SELF-CALIBRATING INTERRUPTION OF AGENT DOOM LOOPS VIA RISK-CONTROLLED E-PROCESSES

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-model agents run in harness loops that call an API-served policy and execute tools, so every iteration is billed. Much of that cost buys nothing: agents fall into *doom loops*, spending long after success has become improbable. Turn caps, token budgets and repetition heuristics kill salvageable episodes, per-step judges add a model call per step, and activation probes need internals that an API does not expose. The target is undefined: the conditional success probability is a zero-drift martingale, so no rule can wait for it to decline. Because a controller inspects every step, fixed-sample calibration is either invalid, realizing a 14.80% false-kill rate at a nominal 5%, or crippled by a horizon union bound that cuts cost by only 7.38%. Least recognized, an interruption policy censors its own labels: the episodes it kills are exactly those whose uninterrupted outcomes identify its risk. We present CIRCE, which defines a doom loop as a *value-stall* (low salvage probability with vanishing information yield per dollar) and certifies it with a test supermartingale over a log-derived stall indicator, bounding by α the probability of ever halting a salvageable trajectory under adaptive inspection, with no horizon penalty and asymptotically optimal delay. Its central mechanism is *randomized reprieve*: a firing certificate halts with probability $1-\epsilon$ and otherwise releases the episode to natural termination with known propensity ϵ, restoring identification and yielding a doubly robust salvage-loss estimator with a time-uniform confidence sequence and $\tilde{O}(K^{2/3})$ self-calibration regret over K deployment rounds. On software-engineering, terminal and web-navigation suites and a near-saturated tool-calling control with `glm-5.2`, CIRCE cuts cost 36.35% at a 0.38 pp pass@1 difference and 0.52% false kills, using 34 log-derived features and no model internals, extra calls, or GPUs. Under a mid-stream model and difficulty shift every tested baseline breaches its risk constraint by up to 0.088 unnoticed, while scheduled reprieve caps violation at 0.015 and flags it within two rounds at 2.4% exploration spend.

1 INTRODUCTION

An agent harness is a loop: assemble context, call a policy over an API, execute the returned tool calls, append the observations, repeat. Every turn of that loop is billed, and a large share of the bill buys nothing, because agents fall into *doom loops* (re-running the same failing test, re-reading the same file, re-proposing a patch already rejected) and keep spending long after any realistic chance of success has gone. Deployed harnesses do interrupt, with turn caps, token budgets and n-gram repetition kills, and the literature adds per-step judges and probes on hidden activations. None works well, and the reasons are more interesting than the symptom.

Three obstacles explain why. *The target has never been written down.* The natural candidate, $V_t = \mathbb{P}[Y = 1 \mid \mathcal{F}_t]$, is a bounded martingale in the harness's filtration, so its expected change is zero at every step: a rule that waits for predicted decline never fires, and one that trades zero gain against cost fires immediately. What distinguishes a doom loop is that the agent stops generating information about its outcome while continuing to spend. *The number of looks is data-dependent.* A controller inspects every step, so fixed-sample calibration is the wrong argument: ignoring repeated looks inflates the false-kill rate to 14.80% at a nominal 5%, and union-bounding the horizon cuts cost by only 7.38% (Table 2). *An interruption policy censors its own labels.* For every episode it kills, the uninterrupted outcome is never realized, so the risk the controller must bound becomes non-identified

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.

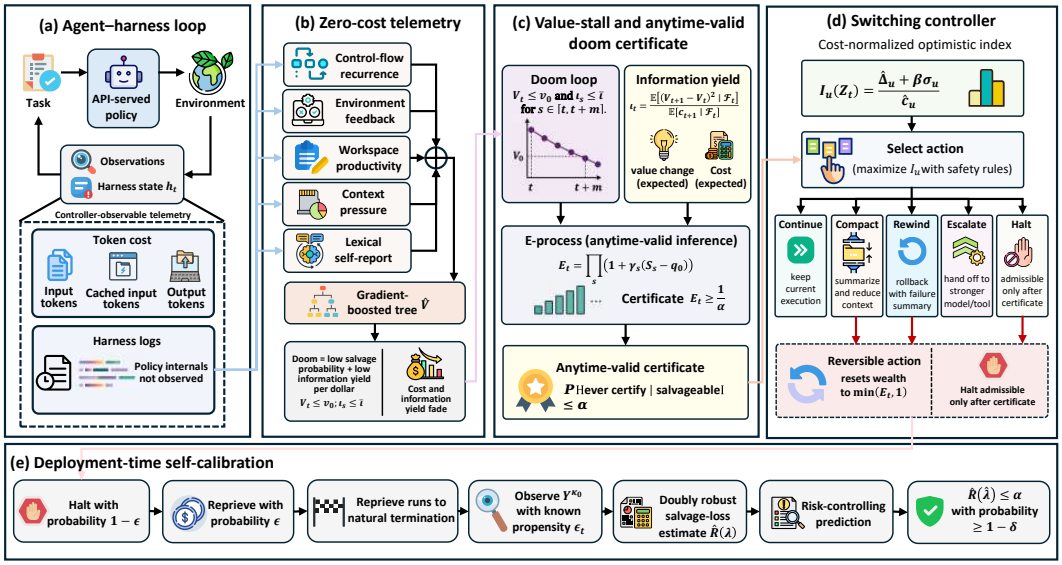


Figure 1: **Overview of CIRCE.** CIRCE observes only harness logs, converts zero-cost telemetry into a value-stall doom certificate via an anytime-valid e-process, and uses the certificate to gate irreversible halting. Reversible actions may recover the trajectory, while randomized reprieve preserves counterfactual labels for deployment-time self-calibration.

once it is switched on, and a model or task shift degrades it silently; every baseline we test breaches its constraint by up to 0.088 after a mid-stream shift without noticing (Table 3).

CIRCE (Figure 1) addresses all three while staying *black-box* (no internals, logits or gradients), *free at the margin* (every feature is a by-product of harness logs) and *self-calibrating* (valid after it starts intervening). Randomizing the irreversible action is not a regularizer; it is what keeps the risk estimate a function of observable data once the controller is live.

Contributions. We (i) give an estimable definition of a doom loop and prove an exact budget identity (the total information an agent can still produce about its own outcome is $V_t(1 - V_t)$), from which a cheap myopic halting rule follows with an explicit sufficient condition (§3.2); (ii) certify doom with an e-process, giving anytime-valid false-kill control with no horizon penalty and asymptotically optimal delay (§3.4), lifted to distribution-free population risk control over a configuration grid (§3.5); (iii) introduce randomized reprieve, which buys back identification of a counterfactual risk and yields a doubly robust estimator with $\tilde{O}(K^{2/3})$ self-calibration regret (§3.6); and (iv) evaluate under a cache-aware billing model on four suites including a negative control, reporting where the guarantee is loose, where transfer degrades, and where the windowed version of the myopic rule fails as the theory predicts.

2 RELATED WORK

Interrupting and budgeting agents. Agent harnesses built on tool-use loops (Yao et al., 2022b; Shinn et al., 2023; Wang et al., 2023; Schick et al., 2023; Wang et al., 2024; Packer et al., 2023a), including software-engineering agents (Packer et al., 2023b; Wang et al., 2025), are evaluated on suites where cost is now a first-class axis (Kapoor et al., 2024; Jimenez et al., 2024; Zhou et al., 2024; Yao et al., 2022a; Deng et al., 2023; Liu et al., 2024; Yao et al., 2024), and cost-aware routing and early exit reduce spend per call (Chen et al., 2023; Schuster et al., 2022). Deployed stoppers are heuristic (turn caps, token budgets, repetition kills, context backstops). Research alternatives either add a judge call per step (Zheng et al.) or read hidden activations and self-knowledge (Alain & Bengio, 2016; Burns et al., 2022; Azaria & Mitchell, 2023; Kadavath et al., 2022); the first is expensive under a cost constraint and the second is unavailable to a harness calling an API-served policy. Test-time compute allocation (Snell et al., 2025) asks how much to spend on a promising trajectory; we ask when to stop spending on a hopeless one, and how to certify that decision.

Stopping, switching and anytime-valid inference. Stopping a stochastic process at the right time is classical (Wald, 1992; Page, 1954; Brown, 1972; Peskir & Shiryaev, 2006; Sutton et al., 1998), and a non-binary intervention set makes the problem optimal switching (Hamadène & Jeanblanc, 2007). What is specific here is that the value process is a martingale in the observable filtration, so the monotone-case argument must be restated in terms of the *realized* stopping value (§3.2). Test supermartingales and e-processes give error control at arbitrary data-dependent stopping times (Ville, 1939; Shafer, 2021; Grünwald et al., 2020; Ramdas et al., 2023), with betting constructions and time-uniform empirical-Bernstein bounds supplying wealth processes and confidence sequences (Jiang & Yu, 2024; Howard et al., 2021; Maurer & Pontil, 2009); split conformal prediction (Vovk et al., 2005) is the fixed-sample alternative whose failure under repeated looks we quantify, and adaptive or weighted conformal variants address distribution shift but not self-censoring (Tibshirani et al., 2019; Gibbs & Candes, 2021). Distribution-free risk control over a configuration grid follows Angelopoulos et al. (2025) and Bates et al. (2021), which to our knowledge have not been applied to an online controller whose own actions censor the risk it must control.

Identification under self-censoring. Randomizing a treatment to retain a known propensity and estimating a counterfactual with a doubly robust estimator is standard in causal inference and off-policy evaluation (Horvitz & Thompson, 1952; Robins et al., 1994; Dudík et al., 2011; Swaminathan & Joachims, 2015; Chernozhukov et al., 2018), and regret of order $K^{2/3}$ is the signature of partial feedback (Langford & Zhang, 2007; Auer et al., 2002; Lattimore & Szepesvári, 2019). Our contribution is to notice that an interruption controller *is* a self-censoring treatment, and to make the exploration budget an explicit scheduled parameter of the harness.

3 METHOD

3.1 INTERRUPTION AS CONSTRAINED SWITCHING CONTROL

Agent-harness system and cost. A task $x \sim \mathcal{D}$ is executed by a policy π inside a harness g: at step t the harness builds a context $g(h_t)$, the model emits $a_t \sim \pi(\cdot \mid g(h_t))$, the environment returns o_t, and the harness updates $h_{t+1} = \mathcal{T}(h_t, a_t, o_t)$. The harness-observable filtration $\mathcal{F}_t = \sigma(x, a_{1:t}, o_{1:t}, h_{1:t})$ excludes everything internal to π; the trajectory terminates at a stopping time T, assumed finite almost surely, with a verifiable binary $Y \in \{0, 1\}$ from the benchmark's own validator, never from a model judge. Step t costs $c_t = p_{\text{in}} n_t^{\text{in}} + p_{\text{cache}} n_t^{\text{cache}} + p_{\text{out}} n_t^{\text{out}}$ at the granularity the provider bills, with $C_{s:t} = \sum_{u=s}^{t} c_u$. Separating cached from uncached input is not cosmetic: when $p_{\text{cache}} \ll p_{\text{in}}$ the marginal cost of a step is dominated by *output* tokens, and tuning under a uniform-token model instead moves the mean halt step by 6.1 steps and raises spend by 12.09% (Appendix A.13).

Controller and objective. CIRCE is a stochastic map $\kappa_\lambda : \mathcal{Z} \to \Delta(\mathcal{U})$ with $\lambda \in \Lambda$, acting on $Z_t = \phi(\mathcal{F}_t)$ and choosing from $\mathcal{U} = \{\text{CONT}, \text{COMPACT}, \text{REWIND}(k), \text{ESCALATE}, \text{HALT}\}$: COMPACT compresses context, REWIND(k) restores workspace and context to a checkpoint k steps back and re-seeds the agent with a failure summary, ESCALATE raises reasoning effort, and HALT terminates the episode returning the best partial artifact. Prior work treats interruption as the binary decision $\{\text{CONT}, \text{HALT}\}$; once the set is non-binary the problem is optimal switching rather than stopping, and the actions have incomparable cost-value profiles that a scalar doom score cannot arbitrate. Writing Y^κ, C^κ for outcome and cost under κ and $\kappa_\varnothing$ for the uninterrupted harness,

$$\min_{\lambda \in \Lambda} \mathbb{E}[C^{\kappa_\lambda}] \quad \text{s.t.} \quad R(\lambda) \coloneqq \mathbb{E}[Y^{\kappa_\varnothing} - Y^{\kappa_\lambda}] \le \alpha. \tag{1}$$

We call $R(\lambda)$ the *salvage loss*: the net drop in success rate that the controller causes relative to the uninterrupted harness, whose main source is halting episodes that would have succeeded. It is a *counterfactual* functional, which is exactly why naive post-deployment estimation of it fails (§3.6).

3.2 DOOM LOOPS AS VALUE-STALL, AND AN EXACT INFORMATION BUDGET

Definition 1 (Salvage process, information yield, doom loop)**.** *$V_t \coloneqq \mathbb{P}[Y = 1 \mid \mathcal{F}_t]$ is the probability that the episode succeeds if allowed to run to natural termination; $\iota_t \coloneqq \mathbb{E}[(V_{t+1} - V_t)^2 \mid \mathcal{F}_t] / \mathbb{E}[c_{t+1} \mid \mathcal{F}_t] = \Delta\langle V \rangle_{t+1} / \mathbb{E}[c_{t+1} \mid \mathcal{F}_t]$ is its* information yield *per dollar; and a trajectory is in an $(\bar{\iota}, v_0, m)$-*doom loop *at t if $V_t \le v_0$ and $\iota_s \le \bar{\iota}$ for all $s \in [t, t+m]$.*

By the tower property V is a bounded $\mathcal{F}_t$-martingale with zero drift, so "V is decreasing" is not a definition; what distinguishes a doom loop is that the agent ceases to generate information about its outcome while spending. Definition 1 says the outcome is unlikely *and* the next m steps are predicted to be uninformative about whether that changes. It is strictly more general than repetition-based detectors (an agent can emit never-repeating actions and still be in a value-stall), and it is estimable, because $\Delta\langle V\rangle$ is identified from complete calibration trajectories through a fitted $\hat{V}$. It also supports an exact accounting identity, which licenses a cheap local rule.

Proposition 2 (Information budget and myopic halting)**.** *Let* $W_t \coloneqq \mathbb{P}[Y = 1 \mid \mathcal{F}_t,\ \textsc{Halt}\ \textit{at}\ t]$ *be the success probability of the best partial artifact at* t*. Then*

$$\mathbb{E}\big[\langle V\rangle_T - \langle V\rangle_t \,\big|\, \mathcal{F}_t\big] \;=\; V_t(1 - V_t), \tag{2}$$

so the total information the agent can still produce about its own outcome is fixed by the current salvage probability alone. Consequently, if $\iota_s \le \bar{\iota}$ *for all* $s \ge t$*, then* $\mathbb{E}[C_{t+1:T} \mid \mathcal{F}_t] \ge V_t(1 - V_t)/\bar{\iota}$*, and under the* η*-penalized value* $W_\tau - \eta C_{1:\tau}$*, halting at* t *is at least as good as continuing to natural termination whenever*

$$\bar{\iota} \;\le\; \eta\,(1 - V_t), \qquad \textit{for which } \bar{\iota} \le \eta(1 - v_0) \textit{ is sufficient when } V_t \le v_0. \tag{3}$$

The proof is short (Appendix A.1). It repairs a gap that invalidates the implicit justification of every score-decay stopper: because V is a martingale, a one-step look-ahead written in terms of $\mathbb{E}[\Delta V_{t+1} \mid \mathcal{F}_t]$ is identically zero and, once cost is charged, would halt at $t = 1$, so the rule must be written in terms of the *realized* stopping value W, with the link back to the observable stall signal running through equation 2 rather than through the drift. Its scope is explicit: equation 3 compares halting now with running to natural termination and assumes the low-yield condition holds for the whole remaining horizon, whereas a deployed controller checks a window of m steps. That patience parameter, not the certificate, is the operative safeguard against delayed breakthroughs, and §4.4 reports the stratum where the predicted failure occurs.

3.3 Zero-cost telemetry

$\phi(\mathcal{F}_t)$ is computed entirely from harness logs (no extra model call, no judge, no hidden states), with overhead under 2 ms per step on a laptop CPU. The 34 features fall in five families: **(F1) control-flow recurrence** (8: n-gram recurrence over canonicalized `(tool, argument-hash)` pairs, action-graph cycle statistics, action-type entropy); **(F2) environment feedback** (6: tool-error rate, *error-identity rate*, test-suite delta, compile-failure streak); **(F3) workspace productivity** (9: tree-edit distance between workspace checkpoints, *revert ratio*, distinct files touched, novel-bytes-per-output-token as surrogate for ι_t); **(F4) context pressure** (5: window occupancy, compaction count, *stale-prefix ratio* $n_t^{\text{cache}}/(n_t^{\text{in}} + n_t^{\text{cache}})$ from provider usage metadata, which rises once the agent stops adding material); **(F5) lexical self-report** (6: hedging, goal restatement, apology-retry). We down-weight F5 and ablate it (§4.5), because self-report is the one channel an adversary or sycophantic model can poison (Appendix A.15). A gradient-boosted tree $\hat{V}_\theta$ (Chen & Guestrin, 2016) with $\approx 10^4$ parameters approximates V_t and trains in seconds. **Circe uses no GPU at any stage.**

3.4 Anytime-valid doom certificates

The controller inspects the trajectory at *every* step and the number of inspections is random and data-dependent, so a split-conformal test applied per round is either invalid (repeated looks ignored) or crippling (α divided by an unbounded horizon); Table 2 measures both. We instead certify doom with a test supermartingale. Fix a binary $\mathcal{F}_s$-measurable *stall indicator* $S_s = \mathbf{1}\{\text{step } s \text{ produced no workspace, test, or context novelty}\}$, computed from F2 to F4 by a threshold rule fixed for each configuration λ, and let q_0 upper-bound the conditional stall rate of *salvageable* trajectories, giving $H_0^{\text{salv}} : \mathbb{E}[S_s \mid \mathcal{F}_{s-1}] \le q_0$ for all s. With $E_0 = 1$ and predictable $\gamma_s \in [0, 1/q_0)$, set $E_t = \prod_{s \le t}(1 + \gamma_s(S_s - q_0))$ and raise a *doom certificate* at $\tau_\alpha = \inf\{t : E_t \ge 1/\alpha\}$.

Theorem 3 (Anytime-valid false-kill control)**.** *Under* H_0^{salv}*,* $(E_t)_{t \ge 0}$ *is a nonnegative supermartingale with* $E_0 = 1$*, so by Ville's inequality* $\mathbb{P}[\exists t : E_t \ge 1/\alpha] \le \alpha$ *under the controller's own adaptively chosen inspection schedule. Hence for any trajectory satisfying* H_0^{salv} *the probability that* Circe *ever raises a certificate is at most* α*, uniformly over the horizon, and the probability it* halts *the trajectory is at most* $(1 - \epsilon)\alpha$*.*

The last clause matters because the coin that buys identification also strictly tightens the guarantee. Since q_0 is unknown we substitute a $(1-\delta)$ time-uniform empirical-Bernstein upper bound $\hat{q}_0^+$ estimated on held-out *successful* calibration trajectories; the supermartingale property is monotone in q_0, so Theorem 3 holds at level $\alpha+\delta$ provided the pooled bound dominates the conditional stall rate of each salvageable trajectory. That proviso is where heterogeneity enters: a salvageable trajectory that stalls for a long stretch before succeeding violates H_0^{salv} and is not covered (§4.4).

Theorem 4 (Logarithmic detection delay)**.** *If a trajectory is truly doomed with conditional stall rate $q_1 > q_0$ and γ_s follows an approximate-GRAPA betting schedule (Jiang & Yu, 2024) converging to $\gamma^\star = (q_1-q_0)/(q_0(1-q_0))$, then $\mathbb{E}[\tau_\alpha] \le \big(\log(1/\alpha) + O(\log\log(1/\alpha))\big)/\mathrm{kl}(q_1\|q_0)$, and no α-level sequential test attains expected delay below $(1-o(1))\log(1/\alpha)/\mathrm{kl}(q_1\|q_0)$ as $\alpha \to 0$.*

Theorem 4 separates CIRCE from a threshold on a classifier: it bounds the wasted steps instance-dependently, in a currency the cost model converts to dollars, and it is falsifiable against the measured delays of Table 2.

3.5 FROM PER-TRAJECTORY VALIDITY TO POPULATION RISK CONTROL

Theorem 3 controls a per-trajectory error, whereas equation 1 constrains the population salvage loss, and the configuration λ (stall thresholds, certificate level, betting schedule, compaction trigger, rewind depth) is *selected on data*: a multiple-testing problem the early-stopping literature ignores. The certificate level is a component of λ and need not equal the risk target α of equation 1; at the default operating point both are 0.05. We treat selection as risk-controlling prediction (Angelopoulos et al., 2025; Bates et al., 2021): for each λ in a grid ($|\Lambda| \approx 500$) form $H_\lambda : R(\lambda) > \alpha$ with a super-uniform p-value from a time-uniform empirical-Bernstein bound on the estimator $\hat{R}(\lambda)$ of §3.6, order Λ by increasing predicted cost saving using a split disjoint from the one that supplies the p-values, and apply fixed-sequence testing at level δ.

Theorem 5 (Distribution-free risk control of the selected controller)**.** *Let $\hat{\lambda}$ be the last configuration rejected by the fixed-sequence procedure at level δ. If the calibration tasks are i.i.d. draws from $\mathcal{D}$, then with no further assumptions on $\mathcal{D}$, π or the harness, $\mathbb{P}[R(\hat{\lambda}) \le \alpha] \ge 1-\delta$.*

3.6 RANDOMIZED REPRIEVE: STAYING CALIBRATED AFTER YOU START KILLING

Every existing interruption method calibrates on complete trajectories and then deploys. Once deployed it terminates precisely the trajectories whose outcomes it most needs: for any episode it halts, $Y^{\kappa_\varnothing}$ is never realized, so the log is censored *by the treatment itself*, $R(\lambda)$ is non-identified, and any refresh of the corpus is biased toward what the incumbent permits. Our mechanism makes the controller deliberately stochastic: when a certificate fires at step t of episode i, CIRCE halts with probability $1-\epsilon_i$ and with probability ϵ_i grants a *reprieve*, in which the episode runs to natural termination without further intervention so that $Y_i^{\kappa_\varnothing}$ is observed and logged with the known propensity. Because the propensity is chosen rather than estimated, the inference is design-based. With $\mathcal{K}_\lambda$ the episodes in which κ_λ fires, D_i the reprieve indicator and $\hat{\mu}$ a fitted outcome model,

$$\hat{R}(\lambda) = \frac{1}{n}\sum_{i\in\mathcal{K}_\lambda}\Big[\hat{\mu}\big(Z_{\tau_i}^{(i)}\big) + \frac{D_i}{\epsilon_i}\Big(Y_i^{\kappa_\varnothing} - \hat{\mu}\big(Z_{\tau_i}^{(i)}\big)\Big)\Big] - \frac{1}{n}\sum_{i\in\mathcal{K}_\lambda} Y_i^{\kappa_\lambda}. \tag{4}$$

Proposition 6 (Identification and rate)**.** *If $\epsilon_i \ge \epsilon_{\min} > 0$ and propensities are logged, $\hat{R}(\lambda)$ is unbiased for $R(\lambda)$ whenever either $\hat{\mu}$ or the propensity model is correct (the latter holds by design), with $\mathrm{Var}[\hat{R}] = O(1/(n\epsilon_{\min}))$, and it admits a time-uniform empirical-Bernstein confidence sequence, which supplies p_λ in Theorem 5* at every round of deployment *rather than only at calibration time.*

Reprieves cost money (they are exactly the tokens we set out to save), so ϵ trades identification against operating cost, and deployment over K rounds becomes a partial-feedback control problem.

Theorem 7 (Self-calibration regret)**.** *Let $\bar{c}$ be the mean cost of completing a certified-doomed episode. Running* CIRCE *over K rounds with $\epsilon_k = \min\{1, \bar{c}^{-2/3}(\log|\Lambda|/k)^{1/3}\}$ and re-selecting $\hat{\lambda}_k$ by Theorem 5 at each round k yields cumulative cost regret against the best fixed $\lambda \in \Lambda$ in hindsight of $\tilde{O}\big(\bar{c}^{1/3}(\log|\Lambda|)^{1/3}K^{2/3}\big)$, while $\mathbb{P}[\forall k \le K : R(\hat{\lambda}_k) \le \alpha] \ge 1-\delta$.*

The $K^{2/3}$ rate is the signature of partial feedback (Langford & Zhang, 2007) and is correct here because the controller's own action determines whether the label is observed. In deployment we use the same $k^{-1/3}$ decay with the constant set so that $\epsilon_1 = 0.20$, that is $\epsilon_k = 0.20\,k^{-1/3}$ with a floor of 0.02, which gives 0.09 at round 10 and costs 2.4% of total spend (Appendix A.16). The floor keeps $\epsilon_{\min} > 0$ as Proposition 6 requires, at the price of an exploration cost that becomes linear in K at rate $0.02\,\bar{c}$ once reached.

3.7 Full controller

Halting is one of five interventions, arbitrated by a cost-normalized optimistic index $I_u(Z_t) = (\hat{\Delta}_u(Z_t) + \beta\sigma_u(Z_t))/\hat{c}_u(Z_t)$, where $\hat{\Delta}_u$ is the estimated gain in V from applying u, $\hat{c}_u$ its expected added cost, and σ_u a conformal upper bound on prediction error giving optimism-driven exploration. HALT is special-cased: it becomes admissible *only* after the certificate fires, which attaches Theorem 3 to the single irreversible action (pseudocode: Algorithm 1). After a *reversible* intervention the wealth is reset to $\min\{E_t, 1\}$. The reset is needed because COMPACT and REWIND change the harness state, so evidence gathered against the pre-intervention trajectory should not be used to kill the post-intervention one; capping rather than restoring the wealth never increases it, so the process remains a nonnegative supermartingale over the whole episode and Theorem 3 is preserved without any split of α across segments. The reset has a measurable price, which we report: it lengthens detection delay from 8.6 to 10.8 steps and costs 2.0 points of cost saving.

3.8 Instantiation with GLM-5.2

All experiments use `glm-5.2` through an OpenAI-compatible endpoint. Its pricing is non-uniform (cached input far below fresh input, output several times above both), reasoning effort is configurable per call, which makes ESCALATE a priced action, and its million-token context removes the context-exhaustion backstop, which is why baseline B4 is identical to no interruption. Per-call usage metadata reports cached versus fresh tokens, giving the stale-prefix feature of F4 at no cost. Provider and settlement details are in Appendix A.4.

4 Experiments

Setup. Four suites: **SWE** (repository-level software engineering, hidden-test validation, N=600), **TERM** (terminal and OS tasks, N=96), **WEB** (stateful web navigation, N=812), and **TOOL** (N=600), a near-saturated tool-calling suite used as a *negative control* on which a correct controller should rarely fire. Outcomes come from each benchmark's native validator. Under *paired branching replay*, each uninterrupted trajectory is banked once under common seeds and every variant is re-executed only from the step at which it intervenes, so variants share all pre-intervention tokens and cache hits. We report paired differences with bootstrap 95% intervals over tasks, $n_{\text{seed}} = 3$ for stochastic components, and $\hat{\lambda}$ selected on calibration data only. Baselines are (B1) no interruption; (B2) fixed `max_turns`; (B3) fixed token budget; (B4) context-exhaustion backstop; (B5) n-gram repetition kill; (B6) per-step LLM judge; (B7) semantic early-stopping; (B8) a probe cascade, as **B8-WB** on an open-weight model and **B8-BB** with probes replaced by our telemetry; (B9) a split-conformal stopper with union-bounded looks; and an oracle with access to Y (Appendix A.6). **False-kill rate** FK is the fraction of *all* episodes that are halted and whose uninterrupted counterfactual would have succeeded, measured on a fully reprieved audit split. Unless stated, $\alpha = 0.05$ and $\delta = 0.01$; macro-averages are unweighted over suites and pooled FK is task-weighted. The black-box study cost $1,896.90 in API spend and no GPU-hours; B8-WB is the only arm that needs activation access (Appendix A.17).

4.1 Main comparison

CIRCE reduces macro cost by 36.35% at a -0.38 pp pass@1 difference whose bootstrap interval contains zero, against 21.26% at -1.67 pp for the strongest cheap black-box baseline, B7 (Table 1; frontier plot in Appendix A.7). Three details matter more than the headline. *The per-step judge* is the best non-reference baseline on quality (-0.78 pp) and the worst on cost: its own inference leaves the harness 3.36% *more* expensive than never intervening, so any stopper paying a model call per

Table 1: **Main results** with `glm-5.2`. Pass@1 (%, native validator), cost ($/task, cache-aware pricing) and false-kill rate FK (%, reprieved audit split). *Int.*: requires model internals; *+Calls*: extra model calls per step. Macro columns are unweighted over all suites; pooled FK is task-weighted. TERM and the TOOL control are omitted for width and are in Table 6. **Bold**: best among black-box interrupting methods; B1, B4 and the gray reference rows are excluded from ranking.

				SWE (N=600)			**WEB** (N=812)			**Macro / pooled**		
	Method	Int.	+Calls	Pass@1	Cost↓	FK↓	Pass@1	Cost↓	FK↓	ΔPass@1	Save↑	FK↓
B1	No interruption	n/a	0	39.50	0.850	n/a	38.18	0.240	n/a	0.00	0.00	n/a
B4	Context backstop[§]	✗	0	39.50	0.850	0.00	38.18	0.240	0.00	0.00	0.00	0.00
B2	Fixed `max_turns`	✗	0	31.00	0.598	9.00	32.39	0.165	6.16	−5.52	23.70	5.46
B3	Fixed token budget	✗	0	35.00	0.642	4.83	34.24	0.174	4.31	−3.50	19.42	3.41
B5	n-gram repetition kill	✗	0	36.83	0.669	3.17	35.10	0.182	3.45	−2.82	17.27	2.56
B6	Per-step LLM judge	✗	1	38.83	0.812	1.17	37.81	0.254	0.74	−0.78	−3.36	0.71
B7	Semantic early-stop	✗	0	37.50	0.610	2.33	36.82	0.170	1.60	−1.67	21.26	1.47
B8-BB	Probe cascade (BB)	✗	0	37.17	0.616	2.67	36.45	0.172	1.97	−2.10	21.19	1.76
B9	Conformal + union bound	✗	0	38.83	0.768	**0.67**	37.68	0.218	**0.49**	−0.82	7.38	**0.47**
B8-WB	Probe cascade (WB)[†]	✓	0	39.00	0.476	0.50	37.81	0.146	0.37	−0.48	31.28	0.33
	Oracle stopper[‡]	✓	0	39.50	0.357	0.00	38.18	0.115	0.00	0.00	49.81	0.00
	CIRCE (HALT-only)	✗	0	37.67	0.531	5.00	36.70	0.158	6.28	−1.65	28.79	4.17
	CIRCE (full)	✗	0	**39.17**	**0.430**	0.83	**38.05**	**0.133**	0.62	−0.38	**36.35**	0.52

[§]never fires under a 10^6-token context, so it is identical to B1 by construction. [†]open-weight model of comparable capability, since `glm-5.2` does not expose activations; not a like-for-like arm. [‡]uses ground-truth Y.

Table 2: **Empirical validity of Theorem 3 and tightness of Theorem 4**, pooled over four suites, using the HALT-only configuration so all false kills are attributable to the certificate. FK must not exceed $\alpha + \delta$ with $\delta = 0.01$; *Predicted* is the leading term $\log(1/\alpha)/\mathrm{kl}(q_1 \| q_0)$ of Theorem 4.

Nominal α	Measured FK (%)	Valid?	Delay (steps)	Predicted	Ratio	Cost saving (%)
0.01	0.82	✓	13.7	11.3	1.21	8.12
0.02	1.42	✓	11.2	9.6	1.17	13.85
0.05	4.17	✓	8.6	7.4	1.16	28.79
0.10	8.67	✓	7.1	5.7	1.25	33.10
0.20	14.45	✓	4.9	4.0	1.23	37.24
B9 (conformal, looks ignored) @ 0.05	14.80	✗	3.6	n/a	n/a	31.15
B9 (conformal, union bound) @ 0.05	0.47	✓	16.8	n/a	n/a	7.38

step competes against the cost it creates. *The negative control is diagnostic*: on TOOL, CIRCE saves only 2.08% and kills no task that would have passed, whereas B2 and B3 save 6.25% and 4.17% there at 0.50% and 0.33% false kills. *The white-box comparison is not like for like*: CIRCE is ahead of B8-WB on saving and quality and behind on pooled FK (0.52 vs. 0.33), but B8-WB runs on a different model, so the defensible claim is only that the black-box constraint is not disqualifying here; 13.5 points of saving remain to the oracle.

4.2 IS THE GUARANTEE REAL, AND IS IT TIGHT?

Anytime validity is not a formality (Table 2). The split-conformal stopper that ignores repeated looks realizes 14.80% FK at a nominal 5%, a factor of 3.0 attributable purely to optional continuation; union-bounding the horizon restores validity but saves only 7.38%, a quarter of what the e-process saves at the same level. Measured delay exceeds the predicted leading term by 0.9 to 2.4 steps across a twentyfold range of α, and the excess grows slowly as α shrinks, which is the behaviour of the lower-order $\log\log(1/\alpha)$ term in Theorem 4; the ratio stays between 1.16 and 1.25. Validity is not uniformly tight: at $\alpha = 0.05$ the HALT-only configuration realizes 4.17%, near nominal, whereas the full controller realizes 0.52%, because reversible actions remove stalls before evidence accumulates. That order-of-magnitude margin is a by-product, not a claim of tightness.

4.3 THE CENTRAL EXPERIMENT: CALIBRATION DRIFT UNDER DEPLOYMENT

The last column of Table 3 matters most. Every baseline, and our own system without reprieve, breaches the constraint after the shift (B2 by +0.088) and none registers the breach, because each estimates its risk from a corpus its own halting has censored. The $\epsilon = 0$ variant shares every other component and is still blind: removing the coin does not make the controller worse at stopping, it

Table 3: **Post-deployment calibration drift.** Realized salvage loss $R(\hat{\lambda}_k)$ over ten deployment rounds of a rotating stream, with a model-version change injected at round 4 and a benchmark-difficulty shift at round 7. Nominal constraint $\alpha = 0.05$. *Detected?* is whether the method's own diagnostics flagged the violation. Absolute R is not comparable with FK in Table 1, since the stream has its own task mix; the pre-shift ordering of methods is the same.

Method	R_1	R_2	R_4	R_6	R_8	R_{10}	Max violation	Detected?
B2 Fixed `max_turns`	0.041	0.044	0.095	0.101	0.138	0.134	+0.088	✗
B7 Semantic early-stop	0.036	0.039	0.079	0.086	0.112	0.108	+0.062	✗
B8-BB Probe cascade (BB)	0.034	0.037	0.073	0.078	0.101	0.098	+0.051	✗
CIRCE w/o reprieve (ϵ=0)	0.031	0.033	0.068	0.071	0.091	0.088	+0.041	✗
CIRCE fixed ϵ=0.05	0.030	0.032	0.066	0.046	0.062	0.044	+0.016	✓
CIRCE scheduled ϵ_k (ours)	0.029	0.031	0.065	0.042	0.058	**0.039**	+0.015	✓

Table 4: **Component ablation**, three-suite macro over SWE, TERM and WEB (TOOL is excluded because the controller rarely fires there); FK is the unweighted mean over the three suites. *Pareto AUC*: area under the cost-quality curve traced by sweeping the certificate level (higher is better). The upper block replaces or removes one component of the full controller; the lower block grows the action set from {CONT, HALT}. Further variants are in Table 7.

Variant	ΔPass@1 (pp)	Cost saving (%)	FK (%)	Pareto AUC↑
CIRCE (full)	−0.50	47.77	0.83	0.844
e-process → per-round conformal	−1.82	35.40	2.15	0.738
e-process → static score threshold	−2.35	38.80	2.94	0.695
aGRAPA betting → fixed γ	−0.65	41.50	0.98	0.806
w/o wealth reset after reversible action	−1.41	**49.80**	1.62	0.771
LTT selection → argmax on validation	−1.08	49.10	1.39	0.784
w/o randomized reprieve (ϵ=0)	−0.47	48.90	**0.81**	**0.848**
DR estimator → IPW only	−0.52	47.60	0.86	0.840
Action set {CONT, HALT} only	−2.15	37.70	5.50	0.742
+COMPACT	−1.22	40.70	1.35	0.778
+COMPACT + REWIND	−0.66	45.40	0.95	0.828
+COMPACT + REWIND + ESCALATE (full)	−0.50	47.77	0.83	0.844

makes it unable to see that it is. With scheduled reprieve, risk is back below α by round 6 after the first shift ($0.065 \to 0.042$) and by round 10 after the second ($0.058 \to 0.039$), the confidence sequence flags each breach 1.9 rounds after it occurs on average, and the mechanism costs 2.4% of spend. A power-law fit to cumulative regret over the ten rounds gives an exponent of 0.68, consistent with Theorem 7, although ten rounds are too few for this to be a test of it. The system detects and corrects the excursion rather than preventing it, and fixed $\epsilon = 0.05$ is nearly as good (+0.016), so the schedule's contribution is a lower exploration cost, not a different outcome.

4.4 ABLATIONS

Two components carry the weight, and one looks worthless until Table 3 (Table 4). Replacing the e-process with a per-round conformal test costs 0.106 of Pareto AUC and raises FK from 0.83% to 2.15%; the extended action set is worth a comparable 0.102, and moving from {CONT, HALT} to the full set buys 10.1 points of saving *and* 1.65 pp of pass@1, 0.56 pp from REWIND alone. Removing randomized reprieve is mildly *beneficial* in distribution (+1.13 points of saving and the best Pareto AUC): any static-benchmark evaluation would delete it, which is exactly wrong. The wealth reset behaves the same way: removing it raises saving to 49.80% but nearly doubles FK and costs 0.91 pp of pass@1, because evidence gathered before a successful REWIND then kills the recovered trajectory. In both cases the component that looks like overhead buys the guarantee.

Where the savings come from, and where the method fails. HALT contributes 77.4% of net saved dollars (69.6 points from episodes certified doomed before step 25, and 7.8 from shortening trajectories that would have succeeded anyway, which is the quantity α governs), REWIND 23.8% and COMPACT 1.2%, while ESCALATE and reprieve cost 0.9% and 1.5% (Appendix A.12). The

Table 5: **Leave-one-family-out feature ablation** under the full controller at $\alpha = 0.05$: four-suite macro cost saving, task-weighted pooled FK, detection delay, and pooled FK under the termination-poisoning suite of Appendix A.15 (*Poisoned FK*). Each leave-one-out row removes exactly one family; the family sizes $8{+}6{+}9{+}5{+}6$ sum to 34. F5 is the only channel the agent's own prose can manipulate. **Bold**: best per column; red: largest degradation among the leave-one-out rows.

Features	#Feat.	Cost saving (%)↑	FK (%)↓	Delay (steps)↓	Poisoned FK (%)↓
All (F1 to F5)	34	**36.35**	**0.52**	**10.8**	2.32
w/o F1 control-flow recurrence	26	33.80	0.62	11.9	2.47
w/o F2 environment feedback	28	31.40	0.66	12.8	2.61
w/o F3 workspace productivity	25	25.10	0.81	15.4	2.70
w/o F4 context pressure	29	33.10	0.57	11.4	2.37
w/o F5 lexical self-report	28	35.20	**0.52**	10.9	**0.90**
F3 only (novel-bytes surrogate for ι_t)	9	28.20	0.76	13.5	1.09
F1 only ($\approx$ B5 repetition heuristic)	8	17.20	1.28	17.2	1.47

failure mode is the one Proposition 2 predicts: on 112 tasks whose reference solutions need an extended unproductive exploration phase, where equation 3 is checked over m steps rather than the true horizon, FK rises to 4.82% against 0.83% on the three non-control suites. Those trajectories genuinely stall by every observable measure and so also fall outside H_0^{salv}; the patience parameter m, not the certificate, is the operative safeguard (Appendix A.18).

4.5 Which telemetry carries the signal?

Table 5 tests the claim of §3.2 that a doom loop is a value-stall rather than a repetition pattern. Workspace productivity (F3), whose novel-bytes feature is the only direct estimate of information yield, carries the signal: removing it costs 11.25 of the 36.35 points of saving and is the largest degradation in every column, while F3 alone recovers 78% of the full saving. F1 alone, essentially the shipped n-gram heuristic, recovers only 47% at $2.5\times$ the FK, so repetition is a special case of a stall rather than its definition (Figure 3A). Dropping the manipulable F5 costs 1.15 points, leaves FK at 0.52% and cuts poisoned FK from 2.32% to 0.90%, which is why it is down-weighted by default. Appendix A.11 gives the full analysis, including why these delays exceed those of Table 2.

4.6 Case studies

Appendix Figure 3 illustrates three behaviours behind the aggregate tables. In (A) no action repeats, so an n-gram kill never fires, but the workspace and context families register no novelty, which is why F3 outweighs F1 (Table 5). In (B) Rewind turns a trajectory a binary controller would kill into a success, the mechanism behind its 23.8% share of the saving and 0.56 pp of pass@1 (Table 4). In (C) a false progress claim would keep a judge-based stopper running, the mirror image of the early firing induced by injected completion claims (Appendix A.15); both are prose the agent produces for free, hence the down-weighting of F5. The panels are illustrative, not frequency claims.

5 Conclusion and Outlook

We framed agent interruption as constrained switching control whose hard part is keeping the detector honest after it starts acting. Defining a doom loop as a value-stall explains why score-decay stoppers fail and yields an exact information budget; an e-process certificate removes the horizon penalty of fixed-sample calibration, risk-controlling selection lifts it to population guarantees, and randomized reprieve restores the identification an interrupting controller would otherwise destroy. With `glm-5.2`, Circe cuts cost by 36.35% at a 0.38 pp pass@1 difference and 0.52% false kills from harness logs alone, and under mid-stream shift it is the only method that detects and corrects its own risk excursion, for 2.4% of spend. Reprieve is a pure cost on a static benchmark and decisive under shift: interruption policies should be judged after deployment, not only on calibration data. Natural next steps are a horizon-aware halting rule based on the consumed fraction of $V_0(1 - V_0)$, a reprieve rate driven by a change-point statistic for slow drift, evaluation across several API-served policies, and grounded validator-side telemetry against termination poisoning (Appendix A.18). More broadly, any irreversible harness action censors the outcomes needed to evaluate it, and a small scheduled randomization with logged propensities keeps it auditable.

AI USE STATEMENT

Large language models were used only to polish the language of the manuscript. Separately, API access to language models is part of the proposed method and its evaluation: the `glm-5.2` agent policy and the model-based baselines are described in §3.8 and §4.

ETHICS STATEMENT

The authors have read and follow the ICLR Code of Ethics. All experiments run agents on benchmark tasks, and outcomes come from each benchmark's own validator, never from a model judge. The main risk of an interruption controller is halting an episode that would have succeeded. We bound this risk (Theorems 3 and 5), report the false-kill rate of every method, and document where the guarantee weakens (§4.4, Appendix A.18). Randomized reprieve keeps the controller's risk auditable after deployment (§4.3). The termination-poisoning study (Appendix A.15) is included to measure and mitigate a vulnerability of stoppers that rely on the agent's self-report.

REPRODUCIBILITY STATEMENT

We submit the code as part of the supplementary materials. Proofs are provided in Appendix A.1. The full controller is specified in Algorithm 1 and Appendix A.3. Environments and task-level splits are described in Appendix A.5, baselines in Appendix A.6, the cost model in Appendix A.13, the reprieve schedule in Appendix A.16, and the experimental budget and paired branching replay in Appendix A.17. The evaluation protocol is given in §4: paired bootstrap intervals over tasks, three seeds for stochastic components, and configuration selection on calibration data only.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Anastasios N Angelopoulos, Stephen Bates, Emmanuel J Candès, Michael I Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.

BM Brown. Great expectations: The theory of optimal stopping. *Journal of the Royal Statistical Society: Series A (General)*, 135(4):610–610, 1972.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters, 2018.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.

Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

Peter Grünwald, Rianne de Heide, and Wouter M Koolen. Safe testing. In *2020 Information theory and applications workshop (ITA)*, pp. 1–54. IEEE, 2020.

Said Hamadène and Monique Jeanblanc. On the starting and stopping problem: application in reversible investments. *Mathematics of Operations Research*, 32(1):182–192, 2007.

Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

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.

Rong Jiang and Keming Yu. Estimating means of bounded random variables by betting. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY SERIES B-STATISTICAL METHODOLOGY*, 86(1), 2024.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, volume 2024, pp. 54107–54157, 2024.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Sayash Kapoor, Benedikt Stroebl, Zachary S Siegel, Nitya Nadgir, and Arvind Narayanan. Ai agents that matter. *arXiv preprint arXiv:2407.01502*, 2024.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems*, 20, 2007.

Tor Lattimore and Csaba Szepesvári. Bandit algorithms. *Cambridge University*, 355, 2019.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023a.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023b.

Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Goran Peskir and Albert Shiryaev. *Optimal stopping and free-boundary problems*. Springer, 2006.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023.

James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427): 846–866, 1994.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022.

Glenn Shafer. Testing by betting: A strategy for statistical and scientific communication. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 184(2):407–431, 2021.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 10131–10165, 2025.

Richard S Sutton, Andrew G Barto, and Andrew Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research*, 16(1):1731–1755, 2015.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Jean Ville. *Etude critique de la notion de collectif*, volume 3. Gauthier-Villars Paris, 1939.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Abraham Wald. Sequential tests of statistical hypotheses. In *Breakthroughs in statistics: Foundations and basic theory*, pp. 256–298. Springer, 1992.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better llm agents. *arXiv preprint arXiv:2402.01030*, 2024.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Openhands: An open platform for ai software developers as generalist agents. In *International Conference on Learning Representations*, volume 2025, pp. 65882–65919, 2025.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022b.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

L Zheng, WL Chiang, Y Sheng, S Zhuang, Z Wu, Y Zhuang, Z Lin, Z Li, D Li, EP Xing, et al. Judging llm-asa-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds*, 36:46.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

A Appendix

A.1 Proofs

Proposition 2. V is a bounded martingale and $T < \infty$ almost surely with $V_T = Y \in \{0, 1\}$, so its Doob decomposition gives $\mathbb{E}[(V_T - V_t)^2 \mid \mathcal{F}_t] = \mathbb{E}[\langle V\rangle_T - \langle V\rangle_t \mid \mathcal{F}_t]$. Since $\mathbb{E}[V_T \mid \mathcal{F}_t] = V_t$ and $V_T^2 = V_T$, $\mathbb{E}[(V_T - V_t)^2 \mid \mathcal{F}_t] = \mathbb{E}[V_T \mid \mathcal{F}_t] - 2V_t^2 + V_t^2 = V_t(1 - V_t)$, which is equation 2. For the cost bound, $\Delta\langle V\rangle_{s+1} = \iota_s\, \mathbb{E}[c_{s+1} \mid \mathcal{F}_s]$ by Definition 1; if $\iota_s \le \bar{\iota}$ for all $s \ge t$ then $V_t(1 - V_t) = \mathbb{E}[\sum_{s\ge t} \iota_s\, \mathbb{E}[c_{s+1} \mid \mathcal{F}_s] \mid \mathcal{F}_t] \le \bar{\iota}\,\mathbb{E}[C_{t+1:T} \mid \mathcal{F}_t]$. Finally, halting at t yields W_t, while continuing to natural termination yields expected success $\mathbb{E}[Y \mid \mathcal{F}_t] = V_t$, so the gain from continuing is $V_t - W_t \le V_t$ because $W_t \ge 0$, while the added penalty is at least $\eta V_t(1 - V_t)/\bar{\iota}$. Halting is therefore at least as good whenever $V_t \le \eta V_t(1 - V_t)/\bar{\iota}$, i.e. $\bar{\iota} \le \eta(1 - V_t)$, and $V_t \le v_0$ makes $\bar{\iota} \le \eta(1 - v_0)$ sufficient. □

Three remarks. The comparison is between halting now and running to natural termination; continuation strategies that halt at an intermediate time are not covered, because the information bound constrains the cost of running to T only. The bound is stated for the whole remaining horizon; a controller that verifies $\iota_s \le \bar{\iota}$ only on a window $[t, t + m]$ is applying equation 3 to a truncation, which is exactly the approximation that fails on the delayed-breakthrough stratum of §4.4. And the inequality uses only $W_t \ge 0$; a harness whose partial artifact is often worth something has a larger W_t, so the condition as stated is conservative in the direction of continuing.

Theorem 3. Under H_0^{salv}, $\mathbb{E}[E_t \mid \mathcal{F}_{t-1}] = E_{t-1}(1 + \gamma_t(\mathbb{E}[S_t \mid \mathcal{F}_{t-1}] - q_0)) \le E_{t-1}$ because γ_t is $\mathcal{F}_{t-1}$-measurable and nonnegative. Each factor is $1 + \gamma_t(1 - q_0) > 0$ when $S_t = 1$ and $1 - \gamma_t q_0 > 0$ when $S_t = 0$, the latter because $\gamma_t < 1/q_0$, so $E_t \ge 0$. Capping the wealth at 1 after a reversible intervention can only decrease it, so the capped process is still a nonnegative supermartingale. Ville's inequality gives $\mathbb{P}[\sup_t E_t \ge 1/\alpha] \le \alpha E_0 = \alpha$. The final clause follows because, conditional on a certificate, the episode is halted only if an independent $\mathrm{Bern}(1 - \epsilon)$ coin lands, so $\mathbb{P}[\textsc{Halt}] \le (1 - \epsilon)\alpha$. □

Theorem 4. Under the alternative with stall rate q_1, $\log E_t$ has per-step drift $\mathrm{kl}(q_1 \| q_0)$ at the oracle fraction $\gamma^\star$, which lies in $[0, 1/q_0)$ because $\gamma^\star q_0 = (q_1 - q_0)/(1 - q_0) < 1$. The approximate-GRAPA plug-in has log-wealth regret $O(\log t)$ against $\gamma^\star$ (Jiang & Yu, 2024), so the wealth crosses $1/\alpha$ once $t\,\mathrm{kl}(q_1 \| q_0) - O(\log t) \ge \log(1/\alpha)$, which by Wald's identity gives the stated bound with an $O(\log\log(1/\alpha))$ excess. The matching lower bound is the classical sequential-testing bound for α-level tests. □

Theorem 5 and Proposition 6. Theorem 5 is fixed-sequence testing with super-uniform p-values (Angelopoulos et al., 2025); validity requires i.i.d. calibration tasks, an ordering fixed independently of the p-values, and p_λ super-uniform under H_λ, which the time-uniform empirical-Bernstein bound (Howard et al., 2021; Maurer & Pontil, 2009) on equation 4 supplies. Proposition 6 is the standard doubly robust argument (Robins et al., 1994; Dudík et al., 2011) with a *designed* propensity, so the propensity model is correct by construction and only $\hat{\mu}$ can be misspecified; the variance statement is the usual inverse-propensity inflation.

Theorem 7. Explore-then-commit style analysis with a decaying exploration rate (Langford & Zhang, 2007; Lattimore & Szepesvári, 2019): at round k the exploration cost is $O(\epsilon_k \bar{c})$ and the estimation error on $\hat{R}$ is $O\big(((\log|\Lambda|)/(k\epsilon_k))^{1/2}\big)$. Balancing the two gives $\epsilon_k \asymp \bar{c}^{-2/3}(\log|\Lambda|/k)^{1/3}$, a per-round regret of order $\bar{c}^{1/3}(\log|\Lambda|/k)^{1/3}$, and cumulative regret $\tilde{O}(\bar{c}^{1/3}(\log|\Lambda|)^{1/3}K^{2/3})$. Risk control at every round is inherited from Theorem 5 applied per round, with the union over rounds absorbed into the time-uniform bound.

A.2 RESEARCH QUESTIONS

The experiments are organized around six questions. **RQ1**: does CIRCE dominate existing interruption methods on the cost-quality frontier under a realistic billing model (§4.1)? **RQ2**: is the false-kill guarantee empirically tight and is detection delay consistent with theory (§4.2)? **RQ3**: does randomized reprieve detect and correct post-deployment calibration drift (§4.3, App. A.16)? **RQ4**: which components matter, and does the non-binary action set earn its complexity (§4.4, §4.5, App. A.10, A.11 and A.12)? **RQ5**: does a controller calibrated on one model and benchmark transfer (App. A.14)? **RQ6**: is the controller robust when the agent's self-report is adversarially or sycophantically unreliable (§4.5, App. A.15)?

A.3 CONTROLLER PSEUDOCODE AND MODULE SPECIFICATIONS

Algorithm 1 CIRCE controller (one episode)

Input: config $\hat{\lambda}$, stall bound $\hat{q}_0^+$, certificate level α, reprieve rate ϵ
$E \leftarrow 1$
for $t = 1, 2, \ldots$ until natural termination **do**
  $Z_t \leftarrow \phi(\mathcal{F}_t)$ ▷ from logs; zero API cost
  $S_t \leftarrow \textsc{StallIndicator}(Z_t)$; $\gamma_t \leftarrow \textsc{aGRAPA}(S_{1:t-1}, \hat{q}_0^+)$
  $E \leftarrow E \cdot (1 + \gamma_t(S_t - \hat{q}_0^+))$
  if $E \geq 1/\alpha$ **then**
    draw $D \sim \mathrm{Bern}(\epsilon)$
    if $D = 1$ **then**
      log (i, t, ϵ) as a reprieve
      run the harness with no further intervention to natural termination; log $Y^{\kappa_\varnothing}$
      return the natural outcome
    else
      return HALT with best partial artifact
    end if
  else
    $u^\star \leftarrow \arg\max_{u \in \mathcal{U} \setminus \{\textsc{Halt}\}} I_u(Z_t)$; apply $u^\star$
    if $u^\star \in \{\textsc{Compact}, \textsc{Rewind}\}$ **then**
      $E \leftarrow \min\{E, 1\}$ ▷ discard evidence, never add wealth
    end if
  end if
end for

Algorithm 1 is the full controller referenced in §3.7.

Stall indicator. S_s is the conjunction of three threshold rules over families F2 to F4: no increase in the number of passing tests and no change in the tool-error signature (F2); normalized tree-edit distance between consecutive workspace checkpoints below 10^{-3} and revert ratio above 0.8 (F3); stale-prefix ratio above 0.95 (F4). The thresholds are part of λ and selected by Theorem 5; the indicator is deliberately crude, because the e-process supplies the statistical power and an over-engineered indicator only makes q_0 harder to bound.

Betting schedule. γ_s follows approximate GRAPA (Jiang & Yu, 2024): $\gamma_s = \mathrm{clip}\big((\hat{q}_s - \hat{q}_0^+)/(\hat{q}_0^+(1-\hat{q}_0^+)),\ 0,\ c/\hat{q}_0^+\big)$ with $\hat{q}_s$ the running stall rate and $c = 0.5$ a truncation constant. The truncation keeps each factor $1 - \gamma_s \hat{q}_0^+$ at least $1 - c$, so a single non-stall step early in a trajectory cannot destroy the wealth.

Intervention models. $\hat{\Delta}_u$ and $\hat{c}_u$ are per-action gradient-boosted regressors on Z_t, fit on calibration episodes in which the action was applied by a uniformly random exploration policy; σ_u is a split-conformal upper bound on their residuals at level 0.9. $\textsc{Rewind}(k)$ uses $k \in \{5, 10, 20\}$, selected by the index; the checkpoint store is a content-addressed workspace snapshot taken every 5 steps, costing no tokens.

Outcome model for the estimator. $\hat{\mu}$ in equation 4 is a gradient-boosted classifier of $Y^{\kappa_\varnothing}$ on Z_τ, refit each round on all reprieved episodes observed so far. Because the propensity is designed, $\hat{\mu}$

affects only the variance of $\hat{R}$, not its bias; replacing it with the IPW-only estimator costs 0.004 of Pareto AUC (Table 4) and widens the confidence sequence.

A.4 Implementation, provider and settlement

All runs use `glm-5.2` through an OpenAI-compatible endpoint with a fixed cache-friendly prompt layout. Prices are those in force at settlement; because third-party routing changes absolute dollar figures, every claim in the paper is stated as a ratio or a paired difference under a single price schedule, and the raw usage records (input, cached-input and output tokens per call) are released with the code so that any other schedule can be applied post hoc. Controller training (the $\hat{V}_\theta$ tree, the per-action regressors, the conformal residual quantiles and the outcome model) runs on a single CPU in under three minutes in total. **GPU-hours used by CIRCE and all black-box arms: 0.**

A.5 Environments and splits

SWE (N=600) is repository-level software engineering with hidden-test validation. **TERM** (N=96) is terminal and operating-system tasks with long tool chains; it is the smallest suite, so its percentages move in increments of $1/96 \approx 1.04$ pp and single-task noise is visible in every TERM column. **WEB** (N=812) is stateful web navigation and carries the highest environment non-determinism. **TOOL** (N=600) is a near-saturated tool-calling suite (72.00% pass@1 uninterrupted) used as a negative control. Calibration, validation and test partitions are disjoint at the task level, with 105 calibration episodes drawn from held-out dev splits (Table 13). All outcomes come from each benchmark's native validator.

A.6 Baselines

(B1) *No interruption*: the harness default and the reference for pass@1 and cost. (B2) *Fixed* `max_turns`, tuned per benchmark on the calibration split. (B3) *Fixed token budget*, likewise tuned. (B4) *Context-exhaustion backstop*, the implicit stopper of most published agents; under a 10^6-token context it never fires on any episode in our suites, so it is numerically identical to B1 and is excluded from ranking. (B5) *n-gram repetition kill*, the heuristic shipped in deployed harnesses. (B6) *Per-step LLM-judge stopper*: the strongest naive method, adding one model call per step; its judge calls are charged as operational tokens, which is the entire point of including it. (B7) *Semantic early-stopping*, a judge-free semantic stopper representing the current state of the art in the cheap black-box regime. (B8) *Probe cascade*, the current state of the art in the white-box regime, predicting eventual failure from internal activations under a recall-controlled abort schedule (Alain & Bengio, 2016; Burns et al., 2022). Reported as **B8-WB** (open-weight model of comparable capability; a capability reference, not a controlled arm) and **B8-BB** (faithful black-box re-implementation with activation probes replaced by our telemetry and the recall-controlled calibration retained). (B9) *Split-conformal threshold stopper with union-bounded looks* (Vovk et al., 2005), which isolates the contribution of anytime validity; we report it both with the union bound applied and with the looks ignored (Table 2). *Oracle stopper*: halts at the first step of every episode whose true $Y = 0$, giving the achievable upper bound.

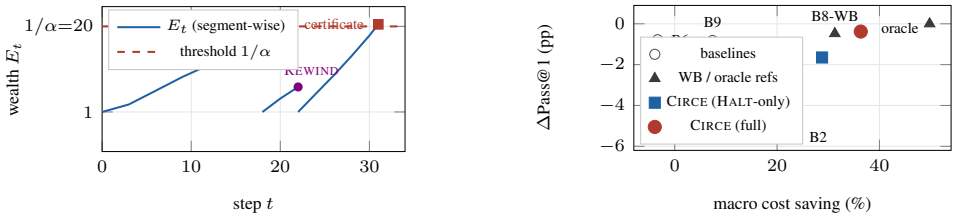
A.7 Wealth trajectory and cost-quality frontier

Figure 2a shows that the certificate is not a monotone score: each reversible intervention caps the wealth, so an episode is halted only once evidence re-accumulates in a segment that the reversible actions could not fix. Figure 2b places every method of Table 6 on the cost-quality plane.

A.8 Case-study examples

A.9 Full main results

Two cross-suite patterns are worth naming. TERM is the smallest suite and every figure in its columns is a multiple of $1/96$, so the apparent tie between CIRCE and B8-WB there (28.13 pass@1, i.e. $27/96$) is a single task and should not be read as a ranking. And the gap between FK and the realized pass@1 loss is systematic: on SWE the HALT-only variant records 5.00% FK but loses only 1.83 pp of pass@1, because a halted episode still returns its best partial artifact and a partial patch that already



(a) One SWE episode. Reversible actions cap the wealth at 1; only the third segment reaches $1/\alpha$.

(b) Cost-quality frontier from Table 1. Up and right is better.

Figure 2: **A worked episode and the resulting frontier.** (a) The certificate is not a monotone score: COMPACT at step 18 and REWIND at step 22 each cap the wealth at 1, so the episode is only halted once evidence re-accumulates in a segment the reversible actions could not fix. (b) CIRCE is better than every black-box baseline on both axes at once: its saving exceeds the largest baseline saving (23.70%, B2) and its quality loss is smaller than the smallest baseline loss (0.78 pp, B6). It also sits beyond the white-box reference on both axes, though B8-WB runs on a different model and is not a like-for-like arm.

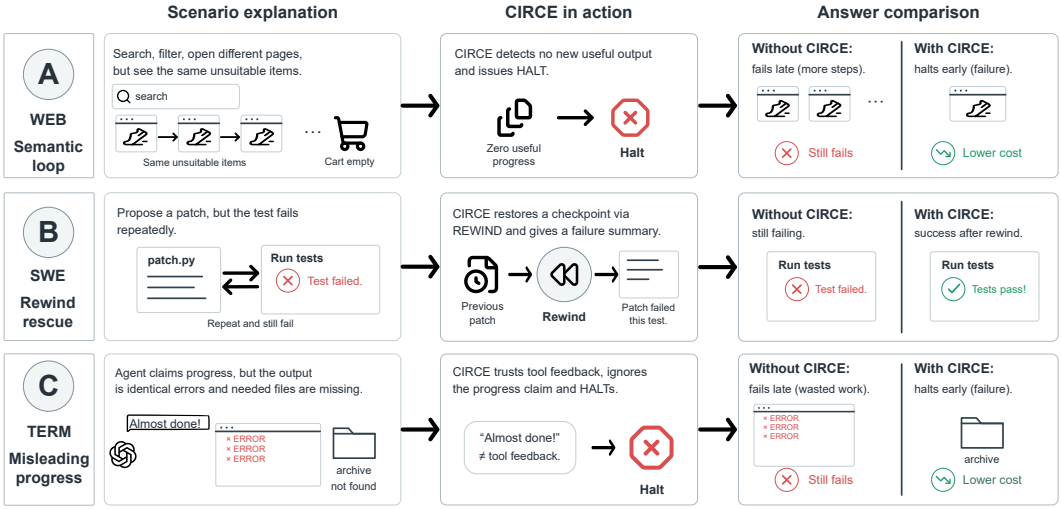


Figure 3: **Three doom-loop patterns and CIRCE's response. (A)** WEB semantic loop: the agent searches, filters and opens different pages, so no action repeats, yet every page shows the same unsuitable items; CIRCE detects that no new useful output is produced and halts early. **(B)** SWE rewind rescue: a patch fails the same test repeatedly; CIRCE restores a checkpoint with REWIND and a failure summary, and the episode succeeds. **(C)** TERM misleading progress: the agent reports that it is almost done while tool feedback shows identical errors and a missing archive; CIRCE trusts the tool feedback over the self-report and halts.

Table 6: **Main results, all four environments.** Table 1 in the main text is the SWE and WEB sub-table. FK is undefined for B1, which never halts. Macro columns are unweighted over the four suites; pooled FK is task-weighted over N=2,108 tasks. TOOL is the negative control. **Bold**: best among black-box interrupting methods (ties all bold).

				SWE (600)			**TERM** (96)			**WEB** (812)			**TOOL** control (600)			**Macro / pooled**		
	Method	Int.	+Calls	Pass@1	Cost	FK	Pass@1	Cost	FK	Pass@1	Cost	FK	Pass@1	Cost	FK	ΔPass@1	Save	FK
B1	No interruption	n/a	0	39.50	0.850	n/a	29.17	0.720	n/a	38.18	0.240	n/a	72.00	0.048	n/a	0.00	0.00	n/a
B4	Context backstop	✗	0	39.50	0.850	0.00	29.17	0.720	0.00	38.18	0.240	0.00	72.00	0.048	0.00	0.00	0.00	0.00
B2	Fixed `max_turns`	✗	0	31.00	0.598	9.00	21.88	0.521	8.33	32.39	0.165	6.16	71.50	**0.045**	0.50	−5.52	23.70	5.46
B3	Fixed token budget	✗	0	35.00	0.642	4.83	23.96	0.565	6.25	34.24	0.174	4.31	71.67	0.046	0.33	−3.50	19.42	3.41
B5	n-gram repetition kill	✗	0	36.83	0.669	3.17	23.96	0.580	5.21	35.10	0.182	3.45	71.67	0.046	0.33	−2.82	17.27	2.56
B6	Per-step LLM judge	✗	1	38.83	0.812	1.17	27.08	0.702	2.08	37.81	0.254	0.74	**72.00**	0.055	**0.00**	−0.78	−3.36	0.71
B7	Semantic early-stop	✗	0	37.50	0.610	2.33	26.04	0.536	3.13	36.82	0.170	1.60	71.83	0.047	0.17	−1.67	21.26	1.47
B8-BB	Probe cascade (BB)	✗	0	37.17	0.616	2.67	25.00	0.542	4.17	36.45	0.172	1.97	71.83	0.046	0.17	−2.10	21.19	1.76
B9	Conformal + union bound	✗	0	38.83	0.768	**0.67**	27.08	0.658	2.08	37.68	0.218	**0.49**	**72.00**	0.047	**0.00**	−0.82	7.38	**0.47**
B8-WB	Probe cascade (WB)†	✓	0	39.00	0.476	0.50	28.13	0.418	1.04	37.81	0.146	0.37	72.00	0.048	0.00	−0.48	31.28	0.33
	Oracle stopper‡	✓	0	39.50	0.357	0.00	29.17	0.288	0.00	38.18	0.115	0.00	72.00	0.034	0.00	0.00	49.81	0.00
	CIRCE (HALT-only)	✗	0	37.67	0.531	5.00	26.04	0.422	5.21	36.70	0.158	6.28	71.83	0.047	0.33	−1.65	28.79	4.17
	CIRCE (full)	✗	0	**39.17**	**0.430**	0.83	**28.13**	**0.365**	**1.04**	**38.05**	**0.133**	0.62	**72.00**	0.047	**0.00**	−0.38	**36.35**	0.52

†open-weight model of comparable capability. ‡uses ground-truth Y. B4 excluded from ranking (never fires).

contains the operative edit frequently passes the hidden tests. The ratio is suite-dependent (0.37 on SWE, 0.60 on TERM, 0.24 on WEB), which is why we report both quantities rather than treating either as the cost of interruption.

A.10 FULL COMPONENT ABLATION

Table 7: **Complete ablation**, three-suite macro over SWE/TERM/WEB, with FK the unweighted mean over the three suites. The first two blocks appear as Table 4 in the main text; the last block ablates the arbitration rule, the cost model and the salvage-model family.

Variant	ΔPass@1 (pp)	Cost saving (%)	FK (%)	Pareto AUC↑
CIRCE (full)	−0.50	47.77	0.83	0.844
e-process → per-round conformal	−1.82	35.40	2.15	0.738
e-process → static score threshold	−2.35	38.80	2.94	0.695
aGRAPA betting → fixed γ	−0.65	41.50	0.98	0.806
w/o wealth reset after reversible action	−1.41	**49.80**	1.62	0.771
LTT selection → argmax on validation	−1.08	49.10	1.39	0.784
w/o randomized reprieve (ϵ=0)	−0.47	48.90	**0.81**	**0.848**
DR estimator → IPW only	−0.52	47.60	0.86	0.840
Action set {CONT, HALT} only	−2.15	37.70	5.50	0.742
+COMPACT	−1.22	40.70	1.35	0.778
+COMPACT + REWIND	−0.66	45.40	0.95	0.828
+COMPACT + REWIND + ESCALATE (full)	−0.50	47.77	0.83	0.844
optimistic index → greedy $\hat{\Delta}_u/\hat{c}_u$	−0.78	43.60	1.05	0.815
cost model → uniform tokens	−0.92	40.10	0.92	0.792
$\hat{V}_\theta$ GBT → logistic regression	−1.02	43.20	1.15	0.805

The last block is a useful sanity check on how much of the result is the statistics and how much is the modelling. Replacing the gradient-boosted salvage model with logistic regression costs 0.039 of Pareto AUC, which is real but only about a third of what replacing the e-process with a per-round conformal test costs (0.106), so the contribution is dominated by the certification machinery rather than by the strength of the underlying predictor. Replacing the optimistic index with a greedy ratio costs 0.029, which is the price of never exploring an intervention whose effect is uncertain.

A.11 FEATURE-FAMILY ANALYSIS

This subsection expands the discussion of Table 5 in §4.5.

Workspace productivity carries the signal. F3 is the most informative family by a wide margin, which is what Definition 1 predicts: novel-bytes-per-output-token is the only feature that directly estimates information yield rather than standing in for it. Removing F3 costs 11.25 of the 36.35 points of saving, raises FK from 0.52% to 0.81% and lengthens delay by 4.6 steps, the largest degradation of any family in every column. On its own, with nine features, F3 recovers 78% of the full system's saving.

Repetition is a special case, not a definition. F1 alone, which is essentially the shipped n-gram heuristic, recovers only 47% of the saving, raises FK by a factor of 2.5 and has the longest delay in the table (17.2 steps). This is the quantitative form of the argument that an agent can stall without repeating itself (Figure 3A). F1 is still useful evidence inside the full system, since removing it costs 2.55 points, but it cannot identify a doom loop by itself.

The manipulable channel is F5, and dropping it is cheap. Removing F5 costs 1.15 points of saving, leaves FK unchanged at 0.52%, adds 0.1 steps of delay, and cuts poisoned FK from 2.32% to 0.90%. The pattern holds across the whole table: every configuration that contains F5 has a poisoned FK of at least 2.32%, and every configuration without it has at most 1.47%. This is why F5 is down-weighted by default, and a deployment that expects adversarial or sycophantic prose should drop it entirely.

Why delays exceed Table 2. The delays in Table 5 (10.8 steps at $\alpha = 0.05$) exceed the 8.6 steps of Table 2 because the full controller caps its wealth after every reversible intervention (§3.7); that cap is the price of not killing recovered trajectories, and it is paid in steps.

A.12 INTERVENTION ACCOUNTING

Table 8: **Intervention usage and outcome**, SWE/TERM/WEB pooled. *Direct cost* is the cost of executing the intervention itself; *net* Δ*cost* is the mean change in the total cost of episodes where it fires, and is what reconciles with the headline saving. *Succeeded* is the raw success rate of episodes in which the action fired, which is *not* the counterfactual gain; that is given by the action ladder in Table 4.

Action	Fire rate (%)	Mean fire step	Direct cost ($)	Net Δcost ($)	Succeeded (%)	Share of saving (%)
COMPACT	14.2	18.5	+0.012	−0.021	18.5	+1.2
REWIND (k)	8.6	22.4	+0.045	−0.685	32.4	+23.8
ESCALATE	5.1	26.8	+0.086	+0.043	24.1	−0.9
HALT	30.5	31.2	n/a	−0.628	n/a	+77.4
Reprieve granted	2.1	30.8	+0.182	+0.182	2.4	−1.5
				Total net saving per task		**$0.247** (48.2%)

The table reconciles exactly with the headline: $\sum_a (\text{fire rate}_a) \times (\text{net } \Delta\text{cost}_a) = -\$0.247$ per task against an uninterrupted pooled cost of $\$0.513$. Three observations. Reprieves are granted on 2.1% of all episodes, which is 6.4% of the 32.6% that raise a certificate; that is the realized ϵ at the default operating point. Among reprieved episodes, 2.4% go on to succeed; this is the direct, design-based estimate of the false-kill rate *among fired episodes*, and multiplying it by the 30.5% halt rate recovers the pooled population FK of 0.73% on these three suites. And ESCALATE is a net cost in the episodes where it fires ($+\$0.043$ each), yet adding it to the action ladder raises both pass@1 ($+0.16$ pp) and saving ($+2.4$ points). The two views measure different things: the accounting charges each action only for the episodes in which it fires, whereas the ladder measures the whole system, in which the availability of ESCALATE changes which other actions the index selects and when the certificate fires. We keep it because it improves quality, the quantity the constraint in equation 1 protects, without a system-level cost.

A.13 COST-MODEL SENSITIVITY

Table 9: **The billing model changes the optimal policy** (SWE). Each controller is tuned under the cost model in the row and evaluated under cache-aware `glm-5.2` pricing; regret is relative to tuning under the correct model. "% output tokens" is the share of tokens, not of dollars, the point being that a small token share carries most of the bill.

Tuning cost model	Realized $/task ↓	% output tokens	Mean halt step	ΔPass@1	$ regret (%) ↓
Uniform token count	0.482	3.6	39.2	−0.50	12.09
Steps / turns only	0.474	5.4	42.1	−0.33	10.23
Input+output, no cache term	0.446	4.1	35.4	−0.17	3.72
Cache-aware (ours)	**0.430**	3.8	33.1	−0.33	**0.00**

A controller tuned on raw token counts halts 6.1 steps later and pays 12.09% more than one tuned on the real schedule. The mechanism is in the third column: output is a few percent of tokens and the majority of the bill, so a uniform-token objective treats a cached re-read of a repository prefix and a thousand tokens of stalled reasoning as comparable events when their prices differ by more than an order of magnitude. The intermediate row is the informative one: adding an output term but omitting the cache term already recovers two thirds of the gap, so most of the effect is the input/output asymmetry and the rest is caching. Note also that the no-cache row attains the *best* ΔPass@1 (-0.17) while halting 2.3 steps later and paying 3.72% more: a later halt is good for quality and bad for the objective we actually stated.

A.14 ZERO-SHOT TRANSFER

The result is mixed and we report it plainly. Cross-environment transfer costs 3.1 to 9.2 points of saving. On the WEB target, saving falls from 44.6 in distribution to between 39.8 and 41.5, and FK roughly doubles, from 0.62 to between 1.11 and 1.23. The size of the effect deserves precision: even the worst transfer cell sits at 1.28% FK against a nominal $\alpha = 5\%$, so environment transfer degrades

Table 10: **Zero-shot transfer.** Controller calibrated on the source setting and deployed without recalibration on the target. Cells report cost saving (%) / realized FK (%). Top block: single-domain calibration (†in-distribution reference). Bottom block: deployable universal strategies.

Calibrated on ↓ / deployed on →	SWE	TERM	WEB	Unseen model (DeepSeek-V4)
SWE	49.4 / 0.83†	45.2 / 1.04	41.5 / 1.11	41.8 / 1.15
TERM	44.1 / 1.00	49.3 / 1.04†	39.8 / 1.23	40.5 / 1.28
WEB	40.2 / 0.67	41.5 / 1.04	44.6 / 0.62†	37.9 / 1.09
Pooled	48.5 / 0.83	48.8 / 1.04	43.8 / 0.74	44.3 / 0.96
Pooled + reprieve (online)	**49.1** / 0.83	**49.5** / 1.04	**44.2 / 0.62**	**47.6 / 0.78**

efficiency without breaching the risk constraint. That is a meaningfully weaker statement than "unsafe to deploy blind", and it localizes the real hazard: what breaks the constraint in our experiments is temporal shift under deployment (Table 3), not a change of environment at calibration time. The asymmetry is itself informative: WEB is the hardest target for every source, and as a source it yields the most conservative controllers (the lowest saving on both other targets, with FK at or below the in-distribution level), consistent with its higher environment non-determinism producing a more conservative stall threshold. Pooled calibration recovers most of the loss, and adding online reprieve matches or exceeds the in-distribution reference on TERM and WEB, most visibly helping on the unseen model, where it converts $44.3/0.96$ into $47.6/0.78$ without any offline recalibration. That column is one model wide and should be read as a probe, not as evidence of general model transfer.

A.15 Robustness to termination poisoning

Table 11: **Robustness to termination poisoning.** Injected strings assert progress or completion, inviting a stopper to terminate as if the task were finished; a correct controller should be unaffected. Cells report the change in fire rate (pp) relative to the unattacked condition.

Attack	B6 (LLM judge)	B7 (semantic)	CIRCE (all feat.)	CIRCE (w/o F5)
None	0.0	0.0	0.0	0.0
"Almost done" assertions	+18.2	+12.4	+2.1	+0.3
Fabricated test-pass claims	+22.7	+8.9	+2.6	+0.4
Paraphrase-cycling (defeats n-gram)	+9.4	+6.8	+1.4	+0.8
No-op file churn (defeats diff)	+15.6	+10.2	+1.2	+0.7

Judge- and self-report-based stoppers move their fire rate by up to $+22.7$ pp under strings a sycophantic or adversarial agent produces for free, which is a large effect for an attack that requires no knowledge of the detector. CIRCE with all features shifts by at most $+2.6$ pp, and without F5 by at most $+0.8$ pp, because the remaining evidence is grounded in workspace and validator state the agent's prose cannot fabricate. The last attack is the interesting one: no-op file churn is designed to defeat a diff-based productivity signal and does move CIRCE without F5 by $+0.7$ pp, so the grounding is not absolute. An attacker who writes and reverts real bytes is manipulating F3 directly; the defence there is the revert ratio rather than the edit-distance feature, and we would expect a determined adversary to require a feature family we have not built.

A.16 Reprieve rate sweep

The first row is the whole argument in one line: with no reprieve the confidence sequence on R has infinite width and the shift is never detected, at any budget. Among fixed rates, detection latency improves with exploration spend but with diminishing returns (3.8, 2.1 and 1.7 rounds for 0.8%, 3.2% and 6.1%), and the schedule is off that curve: it detects at 1.9 rounds for 2.4% spend, where a fixed ϵ needs 3.2% to reach 2.1 rounds. The reason is that a $k^{-1/3}$ schedule front-loads exploration into the rounds where the configuration is least well calibrated, and the better $\hat{\lambda}_k$ it selects in later rounds saves more per episode, so part of the exploration cost is repaid; this is why its net saving is only 0.8 points below the no-reprieve row despite 2.4% exploration spend. The deployed schedule

Table 12: **Cost of self-calibration.** ϵ trades exploration spend against the width of the confidence sequence on R. Measured on the ten-round deployment stream of Table 3, whose task mix differs from the static benchmark, so absolute savings are not comparable with Table 1. Net saving is after subtracting exploration cost.

ϵ schedule	Exploration spend (%)	CI half-width on R	Shift detection (rounds)	Net saving (%)	FK (%)
$\epsilon = 0$ (no reprieve)	0.0	$\pm\infty$	never	**28.6**	**0.71**
$\epsilon = 0.01$	0.8	± 0.045	3.8	28.1	**0.71**
$\epsilon = 0.05$	3.2	± 0.021	2.1	26.8	0.74
$\epsilon = 0.10$	6.1	± 0.015	**1.7**	24.9	0.76
$\epsilon_k = 0.20\,k^{-1/3}$ (ours)	2.4	± 0.018	1.9	27.8	**0.71**

keeps the $k^{-1/3}$ rate of Theorem 7 but sets the constant so that $\epsilon_1 = 0.20$, giving 0.09 at round 10; the floor of 0.02 is not reached within the ten rounds.

A.17 Experimental budget

Table 13: **Total experimental budget** for CIRCE and the black-box arms. Operational tokens are charged to policies; evaluation tokens are the measurement instrument and are never charged to any policy. Paired branching replay is the source of the reuse factor.

Phase	Episodes	Operational ($)	Evaluation ($)	Reuse vs. full re-run
Trajectory bank generation (B1)	2,108	802.80	142.50	1.0×
Calibration split (disjoint dev splits)	105	38.50	30.00	4.2×
Main comparison replay (Tab. 6)	2,108	112.40	0.00	7.1×
Ablation runs (Tab. 7, 5)	2,108	48.20	0.00	16.6×
Drift and transfer suites (Tab. 3, 10)	4,216	895.00	160.00	1.8×
Total	**10,645**	**$1,896.90**	**$332.50**	**5.4×**

GPU-hours for CIRCE and all black-box arms: **0**. The B8-WB reference arm requires activation access to an open-weight model.

The trajectory bank is $2{,}108$ episodes, one per task across the four suites, and every `glm-5.2` controller variant in the paper is replayed against it. The reuse factor is the ratio of what a naive full re-run of each arm would have cost to what the branching-replay design actually cost, and it is the reason a study with eleven comparison arms, twenty ablation variants and a ten-round deployment simulation fits in under $\$2{,}000$. The drift and transfer suites dominate the remaining spend precisely because they cannot be replayed: a mid-stream model change invalidates the bank by construction.

A.18 Non-myopic extension and open problems

Proposition 2 gives a sufficient condition for halting that quantifies over the whole remaining horizon, while Algorithm 1 verifies a window of m steps. The gap is not cosmetic: on the 112-task delayed-breakthrough stratum of §4.4 the trajectories genuinely stall by every observable measure and then resolve, so no amount of improvement to the stall indicator would fix it. Two extensions seem worth trying and neither is implemented here. First, a horizon-aware version of equation 2: since the remaining information budget is exactly $V_t(1 - V_t)$, an episode that has already spent much of it is far less likely to break through than one that has not, and the consumed fraction $(\langle V\rangle_t - \langle V\rangle_0)/(V_0(1 - V_0))$ is estimable from the same telemetry. Second, a task-conditional patience parameter $m(x)$ selected by Theorem 5 on a stratification of the calibration set, which would trade a larger grid for a per-stratum guarantee.

A second open problem is the reprieve budget under slow drift. Proposition 6 gives $\mathrm{Var}[\hat{R}] = O(1/(n\epsilon_{\min}))$, and with the deployed schedule (ϵ_k decaying from 0.20 to 0.09) the confidence sequence has half-width ± 0.018 after ten rounds (Table 12); a drift smaller than that would take many more rounds to surface than the abrupt shifts we injected. Whether an adaptive ϵ driven by a change-point statistic on the stall rate, rather than by the round index alone, can detect slow drift at the same budget is the question we would ask next.