AUDIT HORIZONS: HOW LONG CAN STATEFUL AGENTS ACT BEFORE REVIEW?

Anonymous authors
Paper under double-blind review

ABSTRACT

How many consecutive changes should a coding agent make before a trusted review? We introduce the *audit horizon*: the longest sequence of autonomous actions a fixed agent stack may take before trusted review, under a guarantee on regression risk. We then propose an audit-horizon qualification method: design data choose how deep and how many independent trajectories to run, and fresh trajectories determine the horizon through a valid risk-control test. We prove that one-shot evaluation cannot identify risk that accumulates along a trajectory. Certifying a nonzero audit horizon with high probability therefore requires persistent evidence. However, when earlier observations follow the same distribution regardless of the planned evaluation length, extending a run beyond the target horizon adds cost without strengthening its direct risk bound. On chained upgrades of Flask, Jinja2, and PyJWT with DeepSeek V4 Flash 0731, short-prefix qualification reaches the same audit horizon as full-length evaluation while reducing measured model cost by 53.6% and summed agent–environment time by 42.5%. Across a three-model policy comparison, sharing evaluation depth across packages achieves the highest observed horizon–cost score. Overall, we advocate that future safety metrics should reflect long-term behavior instead of only one-shot workloads.

1 INTRODUCTION

Coding agents modify repositories that subsequent actions inherit. A useful operating decision is therefore how many consecutive changes an agent may make before trusted review. We introduce the *audit horizon*: the longest action block accepted under a statistical guarantee on cumulative regression risk for a fixed model, scaffold, and workload. This connects evaluation to a concrete review interval while separating regression control from task completion.

Task-completion horizons characterize capability through human task duration (Kwa et al., 2025), and reliability metrics distinguish consistency, robustness, predictability, and safety (Rabanser et al., 2026). Persistent software benchmarks expose how agents build on their own changes (Lam et al., 2026; Orlanski et al., 2026). These measurements motivate an evidence-design question: how deeply should each evaluation run proceed, and how many independent runs are needed to support a review interval? Resetting after each action hides the effects of accumulated state; extending every run to maximum depth spends resources on actions beyond the decision.

We propose *audit-horizon qualification*. Design data select evaluation depth and replication; fresh trajectories preserve state within each prefix and support the risk test. Fixed and adaptive depth policies use the same statistical decision rule. The theory explains the role of persistent evidence and shows when extending a run beyond a target horizon adds cost without strengthening its direct risk bound.

On chained upgrades of Flask, Jinja2, and PyJWT with DeepSeek V4 Flash 0731, short prefixes return horizon two, matching full-length evaluation and independent validation, with 53.6% less measured model cost and 42.5% less agent–environment execution time. A 1,188-trajectory comparison across

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.

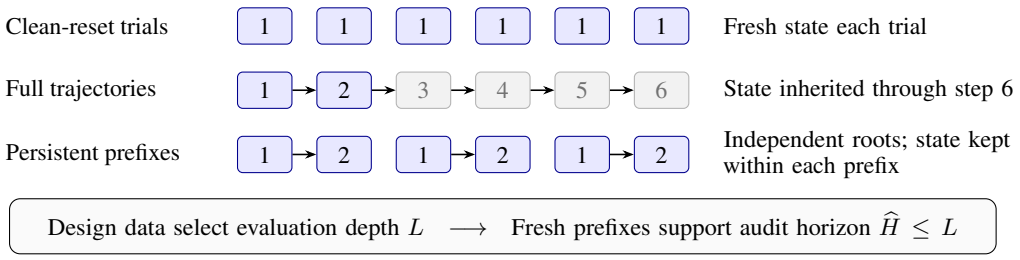

Figure 1: **Acquire evidence for the audit decision.**

DeepSeek V4.1 Flash, Qwen3.8 Flash, and GLM5.3 Flash finds identical outcomes for fixed and adaptive depth, with shared depth attaining the highest observed horizon–cost score.

- **An audit-horizon contract** linking a review interval to a bound on cumulative regression risk.
- **A qualification procedure** that selects prefix depth and replication and assesses the returned horizon using fresh trajectories.
- **Evidence requirements** characterizing reset-data ambiguity, necessary persistent information, and the direct evidence supplied by depth.
- **An empirical evaluation** of matched-horizon resource savings, depth-selection policies, and completed upgrades across three packages.

2 Related Work

Agent horizons and persistent evaluation. Task-completion horizons measure success against human task duration (Kwa et al., 2025). SWE-Chain, SWE-STEPS, SWE-CI, and SlopCodeBench measure upgrades, sequential evolution, maintenance, and degradation (Lam et al., 2026; Shastry et al., 2026; Chen et al., 2026; Orlanski et al., 2026). We use persistent evaluation to select a review interval under a cumulative risk constraint and study the resources needed to support that decision.

Calibrated stopping and intervention. KnowNo calibrates when language-model planners request human help (Ren et al., 2023). Conformal Language Modeling calibrates stopping and filtering when sampling answer sets (Quach et al., 2024); Conformal Thinking selects reasoning-exit thresholds under risk and efficiency objectives (Wang et al., 2026). ToolChain-CRC calibrates accepted trajectory losses and adds online intervention (Opoku & Banahene, 2026). These methods select runtime responses, stopping rules, or interventions. Our selected object is an autonomous action-block length; the design controls how far independent evaluation trajectories run before supporting that length.

Risk control and sample design. Risk-controlling prediction sets and Learn then Test provide finite-sample calibration, including ordered testing (Bates et al., 2021; Angelopoulos et al., 2021). Confidence sequences and active risk control support sequential validity and budgeted label selection (Howard et al., 2021; Xu et al., 2024). Statistical model checking tests probabilistic properties of simulated paths (Younes & Simmons, 2002; Legay et al., 2010). Our qualification rules build on these tools. The observation-design question is how trajectory depth and independent replication determine the evidence available for an audit horizon. Appendix A compares risk targets, intervention policies, and observation costs in more detail.

3 Problem

In this section, we define a number of fundamental concepts for our work.

System and risk. We fix a model, its prompts, scaffold, tools, workload distribution, initial-state law, and persistence rule. A choice is called *prespecified* when it is fixed before the corresponding outcomes are observed. A process law $\theta \in \Theta$ captures the remaining randomness from the model and

the environment under this setup. Let T be the first action at which a specified unacceptable event occurs. The cumulative risk is

$$r_h(\theta) = P_\theta(T \le h), \qquad r_0(\theta) = 0. \tag{1}$$

The event is absorbing. Once it happens, a later fix does not undo it. The finite horizon menu is $\mathcal{H} = \{0, \dots, H_{\max}\}$. Given a risk limit $\alpha \in (0, 1)$ and an error allowance $\delta \in (0, 1)$, evaluation returns $\widehat{H} \in \mathcal{H}$ such that

$$\sup_{\theta \in \Theta} P_\theta\{r_{\widehat{H}}(\theta) > \alpha\} \le \delta. \tag{2}$$

The outer probability covers evaluation data and procedure randomization. Deployment randomness is already included in r_h. A positive return allows one block of at most $\widehat{H}$ actions before trusted review. Horizon zero allows no autonomous action. The return is the largest horizon that the procedure can support.

Persistent prefixes and independent roots. A length-L prefix starts from a fresh draw of the prespecified initial state and keeps the state produced by each action. Its law is $P_\theta^{(L)}$. The *root* is the independent initial draw together with its trajectory. Steps within a root may depend on each other arbitrarily. Observation depth L controls data collection, while $\widehat{H}$ controls the permitted action block. A prefix of depth L provides direct evidence only for horizons $h \le L$. A run can end early after a confirmed event or an unresolved outcome.

Recorded outcomes. Let $Y_{i,h} = \mathbf{1}\{T_i \le h\}$ be the latent cumulative event for root i. The record separates measured regression, measured preservation, and unknown outcomes. For the upper-risk calculation we *upper-code* the record by setting $Z_{i,h} = 1$ from the first measured regression or unresolved outcome onward, and $Z_{i,h} = 0$ otherwise. Correctly observed outcomes and conservative missingness give $Y_{i,h} \le Z_{i,h}$. Regression and upgrade completion are recorded as separate outcomes.

Reset observations and workload strata. A reset experiment restores the specified clean state at its boundary. Its complete observation law is Q_θ. A one-action reset trial observes first-action behavior. Persistent prefixes also observe continuation from states generated by the agent. For a workload split into S prespecified strata with weights $w_s \ge 0$ and $\sum_s w_s = 1$, the target risk is $r_h = \sum_{s=1}^{S} w_s p_{s,h}$, where $p_{s,h}$ is the cumulative risk within stratum s. In software maintenance, strata are starting release windows, the persistent state is the repository written by the agent, the event is a regression in behavior, such as import failure.

4 METHOD

We propose a method for obtaining the audit horizon that separates evaluation design from risk assessment. The design chooses which prefixes to observe; their outcomes support a test that returns an audit horizon. This separation lets the design control evaluation cost while preserving the risk contract.

4.1 SAMPLING

Let $\mathcal{A}$ be a finite registry of prefix depths, root caps, and stopping policies. Independent design data select $\widehat{a} = A(D_{\mathrm{dev}}) \in \mathcal{A}$. These data estimate cumulative risk, prefix cost, and the probability of returning a positive horizon. Our policy comparison scores the projected horizon minus a price fixed from design data times token use (Section 6). Forecasts determine which prefixes to observe, and fresh qualification data supply the evidence for the final decision.

Fixed depth selects a depth and root cap from design data, then uses fresh batches with a cost-sensitive stopping rule. *Adaptive depth* can select the next depth after a completed batch; that depth is committed before any outcome in the next batch is observed. Both can stop when the expected gain from further evidence does not justify its forecast cost. The *shared-depth* comparator selects one depth across a model's package targets, retaining the same setting-specific stopping calculation. Section 6 specifies the finite registry and independent sample used to select between fixed and adaptive depth.

4.2 SAMPLE DIRECTED EVIDENCE

We now show how to compute statistical evidence for two sampling methods.

Fixed stratified samples. Suppose the allocation fixes $n_s \geq 1$ independent roots in each stratum. For a measured horizon h, define

$$\widehat{r}_h^{+} = \sum_{s=1}^{S} \frac{w_s}{n_s} \sum_{i=1}^{n_s} Z_{s,i,h}, \qquad V = \sum_{s=1}^{S} \frac{w_s^2}{n_s}. \tag{3}$$

A one-sided bound at error level ε is

$$\widetilde{U}_h(\varepsilon) = \min\left\{1, \widehat{r}_h^{+} + \sqrt{\frac{V \log(1/\varepsilon)}{2}}\right\}. \tag{4}$$

This is weighted Hoeffding concentration, with unknown outcomes upper-coded. It allows different stratum risks and conditions on an independently selected allocation. Set unobserved coordinates to one. Taking $U_h = \max_{j \leq h} \widetilde{U}_j(\delta/H_{\max})$ gives a simultaneous upper band. Alternatively, the ordered return rule below needs only pointwise level δ (Theorem 5.2).

Sequential balanced samples. The sequential design reads outcomes only after complete rounds containing one fresh root per equally weighted stratum. For a candidate horizon h, let n_h and K_h count roots and upper-coded events in rounds assigned depth at least h. Its evidence process is the uniform likelihood-ratio mixture

$$E_h = \frac{1}{\alpha} \int_0^{\alpha} \left(\frac{q}{\alpha}\right)^{K_h} \left(\frac{1-q}{1-\alpha}\right)^{n_h - K_h} dq. \tag{5}$$

Under the null $r_h \geq \alpha$, this is a nonnegative supermartingale at complete-round boundaries, provided fresh stratum outcomes are conditionally independent with the prespecified means. A skipped horizon receives no update. Predictable depth choices therefore allow different observation counts across horizons. We accept h if its process ever reaches $1/\delta$. Appendix B gives the balanced-round proof for heterogeneous strata.

4.3 COMPUTING THE ACCEPTED HORIZON

For fixed samples, we accept h when the appropriate upper bound is at most α. For sequential samples, an acceptance is permanent once its process crosses the threshold. Return the largest h for which every coordinate $1, \ldots, h$ is accepted. An unobserved coordinate is never accepted. This ordered rule uses the nested risk structure to control the selected horizon. A simultaneous band additionally covers the whole risk curve.

5 THEORY

The following results connect reset-data ambiguity, persistent information, and prefix depth to the audit-horizon decision.

Reset observations can identify the target only when all systems they cannot distinguish have the same persistent risk. Write $g_H(\theta) = (r_1(\theta), \ldots, r_H(\theta))$.

Theorem 5.1 (Reset-equivalent systems)**.** *The vector g_H is identified by Q_θ exactly when it is constant on every set of systems with the same reset law. If $Q_{\theta_0} = Q_{\theta_1}$ and $\|g_H(\theta_0) - g_H(\theta_1)\|_\infty = d$, every reset-only estimator has worst-case expected sup-norm error at least $d/2$, at every sample size.*

This holds because the complete reset data have the same distribution under the two systems, so their different targets cannot both be estimated accurately. A startup/survivor model makes the distinction concrete. Let q be the first-action failure probability and p the conditional failure probability at each later surviving action. Then

$$r_h(q,p) = 1 - (1-q)(1-p)^{h-1}. \tag{6}$$

One-action reset trials observe q but not p, so equal reset behavior can coexist with different cumulative regression risks.

Theorem 5.2 (Validity of horizons)**.** *Condition on independent design data. Suppose that, for each horizon h with $r_h > \alpha$, the probability that h is ever accepted is at most δ. Return the last coordinate in the contiguous accepted prefix. Then $P_\theta\{r_{\widehat{H}}(\theta) > \alpha\} \le \delta$. The premise holds for fixed-time pointwise bounds with the stated coverage, or for valid nonnegative evidence supermartingales crossing $1/\delta$.*

The argument is a containment. Fix the first horizon j with $r_j > \alpha$ under the true law. Any unsafe return must accept j, an event with probability at most δ. The containment allows dependence among the horizon tests, and averaging over design data preserves the guarantee. For the fixed stratified design, (4) supplies the required pointwise coverage. For the balanced sequential design, (5) supplies time-uniform control at the allowed readout points. The construction uses concentration and ordered testing (Hoeffding, 1963; Angelopoulos et al., 2021; Howard et al., 2021).

Validity alone is compatible with always returning horizon zero. To rule this out, call a procedure $(1-\eta)$-*useful* at horizon h under a safe system θ_0 when $P_{\theta_0}(\widehat{H} \ge h) \ge 1-\eta$, with $0 \le \eta < 1-\delta$.

Theorem 5.3 (Necessity of long-term observation)**.** *Suppose $Q_{\theta_0} = Q_{\theta_1}$ and $r_h(\theta_0) < \alpha < r_h(\theta_1)$. Any procedure satisfying (2) that is $(1-\eta)$-useful at h under θ_0 and whose data consist of any number of reset observations and M independent length-$L \ge h$ prefix runs under a fixed design satisfies*

$$M\, D_{\mathrm{KL}}\Big(P_{\theta_0}^{(L)} \,\Big\|\, P_{\theta_1}^{(L)}\Big) \ge \mathrm{kl}(1-\eta, \delta). \tag{7}$$

Here D_{KL} is Kullback–Leibler divergence and kl is its Bernoulli form; Appendix B fixes the definitions and endpoint conventions. When the two systems share the reset law, reset observations drop out of (7) entirely: the required information must come from the M persistent runs themselves. In the survivor model (6), that information concerns p, exactly what clean-start data cannot supply. Appendix B gives the proof.

Proposition 5.4 (Invariance of direct prefix evidence)**.** *Fix n independent roots and a target horizon h. Suppose every assigned length $L \ge h$ induces the same observation law through h, including its missing-outcome rule. We call this* prefix consistency. *Any fixed piece of evidence that is a measurable function only of those first-h observations has the same distribution for every $L \ge h$. If expected execution cost is nondecreasing in L, continuing beyond h weakly increases cost without improving that evidence.*

The proposition follows because the inputs to the evidence have the same distribution. Under the coupling that gives both lengths the same first-h records, its value is identical pathwise. It applies to the direct bounds at a fixed root count and explains why, once a horizon is targeted, additional independent roots yield more information per unit cost than the actions beyond it. Design data select which horizons to observe.

6 Experiments

We evaluate how much persistent evidence is needed to support an audit horizon on chained software upgrade scenario from SWE-Chain (Lam et al., 2026). The evaluation asks three questions: how much cost short prefixes save at an equal returned horizon, how depth-selection policies compare across models and packages, and how much upgrade work is done within the returned horizon.

6.1 Workloads and agent configurations

The workloads comprise six-upgrade release windows from Flask, Jinja2, and PyJWT. Each action applies one upgrade, and the resulting repository is preserved. External tests measure regression and upgrade completion. Table 1 specifies the agent, target population, evaluator, and sampling design for each comparison. The direct resource comparison uses DeepSeek V4 Flash 0731 in OpenCode 1.18.21 and an equal-weight target of 18 windows across the three packages. The policy comparison crosses DeepSeek V4.1 Flash, Qwen3.8 Flash, and GLM5.3 Flash with the three packages. Each model–package setting targets six equally weighted windows and uses a builder followed by one review, with fresh conversation context at each upgrade.

Property	Direct resource comparison	Policy comparison
Models	DeepSeek V4 Flash 0731	DeepSeek V4.1 Flash; Qwen3.8 Flash; GLM5.3 Flash
Target	18 windows, equal weight across packages	Six equal-weight windows per model–package setting
Evaluator	Benchmark installation and tests	Fresh candidate installation, reference test overlay, implementation invariance check
Sampling	Fixed stratified root counts	Balanced six-root rounds
Risk analysis	Simultaneous weighted bounds	Ordered sequential tests
Resources	Measured dollars, tokens, execution time	Policy-consumed tokens and physical archive inventory

Table 1: Evaluation configurations. Each target has maximum depth six. The direct resource comparison measures independently executed designs; the policy comparison exposes requested prefixes from a common trajectory archive.

6.2 Evaluation policies and data allocation

Design data select evaluation depth and number of rounds before qualification. For the direct resource comparison, a finite search over depth and root count selects 100 depth-two roots. The full-length comparator uses 60 depth-six roots, and an independent validation sample contains 200 depth-six roots. All samples cover the same 18-window target; the analysis uses each sample's actual stratum counts. The agent, event contract, and resource accounting are common to both evaluation designs.

The policy comparison includes *fixed depth*, *adaptive depth*, *shared depth*, full 6-depth, and depth one. Fixed depth selects a depth and root cap for each setting from design data. Adaptive depth can choose a new depth after a completed batch. Shared depth selects one depth across a model's three packages, retaining setting-specific caps and stopping. All policies receive the same environment otherwise. Full depth and depth one can opt out under the same stopping calculation. Candidate depths are one through six, root caps are 12, 24, 48, and 60, and assigned work is limited to 360 upgrade opportunities per setting.

Each setting has four disjoint samples: 12 roots for fitting risk and cost forecasts, 12 for selecting the fixed or adaptive variant, 60 for qualification, and 48 for validation. Together these contain 1,188 primary roots, each assigned at most six upgrades. One balanced round contains a fresh root from every window, in randomized order. The independent selection sample chooses adaptive depth only if its mean forecast return is no smaller than fixed depth's and its mean conservative token-cost ratio is below one. This rule selects fixed depth; adaptive depth is also evaluated as a policy comparison. The shared depths selected from design data are two for DeepSeek, two for Qwen, and one for GLM. Appendix D specifies the design search, forecasts, and data allocation.

6.3 Metrics and statistical analysis

All targets use $H_{\max} = 6$, risk limit $\alpha = .40$, and error probability $\delta = .10$. Fixed stratified samples use the simultaneous weighted bounds in (3)–(4); balanced samples use the ordered sequential procedure at complete-round boundaries. Each target receives its own guarantee. Held-out validation assesses the horizon returned during qualification on an independent sample.

We report returned horizon, evaluation resources, and complete nonregressing upgrades through the returned horizon. The policy objective averages $H - \lambda C$ equally across the nine settings, where C is qualification token consumption and each setting's price λ is fixed from design data. Native token totals count prompt and completion usage once, including cache and reasoning components. Missing usage receives lower and upper accounting bounds. Execution time is the sum of agent–environment durations.

For the policy comparison, each policy receives only its requested prefixes from the same 6 deep trajectory history collection. We measure its consumed prefix cost and report the full physical resources for fitting, selection, qualification, and validation separately. Section 8.1 discusses the recorded execution conditions and the interpretation of these measurements.

Design	Roots	Depth	$\widehat{H}$	U_2	Tokens (M)	Cost / hours
Prefix	100	2	2	.3751	676.1	$16.86 / 93.0
Full length	60	6	2	.3762	1,544.1	$36.35 / 161.7
Validation	200	6	2	.3449	n/a	n/a

Table 2: Direct resource comparison with DeepSeek V4 Flash 0731 on the 18-window target. U_2 is the simultaneous stratified upper bound. Resources cover qualification; time is summed agent–environment hours and tokens are reported native usage. Validation supplies an independent risk estimate.

		Qualified H			Tokens (M)		
Model	Task	Fixed	Shared	H_V	Fixed	Shared	J_F
DeepSeek	Flask	1	2	2	63.4–64.9	147.7–150.5	36
DeepSeek	Jinja2	2	2	2	219.0–221.9	219.0–221.9	10
DeepSeek	PyJWT	2	2	2	112.9–117.2	112.9–117.2	7
Qwen	Flask	1	2	2	32.5	77.4	28
Qwen	Jinja2	2	2	3	101.0–102.1	101.0–102.1	5
Qwen	PyJWT	1	1	2	26.0–27.1	83.2–84.3	12
GLM	Flask	0	0	1	0.0	0.0	n/a
GLM	Jinja2	1	1	2	13.7	13.7	11
GLM	PyJWT	2	1	2	146.7	16.0	1

Table 3: Policy comparison across all nine settings. Fixed and adaptive depth share identical executions. H_V is the held-out support horizon. J_F counts complete, nonregressing roots through the fixed-depth horizon (48 roots per setting). Token intervals reflect accounting uncertainty.

7 Results

7.1 Cost of supporting an audit horizon

Short-prefix evaluation reaches horizon two at substantially lower measured cost in the direct resource comparison (Table 2). The prefix, full-length, and validation samples all return H2, with upper risk bounds .3751, .3762, and .3449, all below the risk limit .40. The prefix design reduces measured model cost by 53.6%, execution time by 42.5%, and token usage by 56.2% relative to full length.

Most of the cost is due to suffix execution: upgrades 3-6 consume 59.9% of the full-length design's measured model cost and 63.0% of its execution time. Those positions do not change its H2 bound. This result illustrates Proposition 5.4: once the decision concerns horizon two, later actions add cost without supplying another observation of the first two actions.

Matched-horizon comparisons also show large token savings across the policy choices. In Qwen/Jinja2, fixed and adaptive depth reach H2 using 67.8–69.8% fewer qualification tokens than full length; shared depth matches them. In Qwen/PyJWT, fixed and adaptive depth reach H1 using 67.4–69.2% fewer tokens than shared depth; depth one matches them. Figure 2 shows both measured resource reductions and the horizon–cost outcomes across all nine settings.

7.2 Depth-selection policies across models and packages

Fixed and adaptive depth return H1 or H2 in eight of nine settings and opt out in GLM/Flask (Table 3). The adaptive rule keeps its selected depth on every observed path, therefore delivering identical results. Held-out numerical assessment supports every positive return from these policies and the baselines.

Shared depth returns H2 instead of H1 in DeepSeek/Flask and Qwen/Flask, and H1 instead of H2 in GLM/PyJWT; it ties the other six returns. It achieves the highest mean prespecified horizon–cost score among the compared policies: 1.3511–1.3525, versus 1.1500–1.1512 for fixed and adaptive depth, .8409–.8417 for depth one, and .2081–.2089 for full length. Its observed score is about 17.5% higher than fixed depth's. These ranges reflect request-level token-accounting uncertainty.

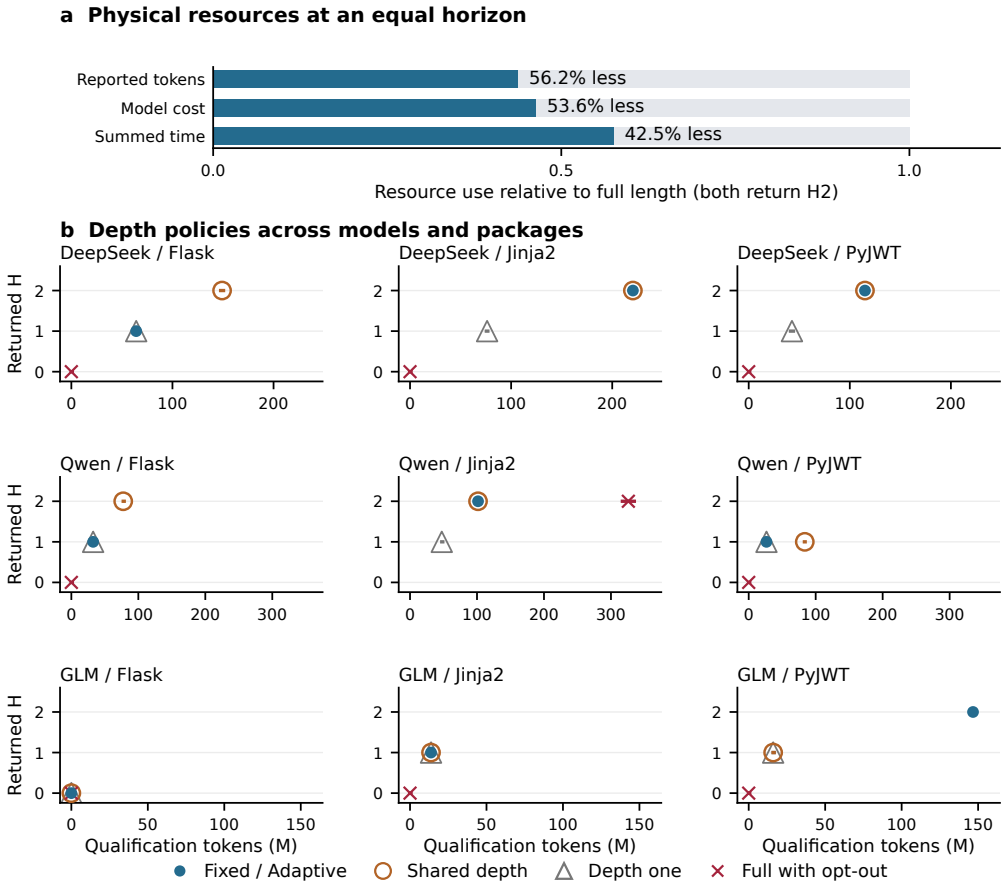

Figure 2: **Resource savings and depth selection.** (a) Independently collected prefixes and full trajectories with DeepSeek V4 Flash 0731 return the same H2. (b) Fixed, adaptive, shared, depth-one, and full-depth policies are compared across nine model–package settings; overlapping markers indicate matching outcomes and horizontal segments show token-accounting bounds. Axis ranges are shared within a model row. Panel (b) measures requested qualification-prefix consumption from the trajectory archive.

The comparison shows that simple depth choices capture the observed benefit. A longer return also has a measurable price: in GLM/PyJWT, fixed and adaptive depth reach H2 for 146.7M tokens, while shared depth reaches H1 for 16.0M. Full length opts out in eight settings and returns H2 in Qwen/Jinja2 under the common cost-sensitive stopping rule.

7.3 Total evaluation resources

The fixed and adaptive qualification paths each consume 715.2–726.0M native tokens across the nine settings. Physical collection uses 4.292–4.688B tokens for fitting, 3.641–3.788B for policy selection, 18.163–19.547B for qualification, and 9.941–11.121B for validation. This archive supplies the prefixes and suffixes needed to compare the policies and assess their returns. Appendix D reports the full resource inventory alongside the per-policy consumption.

8 Conclusion and limitations

Audit-horizon qualification turns stateful-agent evaluation into a concrete review decision. It returns the number of consecutive actions a fixed agent may take under a regression-risk guarantee. The procedure separates evaluation design from risk assessment. Design data guide how deeply and how often to evaluate, while fresh trajectories support the statistical guarantee. Reset trials cannot

determine persistent risk, reliably certifying a nonzero audit horizon requires sufficient persistent evidence, and extending a run beyond a targeted horizon adds cost without strengthening its direct risk bound.

On chained upgrades of Flask, Jinja2, and PyJWT, short-prefix evaluation reaches horizon two with less measured model cost and execution time than full-length evaluation. Across the three-model policy comparison, fixed and adaptive depth produce identical horizons and costs, while shared depth achieves the highest observed horizon–cost score. Together, the theory and experiments identify evaluation depth and independent replication as joint design choices. Resources spent on suffixes can instead support more observations of the horizon being assessed. We advocate that future safety metrics for autonomous agents reflect long-term behavior rather than one-shot workloads alone.

8.1 Limitations

The guarantee assumes fresh roots follow a fixed process law. Budget-admission failures affect 275/432 validation roots in the three-model archive and can alter execution or couple trajectories through shared spending. Upper-coding missing outcomes does not correct such interventions; the numerical checks describe the recorded archive. Uninterrupted deployment requires prospective validation.

Regression control does not imply upgrade completion or protection against untested harms. The DeepSeek V4 Flash 0731 evaluator can retain target-only implementation files. These observations do not identify causal effects on completion (Appendix D).

The direct comparison measures physical qualification resources; seven interrupted steps may contain unmeasured token usage. Its depth-two truncation is retrospective. Model-wise policy costs measure consumed prefixes from a depth-six archive, rather than realized campaign savings. The comparison uses one outer seed; token ranges express accounting uncertainty, not sampling confidence. Full length shares the other policies' right to opt out.

The evidence covers three packages at $\alpha = .40$ and $\delta = .10$ per target, without joint coverage across settings. Returned horizons and policy rankings depend on the sample and do not establish population optima. Each guarantee covers one action block; review need not restore the evaluated initial-state law. Stratified and sequential validity are established by the paper proofs.

AI use statement

We used large language models to assist with literature discovery, experimental code implementation, analysis and visualization of results, theoretical exposition, and manuscript polish and revision. The authors reviewed the AI-assisted contributions and take full responsibility for the methods, results, claims, and final manuscript.

Ethics statement

Experiments use public software packages, published benchmark tasks, isolated compute environments, and external test evaluators. They involve no human subjects or production infrastructure. The audit decision is tied to a specified event and risk contract, with horizon zero as the fallback.

Reproducibility statement

Appendices B–D describe the mathematical arguments, sampling designs, depth-selection rules, simulation settings, token-accounting formulas, and reported aggregate results. They specify the design risk curves, selection thresholds, provider limits, and outcome definitions used in the evaluation configurations.

References

Anastasios Nikolas Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2021. URL https://arxiv.org/abs/2110.01052.

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016. doi: 10.1073/pnas.1510507113. URL https://doi.org/10.1073/pnas.1510507113.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *arXiv preprint arXiv:2101.02703*, 2021. URL https://arxiv.org/abs/2101.02703.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24(2):265–272, 1953. doi: 10.1214/aoms/1177729032. URL https://doi.org/10.1214/aoms/1177729032.

Jialong Chen, Xander Xu, Hu Wei, Chuan Chen, and Bing Zhao. SWE-CI: Evaluating agent capabilities in maintaining codebases via continuous integration. *arXiv preprint arXiv:2603.03823*, 2026. URL https://arxiv.org/abs/2603.03823.

Rémy Degenne and Wouter M. Koolen. Pure exploration with multiple correct answers. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/60cb558c40e4f18479664069d9642d5a-Abstract.html.

Aurélien Garivier and Emilie Kaufmann. Nonasymptotic sequential tests for overlapping hypotheses applied to near-optimal arm identification in bandit models. *Sequential Analysis*, 40(1):61–96, 2021. doi: 10.1080/07474946.2021.1847965.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: Improving safety despite intentional subversion. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16295–16336. PMLR, 2024. URL https://proceedings.mlr.press/v235/greenblatt24a.html.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

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://arxiv.org/abs/1810.08240.

Shervin Javdani, Yuxin Chen, Amin Karbasi, Andreas Krause, Drew Bagnell, and Siddhartha Srinivasa. Near optimal bayesian active learning for decision making. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pp. 430–438. PMLR, 2014. URL https://proceedings.mlr.press/v33/javdani14.html.

Qirui Jin, Lingching Tung, Kenan Li, Qiyang Shi, Yushi She, Huanzhong Jia, Harrison Zhao, Kejing Xia, Zhenbang Du, Jiaxin Pei, Zhenyu Zhang, Zhen Qi, Yuyan Duan, Wenke Lee, and Zijian Jin. ChainSWE: Benchmarking coding agents on multi-bug software maintenance. *arXiv preprint arXiv:2607.02606*, 2026. URL https://arxiv.org/abs/2607.02606.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, et al. Measuring AI ability to complete long software tasks. *arXiv preprint arXiv:2503.14499*, 2025. URL https://arxiv.org/abs/2503.14499.

Man Ho Lam, Chaozheng Wang, Hange Liu, Jingyu Xiao, Haau-sing Li, Jen-tse Huang, Terry Yue Zhuo, and Michael R. Lyu. SWE-Chain: Benchmarking coding agents on chained release-level package upgrades. *arXiv preprint arXiv:2605.14415*, 2026. URL https://arxiv.org/abs/2605.14415.

Axel Legay, Benoît Delahaye, and Saddek Bensalem. Statistical model checking: An overview. In *Runtime Verification*, pp. 122–135. Springer, 2010. doi: 10.1007/978-3-642-16612-9_11. URL https://doi.org/10.1007/978-3-642-16612-9_11.

Charles F. Manski. *Partial Identification of Probability Distributions*. Springer, 2003. doi: 10.1007/b97478. URL https://link.springer.com/book/10.1007/b97478.

Jeffery Opoku and David Banahene. ToolChain-CRC: Conformal risk control for agentic AI under retrieval and tool-use drift. *arXiv preprint arXiv:2606.18467*, 2026. URL https://arxiv.org/abs/2606.18467.

Gabriel Orlanski, Devjeet Roy, Alexander Yun, Changho Shin, Alex Gu, Albert Ge, Dyah Adila, Nicholas Roberts, Frederic Sala, and Aws Albarghouthi. SlopCodeBench: Benchmarking how coding agents degrade over long-horizon iterative tasks. *arXiv preprint arXiv:2603.24755*, 2026. URL https://arxiv.org/abs/2603.24755.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.10193.

Stephan Rabanser, Sayash Kapoor, Peter Kirgis, Kangheng Liu, Saiteja Utpala, and Arvind Narayanan. Towards a science of AI agent reliability. *arXiv preprint arXiv:2602.16666*, 2026. URL https://arxiv.org/abs/2602.16666.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 661–682, 2023. URL https://proceedings.mlr.press/v229/ren23a.html.

K. N. Ajay Shastry, Ganesh Senrayan, Shrey Satapara, Pranoy Panda, and Chaitanya Devaguptapu. Beyond isolated tasks: A framework for evaluating coding agents on sequential software evolution. *arXiv preprint arXiv:2604.03035*, 2026. URL https://arxiv.org/abs/2604.03035.

Toby Shevlane, Sebastian Farquhar, Ben Garfinkel, Mary Phuong, Jess Whittlestone, Jade Leung, Daniel Kokotajlo, Nahema Marchal, Markus Anderljung, Noam Kolt, Lewis Ho, Divya Siddarth, Shahar Avin, Will Hawkins, Been Kim, Iason Gabriel, Vijay Bolina, Jack Clark, Yoshua Bengio, Paul Christiano, and Allan Dafoe. Model evaluation for extreme risks. *arXiv preprint arXiv:2305.15324*, 2023. URL https://arxiv.org/abs/2305.15324.

Wenwen Si, Sooyong Jang, Insup Lee, and Osbert Bastani. Conformal constrained policy optimization for cost-effective LLM agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. URL https://arxiv.org/abs/2511.11828.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009. doi: 10.1007/b13794. URL https://link.springer.com/book/10.1007/b13794.

Andrew J. Wagenmaker, Max Simchowitz, and Kevin Jamieson. Task-optimal exploration in linear dynamical systems. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10641–10652. PMLR, 2021. URL https://proceedings.mlr.press/v139/wagenmaker21a.html.

Xi Wang, Anushri Suresh, Alvin Zhang, Rishi More, William Jurayj, Benjamin Van Durme, Mehrdad Farajtabar, Daniel Khashabi, and Eric Nalisnick. Conformal thinking: Risk control for reasoning on a compute budget. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.03814.

Ziyu Xu, Nikos Karampatziakis, and Paul Mineiro. Active, anytime-valid risk controlling prediction sets. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6eb05d8bc6bd7bb6868c64b5802125bd-Abstract-Conference.html.

Håkan L. S. Younes and Reid G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *Computer Aided Verification*, volume 2404 of *Lecture Notes in Computer Science*, pp. 223–235. Springer, 2002. doi: 10.1007/3-540-45657-0_17. URL https://doi.org/10.1007/3-540-45657-0_17.

A Extended related work

A.1 Horizon metrics and evolving software

Task-completion horizons describe the duration of tasks an agent can solve at a given success rate, using human completion time as the difficulty axis (Kwa et al., 2025). The audit horizon instead counts consecutive actions under a cumulative regression-risk constraint. The reliability framework of Rabanser et al. (2026) further separates consistency, robustness, predictability, and safety. Our contract makes one aspect of this broader reliability problem operational by specifying a review interval.

Persistent software benchmarks supply the evolving states on which this decision depends. SWE-Chain evaluates release-level upgrades that inherit the agent's previous code (Lam et al., 2026); it supplies our experimental workloads. SWE-STEPS evaluates sequential changes and repository maintainability (Shastry et al., 2026), SWE-CI studies maintenance through continuous integration (Chen et al., 2026), and SlopCodeBench measures extension robustness as requirements evolve (Orlanski et al., 2026). ChainSWE adds sequential, dependent bug fixes within a shared repository (Jin et al., 2026). Our evaluation uses independent trajectories to assess which prefix lengths and root counts support the review decision.

A.2 Stopping, human assistance, and trajectory risk

KnowNo uses conformal calibration to construct action sets and request human help when a language-model planner faces ambiguity, including in multi-step tasks (Ren et al., 2023). Its intervention decision depends on the current planning context. Audit-horizon qualification produces a population-level action allowance before the deployment block begins. The allowance specifies when trusted review must occur.

Conformal Language Modeling samples candidate responses until a calibrated stopping rule returns an answer set, with filtering to remove unsuitable candidates (Quach et al., 2024). Conformal Thinking calibrates reasoning exit thresholds to control error while reducing computation (Wang et al., 2026). Conformal Constrained Policy Optimization learns model and tool orchestration under a conformal reliability objective (Si et al., 2026). Their efficiency objectives concern the deployed system's response generation or reasoning. Here the agent stack is fixed, and the resource choice concerns the trajectories collected to evaluate it. Prefix length determines which horizon outcomes become observable; replication supplies independent observations of those outcomes.

ToolChain-CRC makes a complete tool-using trajectory the calibration object and combines step-level risk information with acceptance and intervention (Opoku & Banahene, 2026). Its base loss is the product of a trajectory failure label and an acceptance indicator, so intervention contributes zero accepted loss. The base guarantee controls this loss in expectation over calibration and test trajectories; its drift extension also provides a high-probability bound under additional assumptions. Our contract controls the probability, over evaluation data, of selecting an action block whose population cumulative regression risk exceeds the tolerance. It uses audited prefix outcomes to choose a horizon, whereas ToolChain-CRC calibrates a risk-score threshold and runtime escalation. AI control similarly connects monitoring and trusted intervention to deployment protocols, with intentional subversion as its threat model (Greenblatt et al., 2024); broader model risk evaluation connects evidence to deployment decisions (Shevlane et al., 2023).

A.3 Risk calibration and statistical model checking

Risk-controlling prediction sets calibrate a predictor to satisfy a population-loss constraint with high probability (Bates et al., 2021). Learn then Test expresses configuration selection as multiple hypothesis testing and provides fixed-sequence procedures (Angelopoulos et al., 2021). Our ordered horizon rule applies this principle to nested cumulative events: an unsafe return entails accepting the first unsafe coordinate. The stratified and sequential constructions specify the valid tests for the sampling schedules used here.

Statistical model checking verifies probabilistic temporal properties using simulation and hypothesis testing (Younes & Simmons, 2002; Legay et al., 2010). Regression by a specified action count is

such a path property. Searching across action counts turns these tests into a review-horizon decision. Confidence sequences provide time-uniform inference (Howard et al., 2021), and active, anytime-valid risk control combines adaptive label queries with a label budget (Xu et al., 2024). A label-query policy decides which outcomes to observe on arriving examples. Our depth policy decides how far to execute a fresh trajectory, revealing a nested set of horizon outcomes. Balanced-round sampling preserves the target stratum weights during sequential assessment.

A.4 Identification and decision-directed sampling

Identification asks whether the observation law determines the target; partial identification studies targets compatible with the same evidence (Blackwell, 1953; Manski, 2003; Bareinboim & Pearl, 2016). Our reset-equivalent construction applies this distinction to persistent risk, and the information bound uses a two-point testing argument (Tsybakov, 2009). Reset observations contribute no information for distinguishing the constructed systems; persistent trajectories do.

Decision-directed learning, task-specific exploration, and identification with multiple acceptable answers select observations according to the outputs they support (Javdani et al., 2014; Wagenmaker et al., 2021; Degenne & Koolen, 2019; Garivier & Kaufmann, 2021). Our sampling choices are trajectory depth and independent replication. Under prefix consistency, continuing an observed trajectory beyond a fixed target horizon leaves the direct prefix statistic unchanged. This links evaluation cost to the selected review interval.

B Proofs and statistical interfaces

Divergences and notation. All logarithms are natural. For probability laws P, Q, write

$$D_{\mathrm{KL}}(P\|Q) = \int \log\left(\frac{dP}{dQ}\right) dP$$

when P is absolutely continuous with respect to Q, and set the divergence to infinity otherwise. For Bernoulli probabilities a, b, define

$$\mathrm{kl}(a,b) = a\log\frac{a}{b} + (1-a)\log\frac{1-a}{1-b},$$

Terms with zero numerator contribute zero; a positive numerator and zero denominator give infinite divergence. Total variation is $\mathrm{TV}(P,Q) = \sup_A |P(A)-Q(A)|$, with the supremum over measurable events. The squared Hellinger distance is $d^2_{\mathrm{H}}(P,Q) = 1 - \int \sqrt{dP\,dQ}$.

B.1 Reset identification

Proof of Theorem 5.1. Identification means that there is a function f on attainable reset laws with $g_H(\theta) = f(Q_\theta)$. Such a function exists exactly when g_H is constant whenever the reset laws agree. For the error floor, any number of independent reset observations has the same joint law under θ_0 and θ_1. Let $\widehat{g}$ be any estimator of g_H from those observations. Under the common data law, the triangle inequality gives

$$d \le \|\widehat{g} - g_H(\theta_0)\|_\infty + \|\widehat{g} - g_H(\theta_1)\|_\infty.$$

Taking expectations implies that at least one of the two expected errors is at least $d/2$. The argument includes independent estimator randomization. □

Define the maximal risk separation between reset-equivalent systems whose length-L prefix laws are close in Hellinger distance:

$$\omega_{H,L}(\varepsilon) = \sup_{\substack{Q_{\theta_0}=Q_{\theta_1} \\ d_{\mathrm{H}}(P^{(L)}_{\theta_0}, P^{(L)}_{\theta_1}) \le \varepsilon}} \|g_H(\theta_0) - g_H(\theta_1)\|_2.$$

For fixed independent samples consisting of reset trials and $M \ge 1$ length-L persistent roots, a standard two-point argument gives

$$\inf_{\widehat{g}_H} \sup_{\theta} \mathbb{E}_\theta \|\widehat{g}_H - g_H(\theta)\|_2^2 \ge \frac{1}{16}\omega_{H,L}\left(\frac{1}{\sqrt{8M}}\right)^2. \tag{8}$$

Indeed, the squared Hellinger distance of the persistent product laws is at most M times its one-root value, hence at most $1/8$. Their total variation is at most $1/2$. Reset factors coincide, and Le Cam's bound supplies the constant $1/16$ (Tsybakov, 2009). Apply the argument to approximating pairs if the supremum is not attained.

B.2 Fixed stratified bounds

Condition on the design data and all outcome-independent allocation randomness. Let $a_{s,i} = w_s/n_s$. Independence of the latent roots and Hoeffding's inequality give

$$P\left\{\sum_{s,i} a_{s,i}\mathbb{E}Y_{s,i,h} - \sum_{s,i} a_{s,i}Y_{s,i,h} > t\right\} \le \exp(-2t^2/V), \quad V = \sum_s w_s^2/n_s.$$

The expectation is r_h. Replacing Y by $Z \ge Y$ only increases the weighted empirical risk estimate. Setting $t = \sqrt{V\log(1/\varepsilon)/2}$ proves the pointwise bound (4) for the specified stratum means. A union bound at $\varepsilon = \delta/H_{\max}$ yields simultaneous coverage. Taking a cumulative maximum and assigning one to unobserved coordinates preserves that coverage. Averaging over independent design data and allocation randomness completes the unconditional statement.

For a partially observed target, let O be the set of observed strata and let $m_O = \sum_{s\notin O} w_s$ be the unobserved target mass. With $n_s \ge 1$ for $s \in O$, define

$$\widehat{r}^{+}_{h,O} = m_O + \sum_{s\in O}\frac{w_s}{n_s}\sum_{i=1}^{n_s} Z_{s,i,h}, \qquad V_O = \sum_{s\in O}\frac{w_s^2}{n_s}.$$

The observed weighted sum satisfies the same concentration inequality, and the unobserved risk contributes at most m_O. Thus

$$U_{h,O}(\varepsilon) = \min\left\{1, \widehat{r}^{+}_{h,O} + \sqrt{\frac{V_O\log(1/\varepsilon)}{2}}\right\} \tag{9}$$

is a pointwise upper bound at error level ε. For $O = \varnothing$, $m_O = 1$ and $V_O = 0$, giving $U_{h,O} = 1$.

B.3 Balanced-round sequential evidence

Fix a horizon h and suppress its subscript. Let $\mathcal{F}_{b-1}$ contain all observations available before round b. At that complete round, let $Y_s \in \{0,1\}$ be the independent latent outcomes in its S equally weighted strata, with means p_s conditional on the prior observed history. Assume the prespecified target means remain fixed and $S^{-1}\sum_s p_s \ge \alpha$. For any $q \in [0,\alpha]$, the Bernoulli likelihood-ratio factor is

$$f_q(y) = \left(\frac{q}{\alpha}\right)^y\left(\frac{1-q}{1-\alpha}\right)^{1-y} = 1 + \frac{\alpha-q}{\alpha(1-\alpha)}(\alpha - y).$$

Its expectation in stratum s is nonnegative. Conditional independence and the arithmetic–geometric mean inequality imply

$$\mathbb{E}\left[\prod_{s=1}^{S} f_q(Y_s)\,\middle|\,\mathcal{F}_{b-1}\right] = \prod_{s=1}^{S}\left[1 + \frac{\alpha-q}{\alpha(1-\alpha)}(\alpha - p_s)\right] \le 1.$$

The inequality depends on the mean of the stratum risks p_s. If $Z_s \ge Y_s$, each factor can only decrease, so the inequality also holds for the recorded upper-coded outcomes. Decisions about whether to observe this horizon are made before the round. When the horizon is not observed in a round, its evidence process remains unchanged. Products over observed complete rounds are therefore nonnegative supermartingales. Integrating the fixed q processes against Uniform$(0,\alpha)$ gives (5) by nonnegativity and conditional integration. Ville's inequality bounds the probability of ever crossing $1/\delta$ by δ under the null.

The filtration includes all prior evidence used for sampling. Fresh roots retain their prespecified conditional laws given this history, depths are selected before each round, and evidence is read at complete-round boundaries. These conditions supply the conditional expectation inequality above.

B.4 Ordered horizon selection

Proof of Theorem 5.2. If every horizon is safe, an invalid return is impossible. Otherwise let j be the first horizon with $r_j > \alpha$. It is fixed under the true law. Returning an unsafe horizon requires accepting every coordinate through j, so it requires acceptance of j. By assumption this event has probability at most δ, conditional on independent design data. Integrating over design data proves the claim. Dependence among the horizon tests does not change this containment argument. □

For fixed-time pointwise upper bounds, acceptance of j means $U_j \le \alpha < r_j$, a coverage error of probability at most δ. For the sequential construction, acceptance requires the null evidence process to cross $1/\delta$ at some readout point. Both satisfy the theorem's premise. Simultaneous coverage of the whole numerical risk curve is obtained through the separate band construction.

B.5 Direct-prefix invariance

Proof of Proposition 5.4. Let $O_{1:n,1:h}$ be the observed first-h records and let C_h be the fixed measurable evidence map. Prefix consistency makes the law of these inputs identical for every assigned $L \ge h$. Their image under C_h consequently has the same law. In a common-prefix coupling it is the same value. Nondecreasing expected cost in L gives the resource comparison. Strictly positive expected suffix cost gives a strict cost increase. □

B.6 Proof of the persistent-evidence lower bound

Proof of Theorem 5.3. Map the data to $\phi = \mathbf{1}\{\widehat{H} \ge h\}$. Under θ_0 its success probability is at least $1-\eta$. Under θ_1 it is at most δ, since any return at least h is unsafe. Data processing and monotonicity of binary relative entropy give a KL divergence between the joint data laws of at least $\mathrm{kl}(1-\eta, \delta)$. Independence makes this divergence a sum. The reset observations contribute zero divergence, and the persistent terms sum to the stated left side. □

With an additional independent design sample of law D_θ, the joint-data divergence includes $D_{\mathrm{KL}}(D_{\theta_0} \| D_{\theta_1})$ alongside the persistent-root term. This additive accounting applies the same data-processing argument to the complete observation set.

C Two observation channels for persistent risk

In the startup/survivor model of (6), q is the first-action failure probability and p is the conditional failure probability at each later action, given survival. For $q, p \in (0,1)$,

$$\nabla r_h(q,p) = \left((1-p)^{h-1}, (1-q)(h-1)(1-p)^{h-2}\right).$$

The derivative with respect to p is zero at $h=1$ and positive for $h \ge 2$. For example, $q = .10$ with $p = .05$ gives $r_2 = .145$ and $r_6 \approx .304$, whereas the same q with $p = .30$ gives $r_2 = .370$ and $r_6 \approx .849$. The reset laws agree, while at risk limit .40 their largest safe horizons in the six-action menu are six and two respectively.

Sampling and estimation. Consider two independent sampling channels. The reset channel gives $X_1, \dots, X_m \overset{\text{iid}}{\sim} \mathrm{Bernoulli}(q)$. The survivor channel gives $V_1, \dots, V_{M_s} \overset{\text{iid}}{\sim} \mathrm{Bernoulli}(p)$, where each observation records a later-action failure conditional on reaching that action, using a separate independent trajectory. Here M_s counts survivor-channel observations. Define

$$\widehat{q} = \frac{1}{m}\sum_{i=1}^{m} X_i, \qquad \widehat{p} = \frac{1}{M_s}\sum_{j=1}^{M_s} V_j, \qquad \widehat{r}_h = r_h(\widehat{q}, \widehat{p}),$$

with $m, M_s \ge 1$. The two sample proportions have variances $q(1-q)/m$ and $p(1-p)/M_s$ and covariance zero. The linear expansion

$$r_h(q,p) + \partial_q r_h(q,p)(\widehat{q}-q) + \partial_p r_h(q,p)(\widehat{p}-p)$$

has variance

$$V_h^{(1)} = \frac{(1-p)^{2h-2}q(1-q)}{m} + \frac{(1-q)^2(h-1)^2(1-p)^{2h-4}p(1-p)}{M_{\mathrm{s}}}. \tag{10}$$

For fixed h and interior q, p, this is the leading delta-method variance of the plug-in estimator as both sample sizes grow with a positive finite ratio. Reset observations determine the first term; independent survivor observations determine the second.

Observation depth. A direct bound at $h = 2$ uses the first two observations from each root. For the same n prefix-consistent roots, assignments of depth two and depth six therefore give identical inputs to that bound. Their cost difference is the expenditure on actions after step two.

D Experimental methods and complete results

This appendix specifies the target populations, evaluation designs, statistical procedures, resource measurements, and observation inclusion rules used in Section 6.

D.1 System configuration and execution environment

Agent and inference settings. Both evaluation configurations use OpenCode 1.18.21. For the three-model policy comparison, the agent image includes Node.js 20.20.2, and model inference is served through OpenRouter with the model/provider pairs in Table 7 pinned and fallback disabled. Requests use temperature 1.0, top-p .95, and a maximum output allowance of 65,536 tokens. DeepSeek V4.1 Flash and GLM5.3 Flash request low reasoning effort; Qwen3.8 Flash enables reasoning without an effort-level override. Each upgrade uses a builder and one review with fresh conversation context, while repository state persists across upgrades within the root.

Software environment. The three-model execution uses Linux x86-64 virtual machines with Debian 12. The workflow runtime uses CPython 3.11.13; statistical analysis pins NumPy 2.4.6 and SciPy 1.17.1. Agent and test containers use Debian Bookworm-based images, with Python 3.9 package-specific environments. Flask and Jinja2 tests use pytest 6.2.5; PyJWT tests use pytest 8.3.5. Flask dependency profiles separate legacy, intermediate, and recent release families. Candidate installation and reference-test overlays use the profile selected for the release transition. Container images are fixed by content digest and checked before execution.

Compute resources and parallelism. For the three-model comparison, each trajectory runs in a KVM-accelerated virtual machine with eight virtual CPUs, 16 GiB of RAM, and a 40 GiB writable disk overlay. Host allocations reserve eight CPU cores and 20–24 GiB of RAM. Model inference runs on the provider infrastructure; no local GPU is allocated. Scheduler-managed job arrays parallelize roots. This execution parallelism is separate from the six-root balanced rounds used for statistical updates. Primary jobs have a 12-hour scheduler limit; the separate repeat qualification runs described below use a 24-hour limit.

Timeouts and request recovery. Agent phases have a 1,800-second timeout; test evaluation uses a 180-second limit. Virtual-machine startup has a 600-second readiness deadline. Provider connections use a 20-second connection timeout and a 1,850-second request limit; the transport-recovery configuration also imposes a 120-second low-throughput cutoff. At most three attempts are admitted for the same model input. Retries require a recognized no-output failure or an explicitly diagnosed terminal error, and all attempts remain in resource accounting. The outcome-inclusion rules in Appendix D.7 apply to interrupted trajectories and repeated attempts.

D.2 Workload targets and data allocation

Direct resource measurements. The DeepSeek V4 Flash 0731 target assigns weight $1/18$ to each starting release window. Table 4 gives the collected samples. The generator repeats each window, distributes the remaining roots without replacement, and shuffles their order. The analysis conditions on these outcome-independent counts and uses the weighted bound in (4).

Sample	Roots	Assigned depth	Package counts
Prefix qualification	100	2	(32,34,34)
Full-length comparison	60	6	(19,22,19)
Held-out validation	200	6	(66,66,68)

Table 4: Direct resource comparison: data allocation. Package counts are ordered as Flask, Jinja2, and PyJWT. All three samples cover all 18 target windows.

Model-wise policy evaluation. The three-model, three-package policy comparison has nine targets. Each target is the equal-weight mixture of six release windows. A round contains one fresh root per window; only complete rounds supply a readout. The outer seed is 20260918. Table 5 gives the four disjoint samples. All physical roots are assigned a maximum depth of six. The policy evaluator exposes the requested prefix of each assigned root, after recording the depth decision.

Sample	Roots per target	Total roots	Role
Fitting	12	108	Estimate risk and prefix cost
Policy selection	12	108	Select fixed or adaptive depth
Qualification	60	540	Evaluate depth policies
Validation	48	432	Check the original returned horizons

Table 5: Policy comparison: data allocation. Each complete round contains six roots per model–package target.

D.3 Selection of evaluation depth and run count

Candidate designs. The design registry contains (L, n, n_v), with depth $L \in \{1, \dots, 6\}$, validation count $n_v \in \{100, 150, 200\}$, and qualification counts $n = 20, 30, \dots$ up to $\lfloor 360/L \rfloor$. The maximum feasible count is also included when it lies between grid points. This gives 249 candidates. The comparison design has 60 depth-six roots; validation has depth six for every candidate.

Design risk and cost models. Table 6 lists the three cumulative-risk curves fixed before qualification for the direct resource comparison. The first pools 113 persistent design roots; the second gives equal weight to the three package-specific curves from that design sample; the third uses a separate 36-root reference sample. The first and third curves are the respective count vectors $(16, 25, 40, 45, 52, 54)/113$ and $(7, 15, 20, 24, 25, 29)/36$. The equal-package curve is the coordinatewise average of $(4, 9, 17, 18, 22, 24)/55$, $(10, 11, 16, 19, 21, 21)/46$, and $(2, 5, 7, 8, 9, 9)/12$.

Design curve	1	2	3	4	5	6
113-root pooled	0.1416	0.2212	0.3540	0.3982	0.4602	0.4779
Equal-package	0.1523	0.2731	0.4134	0.4690	0.5355	0.5476
36-root reference	0.1944	0.4167	0.5556	0.6667	0.6944	0.8056

Table 6: Cumulative-risk curves used for depth and run-count selection. Columns give risk through each action. Displayed values are rounded to four decimals.

Projected dollars and summed agent–environment seconds are calculated as

$$K_x(n, L; r) = n \left(c_{x,1} + c_{x,+} \sum_{j=1}^{L-1} (1 - r_j) \right), \qquad x \in \{\$, t\}.$$

The first-action coefficients are $c_{\$,1} = .06800005$ and $c_{t,1} = 1138.335$; later-action coefficients are $c_{\$,+} = .08426698$ and $c_{t,+} = 1259.69805$. These coefficients were estimated from the design executions. The survival factors $1 - r_j$ price an action only when its predecessor prefix has survived.

Simulation and selection rule. Each candidate and risk curve receives 50,000 simulation repetitions. A repetition draws independent method, comparison, and validation samples from the first-failure

distribution $(r_1, r_2-r_1, \ldots, r_6-r_5, 1-r_6)$. The simulator computes returns with one-sided Clopper–Pearson bounds at coordinate error $.10/6$ and risk limit $.40$. Let H_a, H_b, H_v be the simulated method, comparison, and validation returns. A repetition counts as a resource advantage when

$$\mathcal{E}_a = \{0 < H_a \le H_v\} \cap \big[\{H_b > H_v\} \cup \{H_a > H_b\} \\ \cup \{H_a = H_b,\ K_{\$,a} \le .85K_{\$,b},\ K_{t,a} \le .85K_{t,b}\}\big].$$

Here $K_{x,a}$ and $K_{x,b}$ are the scenario-projected costs above. For every risk curve, the candidate must achieve an empirical advantage probability of at least .83 and a lower endpoint of its 95% Wilson interval of at least .80. The implementation uses $z = 1.9599639845$ in that interval. Candidates are ordered by total assigned work $nL + 360 + 6n_v$, then by nL, then by decreasing minimum Wilson lower endpoint across the three curves. The base simulation seed is 2026090301, with a separate deterministic random-number stream for each candidate and risk curve.

One candidate passes: $(L, n, n_v) = (2, 100, 200)$. Its minimum empirical advantage probability is .84666 and minimum Wilson lower endpoint is .84348. This simulator selects the evaluation design. The collected sample is analyzed with its actual stratified counts using (4).

Policy forecasts and selection. Design roots estimate equal-window cumulative risk and the upper endpoint of mean prefix token cost. The price is $\lambda = .02/\overline{C}_2$, where $\overline{C}_2$ is the fitted mean upper cost of a depth-two prefix in millions of tokens. Thirty-two simulated future paths drive continuation forecasts, and 128 simulations score each candidate depth and root cap. The caps are 12, 24, 48, and 60. The separate selection sample supplies 64 resampled paths per target for fixed/adaptive comparison, using paired sample indices for the upper and lower cost endpoints. The empirical-mixture simulation supplies the evaluation forecast; scientific qualification uses the balanced-round schedule above.

The policy comparison includes fixed depth, adaptive depth, a model-specific shared depth across packages, depth one, and full depth six. Their common certifier, stopping rule, price, and root order are specified in Section 6. The conservative selection ratio uses the adaptive policy's upper token-cost endpoint divided by the fixed policy's lower endpoint in each setting, then averages these ratios over settings. Adaptive depth is selected only when this mean is below one and its mean forecast horizon is no smaller.

D.4 Statistical procedures and stopping

The direct resource comparison uses outcome-independent stratified counts. At each horizon, (4) gives a weighted upper bound, with coordinate error $.10/6$ for the simultaneous band. The ordered alternative uses pointwise error .10. Table 8 reports both. The design simulator uses the binomial calculation specified above; the measured stratified samples use their actual window counts.

For model-wise policy evaluation, each readout follows a complete six-root round. The likelihood-ratio mixture in (5) updates only those horizons observed in the round. An acceptance is permanent after the evidence process crosses $1/\delta = 10$. The return is the largest contiguous accepted horizon. Depth is selected before observing a batch, and all policies share the same forecast-based stopping calculation. Appendix B gives the balanced-round and ordered-selection arguments. Each of the nine model–package targets uses $\alpha = .40$ and $\delta = .10$ separately.

Validation applies the risk assessment to a disjoint held-out sample and checks the original qualified horizons. The reported accounting ranges propagate request-level usage uncertainty through the token totals and horizon–cost objective.

D.5 Resource accounting

A complete usage record contains nonnegative integer prompt, completion, and total counts, with the total equal to the sum of the first two. Its token contribution is the prompt count plus completion count, including cache and reasoning components once. For a request without complete usage, the lower contribution is zero and the upper contribution is the sum of its model/provider prompt and completion limits in Table 7. Included prior attempts are counted once. If $\mathcal{K}$ and $\mathcal{U}$ are the sets of

requests with known and unknown usage, the token interval is

$$\left[\sum_{j\in\mathcal{K}}(P_j+C_j),\quad \sum_{j\in\mathcal{K}}(P_j+C_j)+\sum_{j\in\mathcal{U}}(P^{\max}_{m(j)}+C^{\max}_{m(j)})\right].$$

Here P_j, C_j are reported counts and $m(j)$ identifies the request's fixed model/provider pair. Requests use the listed provider with fallback disabled. The completion bound is taken from the provider catalog; the requested output limit is 65,536. The catalog limits in Table 7 are used in the upper-endpoint calculation.

Model	Provider	Prompt limit	Completion limit
DeepSeek V4.1 Flash	DeepSeek	1,048,576	384,000
Qwen3.8 Flash	Alibaba	983,616	131,072
GLM5.3 Flash	Z.ai (FP8)	1,048,576	131,072

Table 7: Per-request token limits used for missing-usage accounting in the three-model policy comparison.

D.6 Complete results

Table 8 gives the H2 results under the weighted stratified analysis. Both the simultaneous-band and ordered-return procedures return H2 for all three samples.

Sample	Weighted empirical risk	Simultaneous upper	Ordered upper
Prefix qualification	.2315	.3751	.3392
Full-length comparison	.1898	.3762	.3296
Held-out validation	.2437	.3449	.3196

Table 8: Direct resource comparison: risk estimates and upper bounds at H2. The simultaneous band uses coordinate error $.10/6$; ordered selection uses pointwise error $.10$.

Retrospectively truncating the same 60 full-length roots at depth two returns H2 for $\$14.57$ and 59.9 summed hours, versus $\$16.86$ and 93.0 hours for the selected 100-root prefix design. This truncation isolates recorded suffix cost while holding root identities fixed.

Table 9 reports each policy's horizon and consumed prefix tokens. Table 10 gives the physical sample inventory across all four data stages.

Setting	Fixed / Adaptive	Shared	Depth one	Full
DeepSeek/Flask	1, 63.4–64.9	2, 147.7–150.5	1, 63.4–64.9	0, 0.0
DeepSeek/Jinja2	2, 219.0–221.9	2, 219.0–221.9	1, 75.5–76.9	0, 0.0
DeepSeek/PyJWT	2, 112.9–117.2	2, 112.9–117.2	1, 41.3–44.2	0, 0.0
Qwen/Flask	1, 32.5	2, 77.4	1, 32.5	0, 0.0
Qwen/Jinja2	2, 101.0–102.1	2, 101.0–102.1	1, 47.4	2, 317.1–334.9
Qwen/PyJWT	1, 26.0–27.1	1, 83.2–84.3	1, 26.0–27.1	0, 0.0
GLM/Flask	0, 0.0	0, 0.0	0, 0.0	0, 0.0
GLM/Jinja2	1, 13.7	1, 13.7	1, 13.7	0, 0.0
GLM/PyJWT	2, 146.7	1, 16.0	1, 16.0	0, 0.0

Table 9: Complete policy-prefix results. Each entry lists the returned horizon and native tokens (M). Zero means the policy opted out. Fixed and adaptive depth have identical executions.

Stage	Roots	Native tokens (billions)
Fitting	108	4.292–4.688
Policy selection	108	3.641–3.788
Qualification	540	18.163–19.547
Validation	432	9.941–11.121

Table 10: Physical sample inventory for the three-model policy comparison. Intervals use the request-level accounting rule in Appendix D.5.

D.7 Evaluator controls and observation inclusion

The direct resource comparison uses the benchmark installation procedure. The model-wise policy comparison installs each saved candidate in a fresh container, overlays the reference test files, and checks that candidate implementation files remain unchanged during evaluation. A reference-release control evaluates the maintainer-written target release against its required test suite. All 42 reference configurations passed. Nine reference-release controls evaluate six consecutive upgrades.

An *admission-failure event* is a model-request attempt for which the shared financial admission mechanism records a denial. This category includes failed spending reservations and accounting-lock failures. A *no-contact attempt* explicitly records that the provider was not contacted. Counts of attempts include retries. A *step accounting flag* records a missing or empty phase request log, an unresolved request, or a denied phase admission. These request and step categories are counted separately from measured test outcomes.

Admission failures occur in 23/108 fitting roots, 41/108 selection roots, 155/540 qualification roots, and 275/432 validation roots. Validation contains 5,609 denied attempts, including 1,462 within the positive fixed/adaptive returned horizons across 170/384 roots. Thirty validation roots have only no-contact attempts. The qualification prefixes requested by fixed and adaptive depth contain 55 denials across 32/120 roots; restricting to their returned horizons gives 50 denials across 30/120 roots. Table 11 gives all eight validation batches.

Batch	Roots with denials	Denied attempts	Only no-contact roots
1	47	749	0
2	43	420	0
3	0	0	0
4	0	0	0
5	26	240	0
6	51	1,141	15
7	54	1,568	6
8	54	1,491	9

Table 11: Validation request-admission records by sequential batch, with 54 roots per batch. The final column counts roots whose recorded attempts all explicitly show no provider contact.

Qualification contains 2,495 measured steps: 576 complete upgrades, 1,841 incomplete upgrades, and 78 unknown outcomes. Among incomplete steps, 1,631 have a completed final test suite. There are 146 first-step incomplete outcomes with a successful final agent process and no step accounting flag. Of the 576 complete steps, 249 already passed all required target tests on entry; 327 closed a measured target-test gap.

Observation inclusion. The direct resource comparison includes all 60 full-length roots, covering all 18 target windows. A 33-root subset with package counts $(19, 14, 0)$ covers ten windows and has unobserved target mass $8/18$. Its upper bound in (9) is therefore at least $8/18 > .40$. For the model-wise comparison, each primary scientific root is counted once. Seven validation setup attempts fail before model work and are retried under the same root identity. Ten repeat qualification trajectories are excluded from the primary comparison. Resource totals include the prior attempts associated with included roots under the request-accounting rule above.