at 174 and 53 cells under the majority of rollouts and under the first rollout.

Scoring each cell once changes how the decision rule reads the executed rollouts (Table 8). On Granite R2 the fixed-batch Accept at $[0.070, 0.227]$ over 256 pairs becomes Undecided at $[-0.337, 0.528]$ under the majority of rollouts and $[-0.325, 0.601]$ under the first rollout, since the 256 pairs resolve 15 comparisons (Proposition 1). The other three rounds return Undecided under every rule. Figure 10 gives the rollout count that independent rollouts would need.

E.7 Exact finite models

The finite-model studies enumerate every binary score table of a declared setting with its exact probability under the product prior, run every schedule on every table, and report exact rational

Table 6: Distinct cells per rule at four budgets on each complete table. Seeded rules report the median over 32 reading orders, with the decision rate and the share of decisions that agree with the complete table (%) in parentheses. Circles mark no decision and daggers a cost past the budget.

Budget	BranchCert-E	Parent-first	Halfspace	Cand.-first	Betting screen	Succ. halving	Hoeffding race	Score-and-keep
Granite R1, 9 cand., 0 qualify, 200 cells								
50	50°	50°	50°	50°	30 (3, 0)	32 (100, 0)	50° (0)	150†
100	100°	100°	100°	100°	43 (6, 0)	53 (100, 0)	88° (0)	150†
150	150°	150°	150°	150°	45 (6, 0)	72 (100, 0)	108° (0)	150
200	154	154	200	199	45 (9, 0)	82 (100, 0)	122° (0)	150
Granite R2, 7 cand., 7 qualify, 160 cells								
40	40	40	40	40°	28° (0)	24 (100, 100)	40° (0)	120†
80	40	40	40	80°	30 (25, 100)	44 (100, 100)	72° (0)	120†
120	40	40	40	120°	30 (81, 100)	60 (100, 100)	88° (0)	120
160	40	40	40	148	30 (94, 100)	70 (100, 100)	98° (0)	120
Granite R3, 12 cand., 0 qualify, 260 cells								
65	65°	65°	65°	65°	47° (0)	36 (100, 0)	65° (0)	195†
130	130°	130°	130°	130°	68° (0)	65 (100, 0)	110° (0)	195†
195	174	174	195°	195°	88 (3, 0)	91 (100, 0)	138° (0)	195
260	174	174	260	254	107 (3, 0)	106 (100, 0)	155° (0)	195
Granite R4, 10 cand., 3 qualify, 220 cells								
55	53	53	55°	55°	32 (6, 100)	31 (100, 75)	55° (0)	165†
110	53	53	58	110°	42 (75, 100)	57 (100, 78)	96° (0)	165†
165	53	53	58	165°	42 (97, 100)	78 (100, 75)	118° (0)	165
220	53	53	58	215	42 (100, 100)	90 (100, 91)	132° (0)	165
Matched Qwen3.5 table, 10 cand., 10 qualify, 220 cells								
55	37	37	38	55°	25 (81, 100)	32 (100, 100)	55° (0)	165†
110	37	37	38	110°	25 (100, 100)	57 (100, 100)	96° (0)	165†
165	37	37	38	165°	25 (100, 100)	78 (100, 100)	118° (0)	165
220	37	37	38	204	25 (100, 100)	90 (100, 100)	132° (0)	165

Table 7: Distinct cells read on the seven live tables, median over 16 reading orders. Granite R6 and R8 come from a later Granite run (Appendix D.1). Circles mark no decision and crosses a wrong verdict.

Table	Size	BranchCert-Σ	Parent-first	DISCERN	Slice-wise	Saving (%)
Granite R1	200	130	154	200	30°	15.6
Granite R2	160	40	40	40	24°	0.0
Granite R3	260	106	174	260	39°	39.1
Granite R4	220	50	53	60	33°	5.7
Matched Qwen3.5 table	220	36	37	40	33°	2.7
Granite R6	40	29	36	40	6°	19.4
Granite R8	60	40	54	60	9°	25.9

quantities. Fifteen settings with two or three candidates, one or two targets, one or two cells per row and primitive probabilities in $\{1/5, 1/2, 4/5\}$ are swept over every budget up to the full table, giving 115 setting and budget pairs over 1,920 weighted tables. Table 9 compares each schedule with the Bellman optimum $V^{\star}_{\beta}$. The planner of Appendix B.2 attains it on 91.3% of the pairs, the parent-first order on 88.7%, and a favorable-head order on 90.4%. On a three-candidate, two-target setting with candidate probability $1/5$ and parent probability $4/5$ the replanning policy's expected cost is $358146/78125 = 4.5843$ cells against $390146/78125 = 4.9939$ for the parent-first order, 8.2% fewer. The source halfspace order is the unit-weight, unit-cost specialization of Algorithms 1 and 3 of Ghuge et al. (2025) with $\epsilon = 3/80$.

Table 8: The fixed-batch paired test on each Granite-4.2-8B round, 40 distinct cells at a duplication factor of 38.4.

	Unsettled	Required pairs		Decision		Complete
Round	cells	Positive	Negative	Drawn pairs	Per cell	table
1	4	2	10	UNDECIDED	UNDECIDED	does not improve
2	1	39	0	ACCEPT	UNDECIDED	improves
3	2	17	16	UNDECIDED	UNDECIDED	does not improve
4	4	51	0	UNDECIDED†	UNDECIDED	improves

†The drawn pairs stay undecided on this round because of one negative control pair of 512.

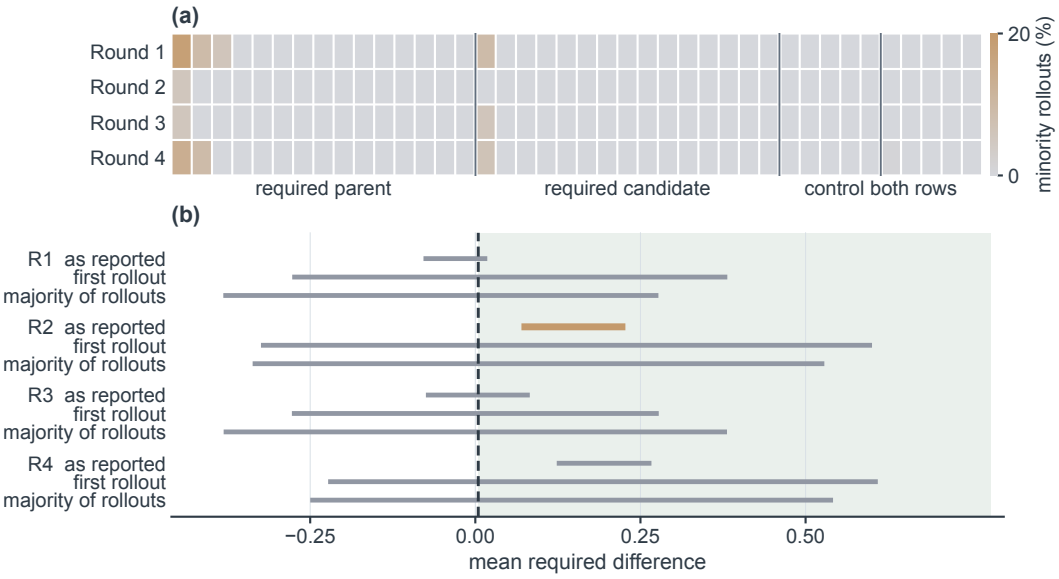

Figure 9: Audited Granite-4.2-8B cells. (a) Share of rollouts taking the minority score. (b) The required interval under three charging rules; the shaded side of the dashed line qualifies.

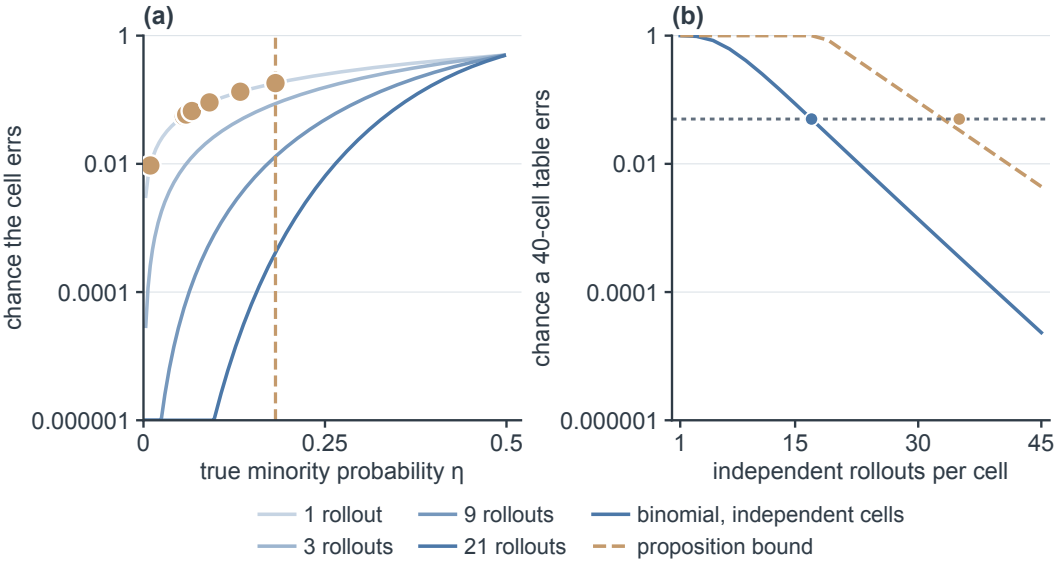

Figure 10: Repeated-rollout calculations under independent rollouts. (a) Cell error against the true minority probability η; dots are observed fractions. (b) Error of a 40-cell table at $\eta = 2/11$; dots mark the first counts below 0.05.

Table 9: Exact finite-model comparison over 115 setting and budget pairs. “At envelope” counts the pairs whose accept probability equals $V_\beta^\star$.

Schedule	At envelope	Share (%)	Largest gap
BranchCert-E read-plan replanning	105	91.3	$19/125$
Favorable-head order	104	90.4	$164/625$
Inverted-prior replanning	97	84.3	$3/5$
Parent-first order	102	88.7	$3/5$
Candidate-first order	58	50.4	$512/625$
Source halfspace order	57	49.6	$16/25$