

Many Steps into One: Reasoning Contraction and the Limits of Safe Memory Reuse for LLM Agents

Jianghong Zhou¹ Hao Rong² Chen Lin¹ Daoan Zhang¹
¹ByteDance ²Scale AI

Abstract

Thought templates, skill libraries, agent memories and caches turn many reasoning steps into one. We call this *reasoning contraction*. A contracted memory saves reasoning steps and errors until its facts change, then quietly answers wrong. Most systems trust memories by counting successes and failures. We prove that such counters are badly tuned change detectors: a symmetric counter notices a partial change only after a power of its false-alarm horizon, and an unclipped one may never notice it. On five LLMs answering multi-hop questions over edited facts, ExpeL- and ACE-style counters serve $7.0\text{--}13.5\times$ more stale answers than certified memories given the same information, and on code tasks with an exact verifier $35\text{--}73\times$ more. Safety also has a price. When facts change and each memory is judged only by its own verified outcomes, one number, the *certification load*, sets how much reasoning any safe memory can skip, and under Zipf demand it yields a scaling law. Our recipe weighs each memory’s failures by its own likelihood ratio, certifies its utility against live reasoning, and restarts the certificate when facts change, with the likelihood-ratio detector as the safety net for unannounced changes; for large edits it also invalidates the memory.

1 Introduction

Everyone builds reasoning memories. Thought templates (Yang et al., 2024a), reasoning and workflow memories (Ouyang et al., 2026; Wang et al., 2025b; Suzgun et al., 2026), behavior handbooks (Didolkar et al., 2025), skill libraries (Wang et al., 2024a, 2025a), answer caches (Bang, 2023; Schroeder et al., 2026), and the memories that agents share (Zhang et al., 2025a) or rewrite themselves from (Zhang et al., 2026a) all do one thing: they compile a derivation of $m \geq 2$ steps into a one-step *contracted memory* for later instances of the problem. We call this *reasoning contraction*.

Trusting such a memory is a sequential test, bound by the limits of change detection and certification.

Live reasoning is costly and fragile. LLM reasoners buy accuracy with computation: voting over sampled chains (Wang et al., 2023), tree search (Yao et al., 2023a), tool calls (Yao et al., 2023b) and self-critique (Shinn et al., 2023; Madaan et al., 2023). Every step is another chance to err. On the unedited facts of our multi-hop benchmark, Qwen3-8B’s live accuracy falls from 0.63 at two hops to 0.31 at three and 0.19 at four (Appendix D). A verified contracted memory answers in one step; on our edited streams it raises Qwen3-8B’s accuracy from 0.379 to 0.648 with fewer calls (Table 3). The price is a new failure mode shared by all these systems: when the facts behind a memory change, it keeps answering, wrongly. This is the utility problem of macro learning (Minton, 1990), now for LLM memories over changing facts.

Memories are trusted by counters, which are badly tuned change detectors. ExpeL and ACE vote memories up and down (Zhao et al., 2024; Zhang et al., 2026b), and validation-gated memories clip a Hebbian strength (Singh and Yu, 2025). Every clipped counter is a cumulative-sum (CUSUM) change detector in disguise, testing an *implied* change that nobody chose (Section 3). A symmetric counter assumes that a stale memory is as often wrong as a fresh one was right, so counting votes equally is provably the wrong detector for *partial* staleness: when an edit leaves a memory failing less than half the time, the counter’s evidence drifts the wrong way, and its delay grows as a power of its false-alarm horizon, not its logarithm (Theorem 1, Figure 1b). Unclipped counters are worse: credit banked while the memory was healthy must be spent before retirement. On KQA-STREAM, multi-hop questions over controlled knowledge-base edits, ExpeL- and ACE-style counters leave about half of the same 156

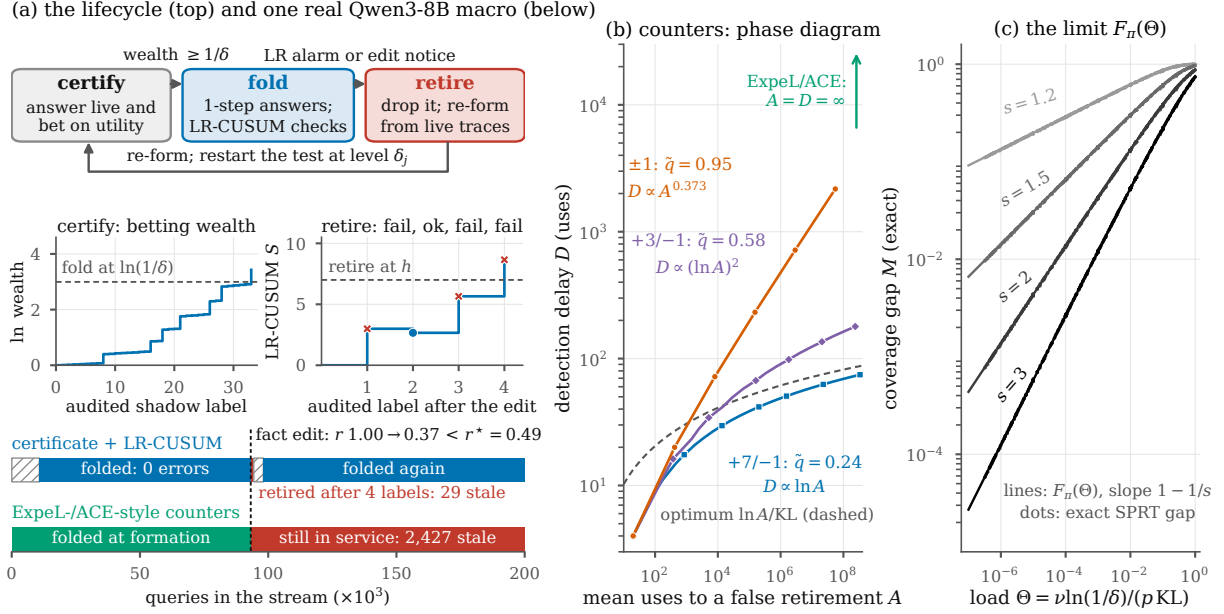


Figure 1: **Trusting a contracted memory is a sequential test.** (a) Top: the lifecycle of a contracted memory (a *macro*). In *shadow* mode a betting certificate scores the macro’s verified answers and *folds* it (one-step answers) once its wealth reaches $1/\delta$; a likelihood-ratio CUSUM (LR-CUSUM) alarm or an edit notice retires it, and the next candidate restarts the test. Middle: both tests on one real macro (Qwen3-8B, KQA-STREAM; per candidate, no notices). Its log-wealth reaches $\ln 20$ ($\delta = 0.05$) after 33 labels; an edit then drops its reliability from 1.00 to 0.37, below break-even ($r^* = 0.49$), and the LR-CUSUM crosses $h = 7$ on the fourth label. Bottom: its timeline (hatched: certification); ExpeL- and ACE-style counters have not retired it by stream end (2,427 stale answers, against 29). (b) Exact detection delay D against uses to a false retirement A for three clipped counters (score per failure/success), failure rate $0.05 \rightarrow 0.25$: the symmetric ± 1 rule is slow by a power of A ; likelihood-ratio weights ($+7/-1$) track the optimum. $\tilde{q}$: the failure rate a rule implicitly tests (Remark 1). (c) The exact coverage gap M (share not answered in one step) of a restarted test for 2,520 settings (dots) lies on the limit $F_\pi(\Theta)$ (lines).

stale memories in service at stream end, against 2% for a likelihood-ratio rule, and need about $60\times$ more uses to retire them (restricted mean; Figure 2). On five open LLMs they serve $7.0\text{--}13.5\times$ more stale answers than our certificate given the same information (no notices; Table 2); on code, larger still (Section 6).

Safety has a price. Can a memory be safe and still skip most reasoning? One number decides: the *certification load* $\Theta = \nu \ln(1/\delta)/(p \text{KL})$, which combines the rate of change (ν), the risk tolerated (δ), the audit rate (p) and how much one verified outcome reveals (KL). Whatever its algorithm, every safe memory must, in the worst case and as $\delta \rightarrow 0$, leave a share $F_\pi(\Theta)$ of queries to multi-step reasoning, and a restarted test achieves it (Theorem 3, Figure 1c). Under Zipf demand it scales as $\kappa_s \Theta^{1-1/s}$ (Corollary 4): popular problems can be contracted; rare ones change before they can be trusted.

The recipe. Three changes turn a counter into a *certificate*, a rule that lets a memory answer in one step only while the risk of trusting a bad one stays below δ (Section 5). (i) Weigh each memory’s failures against its successes by its own likelihood ratio, set by its healthy and stale failure rates; on KQA-STREAM this detects staleness in $1.3\text{--}5.9\times$ fewer uses than the best tested fixed weighting. (ii) Certify utility, not accuracy, by betting on the paired difference with live reasoning on the same query, with no estimate of live accuracy. (iii) Restart a memory’s certificate whenever its answers or the live reasoning it competes with change, so every trust decision rests on current evidence. Where edits are substantial, our deployment also invalidates memories an edit announces, and the likelihood-ratio detector is the safety net for unannounced edits; it cuts stale answers $9\text{--}12\times$ against invalidation alone. On five models it trusts under one below-break-even memory per stream (with notices; counters: 111–595), and, scoring answers alone, makes no false certification at $\delta \leq 0.1$.

Agent memories and skill libraries with a verifier should therefore adopt these changes, and memory benchmarks should edit facts.

Contributions and what is new. Our results apply when facts can change and each memory can be judged only by its own verified outcomes. (1) *A concept*: reasoning contraction, one lens on templates, skills, memories and caches, whose two risky decisions, trust and retirement, are sequential tests bound by the limits of change detection and certification (Figure 1a); done safely, KQA-STREAM’s symbolic macros add 14–33 points over live reasoning. (2) *Two theorems*: the phase diagram of counter maintenance (equal votes: delay $\asymp A^{0.373}$, not $\ln A$, at failure rate $0.05 \rightarrow 0.25$), with the learning rates that fix it (Theorem 1 and Corollary 2), and the certification-load limit with its scaling law $\kappa_s \Theta^{1-1/s}$, matched within $0.945\text{--}1.025\times$ by a restarted test (Theorem 3 and Corollary 4). (3) *A certificate*: a per-template certify–retire–restart lifecycle from standard anytime-valid tools (Yu et al., 2026b), tested on KQA-STREAM (with invalidation, no stale answers under notices; $7.0\text{--}13.5\times$ fewer than counters without), natural Wiki-data edits and code. Where memory systems vote and anytime-valid routing certifies one global threshold, we test each memory’s trust over its lifecycle: memory systems with a verifier should give each memory its own weight and certificate, and price skipped reasoning by Θ .

2 Contraction as a memory lifecycle

Queries, templates, macros. A reasoner serves a stream of queries. Each query instantiates a *template* i , a canonical derivation skeleton such as a relation path in a knowledge base, a lemma chain, or a tool-call sequence. Template i is drawn with probability π_i and has a *live* derivation of $m_i \geq 2$ steps (LLM calls). A *macro* for template i is a one-step operator formed from a verified derivation: a materialized composed relation, a stored lemma, or a distilled direct-answer note. Its *reliability* r is the probability that it answers a random instance of i correctly. On each query the system answers *live* (cost m , correct with probability a), *folded* (cost 1, answer from the trusted macro), or in *shadow* mode (live answer, macro evaluated alongside). A verifier (exact match, execution, or a unit test) labels the macro’s answer on *audited* queries, a fraction $p \in (0, 1]$ of the template’s shadow or folded queries. With error price L (in steps), a

folded query costs $1 + L(1 - r)$ and a live one $m + L(1 - a)$, so folding pays if and only if

$$r > r^* := a - (m - 1)/L. \quad (1)$$

The lifecycle. Contraction is a memory lifecycle (Figure 1a): *form* a candidate from a verified trace, *certify* it in shadow mode, *fold* it, *monitor* it, *retire* it when it goes stale, and *re-form*. Two decisions carry all the risk: *when may a candidate be trusted* (certification) and *when has a trusted macro gone stale* (retirement). Verifier outcomes $Y \in \{0, 1\}$ of a macro are conditionally Bernoulli given the past, with mean equal to its current reliability; this allows adaptive usage and drift. A macro is *stale* once an edit has raised its failure rate: healthy macros fail with probability q_0 , stale ones with $q_1 > q_0$.

Maintenance rules in current systems. ExpeL gives a new insight importance 2, adds 1 per up-vote or edit, subtracts 1 per down-vote and deletes it at 0 (Zhao et al., 2024). ACE keeps unbounded helpful/harmful counts, and its released code flags a bullet when harmful $\geq$ helpful and harmful > 0 (Zhang et al., 2026b); our ACE-style baseline treats a flag as retirement. We idealize their LLM-issued votes as one vote per verified use. Validation-gated Hebbian memories clip a strength to $[0, 1]$ with time decay (Singh and Yu, 2025); we analyze the symmetric variant $c \leftarrow \min\{1, c + \eta_+\}$ on success and $c \leftarrow c - \eta_-$ on failure, retiring below τ .

3 Memory counters are change detectors

A *score rule* assigns a bounded score $f(Y)$ to each verifier outcome, keeps

$$S_n = \max\{0, S_{n-1} + f(Y_n)\}, \quad (2)$$

and retires the macro at the first n with $S_n \geq h$. Clipped Hebbian confidence has this form. With $S = (1 - c)/\eta_-$ the score is $+1$ on a failure and $-\rho$ on a success, $\rho = \eta_+/\eta_-$, and the clip at full confidence is the reflection at 0 (Appendix A). Let P_0, P_1 be the outcome laws of a healthy and a stale macro, with $\mathbb{E}_0 f < 0 < P_0(f > 0)$, and let $\theta_j > 0$ solve $\mathbb{E}_j e^{\theta_j f} = 1$ (the Cramér–Lundberg *adjustment coefficient* under P_j). We measure a rule by A , the mean number of healthy uses before it falsely retires a macro, and by D , its worst-case number of uses to detect a change (Lorden, 1971).

Remark 1 (Implied alternative). Every score rule (2) is exactly Page’s CUSUM (Page, 1954) for the change $P_0 \rightarrow \tilde{P}_0 := e^{\theta_0 f} P_0$ and for no other

change, as for quality-control charts (Munford, 1980; Brook and Evans, 1972). The symmetric Hebbian rule ($\rho = 1$) thus tests failure rate q_0 against its *mirror* $1 - q_0$: it assumes that a stale memory is as wrong as a fresh one was right (reliability $0.95 \rightarrow 0.05$), whereas a fact edit may corrupt only part of a template's instances (reliability $0.95 \rightarrow 0.75$).

Theorem 1 (Phase diagram of maintenance rules). *Let $\mu_1 = \mathbb{E}_1 f$ and $\gamma = \theta_1/\theta_0$. As $A \rightarrow \infty$,*

$$\begin{aligned} \mu_1 > 0 : \quad D &= \frac{\ln A}{\theta_0 \mu_1} + O(1); \\ \mu_1 = 0 : \quad D &= \frac{(\ln A)^2}{(\theta_0 \sigma_1)^2} + O(\ln A), \quad \sigma_1^2 = \text{Var}_1 f; \\ \mu_1 < 0 : \quad D &\asymp A^\gamma \text{ if } P_1(f > 0) > 0, \end{aligned}$$

where $\gamma < 1$ if and only if $\mathbb{E}_1 e^{\theta_0 f} > 1$, e.g. whenever $q_1 > q_0$ for Bernoulli rules. In the first regime the efficiency relative to the optimal delay $\ln A / \text{KL}(P_1 \| P_0)$ (Lorden, 1971) is

$$\text{Eff} = \frac{\theta_0 \mu_1}{\text{KL}(P_1 \| P_0)} = 1 - \frac{\text{KL}(P_1 \| \tilde{P}_0)}{\text{KL}(P_1 \| P_0)}, \quad (3)$$

so $\text{Eff} \leq 1$, with equality if and only if f is proportional to the log-likelihood ratio. When it applies: any clipped counter with a bounded score per verified outcome, when outcomes are i.i.d. before and after one change (constants in Appendix A).

Why three regimes. After a change the score drifts at rate μ_1 . With positive drift it climbs to the threshold $h \approx \ln A / \theta_0$ in linear time, hence $\ln A$. With zero drift it diffuses and needs time h^2 . With negative drift it must make a rare excursion, like a false alarm but at the post-change exponent $\theta_1 < \theta_0$, hence $e^{\theta_1 h} = A^\gamma$. The sign is that of $\theta_0 \mu_1 = \text{KL}(P_1 \| P_0) - \text{KL}(P_1 \| \tilde{P}_0)$: a rule detects a change quickly only if the change moves the outcome law closer to the rule's implied alternative than to the healthy law. For the symmetric rule this requires $q_1 > \frac{1}{2}$. Partial staleness (reliability $0.95 \rightarrow 0.75$) gives $\gamma = \ln 3 / \ln 19 = 0.373$ (Figure 1b). The logarithmic regime is known (Section 7); the power-law regime and the unified diagram are new. *Why it matters:* under partial staleness, equal votes wait A^γ uses when clipped, not $\ln A$, and with positive probability forever when unclipped (Remark 2).

Corollary 2 (Likelihood-ratio learning rates). *A clipped Hebbian rule is the likelihood-ratio CUSUM for $q_0 \rightarrow q_1$, optimal at the false-alarm horizons its thresholds attain, if and only if*

$$\frac{\eta_+}{\eta_-} = \rho(q_0, q_1) := \ln \frac{1 - q_0}{1 - q_1} / \ln \frac{q_1}{q_0}. \quad (4)$$

When it applies: a memory whose healthy and stale failure rates $q_0 < q_1$ can be estimated (if only an interval for q_1 is known, tuning at its lower end is first-order minimax; Appendix A).

We use the Bernoulli likelihood-ratio CUSUM (LR-CUSUM) of Reynolds and Stoumbos (1999) to set memory learning rates. Under partial staleness ($q_0 = 0.05$, $q_1 = 0.25$), $\rho = 0.147$: a failure must weigh about 7 successes, not 1 as in ExpeL and ACE. Scores $+7/-1$ detect in 41 and 54 uses at $A \approx 1.6 \times 10^5$ and 3×10^6 , $5.7\times$ and $13\times$ faster than the symmetric rule (Figure 1b). *Why it matters:* weighting each memory by its own ρ makes it the optimal detector of its $q_0 \rightarrow q_1$; if memories' ρ differ, no global weight is optimal for all.

Remark 2 (Unclipped counters). An unclipped counter such as ExpeL's or ACE's (Section 2) rarely retires a healthy memory ($A = \infty$) but has no finite worst-case delay: credit banked before a change must be spent first, at rate $2q_1 - 1$. Under partial staleness ($q_1 < \frac{1}{2}$) the stale memory is *never* retired with probability at least $1 - 2q_1$ (ACE) or $1 - (q_1/(1 - q_1))^2$ (ExpeL), i.e. 0.50 and 0.89 for reliability $0.95 \rightarrow 0.75$ (Proposition 9 in Appendix A).

4 The price of safety: how far can reasoning contract?

How large a share of queries can *any* safe system answer in one step when facts keep changing?

Model. Queries of template i arrive at rate π_i ($\sum_i \pi_i = 1$), and each is audited with probability p ; an audit reveals one verifier outcome of the candidate it evaluates. Each template's facts change at rate ν . After a change the current macro is invalid, and a candidate must earn trust in the new epoch. A candidate is *good* if its reliability is $\geq r_1$ and *bad* if it is $\leq r_0 < r_1$ (e.g. $r_0 = r^*$ of Equation (1)). A *fold event* is a time when the system starts answering a template in one step with some candidate, formed in this or an earlier epoch. The *coverage gap* M is the long-run fraction of queries *not* answered in one step by a valid, certified macro.

Definition 1 (δ -safety). A policy is δ -safe if, for every epoch, every history before it and every outcome law within it, the probability that the epoch contains a fold event of a candidate that is bad in that epoch is at most δ .

Safety is per epoch, not per candidate: folding each of many candidates blindly with probabil-

ity δ is safe per candidate but eventually folds a bad one. A sequential probability ratio test (SPRT), $\Lambda_n = \sum_{k \leq n} \ln \frac{P_{r_1}(Y_k)}{P_{r_0}(Y_k)}$, that folds when $\Lambda_n \geq \ln(1/\delta)$ and restarts at each change is δ -safe by Ville's inequality (Wald, 1945; Ville, 1939).

Theorem 3 (Fundamental limit of safe contraction). *Let $\text{KL} = \text{KL}(r_1 \| r_0)$ (Bernoulli), define the certification load*

$$\Theta := \frac{\nu \ln(1/\delta)}{p \text{KL}}, \quad (5)$$

and let $F_\pi(\Theta) := \sum_i \pi_i (1 - e^{-\Theta/\pi_i})$. (i) No safe policy does better: *every δ -safe policy has $M \geq (1 - \varepsilon_\delta) F_\pi(\Theta)$, where $\varepsilon_\delta \rightarrow 0$ as $\delta \rightarrow 0$, whether or not changes are announced and however many candidates it forms.* (ii) A restarted test achieves it: *the SPRT restarted at each change is δ -safe and has $M \leq (1 + \ln(r_1/r_0)/\ln(1/\delta)) F_\pi(\Theta)$. When it applies: facts change at random times, each audit reveals one outcome, only audited outcomes are informative (shadow agreement excluded; Theorem 22), and good memories have reliability only r_1 (the worst case; exact model in Appendix B).*

Why $F_\pi(\Theta)$. A good candidate cannot be told apart from a bad one before its log-likelihood ratio reaches $\ln(1/\delta)$, which takes $n^* = \ln(1/\delta)/\text{KL}$ audited uses, i.e. time $n^*/(p\pi_i)$. A query is folded only if its epoch is older than that. Epoch ages are exponential with rate ν , so this happens with probability $e^{-\Theta/\pi_i}$. Popular templates certify quickly and are almost always covered; templates with $\pi_i \lesssim \Theta$ change before certification. A change of measure makes this rigorous (Lai, 1981; Tartakovsky et al., 2014; Kaufmann et al., 2016). Under Zipf demand it becomes a scaling law:

Corollary 4 (Scaling law of one-step reasoning). *If $\pi_i = i^{-s}/\zeta(s)$ with $s > 1$, then for all $\Theta > 0$*

$$\kappa_s \Theta^{1-1/s} - \Theta \leq F_\pi(\Theta) \leq \kappa_s \Theta^{1-1/s}, \quad (6)$$

with $\kappa_s = \Gamma(1/s)/((s-1)\zeta(s)^{1/s})$ ($\kappa_2 \approx 1.38$). Hence the optimal gap is $\kappa_s \Theta^{1-1/s}(1 + o(1))$ for small δ and small load.

Reading the law. Drift, risk, audit rate, reliability gap and verifier noise (through the KL of the observed channel) enter only through Θ , and the exponent $1 - 1/s$ is set by the demand tail alone. *Why it matters:* in this model no safe algorithm, candidate supply or notice beats the limit as $\delta \rightarrow 0$; for given demand only a lower load contracts more. Figure 1c checks the law exactly: the

gap of the restarted SPRT, computed by dynamic programming for 2,520 settings of reliabilities, verifier noise, $\delta \in [10^{-32}, 0.03]$, audit and drift rates, has $M_{\text{SPRT}}/F_\pi(\Theta) \in [0.945, 1.025]$, and the fitted log-log slopes are within 0.008 of $1 - 1/s$ (finite- δ bounds: Appendix B).

5 The recipe: a certificate that needs no live accuracy

The SPRT of Theorem 3 tests $r \leq r^*$, but r^* involves the unknown live accuracy a (Equation (1)), and plugging in an estimate breaks the guarantee.

Proposition 5 (Plug-in tests are not level δ). *If the SPRT uses $r_0 = \hat{a} - \frac{m-1}{L}$ and $r_1 = \hat{a}$ with $\hat{a}$ estimated from n_a live runs, its false-fold probability at break-even can exceed δ . At $a = 0.9$ and $\frac{m-1}{L} = \delta = 0.05$ it is 0.34, 0.25 and 0.08 for $n_a = 20, 100$ and 1000 (exact; Proposition 18).*

Following testing by betting (Shafer, 2021; Waudby-Smith and Ramdas, 2024), with inverse-propensity weights as in B-PAC (Yu et al., 2026b), we bet on the paired utility of each shadow evaluation, with live reasoning on the same instance as control: $U_t = C_t^L - C_t^M + L(Y_t^M - Y_t^L)$, the steps saved plus L times the correctness gained, where C_t^L, C_t^M are the costs and $Y_t^L, Y_t^M \in \{0, 1\}$ the verified correctness of the live and macro answers; $Z_t = U_t/B \geq -1$ for a known bound B .

Theorem 6 (Betting certificate). *Fold when the wealth of predictable bets on the inverse-propensity-weighted Z_t reaches $1/\delta$. (i) Validity. A macro that is never more useful than live reasoning is folded with probability at most δ , whatever the live accuracy and however live and macro errors are correlated. (ii) Rate. If the unweighted Z_t are i.i.d. with law P_Z and audits have rate p , certification takes $(1 + o(1)) \ln(1/\delta)/(p \text{KL}_{\inf}(P_Z))$ uses, and no δ -safe rule that sees only audited utilities is faster (Kaufmann et al., 2016; Garivier and Kaufmann, 2016). Here $\text{KL}_{\inf}(P) := \inf\{\text{KL}(P \| Q) : \mathbb{E}_Q Z \leq 0\}$ over laws Q on the same bounded range: the divergence from P_Z to the nearest useless macro. When it applies: audits have known probabilities, bounded below, independent of the hidden outcome (exact conditions in Appendix C).*

Under the null, reaching $1/\delta$ is rare (Ville's inequality); a useful macro grows log-wealth by $\text{KL}_{\inf}(P_Z)$ per audited pair (Honda and Takemura, 2010). For a certificate restarted at each change, Theorem 3 holds with $\text{KL}_{\inf}(P_Z)$ for KL and unknown a (Corollary 20). Unlike a discordant-pair

Certificate	Needs	Guarantee
Restart only: ledger δ_j , restarts	notices [†]	δ -safe per epoch (Proposition 23); <i>as run</i> , growth $KL_{\text{inf}}/(1.03-1.07)$ on the laws checked, plus $2 \ln j + O(1)$ nats (Appendix C.2)
Ours (recommended): + invalidate on notice missed	notices [†] , some	as above; a missed notice adds the carried-over term of Proposition 23(iii)
No notice: discard at own alarm	none	δ -safe for folds on post-change evidence (Proposition 25); not per epoch if $\alpha_i r_0 > \delta$ (Proposition 24)
Per candidate: level δ each	none	per candidate only (Figure 1a, code)

Table 1: **Restarting at each notice makes certification safe per epoch.** Bold: ours. All retire folded macros by LR-CUSUM (Corollary 2). The *optimum*, an SPRT or grid bet restarted at known change times, attains $F_\pi(\Theta)$ (Theorem 3 and Corollary 20); *as run* is a rule’s own growth rate; *measured* rates are in Section 6. [†]Of every change of the null, including live reasoning’s (Appendix D.5.1).

test (Shawn, 2026), it can certify macros less accurate than live but above break-even (Theorem 19).

From certificate to deployment. The *restart-only* certificate (Table 1) runs a template’s j -th test at level $\delta_j = 6\delta/(\pi^2 j^2)$, so that all its tests together spend at most δ ; it *restarts*, at the next level, at each notice that the template’s answers changed, and retires a folded macro by its own LR-CUSUM (Corollary 2). Our *recommended deployment* (“ours”) also invalidates and re-forms folded macros at each notice, so the LR-CUSUM is the safety net for missed notices. *When it applies:* both are δ -safe per epoch for their own null, paired utility ≤ 0 , when notices cover every change of that null, including changes of live reasoning (Proposition 23). So one δ bounds *false certification*, a fold of a candidate whose expected paired utility is ≤ 0 at the fold, in every epoch, however many successive candidates and edits; later staleness is left to retirement (Section 3).

6 Experiments

Benchmark. KQA-STREAM is built from the KQA Pro knowledge base (Cao et al., 2022): 300 templates of 2–4 relation hops with 60 instances each (18k LLM-verbalized questions) and three sequences of 24 fact edits (Appendix D). The *live* reasoner is a hop-by-hop agent that chooses one relation per LLM call. A *macro* materializes a verified trace’s relation path as an anchor-to-answer table, reached through one routing call; it goes stale when edits change the facts it cached. A *compiled plan* stores the path and re-executes it

Model	stale per 10^4 queries			unretired at stream end		bad folds		
	ExpeL	ACE	ours ^o	ExpeL	ACE	LR	counters	ours
Qwen3-8B	967	946	72	53%	53%	3%	284–400	0.0
Qwen3-4B	795	783	85	57%	51%	0%	111–148	0.1
Granite-8B	657	650	82	22%	22%	0%	282–369	0.4
Mistral-7B	298	294	42	25%	25%	0%	486–595	0.6
OLMo-3-7B	480	445	51	64%	55%	9%	172–211	0.3

Table 2: **Counters serve $7.0\text{--}13.5\times$ more stale answers on every model, even when ours gets no edit notice.** ^oOurs with every notice missed, i.e. the counters’ information; with notices it serves none. Unretired: share of the stale episodes of Figure 2 (66, 53, 18, 8 and 11) still in service at stream end. Bad folds: per stream (ours: with notices). Bold: ours.

on the current knowledge base. Every LLM call is recorded once per model, with exact selective re-runs after edits, and policies are replayed on 200k-query Zipf streams ($s = 1.2$; 10 seeds $\times$ 3 edit sequences; 10% random audits), so all policies see identical queries, edits and audit coins. We use five open models: Qwen3-8B and Qwen3-4B (Yang et al., 2025) (thinking disabled), Granite-3.3-8B (IBM Granite Team, 2025), Mistral-7B (Jiang et al., 2023) and OLMo-3-7B (Team OLMo, 2025). Prompts, seeds and scripts are in our code release.

Policies. *Fold once* folds a verified candidate at formation and never retires it, like a semantic cache. ExpeL- and ACE-style baselines transplant each system’s retention rule onto our macros, one vote per audited outcome. Detector baselines certify with the plug-in SPRT and retire with an LR-CUSUM; the *optimistic* one folds at formation. The certificates of Table 1 run at $\delta = 0.05$; a *notice* is an edit that changes the gold answer of a query routed to the template; a *vCache-style* cache re-implements Schroeder et al. (2026) (Appendix D.2.4; more variants: Table 5). **Metrics:** *stale* answers are wrong one-step answers that were right when the macro was formed; a *bad fold* trusts a candidate below break-even; a *stale episode* runs from the edit that makes a folded macro stale to its retirement or the stream end; J_{20} is LLM calls per query plus $20\bar{m}$ times the error rate ($\bar{m}$: mean live cost; $L = 20\bar{m}$ in Equation (1); lower is better). Audits query the knowledge base at no LLM-call cost.

Counters leave stale memories in service. As Remark 2 predicts, counters serve $7.0\text{--}13.5\times$ more stale answers than ours on every model, even with every notice missed (Table 2). To isolate detection, we replay every macro that restart only folds, which

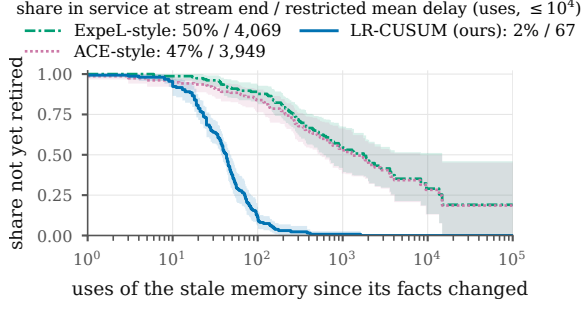


Figure 2: **Counters leave half the stale memories in service at stream end; likelihood ratios retire them about $60\times$ sooner (restricted mean).** Kaplan-Meier share of the 156 stale memories not yet retired (same audits for all rules; counters start at the fold without credit). Legend: share still in service when the stream ends (open episodes, so it differs from the curve’s tail at 10^4 uses) / restricted mean delay ($\leq 10^4$ uses). Bands: 95% template bootstrap.

only detection retires, under each rule with the same audits: 156 stale episodes (Appendix D.3.2). Counters leave 50% and 47% in service at stream end against 2% for the LR-CUSUM, with restricted mean delays of 4,069 and 3,949 uses against 67 (Figure 2); each episode costs counters about 1,350 stale answers, against 38.5. Loss hides this: their J_{20} is close to ours because, skipping certification, they answer 28–76% of queries in one step (ours: 20–53%), so benchmarks must count stale answers.

Detection follows the phase diagram. A false retirement (FR) retires a macro still above break-even. Staleness here is nearly total (mean $q_1 = 0.75$ – 0.90 , $q_0 = 0.04$ – 0.24), and Theorem 1 and Corollary 2 predict three things, all observed. (i) *The symmetric rule detects, but slowly:* with $q_1 > \frac{1}{2}$ its delay is logarithmic, but the LR-CUSUM needs 3.7 – $6.1\times$ fewer uses at 0.8 – 5 FR per 10^4 healthy uses (Figure 6). (ii) *Per-memory weights beat the best tested global one,* as q_0 varies: 1.3 – $5.9\times$ fewer uses (significant on three of five models; Appendix D.3.6); that best ratio is itself small (below 1:1 on four models; predicted $1/\rho$: 0.67 – 2.14 ; Figure 5). (iii) *Partial staleness moves the optimum up:* on 217 real partially stale episodes (predicted $1/\rho = 4.8$) the best tested weight is 5 – 7 , and the symmetric rule needs 1.2 – $1.5\times$ more uses (Appendix D.3.5). The theory thus ranks the rules on real LLM errors.

Restarted certificates are safe in every epoch. Scoring answers alone, as the certificate’s paired

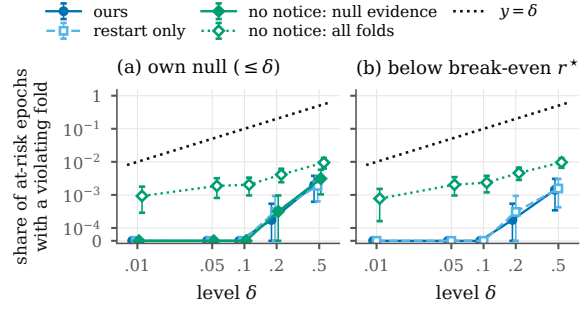


Figure 3: **False certifications stay far below δ .** At-risk epochs with a fold that violates (a) the certificate’s own null, the event δ bounds, or (b) break-even r^* ; answers scored alone; five models. Without notices, folds on null evidence stay at zero up to $\delta = 0.1$, and all folds more than $10\times$ below δ . Bars: 95% template bootstrap.

test does (Appendix D.4.1), ours makes no false certification (Section 5) at $\delta \leq 0.1$ (0 of 5,779 at-risk epochs at $\delta = 0.05$) and stays far below δ up to 0.5, whether “bad” means null at the fold or below break-even (Figure 3); counting routing errors as wrong, 0.26% of at-risk epochs have a bad fold at $\delta = 0.05$ (Appendix D.4.2). Restarts remove the 11 bad folds that candidates good at formation carry across edits without them (standard scoring; Appendix D.3.3). Although 16% of template-versions change live reasoning unannounced, none of the 9,835 answer-scored folds (2 of 8,294 under standard scoring) rests on a test spanning such a change (Appendix D.5.1). δ controls which memories are trusted; how often a trusted memory errs is set by r^* (10–11% of one-step answers, mostly from macros above break-even).

Certified contraction pays. On Qwen3-8B, ours raises accuracy from 0.379 (live) to 0.648 with 21% fewer LLM calls, and certified compiled plans to 0.668 (Table 3), because a macro executes a verified path that live reasoning re-derives with errors. Uncertified reuse (fold once) serves 1,290 stale answers per 10^4 queries; certification cuts this $20\times$, invalidation to zero. Across models ours adds 14–33 points over live reasoning (Table 6).

The detector matters exactly where notices fail. With perfect notices ours serves no stale answers and matches restart only in accuracy (+0.24 points [−0.24, +0.81], pooled). With 80%, 50% or no edits announced, invalidation alone serves 125, 325 and 810 stale answers per 10^4 queries, ours 13, 28 and 66 (Appendix D.4.2), 9–12 $\times$ fewer: invalidation needs the detector wherever notices can be

	Policy	Acc	Calls	Stale	Bad	J_{20}
no memory	live always	.379	3.26	0	0	43.8
no retire	fold once	.534	1.45	1290	34	31.8
	compiled plan, fold once	.639	1.45	95	34	24.8
	vCache-style, $\delta_c = .02$	.377	2.86	30	–	43.5
counters	ExpeL-style counter	.613	1.75	967	284	27.0
	ACE-style counter	.618	1.86	946	400	26.7
detectors	SPRT + LR-CUSUM	.622	2.64	54	0.7	27.3
	optimistic + LR-CUSUM	.632	1.58	573	82	25.5
certified	ours (recommended)*	.648	2.56	0	0.0	25.5
	restart only	.642	2.57	66	0.0	26.0
	no notice (own detector)	.650	2.52	70	0.1	25.3
	compiled plan + betting	.668	2.46	16	0.0	24.1

Table 3: **Certification removes bad folds and invalidation removes stale answers, with fewer calls than live reasoning** (Qwen3-8B, 30 streams of 200k queries). Calls: LLM calls per query. Stale: per 10^4 queries. Bad: bad folds per stream (–: no macros). J_{20} : loss, lower is better ($\bar{m} = 3.26$). Bold: ours. *Perfect notices; with every notice missed it serves 72 stale answers per 10^4 queries. Compiled plans (with restarts) gain +2.6 points [1.7, 3.8] over restart-only macros (paired 95% template bootstrap).

missed.

Natural edits: safety holds, and mild staleness shows its price. On a 1992–2016 timeline of Wikidata temporal qualifiers (four models), where changes are small and partial (median 3.3% of a template’s answers per year), ours keeps bad-fold epochs below δ at every level, the LR-CUSUM detects in $1.1\text{--}1.6\times$ fewer uses than the best tested fixed weight, and counters leave 84% of the same stale memories in service against 37% (Appendices D.4.4 and D.5.3). The price of safety (Section 4) shows: counters make 149–207 bad folds per stream against 0.1 and serve $3\times$ more stale answers than restart only, yet are 3.5 points more accurate because folding pays when staleness is mild; for the same reason invalidation costs ours 1.9 points against restart only.

On code, counters serve $35\text{--}73\times$ more stale answers. On 521 EvalPlus tasks (Liu et al., 2023) (five models), live reasoning writes, tests and repairs a program; a memory is a verified program rerun on its task’s later queries; cost counts steps, an execution being one. Memories go stale under announced requirement revisions and unannounced Python API changes, which also change live reasoning and so exercise the safety net. Ratios are per query, against our certificate given the counters’ information (per candidate, no notices), on all five models ($38\text{--}86\times$ in LLM-call accounting). Here

most stale memories never failed before ($q_0 = 0$, so $\rho \rightarrow 0$), and the per-memory LR-CUSUM leaves them folded for $10\times$ fewer uses than the best fixed weight at matched false retirements ($7.6\text{--}11.4\times$ per model, weights up to 5000:1; Appendix D.6). Ours folds a below-break-even candidate in 12 of 946,678 epochs under test (counters: $4.8\text{--}5.4\%$) and gains 8.7 accuracy points [7.1, 10.5] over live reasoning, with lower J_{20} in both units: it uses 23% fewer steps, while its shadow tests add 17% LLM calls. Zero-utility ties ($84\text{--}86\%$ of audited pairs in LLM-call accounting) make audits nearly uninformative and halve one-step coverage ($0.61 \rightarrow 0.32$), as Theorem 6(ii) predicts: ties shrink KL_{inf} and slow certification.

7 Related work and what is new

Detection. A mismatched CUSUM with positive drift, our first regime and (3), has the known delay $\ln A/(\theta_0\mu_1)$ (Molloy and Ford, 2017; Cooper and Meyn, 2024). We add the zero- and negative-drift regimes, including the finite power law A^{θ_1/θ_0} where Wu and Yang (2016, Thm. 2) state an infinite delay (counterexample: Appendix A), read deployed counters’ implied alternative and set learning rates by it. *Limit.* Ours extends crawler freshness (Azar et al., 2018), where one refresh restores a page, to items whose trust needs $\ln(1/\delta)/\text{KL}$ audited uses; the Zipf law is an occupancy sum (Karlin, 1967) at the load Θ . *Certificate.* With standard betting, inverse-propensity weights and anytime validity, B-PAC (Yu et al., 2026b) certifies one global routing threshold; new is a per-template lifecycle that certifies, retires and restarts. *Concurrent work* extends reasoning contraction to self-modification (Zhou et al., 2026a) and shared multi-agent memory (Zhou et al., 2026b). More: Appendix F.

8 Conclusion

Reusing a reasoning memory safely is a sequential test, and vote counters (ExpeL, ACE) are change detectors tuned to the wrong change. Three rules follow. (1) Weigh a memory’s failures by its own likelihood ratio: at failure rate $0.05 \rightarrow 0.25$ a failure weighs about 7 successes, not 1. (2) Certify utility against live reasoning; the load Θ prices skipped reasoning in the worst case. (3) Restart at every notice (0 false certifications in 5,779 at-risk epochs at $\delta = 0.05$, answers scored alone) and, for substantial edits, invalidate too, with the detector as safety net ($9\text{--}12\times$ fewer stale answers when no-

tices are missed). Memory benchmarks should edit facts and count stale answers.

Limitations

Scope. Our results concern memories whose answers an exact verifier checks, such as knowledge-base execution or unit tests. The evidence comes from three domains and five open models of up to 8B parameters: multi-hop reasoning over a knowledge graph with controlled fact edits, a natural timeline of Wikidata temporal qualifiers (four models; Appendix D.4.4), and EvalPlus code tasks, whose memories go stale under requirement revisions and re-enacted Python API changes (Appendix D.6). Each instance–version is decoded once; seed variance is measured on a subset (Appendix D.4.3). The theorems assume that each audit reveals one verified outcome whose law is set by the memory’s current reliability, and that facts change at random times; Appendices A to C state the exact conditions and Appendix D the replay setting. Per-epoch safety needs change notices that cover every change of the certificate’s paired-utility null, including changes of live reasoning; our notices cover answer changes (Appendix D.5.1). Some change information is necessary on every template whose epochs contain an audit with probability above δ/r_0 (Proposition 24); without an edit log, restarts triggered by the memory’s own detector keep every decision that rests on post-change evidence safe (Proposition 25).

Future work. Five directions extend this scope. (i) A formation-aware limit: on real streams, coverage also depends on how often the live agent produces a verified trace from which a candidate can form, and a limit that models formation would predict real-system coverage. (ii) Noisy and LLM-judge verifiers, which would bring text memories into reach; the certification load already prices noise through the observed channel, and a certificate built for noisy paired labels is the next step (Appendix D.2.3). (iii) Longer and denser natural edit histories, such as full knowledge-base revision logs, where change notices arrive with their own delays and errors, and real library drift on benchmarks with version-specific APIs, where staleness reaches more programs than re-enacted deprecations do. (iv) Larger models. (v) Open problems in the theory: a square-root law for adaptive audit budgets, a joint detection–certification limit that prices stale answers, the exact optimal certifier for

one template, near-optimal retirement for an unknown post-change rate by parallel or GLR rules, and the agreement-aware version of the limit.

Ethical considerations

This work analyzes when stored reasoning may be trusted, and its main practical implication is safety-positive: it bounds how often a memory-augmented reasoner serves stale or spurious shortcuts, and it shows that common counter rules may never retire a stale memory. The datasets we use (KQA Pro, CC BY-SA 4.0; MuSiQue, CC BY 4.0; Wikidata, CC0; EvalPlus, Apache-2.0, with MBPP, CC BY 4.0, and HumanEval, MIT) are public and contain no personal data beyond public facts about notable entities; our synthetic knowledge-base edits, natural-edit timeline and code requirement revisions are released as edit logs for research (under CC BY-SA 4.0 where derived from KQA Pro; our code is MIT-licensed). All questions and tasks are in English or Python. All models are open-weight and were run locally without paid APIs. AI assistants were used to draft proofs, code and text and to run experiments, and independent AI referee agents checked them; the authors take responsibility for all content. The guarantees hold only under the stated assumptions, and deploying certified memory in high-stakes settings still requires a verifier whose errors are understood.

References

- Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. 2024. [AutoMix: Automatically mixing language models](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37.
- John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. 2004. [An integrated theory of the mind](#). *Psychological Review*, 111(4):1036–1060.
- Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. 2025. [Learn Then Test: Calibrating predictive algorithms to achieve risk control](#). *The Annals of Applied Statistics*, 19(2):1641–1662.
- Mohammad Asadolahi, Amir Amini, Samira Talebi, Amirfarhad Farhadi, and Azadeh Zamanifar. 2026. [Memory reward inflation in self-improving LLM agents](#). arXiv preprint arXiv:2608.00017.

- Søren Asmussen. 2003. *Applied Probability and Queues*, 2 edition. Stochastic Modelling and Applied Probability. Springer, New York.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.
- Yossi Azar, Eric Horvitz, Eyal Lubetzky, Yuval Peres, and Dafna Shahaf. 2018. *Tractable near-optimal policies for crawling*. *Proceedings of the National Academy of Sciences*, 115(32):8099–8103.
- Taposh Banerjee and Venugopal V. Veeravalli. 2013. *Data-efficient quickest change detection in minimax settings*. *IEEE Transactions on Information Theory*, 59(10):6917–6931.
- Fu Bang. 2023. *GPTCache: An open-source semantic cache for LLM applications enabling faster answers and cost savings*. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 212–218. Association for Computational Linguistics.
- Melih Bastopcu and Sennur Ulukus. 2021. *Information freshness in cache updating systems*. *IEEE Transactions on Wireless Communications*, 20(3):1861–1874.
- D. Brook and D. A. Evans. 1972. *An approach to the probability distribution of CUSUM run length*. *Biometrika*, 59(3):539–549.
- Qianshu Cai, Yonggang Zhang, Jun Nie, Maohao Ran, Huajiang Zheng, Jun Song, Xinmei Tian, Yike Guo, and Wei Xue. 2026. *A theory of reliable self-evolution for agent harnesses*. *arXiv preprint arXiv:2609.08175*. Version 1 (8 Sep 2026) titled “Safe Harness Self-Evolution: A Theoretical Analysis of Feasibility and Limits”.
- Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. 2024. Large language models as tool makers. In *International Conference on Learning Representations*. ArXiv:2305.17126.
- Shulin Cao, Jiabin Shi, Liangming Pan, Lunyu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. 2022. *KQA Pro: A dataset with explicit compositional programs for complex question answering over knowledge base*. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6101–6119, Dublin, Ireland. Association for Computational Linguistics.
- Yash Chandak, Scott M. Jordan, Georgios Theodorou, Martha White, and Philip S. Thomas. 2020. *Towards safe policy improvement for non-stationary MDPs*. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33.
- Hao Che, Ye Tung, and Zhijun Wang. 2002. *Hierarchical Web caching systems: Modeling, design and experimental results*. *IEEE Journal on Selected Areas in Communications*, 20(7):1305–1314.
- Lingjiao Chen, Matei Zaharia, and James Zou. 2024a. *FrugalGPT: How to use large language models while reducing cost and improving performance*. *Transactions on Machine Learning Research*.
- Lingjiao Chen, Matei Zaharia, and James Zou. 2024b. *How is ChatGPT’s behavior changing over time?* *Harvard Data Science Review*, 6(2).
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Qijia Chen, Andrea Bellucci, Zhida Sun, and Giulio Jacucci. 2026. *SkillDroid: Compile once, reuse forever*. *arXiv preprint arXiv:2604.14872*.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. *Mem0: Building production-ready AI agents with scalable long-term memory*. *arXiv preprint arXiv:2504.19413*.
- Junghoo Cho and Hector Garcia-Molina. 2003. *Effective page refresh policies for Web crawlers*. *ACM Transactions on Database Systems*, 28(4):390–426.
- Nicaise Choungmo Fofack, Philippe Nain, Giovanni Neglia, and Don Towsley. 2014. *Performance evaluation of hierarchical TTL-based cache networks*. *Computer Networks*, 65:212–231.
- Edith Cohen and Scott Shenker. 2002. *Replication strategies in unstructured peer-to-peer networks*. In *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pages 177–190. ACM.
- Austin Cooper and Sean Meyn. 2024. *Quickest change detection using mismatched CUSUM (extended abstract)*. In *Proceedings of the 60th Annual Allerton Conference on Communication, Control, and Computing*, pages 1–7. IEEE. Extended version: *arXiv:2409.07948*.
- Aniket Didolkar, Nicolas Ballas, Sanjeev Arora, and Anirudh Goyal. 2025. *Metacognitive reuse: Turning recurring LLM reasoning into concise behaviors*. *arXiv preprint arXiv:2509.13237*.
- Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, Tianming Liu, Hui Liu, and Zhen Xiang. 2025. *Memory injection attacks on LLM agents via query-only interaction*. *arXiv preprint arXiv:2503.03704*.
- Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sablé-Meyer, Lucas Morales, Luke Hewitt, Luc Cary, Armando Solar-Lezama, and Joshua B. Tenenbaum.

2021. [DreamCoder: Bootstrapping inductive program synthesis with wake-sleep library learning](#). In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI)*, pages 835–850. ACM.
- Ronald Fagin. 1977. [Asymptotic miss ratios over independent references](#). *Journal of Computer and System Sciences*, 14(2):222–250.
- Jean Feng, Scott Emerson, and Noah Simon. 2021. [Approval policies for modifications to machine learning-based software as a medical device: A study of bio-creep](#). *Biometrics*, 77(1):31–44.
- Richard E. Fikes, Peter E. Hart, and Nils J. Nilsson. 1972. [Learning and executing generalized robot plans](#). *Artificial Intelligence*, 3:251–288.
- Saman Forouzandeh, Wei Peng, Parham Moradi, Xinghuo Yu, and Mahdi Jalili. 2026. [Learning hierarchical procedural memory for LLM agents through Bayesian selection and contrastive refinement](#). In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
- Massimo Gallo, Bruno Kauffmann, Luca Muscariello, Alain Simonian, and Christian Tanguy. 2014. [Performance evaluation of the random replacement policy for networks of caches](#). *Performance Evaluation*, 72:16–36.
- Haowen Gao, Haoran Chen, Can Wang, Shasha Guo, Liang Pang, Zhaoyang Liu, Huawei Shen, and Xueqi Cheng. 2026. [SelfEvoSkill: Paired execution audits for skill revision without external outcome supervision](#). arXiv preprint arXiv:2606.14239. Version 1 (12 Jun 2026) titled “SkillAudit: Ground-Truth-Free Skill Evolution via Paired Trajectory Auditing”.
- Aurélien Garivier and Emilie Kaufmann. 2016. [Optimal best arm identification with fixed confidence](#). In *Proceedings of the 29th Annual Conference on Learning Theory (COLT)*, volume 49 of *Proceedings of Machine Learning Research*, pages 998–1027. PMLR.
- Alexander Gnedin, Ben Hansen, and Jim Pitman. 2007. [Notes on the occupancy problem with infinitely many boxes: General asymptotics and power laws](#). *Probability Surveys*, 4:146–171.
- I. J. Good. 1953. [The population frequencies of species and the estimation of population parameters](#). *Biometrika*, 40(3–4):237–264.
- I. J. Good and G. H. Toulmin. 1956. [The number of new species, and the increase in population coverage, when a sample is increased](#). *Biometrika*, 43(1–2):45–63.
- Jonathan Gratch and Gerald DeJong. 1992. [COMPOSER: A probabilistic solution to the utility problem in speed-up learning](#). In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI)*, pages 235–240. AAAI Press.
- Russell Greiner. 1996. [PALO: A probabilistic hill-climbing algorithm](#). *Artificial Intelligence*, 84(1–2):177–208.
- Peter Hall and Christopher C. Heyde. 1980. *Martingale Limit Theory and Its Application*. Academic Press, New York.
- Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2025. [Training large language models to reason in a continuous latent space](#). In *Conference on Language Modeling (COLM)*.
- Andrew Heathcote, Scott Brown, and D. J. K. Mewhort. 2000. [The power law revealed: The case for an exponential law of practice](#). *Psychonomic Bulletin & Review*, 7(2):185–207.
- Junya Honda and Akimichi Takemura. 2010. [An asymptotically optimal bandit algorithm for bounded support models](#). In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, pages 67–79.
- Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. 2021. [Time-uniform, nonparametric, nonasymptotic confidence sequences](#). *The Annals of Statistics*, 49(2):1055–1080.
- Jianguo Huang, Hao Zeng, Bingyi Jing, Hongxin Wei, and Bo An. 2026. [Conditional performance guarantee for large reasoning models](#). arXiv preprint arXiv:2601.22790.
- Marcus Hutter. 2021. [Learning curve theory](#). arXiv preprint arXiv:2102.04074.
- IBM Granite Team. 2025. [Granite 3.3 8B instruct](#). Model card.
- Donald L. Iglehart. 1972. [Extreme values in the GI/G/1 queue](#). *The Annals of Mathematical Statistics*, 43(2):627–635.
- Predrag R. Jelenković. 1999. [Asymptotic approximation of the move-to-front search cost distribution and least-recently used caching fault probabilities](#). *The Annals of Applied Probability*, 9(2):430–464.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. [Mistral 7B](#). *Preprint*, arXiv:2310.06825.
- Samuel Karlin. 1967. [Central limit theorems for certain infinite urn schemes](#). *Journal of Mathematics and Mechanics*, 17(4):373–401.
- Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2023. [RealTime QA: What’s the answer right now?](#)

- In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 36.
- Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. 2016. [On the complexity of best-arm identification in multi-armed bandit models](#). *Journal of Machine Learning Research*, 17(1):1–42.
- Hamed Khosravi and Xiaoming Huo. 2026. [Conformal selective acting: Anytime-valid risk control for RLVR-trained LLMs](#). arXiv preprint arXiv:2605.20270.
- Elliot Kim, Avi Garg, Kenny Peng, and Nikhil Garg. 2025. [Correlated errors in large language models](#). In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 30038–30066. PMLR.
- Andrey Kolobov, Yuval Peres, Cheng Lu, and Eric J. Horvitz. 2019. [Staying up to date with online content changes using reinforcement learning for scheduling](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32.
- Richard E. Korf. 1985. [Macro-operators: A weak method for learning](#). *Artificial Intelligence*, 26(1):35–77.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with PagedAttention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pages 611–626.
- Tze Leung Lai. 1981. [Asymptotic optimality of invariant sequential probability ratio tests](#). *The Annals of Statistics*, 9(2):318–333.
- Tze Leung Lai. 1998. [Information bounds and quick detection of parameter changes in stochastic systems](#). *IEEE Transactions on Information Theory*, 44(7):2917–2929.
- John E. Laird, Paul S. Rosenbloom, and Allen Newell. 1986. [Chunking in Soar: The anatomy of a general learning mechanism](#). *Machine Learning*, 1(1):11–46.
- Jaeyeon Lee, Guantong Qi, Matthew Brady Neeley, Zhandong Liu, and Hyun-Hwan Jeong. 2025. [ConSol: Sequential probability ratio testing to find consistent LLM reasoning paths efficiently](#). arXiv preprint arXiv:2503.17587.
- Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, and 20 others. 2025. [MemOS: A memory OS for AI system](#). arXiv preprint arXiv:2507.03724.
- Adam Liška, Tomáš Kočíský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d’Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. 2022. StreamingQA: A benchmark for adaptation to new knowledge over time in question answering models. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 13604–13622.
- Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*. ArXiv:2305.01210.
- Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024. [Can language models learn to skip steps?](#) In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37.
- Xutong Liu, Baran Atalar, Xiangxiang Dai, Jinhang Zuo, Siwei Wang, John C. S. Lui, Wei Chen, and Carlee Joe-Wong. 2026. [Semantic caching for low-cost LLM serving: From offline learning to online adaptation](#). In *Proceedings of IEEE INFOCOM 2026 – IEEE Conference on Computer Communications*.
- Gordon D. Logan. 1988. [Toward an instance theory of automatization](#). *Psychological Review*, 95(4):492–527.
- G. Lorden. 1971. [Procedures for reacting to a change in distribution](#). *The Annals of Mathematical Statistics*, 42(6):1897–1908.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*. ArXiv:2303.17651.
- Muhammad Mansoor, Tahir Ahmad, and Yeo-Chan Yoon. 2026. [Risk-constrained freshness-aware semantic caching for open-web retrieval-augmented LLMs](#). arXiv preprint arXiv:2607.04281.
- Valentina Martina, Michele Garetto, and Emilio Leonardi. 2014. [A unified approach to the performance analysis of caching systems](#). In *Proceedings of IEEE INFOCOM 2014 – IEEE Conference on Computer Communications*, pages 2040–2048.
- David A. McAllester and Robert E. Schapire. 2000. [On the convergence rate of Good-Turing estimators](#). In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory (COLT)*, pages 1–6.

- Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. 2023. [The quantization model of neural scaling](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 28699–28722.
- Steven Minton. 1990. [Quantitative results concerning the utility of explanation-based learning](#). *Artificial Intelligence*, 42(2–3):363–391.
- Tom M. Mitchell, Richard M. Keller, and Smadar T. Kedar-Cabelli. 1986. [Explanation-based generalization: A unifying view](#). *Machine Learning*, 1(1):47–80.
- Yunxiang Mo, Donghao Zhao, and Hejia Geng. 2026. [Stable answers, unfinished reasoning: Why self-consensus is not a safe early-exit signal](#). arXiv preprint arXiv:2609.09989.
- Timothy L. Molloy. 2021. [Misspecified and asymptotically minimax robust quickest change diagnosis](#). *IEEE Transactions on Automatic Control*, 66(2):857–864.
- Timothy L. Molloy and Jason J. Ford. 2017. [Misspecified and asymptotically minimax robust quickest change detection](#). *IEEE Transactions on Signal Processing*, 65(21):5730–5742.
- George V. Moustakides. 1986. [Optimal stopping times for detecting changes in distributions](#). *The Annals of Statistics*, 14(4):1379–1387.
- A. G. Munford. 1980. [A control chart based on cumulative scores](#). *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(3):252–258.
- Allen Newell and Paul S. Rosenbloom. 1981. Mechanisms of skill acquisition and the law of practice. In John R. Anderson, editor, *Cognitive Skills and Their Acquisition*, pages 1–55. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Mesrob I. Ohannessian and Munther A. Dahleh. 2012. [Rare probability estimation under regularly varying heavy tails](#). In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, volume 23 of *Proceedings of Machine Learning Research*, pages 21.1–21.24. PMLR.
- Alon Orlitsky, Ananda Theertha Suresh, and Yihong Wu. 2016. [Optimal prediction of the number of unseen species](#). *Proceedings of the National Academy of Sciences*, 113(47):13283–13288.
- Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. 2026. [ReasoningBank: Scaling agent self-evolving with reasoning memory](#). In *International Conference on Learning Representations (ICLR)*.
- E. S. Page. 1954. [Continuous inspection schemes](#). *Biometrika*, 41(1–2):100–115.
- Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. [Generative agents: Interactive simulators of human behavior](#). In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*. ACM.
- Aleksandr Podkopaev and Aaditya Ramdas. 2022. [Tracking the risk of a deployed model and detecting harmful distribution shifts](#). In *International Conference on Learning Representations (ICLR)*.
- Hongji Pu, Xinyuan Song, and Liang Zhao. 2026. [SkillOps: Managing LLM agent skill libraries as self-maintaining software ecosystems](#). arXiv preprint arXiv:2605.13716.
- Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. 2023. [Game-theoretic statistics and safe anytime-valid inference](#). *Statistical Science*, 38(4):576–601.
- Marion R. Reynolds, Jr. and Zachary G. Stoumbos. 1999. [A CUSUM chart for monitoring a proportion when inspecting continuously](#). *Journal of Quality Technology*, 31(1):87–108.
- Leo Richter, Xuanli He, Pasquale Minervini, and Matt J. Kusner. 2025. [An auditing test to detect behavioral shift in language models](#). In *International Conference on Learning Representations (ICLR)*.
- Shuvom Sadhuka, Drew Prinster, Clara Fannjiang, Gabriele Scalia, Bonnie Berger, Aviv Regev, and Hanchen Wang. 2025. [E-valuator: Reliable agent verifiers with sequential hypothesis testing](#). arXiv preprint arXiv:2512.03109.
- Luis Gaspar Schroeder, Aditya Desai, Alejandro Cuadron, Kyle Chu, Shu Liu, Mark Zhao, Stephan Krusche, Alfons Kemper, Matei Zaharia, and Joseph E. Gonzalez. 2026. [vCache: Verified semantic prompt caching](#). In *International Conference on Learning Representations (ICLR)*.
- Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. 2022. [Confident adaptive language modeling](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35.
- Arsenios Scrivens. 2026. [Information-theoretic limits of safety verification for self-improving systems](#). arXiv preprint arXiv:2603.28650.
- Glenn Shafer. 2021. [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.
- Zayx Shawn. 2026. [PACE: Anytime-valid acceptance tests for self-evolving agents](#). arXiv preprint arXiv:2606.08106.

- Jaehyeok Shin, Aaditya Ramdas, and Alessandro Rinaldo. 2024. [E-detectors: A nonparametric framework for sequential change detection](#). *The New England Journal of Statistics in Data Science*, 2(2):229–260.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*. ArXiv:2303.11366.
- David Siegmund. 1985. *Sequential Analysis: Tests and Confidence Intervals*. Springer Series in Statistics. Springer, New York.
- Baris Simsek. 2026. [When to forget: A memory governance primitive](#). arXiv preprint arXiv:2604.12007.
- Pragya Singh and Stanley Yu. 2025. [Validation-gated Hebbian learning for adaptive agent memory](#). In *Proceedings of the First Workshop on Knowledge Graphs and Agentic Systems Interplay (NORA 2025)*, co-located with NeurIPS 2025, volume 4162 of *CEUR Workshop Proceedings*, pages 42–54.
- Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. [Stop overthinking: A survey on efficient reasoning for large language models](#). *Transactions on Machine Learning Research*.
- Changzhi Sun, Xiangyu Chen, Jixiang Luo, Dell Zhang, and Xuelong Li. 2025. [Beyond heuristics: A decision-theoretic framework for agent memory management](#). arXiv preprint arXiv:2512.21567.
- Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. 2026. [Dynamic Cheat-sheet: Test-time learning with adaptive memory](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7080–7106. Association for Computational Linguistics.
- Niels A. Taatgen and Frank J. Lee. 2003. [Production compilation: A simple mechanism to model complex skill acquisition](#). *Human Factors*, 45(1):61–76.
- Prasad Tadepalli and Balas K. Natarajan. 1996. [A formal framework for speedup learning from problems and solutions](#). *Journal of Artificial Intelligence Research*, 4:445–475.
- Alexander Tartakovsky, Igor Nikiforov, and Michèle Basseville. 2014. *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. Chapman and Hall/CRC.
- Team OLMo. 2025. OLMo 3. Allen Institute for AI. Model card: <https://huggingface.co/allenai/Olmo-3-7B-Instruct>.
- Philip Thomas, Georgios Theodorou, and Mohammad Ghavamzadeh. 2015. [High confidence policy improvement](#). In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 2380–2388. PMLR.
- Philip S. Thomas, Bruno Castro da Silva, Andrew G. Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. 2019. [Preventing undesirable behavior of intelligent machines](#). *Science*, 366(6468):999–1004.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [MuSiQue: Multi-hop questions via single-hop question composition](#). *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Jean Ville. 1939. *Étude critique de la notion de collectif*. Gauthier-Villars, Paris.
- Abraham Wald. 1945. [Sequential tests of statistical hypotheses](#). *The Annals of Mathematical Statistics*, 16(2):117–186.
- Abraham Wald and Jacob Wolfowitz. 1948. [Optimum character of the sequential probability ratio test](#). *The Annals of Mathematical Statistics*, 19(3):326–339.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024a. [Voyager: An open-ended embodied agent with large language models](#). *Transactions on Machine Learning Research*.
- Haiming Wang, Huajian Xin, Chuanyang Zheng, Lin Li, Zhengying Liu, Qingxing Cao, Yinya Huang, Jing Xiong, Han Shi, Enze Xie, Jian Yin, Zhenguo Li, Heng Liao, and Xiaodan Liang. 2024b. [LEGO-Prover: Neural theorem proving with growing libraries](#). In *International Conference on Learning Representations (ICLR)*.
- Xi Wang, Anushri Suresh, Alvin Zhang, Rishi More, William Jurayj, Benjamin Van Durme, Mehrdad Farajtabar, Daniel Khashabi, and Eric Nalisnick. 2026. [Conformal Thinking: Risk control for reasoning on a compute budget](#). In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*. ArXiv:2203.11171.
- Zhiruo Wang, Graham Neubig, and Daniel Fried. 2024c. [TroVE: Inducing verifiable and efficient toolboxes for solving programmatic tasks](#). In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 51177–51191. PMLR.

- Zora Zhiruo Wang, Apurva Gandhi, Graham Neubig, and Daniel Fried. 2025a. [Inducing programmatic skills for agentic tasks](#). arXiv preprint arXiv:2504.06821.
- Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2025b. [Agent Workflow Memory](#). In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 63897–63911. PMLR.
- Ian Waudby-Smith and Aaditya Ramdas. 2024. [Estimating means of bounded random variables by betting](#). *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(1):1–27.
- Jingxian Wu and Jing Yang. 2016. [Quickest change detection with mismatched post-change models](#). arXiv preprint arXiv:1601.06868.
- Michael Wu and Arquimedes Canedo. 2026. [Invalidation contracts for cross-episode agent memory](#). arXiv preprint arXiv:2609.00243.
- Xuening Wu, Shenqin Yin, Yanlan Kang, Xinhang Zhang, Qianya Xu, Zeping Chen, and Wenqiang Zhang. 2025. [SGM: A statistical Gödel machine for risk-controlled recursive self-modification](#). arXiv preprint arXiv:2510.10232.
- Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. [TokenSkip: Controllable chain-of-thought compression in LLMs](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3351–3363. Association for Computational Linguistics.
- Haoran Xu, Jiacong Hu, Ke Zhang, Lei Yu, Yuxin Tang, Xinyuan Song, Yiqun Duan, Lynn Ai, and Bill Shi. 2025. [SEDM: Scalable self-evolving distributed memory for agents](#). arXiv preprint arXiv:2509.09498.
- Qunzhi Xu, Yajun Mei, and George V. Moustakides. 2021. [Optimum multi-stream sequential change-point detection with sampling control](#). *IEEE Transactions on Information Theory*, 67(11):7627–7636.
- Ziyu Xu, Nikos Karampatziakis, and Paul Mineiro. 2024. [Active, anytime-valid risk controlling prediction sets](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37.
- An Yang, Anfeng Li, Baosong Yang, et al. 2025. [Qwen3 technical report](#). arXiv:2505.09388.
- Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E. Gonzalez, and Bin Cui. 2024a. [Buffer of Thoughts: Thought-augmented reasoning with large language models](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37.
- Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024b. [Do large language models latently perform multi-hop reasoning?](#) In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10210–10229. Association for Computational Linguistics.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems*. ArXiv:2305.10601.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. [ReAct: Synergizing reasoning and acting in language models](#). In *International Conference on Learning Representations*. ArXiv:2210.03629.
- Roy D. Yates, Philippe Ciblat, Aylin Yener, and Michèle Wigger. 2017. [Age-optimal constrained cache updating](#). In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pages 141–145.
- Chengxin Yu, Zhaoxin Fan, Faguo Wu, Hongwei Zheng, Yun Zhou, and Zhiyu Li. 2026a. [CoMem: Collective-individual memory synergy for evolutionary multi-agent systems](#). arXiv preprint arXiv:2609.15009.
- Chengyao Yu, Hao Zeng, Youxin Zhu, Jianguo Huang, Huajun Zeng, and Bingyi Jing. 2026b. [Anytime safe PAC efficient reasoning](#). In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*.
- Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. [Distilling System 2 into System 1](#). arXiv preprint arXiv:2407.06023.
- Lifan Yuan, Yangyi Chen, Xingyao Wang, Yi R. Fung, Hao Peng, and Heng Ji. 2024. [CRAFT: Customizing LLMs by creating and retrieving from specialized toolsets](#). In *International Conference on Learning Representations*. ArXiv:2309.17428.
- Hao Zeng, Jianguo Huang, Bingyi Jing, Hongxin Wei, and Bo An. 2025. [On the provable performance guarantee of efficient reasoning models](#). arXiv preprint arXiv:2510.09133.
- Guibin Zhang, Muxin Fu, Kun Wang, Guancheng Wan, Miao Yu, and Shuicheng Yan. 2025a. [G-Memory: Tracing hierarchical memory for Multi-Agent systems](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pages 14587–14617.
- Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. 2026a. [Darwin Gödel machine: Open-ended evolution of self-improving agents](#). In *International Conference on Learning Representations (ICLR)*.

- Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025b. **AdaptThink: Reasoning models can learn when to think**. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3716–3730. Association for Computational Linguistics.
- Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. 2026b. **Agentic Context Engineering: Evolving contexts for self-improving language models**. In *International Conference on Learning Representations (ICLR)*. Code: <https://github.com/ace-agent/ace>.
- Qizheng Zhang, Michael Wornow, Gerry Wan, and Kunle Olukotun. 2025c. **Agentic Plan Caching: Test-time memory for fast and cost-efficient LLM agents**. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38.
- Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Zhuo Li, Yujie Zheng, Weinan Zhang, Ying Wen, Zhiyu Li, Feiyu Xiong, Yutao Qi, Bo Tang, and Muning Wen. 2026c. **MemRL: Self-evolving agents via runtime reinforcement learning on episodic memory**. arXiv preprint arXiv:2601.03192.
- Xing Zhang, Yanwei Cui, Guanghui Wang, Ziyuan Li, Wei Qiu, Bing Zhu, and Peiyang He. 2026d. **Ratchet: How reliable must an LLM judge be to retire a skill?** arXiv preprint arXiv:2605.22148.
- Yaolun Zhang, Yiran Wu, Yijiong Yu, Qingyun Wu, and Huazheng Wang. 2026e. **Live-Evo: Online evolution of agentic memory from continuous feedback**. arXiv preprint arXiv:2602.02369.
- Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. **ExpEL: LLM agents are experiential learners**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642.
- Wanjuan Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. **MemoryBank: Enhancing large language models with long-term memory**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731.
- Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. **MQuAKE: Assessing knowledge editing in language models via multi-hop questions**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 15686–15702.
- Jianghong Zhou, Hao Rong, and Chen Lin. 2026a. **Certification as a rate limit on self-improving LLM agents**. Under review.
- Jianghong Zhou, Hao Rong, Chen Lin, and Daoan Zhang. 2026b. **More verifiers are not always better: Robust shared memory for multi-agent LLM systems**. Under review.
- Banghua Zhu, Ying Sheng, Lianmin Zheng, Clark Barrett, Michael I. Jordan, and Jiantao Jiao. 2023. **Towards optimal caching and model selection for large model inference**. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.
- Jinchang Zhu, Jindong Li, Cheng Zhang, Jiahong Liu, and Menglin Yang. 2026. **HeLa-Mem: Hebbian learning and associative memory for LLM agents**. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13757–13769. Association for Computational Linguistics.
- Tijana Zrnic and Emmanuel J. Candès. 2024. **Active statistical inference**. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 62993–63010. PMLR.

A Proofs for Section 3

Setting. Outcomes $Y_1, Y_2, \dots$ are independent, with law P_0 before an unknown change time t_c and P_1 from t_c on. A score rule uses a bounded score $X = f(Y)$ with $\bar{x} = \text{ess sup } X$ and $\bar{b} = \text{ess sup } (-X)$, and runs $S_0 = 0$, $S_n = \max\{0, S_{n-1} + X_n\}$, $\tau_h = \inf\{n : S_n \geq h\}$. We assume $\mu_0 := \mathbb{E}_0 X < 0 < P_0(X > 0)$. The ARL is $A(h) = \mathbb{E}_\infty \tau_h$. Lorden’s worst-case delay $\bar{D}(h) = \sup_{t_c} \text{ess sup } \mathbb{E}_{t_c}[(\tau_h - t_c + 1)^+ | Y_{<t_c}]$ equals the zero-start delay $D(h) = \mathbb{E}_1 \tau_h$. The reason is that $s \mapsto S_n(s)$ is monotone, so the worst pre-change state is $S = 0$. We write $W_n = \sum_{k \leq n} X_k$, $\varphi_j(\theta) = \ln \mathbb{E}_j e^{\theta X}$, and $\theta_j > 0$ for the positive root of φ_j .

Hebbian rules and the exact level. Start from $c_0 = 1$, update $c \leftarrow \text{clip}_{[0,1]}(c + \eta_+ \mathbf{1}\{Y = 1\} - \eta_- \mathbf{1}\{Y = 0\})$, and retire at $c < \tau$. With $S = (1 - c)/\eta_-$ and $\rho = \eta_+/\eta_-$, the upper clip is the reflection at 0. The lower clip can bind only on a step that retires under both descriptions. So the rule is (2) with score $+1$ on failure, $-\rho$ on success, and retirement at $S > (1 - \tau)/\eta_-$. For $\eta_+ = \eta_- = \eta$ ($\rho = 1$), S is integer and the retirement time is τ_H with $H = \lfloor (1 - \tau)/\eta \rfloor + 1$. A first-passage recursion ($t_0 = 1/q$, $t_k = 1/q + \lambda t_{k-1}$, $\lambda = (1 - q)/q$) gives the exact run length

$$\mathbb{E}_q \tau_H = \frac{1}{q(\lambda - 1)} \left[\frac{\lambda(\lambda^H - 1)}{\lambda - 1} - H \right]$$

for $q \neq \frac{1}{2}$, and $\mathbb{E}_{1/2}\tau_H = H(H+1)$. This agrees with an exact Markov-chain solve to 10^{-46} relative error. Floating-point implementations of “ $c < \tau$ ” can retire one failure early (e.g. $\eta = 0.05, \tau = 0.9$), so we recommend an integer counter.

Proof of Remark 1. By convexity, φ_0 has a unique positive root θ_0 and $\varphi'_0(\theta_0) > 0$. Then $\tilde{P}_0(dy) = e^{\theta_0 f(y)} P_0(dy)$ is a probability measure with $\ln(d\tilde{P}_0/dP_0) = \theta_0 f$. Multiplying by $\theta_0 > 0$ commutes with $\max\{0, \cdot\}$, so $\theta_0 S_n$ is Page’s log-likelihood CUSUM for $P_0 \rightarrow \tilde{P}_0$ with threshold $\theta_0 h$. Its Lorden optimality for that change is due to Moustakides (1986). It is the likelihood-ratio CUSUM of no other alternative because $\ln(dQ/dP_0) = \kappa f$ with $E_0 e^{\kappa f} = 1$ forces $\kappa = \theta_0$. For the ± 1 rule $e^{\theta_0} = (1 - q_0)/q_0$, so $\tilde{q} = q_0 e^{\theta_0} = 1 - q_0$. More generally, the rule “ $+a$ on failure, $-b$ on success” is the exact CUSUM along the curve $\{(q_0, q_1) : \rho(q_0, q_1) = b/a\}$.

Lemma 7 (Page cycles and Lundberg bounds). *Let $N_h = \inf\{n \geq 1 : W_n \leq 0 \text{ or } W_n \geq h\}$, $p_h = \mathbb{P}(W_{N_h} \geq h)$ and $N_- = \inf\{n \geq 1 : W_n \leq 0\}$. (a) If $\mathbb{E}N_h < \infty$ and $p_h > 0$, then $\mathbb{E}\tau_h = \mathbb{E}N_h/p_h$. (b) If $\mathbb{E}X < 0$ with adjustment coefficient θ , and $c_\theta = \mathbb{E}[1 - e^{\theta X}; X < 0]$, then $c_\theta e^{-\theta(h+\bar{x})} \leq p_h \leq e^{-\theta h}$ and $\mathbb{P}(\sup_n W_n \geq x) \leq e^{-\theta x}$. (c) If $\mathbb{E}X < 0$, then $\mathbb{E}N_- = 1/\mathbb{P}(\sup_n W_n = 0) < \infty$.*

Proof. (a) Until N_h the walk and S coincide. On a downward exit S restarts at 0, so τ_h is a geometric(p_h) sum of i.i.d. cycle lengths, and Wald’s identity applies (Page, 1954; Siegmund, 1985). (b) $e^{\theta W_n}$ is a martingale bounded up to N_h , so $1 = \mathbb{E}e^{\theta W_{N_h}}$. Split into upward exits (value in $[e^{\theta h}, e^{\theta(h+\bar{x})})$) and downward exits (value ≤ 1 , and $= e^{\theta X_1}$ on $\{X_1 < 0\}$). The maximal inequality follows from optional stopping at the first passage above x . (c) Duality: $\{W_1 > 0, \dots, W_n > 0\}$ is the event that n is a strict ascending ladder epoch of the reversed walk. The number of such epochs is geometric with mean $1/\mathbb{P}(\sup W = 0)$. $\square$

Proof of Theorem 1. *ARL.* By Lemma 7(a)–(c) under P_0 , $e^{\theta_0 h} \leq A(h) \leq (\mathbb{E}_0 N_-/c_0) e^{\theta_0(h+\bar{x})}$, so $\theta_0 h \leq \ln A \leq \theta_0 h + \Delta_0$ with an explicit constant Δ_0 . (i) $\mu_1 > 0$. Since $S_n \geq W_n$, τ_h is at most the first passage of W above h , and Wald gives $D \leq (h + \bar{x})/\mu_1$. Conversely $\mu_1 D = \mathbb{E}_1 W_{\tau_h} \geq h - \mathbb{E}_1 |\inf_n W_n| \geq h - 1/\beta_1$, where $\beta_1 > 0$ solves $\mathbb{E}_1 e^{-\beta_1 X} = 1$ (maximal inequality for $-W$). Substituting $h = (\ln A + O(1))/\theta_0$ gives $D = \ln A/(\theta_0 \mu_1) + O(1)$. (ii)

$\mu_1 = 0$. $S_n^2 - n\sigma_1^2$ is a supermartingale, because $\mathbb{E}[\max(0, s + X)^2] \leq s^2 + \sigma_1^2$. Hence $D \geq h^2/\sigma_1^2$. Wald’s second identity on a Page cycle, together with $p_h \geq \mathbb{E}_1 X^-/(h + \bar{x})$, gives $D \leq [(h + \bar{x})^2 + \bar{b}^2(h + \bar{x})/\mathbb{E}_1 X^-]/\sigma_1^2$. (iii) $\mu_1 < 0$. Apply the ARL argument under P_1 with θ_1 : $e^{\theta_1 h} \leq D \leq (\mathbb{E}_1 N_-/c_1) e^{\theta_1(h+\bar{x})}$. Since $e^{\theta_1 h} = (e^{\theta_0 h})^\gamma$, this gives $D \asymp A^\gamma$. Moreover $\gamma < 1$ iff $\varphi_1(\theta_0) > 0$, which holds when the score law under P_1 stochastically dominates that under P_0 . For Bernoulli rules this is equivalent to $q_1 > q_0$. If $P_1(X > 0) = 0$ the rule never alarms after the change. *Efficiency.* Because $\ln(d\tilde{P}_0/dP_0) = \theta_0 f$,

$$\begin{aligned} \theta_0 \mu_1 &= \mathbb{E}_1 \ln \frac{d\tilde{P}_0}{dP_0} \\ &= \mathbb{E}_1 \ln \frac{dP_1}{dP_0} - \mathbb{E}_1 \ln \frac{dP_1}{d\tilde{P}_0} \\ &= \text{KL}(P_1 \| P_0) - \text{KL}(P_1 \| \tilde{P}_0). \end{aligned}$$

Dividing the optimal delay $\ln A/\text{KL}(P_1 \| P_0)$ (Lorden, 1971) by $\ln A/(\theta_0 \mu_1)$ gives (3). Equality holds iff $\tilde{P}_0 = P_1$, i.e. $\theta_0 f = \ln dP_1/dP_0$. $\square$

Proof of Corollary 2. The Bernoulli log-likelihood ratio is $\ln(q_1/q_0) > 0$ on a failure and $\ln \frac{1-q_1}{1-q_0} < 0$ on a success. A rule with scores $(+1, -\rho)$ is proportional to it iff $\rho = \rho(q_0, q_1)$, and the retirement threshold in log-likelihood units is $\ln(q_1/q_0)(1 - \tau)/\eta_-$. The symmetric rule is optimal iff $q_1 = 1 - q_0$.

Unknown degradation.

Proposition 8 (First-order minimax tuning). *Let $q_1 \in [q_-, q_+]$ with $q_0 < q_-$. For every detector with $\text{ARL} \geq A$, the worst-case delay over this interval is at least $(1 + o(1)) \ln A/\text{KL}(q_- \| q_0)$, and the likelihood-ratio rule tuned at q_- attains it.*

Proof. The lower bound is Lorden’s bound at $q_1 = q_-$ (Lorden, 1971). The rule tuned at q_- has scores equal to the log-likelihood ratio of q_- against q_0 , so $\theta_0 = 1$, and its post-change drift $\mu_1(q_1)$ increases in q_1 because the failure score is positive and the success score negative. By Theorem 1(i) its delay is $\ln A/\mu_1(q_1) + O(1) \leq \ln A/\mu_1(q_-) + O(1)$, and $\mu_1(q_-) = \text{KL}(q_- \| q_0)$. $\square$

Proposition 9 (Unclipped counters; restates Remark 2). *Consider an unclipped counter $C_n = C_{n-1} + \mathbf{1}\{Y_n = 1\} - \mathbf{1}\{Y_n = 0\}$ started at c_0 and deleted at $C \leq 0$ (ExpeL: $c_0 = 2$), or the majority rule “flag when harmful $\geq$ helpful” (ACE’s released code). (a) $A = \infty$, since false deletion*

has probability $(q_0/(1 - q_0))^{c_0} < 1$, but no worst-case delay is finite: for $q_1 > \frac{1}{2}$ the delay after a change equals the credit banked before it divided by $2q_1 - 1$. (b) Under partial staleness ($q_1 < \frac{1}{2}$) the stale macro is never flagged (ACE) or deleted (ExpeL) with probability at least $1 - 2q_1$ (majority rule) or $1 - (q_1/(1 - q_1))^{c_0}$, for every change time (averaged over the pre-change history).

Proof of Proposition 9. (a) Without a cap, a ± 1 walk with up-probability $1 - q_0 > \frac{1}{2}$ started at $c_0 \geq 1$ hits 0 with probability $\lambda_0^{-c_0} < 1$, so $A = \infty$. Given the pre-change history on $\{\tau \geq t_c\}$, the post-change walk starts at C_{t_c-1} with drift $-(2q_1 - 1)$. Wald’s identity gives the conditional delay $C_{t_c-1}/(2q_1 - 1)$, and the all-success history makes $C_{t_c-1} = c_0 + t_c - 1$ arbitrarily large, so $\bar{D} = \infty$. (b) For $q_1 < \frac{1}{2}$, the probability that a walk from c never hits 0 is $h_1(c) = 1 - \lambda_1^{-c}$. The process $h_1(C_n)\mathbf{1}\{\tau > n\}$ is a P_∞ -submartingale, so $\mathbb{P}_{t_c}(\tau = \infty)$ is nondecreasing in t_c . Its value at $t_c = 1$ is $1 - \lambda_1^{-c_0}$, and for the majority rule it is $(1 - q_1)h_1(1) = 1 - 2q_1$. Exact dynamic programming reproduces these values to 8 digits.

Time decay is a time-to-live. Suppose confidence also decays as $c \leftarrow (1 - \lambda)c$ at every query, and the template is used with probability ϕ per query. Then any run of $k^* = \lfloor \ln \tau / \ln(1 - \lambda) \rfloor + 1$ consecutive queries that do not use the template retires the macro, whatever its reliability. The expected waiting time for such a run is $((1 - \phi)^{-k^*} - 1)/\phi$ queries, i.e. $(1 - \phi)^{-k^*} - 1$ uses. For $\lambda = 10^{-3}$ and $\tau = 0.85$ we get $k^* = 163$, so a perfectly reliable macro used by 1% of queries is retired after at most 4.15 uses on average.

The negative-drift regime in prior work: a correction. Wu and Yang (2016) study Page’s CUSUM run with a mismatched post-change density g . Their eq. (26) defines it by the restarted recursion $\tilde{C}_{n+1} = \max(1, \tilde{C}_n) \tilde{\Lambda}_{n+1}$, i.e. (2) with $f = \ln(g/f_0)$, for which $\theta_0 = 1$ and $\tilde{P}_0 = g$. Their Thm. 2 states that if $\text{KL}(P_1\|P_0) < \text{KL}(P_1\|g)$ (their $D_{10} < \tilde{D}_{11}$), the average detection delay is infinite for every false-alarm level below one, and they stress that the statement is not asymptotic. This is our regime (iii), and there the delay is finite whenever $P_1(f > 0) > 0$: $D(h) \leq (\mathbb{E}_1 N_- / c_1) e^{\theta_1(h+\bar{x})}$ (proof of Theorem 1). The proof of their Thm. 2 writes the CUSUM stopping time as a minimum over restart times, $\tilde{\tau}_c = \min_v \tau_v(A)$ (their eq. (77)), where $\tau_v(A)$ is the one-

sided test started at v and never restarted, and their eq. (78) then equates $\mathbb{E}_k(\min_v \tau_v(A) - k)^+$ with $\min_v \mathbb{E}_k(\tau_v(A) - k)^+$. Their Lemma 6 is correct: each $\tau_v(A)$ never stops with positive probability, so its mean delay is infinite. But in general only $\mathbb{E} \min \leq \min \mathbb{E}$ holds, and here the inequality is strict: the CUSUM alarms at the first success of a geometric number of Page cycles (Lemma 7(a)), each of finite mean length. *Counterexample.* The symmetric rule with $q_0 = 0.05$ is the CUSUM with g the failure rate 0.95. At $q_1 = 0.25$, $\text{KL}(P_1\|P_0) = 0.225 < \text{KL}(P_1\|g) = 1.697$, so their condition holds; their regularity conditions (their (5) and (47)) hold for i.i.d. data by the strong law. A single test $\tau_v(A)$ with threshold $h = 4$ never stops with probability $1 - 3^{-4} = 0.988$, yet the CUSUM’s exact Lorden delays are $D(4) = 232$ and $D(6) = 2172$ (zero start is the worst case), and Monte Carlo gives 231.9 and 2160 (2×10^5 and 4×10^4 runs). Under their own Bayesian criterion, with the change at a geometric(0.1) time, $\mathbb{E}[\tilde{\tau}_c - \nu \mid \tilde{\tau}_c \geq \nu] = 230.8$ and 2170.8. What their theorem does capture is the first-order statement $D/\ln A \rightarrow \infty$ in this regime; Theorem 1(iii) gives its rate, $D \asymp A^\gamma$ (checks/wuyang_check.py in our code release).

B Proofs for Section 4

Assumptions.

- **(A1) Demand.** Queries of template i form independent Poisson processes of rate π_i , with $\pi_1 \geq \pi_2 \geq \dots$ and $\sum_i \pi_i = 1$.
- **(A2) Audits.** Each query is audited independently with probability p . An audit yields *one* verifier outcome, of the candidate that the policy chose to evaluate on that query, revealed after answering. Nothing else about the candidates’ correctness is observed; in particular, the policy does not see whether an unaudited shadow answer agrees with the live one. (Under exact-match scoring such agreement is informative, and Theorem 22 shows that it can speed up certification.)
- **(A3) Changes.** Facts change at the points of an independent Poisson process of rate ν per template. Changes may or may not be announced to the policy. A macro formed in an earlier epoch never counts as coverage.
- **(A4) Outcomes.** In environment E_r the audited outcomes of the current epoch are i.i.d.

Bernoulli(r), whichever candidate they evaluate, and independent of everything else, denoted $\mathcal{G}$ (arrivals, audits, change times, other templates and epochs, internal randomization).

- **(A5) Policies.** Policies are non-anticipating: a decision at time t depends only on $\mathcal{G}$ -information up to t and on outcomes revealed before t .

Precise form of Theorem 3. Assume (A1)–(A5) and that in every epoch the audited outcomes are i.i.d. Bernoulli(r_1), whichever candidate they evaluate (environment E_{r_1} , the least-favorable good case). (i) For every demand law π and all ν, p, δ , every policy that is δ -safe in the sense of Definition 1 has $\liminf M \geq (1 - \varepsilon_\delta)F_\pi(\Theta)$ almost surely, whether or not changes are announced and however many candidates it forms, where $\varepsilon_\delta = O(\sqrt{\ln \ln(1/\delta)/\ln(1/\delta)})$ depends only on (δ, r_0, r_1) . (ii) With announced changes, the SPRT that discards all evidence and restarts at each change is δ -safe and has $M \leq (1 + \ln(r_1/r_0)/\ln(1/\delta))F_\pi(\Theta)$. Hence $M^* = (1 + o(1))F_\pi(\Theta)$ as $\delta \rightarrow 0$, uniformly in π, ν, p . The main text calls the environment “good memories are only barely good”; the converse needs no other restriction on the policy than non-anticipation (A5) and the information structure (A2).

Notation: $x = \nu/p$, $\rho_i = \pi_i/(\pi_i + x)$, $a_\pm$ the log-likelihood-ratio increments on success and failure, $w = a_+ + a_-$, $\text{KL} = \text{KL}(r_1 \| r_0)$, $L = \ln(1/\delta)$, $n^* = L/\text{KL}$, $\Theta = xn^*$ and $\beta = 1 - 1/s$. Define $M_{\text{fix}}(n, x) = \sum_i \pi_i(1 - \rho_i^n)$.

Lemma 10 (Exact stationary gap). *If the policy folds when the N_e -th outcome of epoch e is observed, with (N_e) i.i.d. copies of N independent of arrivals, audits and changes, then the long-run unfolded fraction of template i is $u_i = 1 - \mathbb{E}[\rho_i^N]$ a.s. and in mean, and $M = \sum_i \pi_i u_i$.*

Proof. Epochs are i.i.d. renewal cycles of mean length $1/\nu$ with π_i/ν queries on average. A query at time t of an epoch is unfolded iff fewer than N audited queries preceded it. By the Mecke formula, $\mathbb{E}U_e = \pi_i \mathbb{E} \int_0^\ell \mathbb{P}(\text{Poi}(p\pi_i t) < N) dt$ with $\ell \sim \text{Exp}(\nu)$. The count $\text{Poi}(p\pi_i E)$ with $E \sim \text{Exp}(\nu)$ is geometric with $\mathbb{P}(=k) = (1 - \rho_i)\rho_i^k$, so $\mathbb{P}(\text{Poi}(p\pi_i E) < N) = 1 - \mathbb{E}\rho_i^N$. Apply renewal-reward per template and sum, using Fatou for the lower bound and the superposition property for the upper bound. $\square$

Lemma 11 (Zipf sums). *For $\pi_i = i^{-s}/\zeta(s)$, $s > 1$, real $n \geq 1$ and $x > 0$,*

$$K_{s,n}x^\beta - nx \leq M_{\text{fix}}(n, x) \leq K_{s,n}x^\beta,$$

where $K_{s,n} = \Gamma(\frac{1}{s})C_{s,n}$ and $C_{s,n} = \zeta(s)^{-1/s}\Gamma(n+1-\frac{1}{s})/((s-1)\Gamma(n))$. For integer n the latter equals the occupancy form $\frac{\zeta(s)^{-1/s}}{s} \sum_{k \leq n} \Gamma(k+1-\frac{1}{s})/k!$. Likewise $\kappa_s \Theta^\beta - \Theta \leq F_\pi(\Theta) \leq \kappa_s \Theta^\beta$ with $\kappa_s = \Gamma(\frac{1}{s})/((s-1)\zeta(s)^{1/s})$.

Proof. The summand $g(\pi) = \pi(1 - (\pi/(\pi+x))^n)$ is nondecreasing in π and bounded by nx . Hence the sum lies between $\int_1^\infty$ and $\int_0^\infty$ of $g(t^{-s}/\zeta(s)) dt$, and the two differ by at most nx . Substituting $y = x\zeta(s)t^s$ and integrating by parts turns $\int_0^\infty$ into a Beta integral, which gives $K_{s,n}x^\beta$. The identity $\sum_{k \leq n} \Gamma(k+a)/k! = \Gamma(n+a)/(a\Gamma(n))$ follows by induction. The bounds on F_π follow in the same way with $g(\pi) = \pi(1 - e^{-\Theta/\pi}) \leq \Theta$. $\square$

Lemma 12 (From Poisson to exponential). *For $n \geq 1$ and $y \geq 0$, $(1 - C_*/n)(1 - e^{-ny}) \leq 1 - (1+y)^{-n} \leq 1 - e^{-ny}$ with $C_* = 1.1047$. Consequently $(1 - C_*/n)F_\pi(nx) \leq M_{\text{fix}}(n, x) \leq F_\pi(nx)$. The function F_π is concave and nondecreasing with $F_\pi(0) = 0$. For random N , $1 - \mathbb{E}\rho^N \leq 1 - \rho^{\mathbb{E}N}$ by Jensen’s inequality.*

Proof of Theorem 3: converse. Fix template i and let $\mathcal{H}_e$ be the history up to the start of epoch e . We bound the expected unfolded count of one epoch given $\mathcal{H}_e$, and then average over epochs with a martingale strong law.

One epoch. Condition on $\mathcal{H}_e$. Let $T_1 < T_2 < \dots$ be the audit times of template i from the epoch start (continued past its end), Y_k the audited outcomes, σ the first fold event of the epoch, of a candidate formed in any epoch ($\sigma = \infty$ if there is none), and $N = \#\{k : T_k < \sigma\}$. By (A5), N is a stopping time for $\mathcal{F}_m = \sigma(\mathcal{H}_e, \mathcal{G}, Y_1, \dots, Y_m)$, and σ is $\mathcal{F}_N$ -measurable. By (A4), $d\mathbb{P}_{r_1}/d\mathbb{P}_{r_0}|_{\mathcal{F}_m} = e^{W_m}$, where W_m is the log-likelihood-ratio walk; this holds whichever candidates the policy forms or audits, because every audit reveals one outcome with the same law. Fix an integer $n \leq n^*$ and $c \geq 0$. Let $Q^{<n}$ be the number of the epoch’s queries preceded by fewer than n of its audits, and let R be the number of those that arrive after the first fold event σ . Every query answered by a candidate of the epoch arrives after σ , so the unfolded

count satisfies $U \geq Q^{<n} - R$. Moreover, R is a function of σ and $\mathcal{G}$, hence $\mathcal{F}_N$ -measurable even if the policy later unfolds and refolds, and $R > 0$ implies $N \leq n - 1$.

- *Low evidence.* Wald's likelihood-ratio identity for the $\mathcal{F}_N$ -measurable $R \geq 0$ gives $\mathbb{E}_{r_1}[R; W_N \leq c, N < \infty] \leq e^c \mathbb{E}_{r_0}[R]$. Under r_0 all candidates are bad, so by Definition 1 the epoch has a fold event with probability at most δ given $\mathcal{H}_e$. Given $\mathcal{F}_\sigma$, the strong Markov property of the Poisson arrival, audit and change processes bounds the mean number of queries between σ and the earlier of the n -th audit and the epoch end by $\min(n/p, \pi_i/\nu)$. Hence $\mathbb{E}_{r_0}[R \mid \mathcal{H}_e] \leq \delta \min(n/p, \pi_i/\nu)$.
- *High evidence.* On $\{W_N > c, R > 0\}$ we have $\max_{k \leq n-1} W_k > c$. Since $R \leq Q^{<n}$ and $Q^{<n}$ is $\mathcal{G}$ -measurable, $\mathbb{E}_{r_1}[R; W_N > c \mid \mathcal{H}_e] \leq \mathbb{E}Q^{<n} \cdot \beta_{n-1}(c)$, where $\beta_m(c) = \mathbb{P}_{r_1}(\max_{k \leq m} W_k \geq c)$.

Since $\mathbb{E}Q^{<n} = \frac{\pi_i}{\nu}(1 - \rho_i^n)$ by the computation in Lemma 10, the unfolded count U_e of epoch e satisfies $\mathbb{E}[U_e \mid \mathcal{H}_e] \geq \frac{\pi_i}{\nu} b_i$, where

$$b_i := (1 - \beta_{n-1}(c))(1 - \rho_i^n) - e^c \delta \min(1, \frac{n\nu}{p\pi_i}).$$

Many epochs. Let Q_e be the number of queries of template i in epoch e . The Q_e are i.i.d. geometric with mean π_i/ν whatever the policy does, so $\mathbb{E}Q_e^2 < \infty$. The differences $D_e = U_e - \mathbb{E}[U_e \mid \mathcal{H}_e]$ are martingale differences with $\mathbb{E}D_e^2 \leq \mathbb{E}Q_e^2$, because $0 \leq U_e \leq Q_e$. Hence $\sum_e D_e/e$ converges a.s. by the L^2 martingale convergence theorem, and Kronecker's lemma gives $\frac{1}{K} \sum_{e \leq K} D_e \rightarrow 0$ a.s. (Hall and Heyde, 1980, Thm. 2.18). With the ordinary strong law for (Q_e) , $\liminf_K \sum_{e \leq K} U_e / \sum_{e \leq K} Q_e \geq b_i$ a.s. The last, incomplete epoch contributes a vanishing share because $Q_K/K \rightarrow 0$ a.s., so the long-run unfolded fraction satisfies $\liminf u_i \geq b_i$ a.s.

Choice of n and c . Take $n-1 = \lfloor (1-\varepsilon)n^* \rfloor$ and $c = (1-\varepsilon/2)L$, with $\varepsilon \rightarrow 0$ and $\varepsilon\sqrt{L}/\sqrt{\ln L} \rightarrow \infty$. Then $\beta_{n-1}(c) \rightarrow 0$ by Azuma's maximal inequality for W (the mean of W_{n-1} is at most $(1-\varepsilon)L$), and $e^c \delta = \delta^{\varepsilon/2} \rightarrow 0$. Lemma 12 and concavity give $1 - \rho_i^n \geq (1 - C_*/n)(1-\varepsilon)(1 - e^{-\Theta/\pi_i})$, and $\min(1, \Theta/\pi_i) \leq \frac{e}{e-1}(1 - e^{-\Theta/\pi_i})$. Hence $\liminf u_i \geq (1 - \varepsilon_\delta)(1 - e^{-\Theta/\pi_i})$ with $\varepsilon_\delta =$

$O(\sqrt{\ln L/L})$, uniformly in i, π, ν and p . Summing over templates with Fatou's lemma for series gives $\liminf M \geq (1 - \varepsilon_\delta)F_\pi(\Theta)$ a.s.

Announcements and candidates. The argument used announcements only through $\mathcal{F}_m$, which contains the change times through $\mathcal{G}$. A policy that does not observe changes is non-anticipating for a smaller filtration, so the bound holds for it a fortiori. The number of candidates entered only through the likelihood ratio, which (A2) and (A4) make independent of which candidate is audited. $\square$

Proof of Theorem 3: achievability. At each announced change the policy discards all candidates' evidence and restarts one SPRT, so every fold event of an epoch is a crossing of that epoch's test. Under $r \leq r_0$, $\mathbb{E}e^Z \leq 1$ for each increment of the SPRT statistic. So e^{Λ_n} is a nonnegative supermartingale, and Ville's inequality gives $\mathbb{P}(\sup_n \Lambda_n \geq L) \leq \delta$ under arbitrary drift below r_0 and adaptive audit schedules. Under $r \geq r_1$, Wald's identity with overshoot $< a_+$ gives $\mathbb{E}N \leq (L + a_+)/\text{KL}$. By Lemma 10, Jensen's inequality and Lemma 12, $M_{\text{SPRT}} \leq F_\pi(x\mathbb{E}N) \leq (1 + a_+/L)F_\pi(\Theta)$, where the last step uses concavity of F_π . $\square$

The per-template gap of the SPRT can be bracketed exactly, which sharpens the Jensen step above and makes the limit quantitative at practical δ .

Lemma 13 (Finite- δ bracket). *Let Z be the SPRT increment, $a_+ = \ln(r_1/r_0)$ its value on a success, and let $\xi_i > 0$ solve*

$$\mathbb{E}_{r_1} e^{\xi_i Z} = 1 + \nu/(p\pi_i). \quad (7)$$

If every epoch runs the SPRT at level δ , the unfolded fraction u_i of template i (so $M = \sum_i \pi_i u_i$) satisfies

$$1 - \delta^{\xi_i} \leq u_i \leq 1 - \delta^{\xi_i} e^{-\xi_i a_+}.$$

Every per-epoch δ -safe certifier that folds at a stopping time of the audited outcomes has $u_i \geq 1 - \delta^{\xi_i/(1+\xi_i)}$. As $\nu/(p\pi_i) \rightarrow 0$ with Θ/π_i fixed, $\xi_i \ln(1/\delta) \rightarrow \Theta/\pi_i$, and all three bounds tend to $1 - e^{-\Theta/\pi_i}$.

Proof of Lemma 13. Write $L = \ln(1/\delta)$ and $\sigma^2 = \text{Var}_{r_1} Z$. The right side of Equation (7) is $1/\rho_i$, and N is the SPRT sample size in audited outcomes. By Lemma 10, $u_i = 1 - \mathbb{E}\rho_i^N$, since audited outcomes of template i arrive at rate $p\pi_i$ and the epoch ends at rate ν . The function $\psi(\xi) = \ln \mathbb{E}_{r_1} e^{\xi Z}$ is convex with $\psi(0) = 0$, $\psi'(0) = \text{KL} > 0$ and $\psi(\xi) \rightarrow \infty$,

so Equation (7) has a unique root. Hence $M_n = \rho_i^n e^{\xi_i \Lambda_n}$ is a mean-one $\mathbb{P}_{r_1}$ -martingale, as in Wald's fundamental identity (Siegmund, 1985). *Bracket.* Before the crossing $M_n \leq e^{\xi_i L}$, and at the crossing $L \leq \Lambda_N < L + a_+$. Since $N < \infty$ a.s., dominated convergence gives $\mathbb{E}M_N = 1$, and the bounds on Λ_N give $\delta^{\xi_i} e^{-\xi_i a_+} \leq \mathbb{E}\rho_i^N \leq \delta^{\xi_i}$. *Converse.* Write $\rho_i^N = M_N^{1/(1+\xi_i)} (\rho_i^N e^{-\Lambda_N})^{\xi_i/(1+\xi_i)}$. By Hölder's inequality with exponents $1 + \xi_i$ and $\frac{1+\xi_i}{\xi_i}$, $\mathbb{E}_{r_1}[\rho_i^N; N < \infty]$ is at most $\mathbb{E}_{r_1}[M_N; N < \infty]^{1/(1+\xi_i)}$ times $\mathbb{E}_{r_1}[\rho_i^N e^{-\Lambda_N}; N < \infty]^{\xi_i/(1+\xi_i)}$. The first factor is ≤ 1 by Fatou's lemma. By change of measure the second expectation is $\mathbb{E}_{r_0}[\rho_i^N; N < \infty]$, the probability of folding a bad candidate within the epoch, which is $\leq \delta$. *Limit.* Expanding $\psi(\xi) = \text{KL}\xi + \sigma^2 \xi^2/2 + O(\xi^3)$ and $\ln(1+y) = y + O(y^2)$ in Equation (7) gives $\xi_i L = \frac{\Theta}{\pi_i} (1 + O(\frac{\nu}{p\pi_i} (1 + \frac{\sigma^2}{\text{KL}^2})))$, and $\xi_i a_+ \rightarrow 0$. $\square$

The bracket is two-sided and sharper than the Jensen bound $u_i \leq 1 - \rho_i^{\mathbb{E}N}$ used for Theorem 3. For $(r_0, r_1) = (0.85, 0.95)$, $\delta = 10^{-2}$ and $\Theta/\pi_i = 0.908$, exact dynamic programming gives $u_i = 0.5610 \in [0.5557, 0.5643]$. Here $\xi_i = 0.1762$, the converse bound is 0.4984, the first-order value $1 - e^{-\Theta/\pi_i}$ is 0.5966, and the Jensen bound is 0.5998.

Universality. Suppose $G(\theta) = \sum_{i:\pi_i \leq \theta} \pi_i$ is regularly varying with index $1 - \alpha \in (0, 1)$. Writing $F_\pi(\Theta) = 1 - e^{-\Theta} + \int_\Theta^\infty G(\Theta/y) e^{-y} dy$ and applying Potter's bounds with dominated convergence gives $F_\pi(\Theta) \sim \Gamma(\alpha) G(\Theta)$ as $\Theta \rightarrow 0$. For Zipf demand, $\alpha = 1/s$.

A finite-size law for cold starts. The steady state of Theorem 3 is reached only after about $1/\nu$ queries. The following law interpolates between a cold start and the drift floor. It is stated for a fixed certification sample size n , not for sequentially tested (safe) folding; extending it to safe folding is open.

Theorem 14 (One law for cold start and drift). *Let demand be Zipf(s), start all templates cold at time 0, let facts change at rate ν , and fold after n audited uses. The probability that the query at time T is not answered in one step satisfies $M(T, \nu) = (pT)^{-(1-\frac{1}{s})} \Phi_{s,n}(\nu T) (1 + o(1))$ with relative error $O(m^{1/s} \ln(1/m))$, $m = \max(1/(pT), \nu/p)$, where $\Phi_{s,n}(y) = C_{s,n} [e^{-y} + y^{1-\frac{1}{s}} \gamma_{\text{inc}}(\frac{1}{s}, y)]$, $C_{s,n}$ is as in Lemma 11, and γ_{inc} is the lower incomplete gamma function.*

At $y = \nu T = 0$ this is the static amortization law $M(T) \sim C_{s,n}(pT)^{-(1-1/s)}$, a corollary of Karlin (1967) and Ohannessian and Dahleh (2012, Prop. 7) summed over $r < n$. It is a power law of practice (Newell and Rosenbloom, 1981; Heathcote et al., 2000) whose exponent $1 - 1/s$, shared by memorization learning curves (Hutter, 2021; Michaud et al., 2023), is set by the task distribution. As $y \rightarrow \infty$ it tends to $\Gamma(\frac{1}{s}) C_{s,n} (\nu/p)^{1-1/s} = \kappa_s \Theta^{1-1/s} (1 + O(1/n))$ with $\Theta = n\nu/p$, the steady state of Corollary 4. Appendix D.1 tests the static branch on natural template streams.

Proof of Theorem 14. With changes at rate ν from a cold start at 0, the age of the current epoch at time T is $A_T = \min(T, E_\nu)$, and the audited count of template i given A_T is $\text{Poi}(p\pi_i A_T)$. For audited exposure a (i.e. $a = pA_T$), the sum $S_n(a) = \sum_i \pi_i \mathbb{P}(\text{Poi}(\pi_i a) < n)$ satisfies $|S_n(a) - C_{s,n} a^{-\beta}| \leq 2n/a$. The summand is unimodal in i with maximum at most n/a , and its integral is $C_{s,n} a^{-\beta}$. Averaging $(pA_T)^{-\beta}$ gives $p^{-\beta} \mathbb{E}A_T^{-\beta} = (pT)^{-\beta} [e^{-y} + y^\beta \gamma_{\text{inc}}(1-\beta, y)]$ with $y = \nu T$, which is the stated $\Phi_{s,n}$. The error bound $O(m^{1/s} \ln(1/m))$ with $m = \max(1/(pT), \nu/p)$ follows by integrating $\min(2n/a, 1)$ against the law of pA_T . $\square$

C Certifying utility by betting

The SPRT of Section 4 tests reliability against $r_0 = r^*$ (Equation (1)), which depends on the live accuracy a . Here we certify the macro's *net utility relative to live* on paired shadow evaluations instead, so a never enters the null. The tools are testing by betting (Shafer, 2021; Ramdas et al., 2023; Waudby-Smith and Ramdas, 2024) with inverse-propensity audits, as in Xu et al. (2024) and Yu et al. (2026b).

Setup. Fix a template and a frozen candidate. At shadow evaluation $t = 1, 2, \dots$ both paths run on the same instance. Live has cost C_t^L and correctness $Y_t^L \in \{0, 1\}$; the macro has C_t^M and Y_t^M . The paired utility of folding is

$$U_t = C_t^L - C_t^M + L(Y_t^M - Y_t^L). \quad (8)$$

Given a known B with $U_t \geq -B$, put $Z_t = U_t/B \in [-1, M]$. In the paired Bernoulli model ($C^L \equiv m$, $C^M \equiv 1$), let $\eta = (m - 1)/L < 1$, $D_t = Y_t^M - Y_t^L$ and $B = L(1 - \eta)$. Then $Z_t = (\eta + D_t)/(1 - \eta)$ and $\mathbb{E}Z_t = (r - r^*)/(1 - \eta)$.

Correctness is revealed only on audited instances ($A_t = 1$). Let $\mathcal{F}_{t-1}$ be everything that happened before t , observed or not, and let W_t be pre-audit side information (costs, both answers, a judge score). Put $\mathcal{H}_t = \mathcal{F}_{t-1} \vee \sigma(W_t)$, and let $\mathcal{G}_{t-1} \subseteq \mathcal{F}_{t-1}$ be the system's records. We assume:

- (A1) the audit probability p_t is $\mathcal{H}_t$ -measurable and computable, and $p_t \geq \pi_t$ on $\{p_t > 0\}$ for a $\mathcal{G}_{t-1}$ -measurable floor $\pi_t \in (0, 1]$;
- (A2) $\mathbb{P}(A_t = 1 \mid \mathcal{H}_t, Z_t) = p_t$;
- (A3) a computable surrogate $g_t \in [-1, \bar{G}]$ satisfies $Z_t = g_t$ on $\{p_t = 0\}$.

The estimator $\hat{Z}_t = g_t + A_t(Z_t - g_t)/p_t$ (with $0/0 := 0$) is observable. Taking $g \equiv 0$ gives IPW, and $p \equiv 1$ gives full information. The candidate is inadmissible under

$$H_0 : \mathbb{E}[Z_t \mid \mathcal{F}_{t-1}] \leq 0 \text{ a.s. for all } t. \quad (9)$$

H_0 leaves arbitrary the joint law of (C^L, C^M, Y^L, Y^M) , including the live-macro error dependence, and its dependence on the past. Drifting live accuracy, shared difficulty and adaptive query selection are therefore all allowed.

Test. Let $\kappa_t = (1 + \bar{G}(1 - \pi_t))/\pi_t$, which is $1/\pi_t$ for IPW. Take $\mathcal{G}_{t-1}$ -measurable bets $\lambda_t^{(j)} \in [0, 1/\kappa_t]$ and weights $w_j \geq 0$ with $\sum_j w_j \leq 1$. Fold at $\tau = \inf\{n : K_n \geq 1/\delta\}$, where

$$K_n = \sum_j w_j \prod_{t \leq n} (1 + \lambda_t^{(j)} \hat{Z}_t). \quad (10)$$

The *grid mixture* at audit rate p uses $\lambda^{(j)} = p c_j$, $c_j = (1 + \varepsilon)^{-j}$, $w_j = 1/(j(j+1))$ and $\varepsilon = 1/\ln(1/\delta)$.

Theorem 15 (Anytime validity). *Under (A1)–(A3) and H_0 , (K_n) is a nonnegative supermartingale with $K_0 \leq 1$. Hence $\mathbb{P}(\exists n : K_n \geq 1/\delta) \leq \delta$ for every schedule of shadow evaluations and every stopping or abandonment rule.*

Proof. Positivity. If $A_t = 0$, then $\hat{Z}_t = g_t \geq -1 \geq -\kappa_t$. If $A_t = 1$, then $\hat{Z}_t = g_t(1 - 1/p_t) + Z_t/p_t \geq \bar{G} - (1 + \bar{G})/p_t$. This bound increases in p_t , so it is $\geq -\kappa_t$ because $p_t \geq \pi_t$. Hence every factor is $\geq 1 - \lambda_t^{(j)} \kappa_t \geq 0$. *Unbiasedness.* On $\{p_t > 0\}$, (A2) gives $\mathbb{E}[\hat{Z}_t \mid \mathcal{H}_t, Z_t] = Z_t$. On $\{p_t = 0\}$, $\hat{Z}_t = g_t = Z_t$ by (A3). Since $\mathcal{F}_{t-1} \subseteq \mathcal{H}_t$,

$\mathbb{E}[\hat{Z}_t \mid \mathcal{F}_{t-1}] = \mathbb{E}[Z_t \mid \mathcal{F}_{t-1}] \leq 0$. *Supermartingale.* Because $\lambda_t^{(j)}$ is $\mathcal{F}_{t-1}$ -measurable, each product in Equation (10) is a nonnegative supermartingale, and so is their weighted sum, by monotone convergence. *Ville.* Let $\sigma = \inf\{n : K_n \geq 1/\delta\}$. Optional stopping at $\sigma \wedge n$ gives $1 \geq \mathbb{E}K_{\sigma \wedge n} \geq \delta^{-1} \mathbb{P}(\sigma \leq n)$. Let $n \rightarrow \infty$ (Ville, 1939). $\square$

Remark 3. Validity has three limits.

- Bets may not use W_t .
- (A2) excludes outcome-dependent labelling, such as labels that arrive through user complaints.
- Drift is covered only inside H_0 , so folded macros must still be monitored.

Audits can also be triggered by disagreement. Audit only when the two answers differ, and set $g_t = (C_t^L - C_t^M)/B$ on agreement, which (A3) allows only under exact-match scoring (condition (E) of Appendix C.1). Then $p_t \in \{0, 1\}$ and $\kappa_t = 1$, so the test keeps full-information power at a label cost of $\mathbb{P}(\text{answers differ}) \mathbb{E}\tau$. Unlike a sign test on discordant pairs (Shawn, 2026), concordant pairs here carry the deterministic gain $\eta/(1 - \eta)$. A macro slightly *less* accurate than live is therefore still certifiable when $r > r^*$.

The information constant. For a law P of Z on $[-1, M]$ with mean μ , let $g_P(c) = \mathbb{E}_P \ln(1 + cZ)$ and $g^*(P) = \max_{c \in [0, 1]} g_P(c)$, attained at c^* . Let $\text{KL}_{\inf}(P) = \inf\{\text{KL}(P \parallel Q) : \mathbb{E}_Q Z \leq 0\}$ over laws Q on $[-1, M]$.

Lemma 16 (Growth equals information). (The dual form of $\text{KL}_{\inf}$ is due to Honda and Take-mura (2010); we include a short proof for bounded Z .) *$g^*(P) = \text{KL}_{\inf}(P)$, and $g^* > 0$ iff $\mu > 0$. If $\mu > 0$, the infimum is attained by $Q^*(dz) = P(dz)/(1 + c^*z) + w \delta_{-1}(dz)$ with $w = 1 - \mathbb{E}_P[(1 + c^*Z)^{-1}] \geq 0$.*

Proof. ($\leq$). Let $\mathbb{E}_Q Z \leq 0$ and $\ell = dP/dQ$. For every $c \in [0, 1]$, Jensen gives $\mathbb{E}_P \ln \frac{1+cZ}{\ell} \leq \ln \mathbb{E}_Q[(1 + cZ)\mathbf{1}_{\ell>0}] \leq \ln(1 + c\mathbb{E}_Q Z) \leq 0$. ($\geq$). If $\mu \leq 0$, both sides are 0. Otherwise $c^* > 0$. If $c^* < 1$, differentiation under $\mathbb{E}$ is legitimate because $|Z/(1 + cZ)| \leq \max(\frac{1}{c}, \frac{1}{1-c})$. The first-order condition $\mathbb{E}_P[Z/(1 + c^*Z)] = 0$ then shows that $w = 0$ and that Q^* has mass 1 and mean 0. If $c^* = 1$, then $P(\{-1\}) = 0$, and the left derivative $\mathbb{E}_P[Z/(1 + Z)] = w$ is ≥ 0 , so Q^*

again has mass 1 and mean 0. In both cases $\text{KL}(P\|Q^*) = \mathbb{E}_P \ln(1 + c^*Z) = g^*$. $\square$

Since Q^* lives on $\text{supp}P \cup \{-1\}$, in the paired Bernoulli model KL_{inf} equals the information of the three-point model. Locally, $\text{KL}_{\text{inf}} = \frac{\mu^2}{2\sigma^2}(1 + O(\mu))$ with $\sigma^2 = \text{Var}(Z)$.

Observation models. A lower bound on certification time depends on what the rule observes. Let U be an independent randomization of the rule, and let $\xi_1, \xi_2, \dots$ be i.i.d. uniform audit coins, independent of U and of all shadow outcomes.

(O1) *Audited utilities only.* After t shadow evaluations the rule's records are $\mathcal{R}_t = \sigma(U, (A_s, A_s Z_s)_{s \leq t})$, and $A_t = \mathbf{1}\{\xi_t \leq p_t\}$, where p_t is a fixed (law-independent) function of the records $\mathcal{R}_{t-1}$. Unaudited shadow outputs, including whether the two answers agree, are neither used nor allowed to trigger audits.

(O2) *Audited utilities and agreement.* The records also contain the agreement indicators G_s , and $A_t = \mathbf{1}\{\xi_t \leq p_t(G_t)\}$, where $p_t(\cdot)$ is a fixed function of the records before t ; audits may thus depend on G_t but not on Z_t (Appendix C.1).

The IPW certificate ($g \equiv 0$) with audits that do not depend on W_t is an (O1) rule. So is our restart-only certificate, which updates its wealth only on audited pairs.

Theorem 17 (Certification time and optimality). *Let Z_t be i.i.d. P with $\mu > 0$. (a) Suppose audits are i.i.d. Bernoulli(p) and independent of (Z_t) . Under the grid mixture, the label count N_p has the law of the full-information stopping time τ_1 , $\mathbb{E}\tau_p = \mathbb{E}\tau_1/p$, and*

$$\mathbb{E}N_p \leq \frac{\ln \frac{1}{\delta} + 2 \ln \ln \frac{1}{\delta} + O(1)}{\text{KL}_{\text{inf}}(P)},$$

where $O(1)$ depends only on c^* and M . (b) Consider a rule of type (O1) with $p_t \leq \bar{p}$, folding at a stopping time τ of $(\mathcal{R}_t)$. Let $\mathbb{P}_Q(\tau < \infty) \leq \delta$ for every i.i.d. law Q of (Z_t) on $[-1, M]$ with $\mathbb{E}_Q Z \leq 0$, and let $\mathbb{P}_P(\tau < \infty) = 1$. Then

$$\begin{aligned} \mathbb{E}_P \sum_{t \leq \tau} A_t &\geq \ln(1/\delta) / \text{KL}_{\text{inf}}(P), \\ \mathbb{E}_P \tau &\geq \ln(1/\delta) / (\bar{p} \text{KL}_{\text{inf}}(P)). \end{aligned}$$

Hence, among (O1) rules, $\mathbb{E}\tau_p = (1 + o(1)) \ln(1/\delta) / (p \text{KL}_{\text{inf}}(P))$ is optimal as $\delta \rightarrow 0$.

Here P is the law of the unweighted Z_t . The audit rate enters only through the factor $1/p$ (proof of (a)); expressing the rate through the law of the weighted increment and dividing by p again would count p twice. Part (b) does not extend beyond (O1): a rule that also sees whether the answers agree can be faster (Proposition 21 and Theorem 22).

Proof. (a) With $\lambda^{(j)} = pc_j$, the factor $1 + c_j A_t Z_t$ equals 1 on unaudited steps. The wealth therefore moves only at audits, whose outcomes are i.i.d. P and independent of the Geometric(p) gaps. So $N_p \stackrel{d}{=} \tau_1$, and Wald's identity gives $\mathbb{E}\tau_p = \mathbb{E}N_p/p$. Pick j^* with $c_{j^*} \in [c^*/(1 + \varepsilon), c^*]$. By concavity, $g(c_{j^*}) \geq g^*/(1 + \varepsilon)$. Since $K_n \geq w_{j^*} \prod_{t \leq n} (1 + c_{j^*} Z_t)$, Wald's identity with overshoot $\leq \ln(1 + M)$ gives

$$\begin{aligned} \mathbb{E}\tau_1 &\leq \frac{1+\varepsilon}{g^*} [\ln \frac{1}{\delta} + \ln(j^*(j^* + 1)) \\ &\quad + \ln(1 + M)]. \end{aligned}$$

Finally, $j^* \leq 1 + 2 \ln(1/c^*) \ln(1/\delta)$.

(b) Take Q^* from Lemma 16, so that $dP/dQ^* = 1 + c^*z$ on $\text{supp}P$, and put $\ell = \ln(1 + c^*Z)$. Under an i.i.d. law Q of (Z_t) the records $\mathcal{R}_n$ have, with respect to the law of U times a fixed dominating measure, the density $\prod_{t \leq n} p_t^{A_t} (1 - p_t)^{1-A_t} q(Z_t)^{A_t}$, where q is the density of Q . Under (O1) the audit factors are the same functions of the records under P and Q^* , because the coins ξ_t and the map $\mathcal{R}_{t-1} \mapsto p_t$ do not depend on the law; they cancel. The likelihood ratio of $\mathcal{R}_n$ is therefore $\exp \sum_{t \leq n} A_t \ell(Z_t)$. (This is the step that fails outside (O1): if the records contain W_t , or audits depend on W_t , the ratio also contains the likelihood ratio of W_t , whose law may differ between P and every null.) Since Z_t and ξ_t are independent of $\mathcal{R}_{t-1}$ and of each other, $\mathbb{E}_P[A_t \ell(Z_t) \mid \mathcal{R}_{t-1}] = p_t g^*$. By optional stopping at $\tau \wedge n$, the KL divergence between the laws of the records up to $\tau \wedge n$ is $g^* \mathbb{E}_P \sum_{t \leq \tau \wedge n} A_t$. Data processing on $\{\tau \leq n\}$, whose Q^* -probability is $\leq \delta$, together with $\text{kl}(x, y) \geq x \ln \frac{1}{y} - h(x)$, bounds this below by $x_n \ln \frac{1}{\delta} - h(x_n)$, where $x_n = \mathbb{P}_P(\tau \leq n) \rightarrow 1$. Now let $n \rightarrow \infty$ (Kaufmann et al., 2016; Garivier and Kaufmann, 2016). Finally, $\mathbb{E}_P \sum_{t \leq \tau} A_t = \mathbb{E}_P \sum_{t \leq \tau} p_t \leq \bar{p} \mathbb{E}_P \tau$. $\square$

Why the plug-in reliability test fails. The SPRT is a Kelly bet: $e^{\Lambda_n} = \prod_{k \leq n} (1 + b(Y_k - r_0))$ with $b = \frac{r_1 - r_0}{r_0(1 - r_0)}$. With $r_0 = r^*$ it is the betting test on $(Y^M - r^*)/r^*$, and it is valid only if a is known.

Proposition 18 (Plug-in tests are not level δ ; re-states Proposition 5). *Let $\hat{a} = k/n_a$ with $k \sim \text{Bin}(n_a, a)$, estimated from independent live runs. Run the one-sided SPRT of Section 4 with $r_0 = \hat{a} - \eta$ and $r_1 = \hat{a}$ on i.i.d. macro outcomes at the break-even $r = r^*$. Let $E = \{\eta < \hat{a} < 1\}$ and*

$$r_{\text{mid}} = \ln \frac{1-r_0}{1-r_1} / \ln \frac{r_1(1-r_0)}{r_0(1-r_1)} \in (r_0, r_1).$$

Then:

$$(i) \mathbb{P}(\text{fold}) \geq \mathbb{P}(E, r_{\text{mid}} < r^*) \geq \mathbb{P}(\eta < \hat{a} \leq r^*);$$

$$(ii) \mathbb{P}(\text{fold}) \geq \mathbb{E}[\mathbf{1}_E \mathbf{1}_{\{r_0 < r^*\}} (\delta r_0 / r_1)^\theta].$$

In (ii), $\theta = 0$ if $r^* \geq r_{\text{mid}}$. Otherwise $\theta \in (0, 1)$ solves $r^*(r_1/r_0)^\theta + (1 - r^*)(\frac{1-r_1}{1-r_0})^\theta = 1$. For large n_a , the middle term of (i) is $\approx \Phi(-\frac{\eta}{2}\sqrt{n_a/(a(1-a))})$.

Proof. Condition on $\hat{a} \in E$. The increments X_k of Λ are i.i.d. with mean $\text{KL}(r^*||r_0) - \text{KL}(r^*||r_1)$. This mean is affine in r^* and vanishes at r_{mid} . If $r^* \geq r_{\text{mid}}$, the drift is nonnegative and the steps are non-degenerate, so $\limsup \Lambda_n = \infty$ and the test folds a.s. On E , $\hat{a} \leq r^*$ implies $r_{\text{mid}} < r_1 \leq r^*$, which proves (i). For (ii), let $r_0 < r^* < r_{\text{mid}}$. The map $\theta \mapsto \mathbb{E}e^{\theta X}$ is strictly convex. It equals 1 at 0 with negative slope and equals $1 + b(r^* - r_0) > 1$ at 1, so the root θ exists and $e^{\theta \Lambda_n}$ is a martingale. At the fold time σ , $\Lambda_\sigma \leq \ln \frac{1}{\delta} + \ln \frac{r_1}{r_0}$; on $\{\sigma = \infty\}$, $\Lambda_n \rightarrow -\infty$. Bounded convergence therefore gives $1 = \mathbb{E}[e^{\theta \Lambda_\sigma}; \sigma < \infty] \leq (r_1/(\delta r_0))^\theta \mathbb{P}(\sigma < \infty | \hat{a})$. $\square$

n_a	10	20	50	100	200	500	1000
$\delta = 0.05$	.28	.34	.30	.25	.19	.11	.081
$\delta = 0.01$	.27	.33	.26	.20	.13	.058	.030
bound (i)	.26	.32	.23	.20	.10	.025	.003

Table 4: Exact false-fold probability of the plug-in SPRT for a macro at break-even, computed by dynamic programming ($a = 0.9$, $(m-1)/L = 0.05$, horizon 2000 uses).

Table 4 shows that the excess over δ persists long after the middle term of (i) drops below δ . At $n_a = 1000$ and $\delta = 0.01$, bound (ii) gives 0.035 and the exact value is 0.034. Macros slightly below break-even behave similarly. At shortfall 0.04 with $n_a \leq 20$ live labels, the rate is 0.1–0.4 across $a \in [0.5, 0.9]$ and $\eta \in [0.05, 0.2]$. On the full KQA-STREAM tables (Qwen3-8B, 30 streams)

both certificates stay far below δ on below-break-even candidates. The plug-in SPRT folds 0.32% of them (477 of 147,349 encounters) and the betting test 0.017% (33 of 190,187), while counter rules fold all of them. The failure of the plug-in test is a boundary phenomenon: it needs candidates just below break-even and live accuracy estimated from few labels. Most spurious candidates in our streams are far below break-even: the ones that counters fold fall short by a median of 0.68.

Pairing. Write $P_1 \leq_{\text{cx}} P_2$ if $\mathbb{E}\phi(Z_1) \leq \mathbb{E}\phi(Z_2)$ for all convex ϕ .

Theorem 19 (Pairing gain). *If $P_1 \leq_{\text{cx}} P_2$ on $[-1, M]$, then $\text{KL}_{\text{inf}}(P_1) \geq \text{KL}_{\text{inf}}(P_2)$, strictly if $P_1 \neq P_2$ and $\mu > 0$. Now take the paired Bernoulli model with marginals (a, r) fixed and $r > r^*$. Then KL_{inf} is strictly increasing in $\gamma = \text{Cov}(Y^L, Y^M)$. Pairing therefore certifies faster than an unpaired design (live outcome from an independent instance) iff $\gamma > 0$, that is, when live and macro fail on the same instances.*

Proof. For $c < 1$, $z \mapsto \ln(1 + cz)$ is concave and bounded, so $g_{P_1}(c) \geq g_{P_2}(c)$. Let $c \uparrow 1$ and maximize. For strictness, Strassen’s theorem gives a coupling $Z_2 = Z_1 + \xi$ with $\mathbb{E}[\xi | Z_1] = 0$ and $\xi \neq 0$ with positive probability. At $c = c^*(P_2)$ the map $\ln(1 + cz)$ is strictly concave where finite, so conditional Jensen is strict. Raising γ by Δ moves mass Δ from each of $D = \pm 1$ to $D = 0$. This changes $\mathbb{E}\phi(D)$ by $-\Delta[\phi(1) + \phi(-1) - 2\phi(0)] \leq 0$ for convex ϕ , and Z is affine in D . $\square$

At $a = r = 0.9$ and $\eta = 0.05$, pairing raises KL_{inf} by $1.85\times$ at correlation 0.5 and by $7.56\times$ at correlation 1. Near break-even, it beats even the SPRT that *knows* a iff the correlation exceeds $\sqrt{a(1-a)}/(2\sqrt{r^*(1-r^*)})$, which is 0.42 here.

The scaling law with KL_{inf} .

Corollary 20 (Coverage gap with KL_{inf}). *Assume (A1)–(A3) and (A5) of Appendix B, and replace (A4) by: in every epoch the normalized paired utilities $Z_t = U_t/B$ of the audited shadow evaluations are i.i.d. from a finitely supported law P_Z on $[-1, M]$ with positive mean, whichever candidate they evaluate, and independent of $\mathcal{G}$. Let*

$$\Theta_U := \frac{\nu \ln(1/\delta)}{p \text{KL}_{\text{inf}}(P_Z)}.$$

(i) Every policy of type (O1) that is δ -safe in the sense of Definition 1, with “bad” meaning

$\mathbb{E}Z \leq 0$, has $\liminf M \geq (1 - \varepsilon_\delta)F_\pi(\Theta_U)$ almost surely, with $\varepsilon_\delta = O(\sqrt{\ln \ln(1/\delta)/\ln(1/\delta)})$ depending only on (δ, P_Z) . (ii) With announced changes, the grid-mixture betting test of Theorem 17(a), restarted at each change with all earlier evidence discarded, is δ -safe and has $M \leq (1 + (2 \ln \ln \frac{1}{\delta} + O(1))/\ln \frac{1}{\delta})F_\pi(\Theta_U)$. Thus Theorem 3 holds with $\text{KL}_{\inf}(P_Z)$ in place of $\text{KL}(r_1 \| r_0)$, and the live accuracy a need not be known.

Proof. (i) Repeat the converse proof of Theorem 3 with three substitutions. *The null environment.* Let Q^* be the least-favorable null of Lemma 16; it has mean 0, so in the environment where the epoch's audited utilities are i.i.d. Q^* every candidate is bad, and δ -safety bounds the probability of a fold event in the epoch by δ given $\mathcal{H}_e$. *The likelihood ratio.* By Lemma 16, $dP_Z/dQ^* = 1 + c^*z > 0$ on $\text{supp}P_Z$, so $P_Z \ll Q^*$. Under (O1) the records add to the audited utilities only the audit coins and the internal randomization, whose laws do not depend on the environment, exactly as in the proof of Theorem 17(b). Hence $d\mathbb{P}_{P_Z}/d\mathbb{P}_{Q^*}$ restricted to $\mathcal{F}_m$ is e^{W_m} with $W_m = \sum_{k \leq m} \ell(Z_k)$ and $\ell(z) = \ln(1 + c^*z)$, whichever candidates the policy forms or audits. Wald's likelihood-ratio identity then gives the low-evidence bound $\mathbb{E}_{P_Z}[R; W_N \leq c, N < \infty] \leq e^c \mathbb{E}_{Q^*}[R]$ as before. *The drift.* Under P_Z the increments $\ell(Z_k)$ are i.i.d., bounded (the support is finite and $1 + c^*z > 0$ on it) and have mean $g^* = \text{KL}_{\inf}(P_Z)$. With $n^* = \ln(1/\delta)/\text{KL}_{\inf}(P_Z)$, $n - 1 = \lfloor (1 - \varepsilon)n^* \rfloor$ and $c = (1 - \varepsilon/2) \ln(1/\delta)$, Azuma's maximal inequality gives $\beta_{n-1}(c) \rightarrow 0$, and the rest of the proof (one epoch, many epochs, choice of n and c) applies verbatim with Θ replaced by Θ_U . (ii) After each change the test restarts from wealth 1 on the epoch's own evidence, so Theorem 15 and Ville's inequality bound the probability of a fold of a null candidate in the epoch by δ , which is Definition 1. The wealth moves only at audits (Theorem 17(a)), so the test folds at the N_e -th audited outcome of epoch e , where the N_e are i.i.d., independent of arrivals, audits and changes, and have the law of N_p in Theorem 17(a). Lemma 10 gives $u_i = 1 - \mathbb{E}\rho_i^N$, and Lemma 12 gives $u_i \leq 1 - \rho_i^{\mathbb{E}N} \leq 1 - e^{-x\mathbb{E}N/\pi_i}$ with $x = \nu/p$. By Theorem 17(a), $x\mathbb{E}N \leq (1 + \eta_\delta)\Theta_U$ with $\eta_\delta = (2 \ln \ln \frac{1}{\delta} + O(1))/\ln \frac{1}{\delta}$. Since F_π is concave and nondecreasing with $F_\pi(0) = 0$, $F_\pi((1 + \eta)\Theta_U) \leq (1 + \eta)F_\pi(\Theta_U)$; summing $\pi_i u_i$ gives the bound. $\square$

Pairing, disagreement-triggered audits and cost-aware utilities thus all act through one constant, $\text{KL}_{\inf}(P_Z)$, in the certification load. Under Zipf demand, a pairing gain $\rho_g = \text{KL}_{\inf}^{\text{pair}}/\text{KL}_{\inf}^{\text{unpair}}$ lowers the coverage gap by the factor $\rho_g^{1-1/s}$.

C.1 Agreement-aware certification

Shadow mode runs both paths, so the system sees both answers before any audit. When the answers agree, their correctness labels agree, provided scoring depends on the answer alone:

- (E) *Exact-match scoring.* There are $G_t \in \{0, 1\}$ and $z_{0,t}$, observed before the audit decision (functions of W_t), such that $G_t = 1$ implies $Z_t = z_{0,t}$.

(E) holds if correctness is a deterministic function of the instance and the final answer, $Y = \phi_t(\text{answer})$ (exact match to a gold answer, or execution against fixed tests), if $G_t = 1$ exactly when the two final answers are identical as inputs to ϕ_t , and if both costs are observed; then $G_t = 1$ forces $Y_t^M = Y_t^L$, and $z_{0,t} = (C_t^L - C_t^M)/B$. It fails for stochastic or rubric-based judges, for judges that read more than the final answer, and whenever a path's score depends on anything other than its answer. KQA-STREAM is such a case: a macro answer counts as correct only if the query was also routed to the right template, so on a misrouted query whose macro answer equals a correct live answer, $G_t = 1$ but $Y_t^M = 0 \neq Y_t^L = 1$. The certificates of this subsection are therefore not valid for the KQA-STREAM labels as defined; they are valid for a variant that scores a macro by its answer alone.

Proposition 21 (Agreement breaks the (O1) bound). *Assume (E) in the paired model with constant costs, so $z_0 = \eta/(1 - \eta)$. Let $V_t = z_0$ if $G_t = 1$ and $V_t = -1$ otherwise. For every $\lambda \in [0, 1]$, $\prod_{s \leq t} (1 + \lambda V_s)$ is a test supermartingale under H_0 , for any audit mechanism. If G_t is i.i.d. with $\mathbb{P}(G_t = 1) = g$, the best constant bet grows the log-wealth by $\text{kl}(g \| \frac{1}{1+z_0})$ per shadow run if $g > \frac{1}{1+z_0}$, and by 0 otherwise. If $g = 1$, the bet $\lambda = 1$ folds after $\lceil \ln(1/\delta)/\ln(1+z_0) \rceil$ shadow runs and uses no labels, whereas every (O1) rule needs $\mathbb{E}_P \tau \geq \ln(1/\delta)/(p \ln(1+z_0))$.*

Proof. By (E), $V_t = Z_t$ if $G_t = 1$, and $V_t = -1 \leq Z_t$ otherwise. So $V_t \leq Z_t$ pointwise, and $\mathbb{E}[V_t | \mathcal{F}_{t-1}] \leq \mathbb{E}[Z_t | \mathcal{F}_{t-1}] \leq 0$ whatever the

audits do. The factors satisfy $1 + \lambda V_t \geq 1 - \lambda \geq 0$, and V_t is observed without a label. For i.i.d. G_t , V_t has the two-point law $g\delta_{z_0} + (1 - g)\delta_{-1}$. By Lemma 16 the best constant bet grows at rate $\text{KL}_{\inf}$ of this law, and its least-favorable null lives on $\{z_0, -1\}$. A null law $q\delta_{z_0} + (1 - q)\delta_{-1}$ has mean $qz_0 - (1 - q) \leq 0$ iff $q \leq 1/(1 + z_0)$, and $\text{kl}(g\|q)$ decreases in q on $[0, g]$, which gives the rate. If $g = 1$, the wealth is $(1 + z_0)^n$. For (O1) rules, $P_Z = \delta_{z_0}$ and $\text{KL}_{\inf}(\delta_{z_0}) = \ln(1 + z_0)$, so Theorem 17(b) applies. $\square$

At $\eta = \delta = 0.05$ and $p = 0.1$ the agreement bet folds after 59 shadow runs, while (O1) rules need at least 584 (the grid-mixture IPW certificate needs 910). Under the least-favorable null $q = 1/(1 + z_0)$ with $Z = -1$ on disagreement, the bet $\lambda = 1$ folds with probability $q^{59} = 0.048 \leq \delta$. The rule is powerless unless agreement is nearly certain ($g > 0.95$ here), because an unaudited disagreement is charged the worst case -1 . The certificate below also bets on audited disagreements, with inverse-propensity weights.

The agreement-aware certificate. Assume (E), (A1), (A2), and $p_t \geq \pi_t > 0$ on $\{G_t = 0\}$ (agreements need no audits). Let

$$\mathcal{C}_t = \{(\lambda, u) : \lambda \geq 0, u \geq -1, \lambda + u \leq \pi_t(1 + u)\}.$$

For $\mathcal{F}_{t-1}$ -measurable $(\lambda_t, u_t) \in \mathcal{C}_t$, the *agreement-aware factor* is

$$F_t = \begin{cases} 1 + \lambda_t z_{0,t}, & G_t = 1, \\ 1 + u_t + A_t(\lambda_t Z_t - u_t)/p_t, & G_t = 0. \end{cases}$$

Equivalently, $F_t = 1 + \lambda_t X_t$ with $X_t = z_{0,t}$ on agreement and $X_t = v_t + A_t(Z_t - v_t)/p_t$ on disagreement, where $v_t = u_t/\lambda_t$ (for $\lambda_t > 0$): an inverse-propensity estimator whose surrogate is exact on agreement and the constant v_t on disagreement. The *conservative* increment

$$X_t^0 = A_t Z_t + (1 - A_t)[G_t z_{0,t} - (1 - G_t)]$$

(audited: Z_t ; unaudited agreement: $z_{0,t}$; unaudited disagreement: -1 , or any known lower bound on Z_t) is bet on with $\lambda \in [0, 1]$ and no weights. As before, $K_n = \sum_k w_k \prod_{t \leq n} F_t^{(k)}$ with $\sum_k w_k \leq 1$, and the certificate folds when $K_n \geq 1/\delta$.

For the rate, take the i.i.d. model: (G_t, Z_t) are i.i.d. with law P satisfying (E) with a constant $z_0 \geq 0$, $g = P(G = 1)$, Z given $G = 0$ has a finitely supported law P_{dis} on $[-1, M]$, and audits

are i.i.d. Bernoulli(p), independent of everything else. A law Q of (G, Z) that satisfies (E) with the same z_0 is *structural*; write $q = Q(G = 1)$ and R for the law of Z given $G = 0$. Define

$$D^*(P; p) = \inf \{ \text{kl}(g\|q) + p(1 - g)\text{KL}(P_{\text{dis}}\|R) \},$$

over structural Q with $qz_0 + (1 - q)\mathbb{E}_R Z \leq 0$. The first term is the information in the agreement frequency, the second the information in the audited disagreements. Let $\Gamma(\lambda, u) = \mathbb{E}_P \ln F$ be the growth rate of the constant bet $(\lambda, u) \in \mathcal{C}$ (with $\pi_t = p$; for $p = 1$, where Γ does not depend on u , we also require $u \leq 0$, so that $\mathcal{C}$ is compact), and call (λ, u) *strictly feasible* if $\lambda > 0$, $u > -1$ and $\lambda + u < p(1 + u)$.

Theorem 22 (Agreement-aware certificate). (i) Validity. Under (E), (A1), (A2), $p_t \geq \pi_t > 0$ on $\{G_t = 0\}$ and H_0 , (K_n) is a test supermartingale, so the certificate folds with probability at most δ . The conservative certificate is a test supermartingale under (E) and H_0 alone, for any audit mechanism, including outcome-dependent audits.

(ii) Information. In the i.i.d. model, $D^*(P; p) = \max_{\mathcal{C}} \Gamma$; $D^* > 0$ iff $\mathbb{E}_P Z > 0$; some maximizer has $u^* \leq 0$; and

$$D^* \geq \max \left\{ p \text{KL}_{\inf}(P_Z), \text{KL}_{\inf}(P_{X^0}), \text{kl}\left(g \parallel \frac{1}{1+z_0}\right) \mathbf{1}_{\{g > \frac{1}{1+z_0}\}} \right\}.$$

(iii) Converse. Consider a rule of type (O2) with $p_t(0) \leq p$ that folds at a stopping time τ of its records. Suppose $\mathbb{P}_Q(\tau < \infty) \leq \delta$ for every i.i.d. structural Q with $\mathbb{E}_Q Z \leq 0$, and $\mathbb{P}_P(\tau < \infty) = 1$. Then $\mathbb{E}_P \tau \geq \ln(1/\delta)/D^*(P; p)$.

(iv) Achievability. For any finite set of strictly feasible (λ_k, u_k) with $u_k \leq 0$ and weights w_k , the mixture has

$$\mathbb{E}_P \tau \leq \min_{k: \Gamma_k > 0} \frac{\ln \frac{1}{\delta} + \ln \frac{1}{w_k} + \ln \frac{1+M}{p}}{\Gamma(\lambda_k, u_k)}.$$

With grids refined as $\delta \rightarrow 0$ and $\ln(1/w_k) = o(\ln \frac{1}{\delta})$, $\mathbb{E}_P \tau = (1 + o(1)) \ln(1/\delta)/D^*(P; p)$. The conservative certificate with the grid mixture of Theorem 17(a) (with $p = 1$) has $\mathbb{E}_P \tau \leq (\ln \frac{1}{\delta} + 2 \ln \ln \frac{1}{\delta} + O(1))/\text{KL}_{\inf}(P_{X^0})$.

Proof. (i) Nonnegativity. Since $\pi_t \leq 1$ and $1 + u \geq 0$, $\lambda \leq \pi_t(1 + u) - u \leq 1$, so $1 + \lambda z_{0,t} \geq 1 - \lambda \geq 0$ on agreement. On disagreement without audit, $F_t = 1 + u_t \geq 0$. With audit, $Z_t \geq -1$

gives $F_t \geq 1 + u_t - (\lambda_t + u_t)/p_t$. This is $\geq 1 + u_t \geq 0$ if $\lambda_t + u_t \leq 0$; otherwise it increases in p_t and is $\geq 1 + u_t - (\lambda_t + u_t)/\pi_t \geq 0$ because $(\lambda_t, u_t) \in \mathcal{C}_t$. Mean. On $\{G_t = 1\}$, (E) gives $F_t = 1 + \lambda_t Z_t$. On $\{G_t = 0\}$, (A2) and $p_t > 0$ give $\mathbb{E}[F_t \mid \mathcal{H}_t, Z_t] = 1 + u_t + \lambda_t Z_t - u_t = 1 + \lambda_t Z_t$. Since λ_t, u_t are $\mathcal{F}_{t-1}$ -measurable and $\lambda_t \geq 0$, $\mathbb{E}[F_t \mid \mathcal{F}_{t-1}] = 1 + \lambda_t \mathbb{E}[Z_t \mid \mathcal{F}_{t-1}] \leq 1$. The mixture and Ville's inequality follow as in Theorem 15. For the conservative certificate, $X_t^0 = Z_t$ if $A_t = 1$, $X_t^0 = 0$ if $A_t = 0$ and $G_t = 1$ (by (E)), and $X_t^0 = -1 \leq Z_t$ otherwise. So $X_t^0 \leq Z_t$ pointwise, whatever the audits depend on, X_t^0 is observable, and $1 + \lambda X_t^0 \geq 1 - \lambda \geq 0$.

(ii) Let $O = (G, A, AZ)$. Because A is independent Bernoulli(p) under P and under every structural Q , and $AZ = Az_0$ on $\{G = 1\}$, the chain rule gives $\text{KL}(P_O \parallel Q_O) = \text{kl}(g \parallel q) + p(1 - g)\text{KL}(P_{\text{dis}} \parallel R)$. So D^* is the infimum of $\text{KL}(P_O \parallel Q_O)$ over structural nulls. *Weak duality.* The computation in (i) with $p_t = p$ gives $\mathbb{E}_Q F = 1 + \lambda \mathbb{E}_Q Z \leq 1$ for every $(\lambda, u) \in \mathcal{C}$ and structural null Q . As in Lemma 16, Jensen then gives $\Gamma(\lambda, u) = \mathbb{E}_P \ln F \leq \text{KL}(P_O \parallel Q_O)$. Hence $\sup_{\mathcal{C}} \Gamma \leq D^*$. If $\mathbb{E}_P Z \leq 0$, P is itself a structural null, so $D^* = 0 = \Gamma(0, 0)$. Let $\mathbb{E}_P Z > 0$. The set $\mathcal{C}$ is a compact polygon, and Γ is concave and upper semicontinuous with values in $[-\infty, \infty)$, so it has a maximizer (λ^*, u^*) . For $\lambda \leq p$, $(\lambda, 0) \in \mathcal{C}$ and $\frac{d}{d\lambda} \Gamma(\lambda, 0)|_{\lambda=0} = \mathbb{E}_P Z > 0$, so $\Gamma^* := \Gamma(\lambda^*, u^*) > 0$ and $\lambda^* > 0$. *Case $g = 1$.* Then $\Gamma = \ln(1 + \lambda z_0)$, which is maximal at $(1, -1) \in \mathcal{C}$ with value $\ln(1 + z_0)$. The structural law with $q = 1/(1 + z_0)$ and $R = \delta_{-1}$ has mean 0 and $\text{KL}(P_O \parallel Q_O) = \ln(1 + z_0)$, so $D^* = \Gamma^*$ and $u^* = -1$. *Case $g < 1$.* Since Γ^* is finite and P has finite support, all factors are positive at (λ^*, u^*) on the support of P_O , and Γ is differentiable nearby, with $F_d(z) := 1 + u + (\lambda z - u)/p$,

$$\begin{aligned} \partial_\lambda \Gamma &= \frac{gz_0}{1 + \lambda z_0} + (1 - g) \mathbb{E}_{P_{\text{dis}}} \frac{Z}{F_d}, \\ \partial_u \Gamma &= (1 - g)(1 - p) \left[\frac{1}{1 + u} - \mathbb{E}_{P_{\text{dis}}} \frac{1}{F_d} \right]. \end{aligned}$$

The constraint $\lambda \geq 0$ is inactive. Write the others as $c_1 = 1 + u \geq 0$ and $c_2 = p(1 + u) - u - \lambda \geq 0$. The KKT conditions for a concave function on a polytope give $\nu_1, \nu_2 \geq 0$ with $\nu_i c_i = 0$, $\partial_\lambda \Gamma = \nu_2$ and $\partial_u \Gamma = \nu_2(1 - p) - \nu_1$. If $p < 1$, $\Gamma = -\infty$ at $u = -1$, so $\nu_1 = 0$. If $p = 1$, $\partial_u \Gamma \equiv 0$; take the maximizer with $u^* = 0$, where $c_1 > 0$, so $\nu_1 = 0$ and the constraint $u \leq 0$ gets multiplier

0. If $\nu_2 > 0$, then $c_2 = 0$, so $F_d(-1) = 0$ and $P_{\text{dis}}(\{-1\}) = 0$. Define a measure Q_O^* on the values of O : mass $g/(1 + \lambda^* z_0)$ on agreement, $(1 - g)(1 - p)/(1 + u^*)$ on an unaudited disagreement, and $(1 - g)pP_{\text{dis}}(z)/F_d(z) + p\nu_2 \mathbf{1}_{\{z=-1\}}$ on an audited disagreement with $Z = z$. On the support of P_O , $dP_O/dQ_O^* = F^*$, and the added atom lies outside it. *Mass.* Since $F - 1$ is linear in (λ, u) , $\lambda \partial_\lambda \Gamma + u \partial_u \Gamma = \mathbb{E}_P[1 - 1/F] = 1 - (Q_O^*(\text{all}) - p\nu_2)$. By the KKT conditions the left side equals $\nu_2(\lambda^* + u^*(1 - p))$, and this is $p\nu_2$ because $\nu_2 > 0$ forces $\lambda^* = p(1 + u^*) - u^*$. So Q_O^* has mass 1. *Structure.* For $p < 1$, the u -condition with $\nu_1 = 0$ reads $(1 - g)/(1 + u^*) - (1 - g)\mathbb{E}_{P_{\text{dis}}}[1/F_d] = \nu_2$, that is, $Q_O^*(\text{unaudited disagreement})/(1 - p) = Q_O^*(\text{audited disagreement})/p =: 1 - q^*$. With the mass identity, the agreement mass is q^* . So Q_O^* is the observation law (the audit coin is Bernoulli(p) under both laws) of the structural Q^* with $Q^*(G = 1) = q^*$ and $R^* = Q_O^*(\text{audited disagreement}, \cdot)/(p(1 - q^*))$. For $p = 1$ there are no unaudited disagreements, and $1 - q^*$ is the audited disagreement mass. *Mean.* $\mathbb{E}_{Q^*} Z = q^* z_0 + (1 - q^*)\mathbb{E}_{R^*} Z = \partial_\lambda \Gamma - \nu_2 = 0$. So Q^* is a structural null and $D^* \leq \text{KL}(P_O \parallel Q_O^*) = \mathbb{E}_P \ln F^* = \Gamma^* \leq D^*$. *Sign of u^* .* For $p < 1$, the structure identity gives $1 + u^* = (1 - g)/(1 - q^*)$, and $q^* = g/(1 + \lambda^* z_0) \leq g$ because $z_0 \geq 0$. Hence $u^* \leq 0$. For $p = 1$, u does not enter Γ , and $u^* = 0$ is a maximizer. *Lower bounds.* Under a structural null Q , the pair (A, AZ) is a function of O with law Bernoulli(p) $\otimes Q_Z$ and $\mathbb{E}_Q Z \leq 0$, so data processing gives $\text{KL}(P_O \parallel Q_O) \geq p\text{KL}(P_Z \parallel Q_Z) \geq p\text{KL}_{\text{inf}}(P_Z)$. Likewise X^0 is a function of O with $\mathbb{E}_Q X^0 \leq \mathbb{E}_Q Z \leq 0$, which gives $\text{KL}_{\text{inf}}(P_{X^0})$. Finally G is a function of O , and $\mathbb{E}_R Z \geq -1$ forces $q \leq 1/(1 + z_0)$, which gives the kl term. Take the infimum over Q .

(iii) Let Q^* be the structural null from (ii). The records after n runs are $(U, (G_t, A_t, A_t Z_t)_{t \leq n})$. Their density under an i.i.d. structural law is a product of $Q(G_t)$, of the audit factors $p_t(G_t)^{A_t}(1 - p_t(G_t))^{1 - A_t}$, and of the density of R at Z_t when $A_t = 1$ and $G_t = 0$; on $\{G_t = 1\}$, $A_t Z_t = A_t z_0$ carries no further information. The audit factors are the same under P and Q^* . The log-likelihood ratio is therefore $\sum_{t \leq n} \ell_t$ with

$$\ell_t = \ln \frac{P(G_t)}{Q^*(G_t)} + A_t(1 - G_t) \ln \frac{dP_{\text{dis}}}{dR^*}(Z_t),$$

which is bounded because the supports are finite and $R^* > 0$ on $\text{supp } P_{\text{dis}}$. Its conditional mean

under P given the records before t is $\text{kl}(g\|q^*) + (1 - g)p_t(0)\text{KL}(P_{\text{dis}}\|R^*) \leq D^*$. Optional stopping at $\tau \wedge n$ and data processing on $\{\tau \leq n\}$ give $D^*\mathbb{E}_P \tau \geq x_n \ln(1/\delta) - h(x_n)$ with $x_n = \mathbb{P}_P(\tau \leq n) \rightarrow 1$, as in Theorem 17(b).

(iv) For a strictly feasible (λ_k, u_k) with $u_k \leq 0$, the factors lie between a positive constant and $(1 + M)/p$, since $1 + u_k(1 - 1/p) + \lambda_k M/p \leq (1 + M)/p$. The log-wealth S_n of this bet has i.i.d. bounded increments of mean $\Gamma_k > 0$. Since $K_n \geq w_k e^{S_n}$, τ is at most the first n with $S_n \geq \ln(1/(\delta w_k))$, and Wald’s identity with overshoot at most $\ln((1 + M)/p)$ gives the bound. For the limit, take $\theta \in (0, 1)$ and a maximizer with $u^* \leq 0$. The point $\theta(\lambda^*, u^*)$ has $\lambda = \theta\lambda^* > 0$, $u = \theta u^* \in (-1, 0]$ and $c_2 = (1 - \theta)p + \theta c_2(\lambda^*, u^*) \geq (1 - \theta)p > 0$, so it is strictly feasible, and $\Gamma(\theta\lambda^*, \theta u^*) \geq \theta\Gamma^* + (1 - \theta)\Gamma(0, 0) = \theta\Gamma^*$ by concavity. Γ is continuous on the strictly feasible set, so fine grids contain points with $\Gamma \geq \theta^2\Gamma^*$; let $\theta \rightarrow 1$. The conservative certificate is Theorem 17(a) applied to the i.i.d. $X_t^0 \in [-1, M]$ with $p = 1$. $\square$

Implementation. A replay policy implements the certificate as follows. The agreement indicator is $G_t = 1$ iff both final answers are non-empty and identical under the scorer’s own comparison, and (E) must hold for the benchmark’s labels. On a shadow run with live cost c_t , put $z_{0,t} = (c_t - 1)/B$ (clipped to $[-1, 1]$ as for Z_t), and on an audited disagreement compute Z_t as in the IPW certificate. The bets are $\lambda = c(\pi(1 - s) + s)$ and $u = -s$, where π is the audit floor, c runs over the grid $c_j = (1 + \varepsilon)^{-j}$ of Theorem 17(a) with weights $1/(j(j + 1))$ (truncated at $c_j \geq 1/200$), and s runs over $\{0, 0.01, 0.03, 0.1, 0.3, 0.6, 0.9, 0.97, 0.99\}$ with weight $1/2$ for $s = 0$ and $1/16$ for each other value. Every grid point is strictly feasible, because $\lambda + u - \pi(1 + u) = -(1 - c)(\pi(1 - s) + s) < 0$. The factors are $1 + \lambda z_{0,t}$ on agreement (audited or not), $1 - s$ on an unaudited disagreement, and $1 - s + (\lambda Z_t + s)/p_t$ on an audited disagreement. The wealth is updated on every shadow run, not only on audited ones. The slice $s = 0$ is the IPW certificate with the exact surrogate $z_{0,t}$ on agreement. By concavity, $g \ln(1 + cpz_0) \geq pg \ln(1 + cz_0)$, so the growth of its bet c is at least that of the IPW bet c . The bound of Theorem 17(a) therefore holds for the mixture with an extra $\ln 2$ in the numerator.

Numerical checks. On 274 random structural laws ($p \in \{0.05, 0.1, 0.3, 0.7, 1\}$), $\max_C \Gamma$ and a direct convex program over structural nulls agree to $3 \cdot 10^{-12}$. The constructed Q^* has mass 1, mean 0 and the structural ratio to within $5 \cdot 10^{-8}$, $u^* \leq 10^{-8}$, and D^* never falls below the three lower bounds in (ii). The median of $D^*/(p\text{KL}_{\text{inf}}(P_Z))$ is only 1.02: agreement adds little information unless it is frequent. In the paired model with $p = 0.1$ the ratio is 1.00 for live accuracy 0.38 and macro reliability 0.97 (as on KQA-STREAM), 1.18 for $a = r = 0.9$ with correlated errors, and 5.2 when the answers agree with probability 0.99. In 200 simulated runs per setting, the grid above certified in 1.07, 1.03 and 0.29 times the time of the IPW grid mixture at $\delta = 10^{-2}$, and in 1.02, 0.88 and 0.23 times at $\delta = 10^{-6}$. The conservative certificate has rate 0 in the first two settings and 0.027 (against $D^* = 0.030$) in the third. Under five structural nulls, including the least-favorable one, the certificate folded in 2.4–3.3% of 4,000 runs at $\delta = 0.05$ (horizon 4,000).

C.2 Changes, carried-over evidence and the lifetime ledger

Definition 1 counts every fold event of a bad candidate in an epoch, including candidates formed in earlier epochs. We separate what is proved for the *idealized* certificate from what holds for the two implemented ledgers: without restarts and with restarts (the restart-only certificate of Section 5).

Idealized. Changes are announced, all certification evidence is discarded at each change, and one test per template (SPRT, or the grid mixture of Theorem 17) restarts in each epoch at level δ ; if several candidates are tested within an epoch, they share δ , e.g. through a per-epoch ledger. Every fold in an epoch then rests on that epoch’s evidence alone, Ville’s inequality gives Definition 1 for every fold event, and Theorem 3(ii) and Corollary 20 describe this certificate.

Ledger with restarts (restart only). The replay notifies the policy of every edit that changes the gold answer of some instance routed to a template; epochs of a template are the intervals between such edits, within which the macro’s reliability is constant. At each notification, a candidate under test discards its wealth and its shadow statistics and restarts its test with the next ledger index, so the lifetime ledger still sums to δ . Folded macros are not reset; they are monitored by the LR-CUSUM.

By Proposition 23(iv) this certificate satisfies Definition 1 for these epochs *if its own null is constant within them*. The null $\mathbb{E}Z \leq 0$ also depends on live reasoning, whose accuracy and cost can change without any answer change; across such a silent change only Proposition 23(iii) applies (Appendix D.5.1 measures how often this happens and evaluates a variant that also restarts on live-trace changes). It needs to know which templates an edit touches; a coarser change log that also flags edits without effect on a template’s answers triggers extra restarts, which cost coverage but not safety, because every restart consumes a fresh index.

Ledger without restarts. Here the policy is not told when an edit occurs. A candidate keeps its wealth (or its SPRT statistic) across edits; it is neither reset nor discarded. The j -th candidate of a template, counted over the whole stream, is tested at $\delta_j = 6\delta/(\pi^2 j^2)$ with seven fixed bets. For a candidate j and its shadow evaluations $t = 1, 2, \dots$, let ζ_j be the first t with $\mathbb{E}[Z_t | \mathcal{F}_{t-1}] > 0$ ($\zeta_j = \infty$ if there is none). Call a fold of candidate j at evaluation $t < \zeta_j$ a *fold on null evidence*: the candidate has been at or below break-even at every evaluation so far.

Proposition 23 (What the lifetime ledger guarantees). *Let each candidate’s wealth be a mixture of test supermartingales, as in Theorem 15, started at 1 at its formation and compared with $1/\delta_j$.*

- (i) *Lifetime. With probability at least $1 - \delta$, over the whole stream no candidate of the template folds on null evidence. The expected number of such folds per template is at most δ .*
- (ii) *Fresh candidates. Given the history at the start of an epoch, the candidates formed from then on fold on null evidence with total probability at most $\sum_{j>J} \delta_j \leq \delta$, where J is the number of earlier candidates.*
- (iii) *Carried-over candidates. Let candidate j be unfolded at a stopping time c with wealth K_c , and let $e \geq c$ be a stopping time such that the null holds at its evaluations in $[c, e)$. Then it folds in $[c, e)$ with conditional probability at most $\min\{1, \delta_j K_c\}$ given $\mathcal{F}_c$. Equality holds (with $e = \infty$) if, after c , its wealth is a martingale that tends to 0 unless it reaches $1/\delta_j$, and it reaches $1/\delta_j$ without overshoot.*
- (iv) *Per epoch. If the library holds at most one candidate per template, the probability that*

an epoch contains a fold of a candidate that is bad in that epoch is at most $\sum_{j>J} \delta_j + \delta_j K_c \leq \delta + \delta_j (K_c - 1)^+$, where candidate J is carried over with wealth K_c at the epoch start ($\delta_j K_c := 0$ if there is none, in particular if $J = 0$). The ledger satisfies Definition 1 if, at each change, carried-over candidates are discarded or restart their wealth at 1 with the next ledger index (a restarted test then counts as a fresh candidate in (ii)); otherwise it does not in general.

Proof. (i) Fix j and condition on the history at the formation of candidate j . Since $\mathbb{E}[Z_t | \mathcal{F}_{t-1}]$ is $\mathcal{F}_{t-1}$ -measurable, $\{\zeta_j > t\} \in \mathcal{F}_{t-1}$. Let $\tilde{K}_t$ be the wealth with factors replaced by 1 from evaluation ζ_j on. Then $\mathbb{E}[\tilde{K}_t | \mathcal{F}_{t-1}] = \tilde{K}_{t-1}(\mathbf{1}_{\{\zeta_j > t\}} \mathbb{E}[F_t | \mathcal{F}_{t-1}] + \mathbf{1}_{\{\zeta_j \leq t\}}) \leq \tilde{K}_{t-1}$, because $\mathbb{E}[F_t | \mathcal{F}_{t-1}] \leq 1$ whenever the null holds at t (Theorem 15). A fold on null evidence implies $\sup_t \tilde{K}_t \geq 1/\delta_j$, which has conditional probability at most δ_j by Ville’s inequality. Sum over j ; $\sum_j \delta_j = \delta$. Each candidate folds at most once, which gives the expectation bound. (ii) The same argument, conditioned on the history at the epoch start, for $j > J$. (iii) Stop the factors at e and at the first non-null evaluation, as in (i). Given $\mathcal{F}_c$, the stopped wealth $(K_{t \wedge e})_{t \geq c}$ is a nonnegative supermartingale started at K_c ; apply Ville’s inequality to $K_{t \wedge e}/K_c$. In the equality case, the martingale stopped at $1/\delta_j$ is bounded by $1/\delta_j$, so bounded convergence gives $K_c = \mathbb{E}[K_\infty | \mathcal{F}_c] = \delta_j^{-1} \mathbb{P}(\text{fold after } c | \mathcal{F}_c)$. (iv) A fold in the epoch of a candidate that is bad in the epoch is either a fold on null evidence of a candidate formed in the epoch, covered by (ii), or a fold of the carried-over candidate, covered by (iii) with c the epoch start and e its end. Previously folded macros cause no fold event. Add the bounds and use $\sum_{j \geq J} \delta_j \leq \delta$. $\square$

In the paired model the utility null $\mathbb{E}Z \leq 0$ is the reliability condition $r \leq r^*$, so “bad” in Definition 1 (with $r_0 = r^*$) and “null” coincide, up to the online estimates of L and m that the certificate uses; since $r^* = a - (m - 1)/L$ moves with live reasoning, both can change inside an epoch of answer changes.

Numerical check of (iii). We simulated a candidate with the deployed seven-bet mixture ($p = 0.1$, $j = 1$, $\delta = 0.05$) that sees a good utility law for 60 shadow runs and then a null law with mean exactly 0. Of the candidates still unfolded at the

change (mean wealth 4.4), 11.1% folded after it, against the bound $\mathbb{E} \min(1, \delta_1 K_c) = 13.4\%$ and $\delta = 5\%$. After 120 good runs the numbers are 22.8% and 28.1%. So carried-over evidence can defeat per-epoch safety, as (iv) allows.

Restarts without change information. The restarts above use the edit log. The next result shows when per-epoch safety is out of reach without change information; the one after it gives the guarantee of restarts triggered by the memory's own change detector, which needs none.

Under (A1)–(A3) of Appendix B, audits of template i form a Poisson process of rate $\lambda_i := p\pi_i$, independent of the changes. Let

$$\alpha_i := \frac{\lambda_i}{\lambda_i + \nu}$$

be the probability that an epoch of template i contains at least one audit. By memorylessness this holds given any history at the epoch start, and the number of audits in an epoch is geometric, $\mathbb{P}(K \geq k) = \alpha_i^k$. In the notation of Lemma 10, $\alpha_i = \rho_i$.

Proposition 24 (Per-epoch safety without change information). *Let the audited outcomes of an epoch be i.i.d. from the epoch's law, whichever candidate they evaluate, and let a policy observe neither change times nor any signal of them, and fold only at audited outcomes of template i . (i) Upper bound. For every epoch and every history before it, the probability that the epoch contains a fold of template i is at most α_i . So if $\alpha_i \leq \delta$, Definition 1 holds for template i whatever the policy. (ii) Lower bound. Suppose that, with positive probability, the policy reaches a history h after which its next audited outcome of template i , whenever it arrives, folds a candidate if it lies in a set F . Then for every bad law Q , given that an epoch starts after h and before that outcome, the probability that this epoch contains a fold of a bad candidate is at least $\alpha_i Q(F)$, and at least $Q(F)$ given in addition that the epoch contains an audit. The policy is therefore not δ -safe if $\alpha_i Q(F) > \delta$. Consequently, without change information, among certificates whose fold decisions for template i are deterministic functions of its own audited outcomes, (a) every reliability certificate that folds with positive probability violates Definition 1 at every level $\delta < \alpha_i r_0$; (b) every paired-utility certificate with Z taking finitely many values in $[-1, M]$ that folds with positive probability violates it at every level $\delta < \alpha_i/(1 + M)$.*

Proof. (i) A fold of template i in the epoch requires an audit of template i in the epoch. Given the history at the epoch start s , the audits after s form a Poisson process of rate λ_i and the epoch length is exponential with rate ν , independent of each other and of the history ((A1)–(A3)), so the first audit after s precedes the epoch end with probability $\lambda_i/(\lambda_i + \nu) = \alpha_i$. (ii) The policy's action at the next audited outcome is the same function of h and of that outcome whether or not a change occurred, because it observes neither. Changes are the points of a Poisson process independent of the history, so an epoch starts between h and the next audited outcome with positive probability, and Definition 1 must hold for the histories at the start of such epochs; these histories consist of h and the elapsed time. Given such a history, the next audited outcome is the first audit of the epoch, and it falls in the epoch with probability α_i , by the argument of (i). Under Q every candidate is bad in the epoch, and this outcome lies in F with probability $Q(F)$, independently of its arrival time and of the history; on the intersection of the two events a bad candidate folds in the epoch. For (a) and (b), outcomes take finitely many values, so some history h just before the policy's first fold has positive probability; take F the set of outcomes on which the policy folds after h , which is not empty and, by determinism, does not depend on when the next audit arrives. (a) If $1 \in F$, take $Q = \text{Bernoulli}(r_0)$, so $Q(F) \geq r_0$; otherwise $0 \in F$ and $Q = \text{Bernoulli}(0)$ gives $Q(F) = 1$. (b) Take $z \in F$. If $z \leq 0$, the point mass at z is bad and $Q(F) = 1$. If $z > 0$, the law with mass $1/(1+z)$ at z and $z/(1+z)$ at -1 has mean 0, so it is bad, and $Q(F) \geq 1/(1+z) \geq 1/(1+M)$. $\square$

So, for a reliability certificate without change information, per-epoch safety is impossible on every template that is likely enough to be audited in an epoch, $\alpha_i r_0 > \delta$, i.e. with more than $\delta/(r_0 - \delta)$ audits per epoch on average, and automatic on every template with $\alpha_i \leq \delta$; in between it depends on the rule. With $M = 1$ (our clipped utilities) or $r_0 = \frac{1}{2}$, and $\delta = 0.05$, the impossibility covers every template with more than one audit per nine epochs on average. The templates it leaves out are barely coverable even with change information: by Lemma 10, the restarted SPRT of Theorem 3(ii), which needs at least $n_{\min} = \lceil \ln(1/\delta)/\ln(r_1/r_0) \rceil \geq 1$ audits in an epoch to fold, answers in one step a share at most

$\mathbb{E}\rho_i^N \leq \alpha_i^{n_{\min}} \leq (\delta/r_0)^{n_{\min}}$ of the queries of a template with $\alpha_i r_0 \leq \delta$. What remains attainable without change information is safety for every fold that rests on evidence collected after the change (Proposition 25).

Exact value for a stationary rule. The lower bound of (ii) is attained by a rule whose only chance to fold is the next audited outcome. A rule that, from the epoch start on, folds at the first audited outcome in F , whenever it comes, has an exact bad-fold probability: outcomes are i.i.d. within the epoch and independent of the audit times, so the audits whose outcome lies in F form a Poisson process of rate $\lambda_i Q(F)$, and the first of them precedes the epoch end with probability

$$\beta_i(F) = \frac{\lambda_i Q(F)}{\lambda_i Q(F) + \nu} \in [\alpha_i Q(F), \alpha_i].$$

For fold-at-first-success under the bad law Bernoulli(r_0) this is $\lambda_i r_0 / (\lambda_i r_0 + \nu)$, so this rule is δ -safe if and only if $\lambda_i / \nu \leq \delta / (r_0(1 - \delta))$, i.e. 0.0658 at $r_0 = 0.8$ and $\delta = 0.05$, slightly below the threshold $\delta / (r_0 - \delta) = 0.0667$ beyond which (a) excludes every rule. The two bounds of the proposition bracket the exact value, and (a) cannot be sharpened to β_i for every rule, because a rule may stop folding after an audited failure.

Numerical check. At $\lambda_i = 0.01$, $\nu = 1$, $\delta = 0.05$ and $r_0 = 0.8$, $\alpha_i = 0.0099 < \delta$, so every audit-driven rule is δ -safe there; the next-audit rule folds a bad candidate with probability $\alpha_i r_0 = 0.0079$, and fold-at-first-success with $\lambda_i r_0 / (\lambda_i r_0 + \nu) = 0.0079$ (they differ in the fifth decimal). Monte Carlo over 2×10^6 epochs per setting (Poisson audits and changes, bad law Bernoulli(r_0)) reproduces the exact value of fold-at-first-success at 49 settings with $r_0 \in \{0.5, 0.8, 0.95, 1\}$ and λ_i / ν from 10^{-3} to 100: 47 of the 49 95% intervals contain it, the largest relative deviation is 2.1%, and an event-driven simulator (2×10^5 epochs) agrees on all 10 rows it was run on. Every exact value lies between $\alpha_i r_0$ and α_i , with equality to α_i at $r_0 = 1$, and the δ -safety boundary falls where predicted: at $r_0 = 0.8$ the probability is 0.0234 at $\lambda_i / \nu = 0.03$ and 0.0500 at 0.0658. For the next-audit rule, Monte Carlo reproduces $\alpha_i r_0$ to within $3 \cdot 10^{-4}$ at five settings with α_i from 0.0099 to 0.99. For an SPRT without restarts ($r_1 = 0.95$, $r_0 = 0.5$, $\delta = 0.05$), bad epochs that start one success from the threshold contain a bad fold with frequency $\alpha_i r_0$ up to Monte Carlo

error (more for $\alpha_i \geq 0.9$, through later audits), which exceeds δ from $\alpha_i = 0.2$ on; in a stream of epochs the frequency stays below α_i at every tested $\alpha_i \in [0.02, 0.98]$ (checks/prop24_check.py in our code release).

Proposition 25 (Detector-triggered restarts). *Let the library hold at most one candidate per template, and let each candidate restart its test (wealth 1, the next ledger index) at stopping times of the system's own records, for instance at the alarms of an LR-CUSUM on its audited shadow outcomes, or be discarded then. Call the interval from a candidate's last restart (or its formation) to its decision its test segment. Then: (i) statement (i) of Proposition 23 holds; (ii) for every epoch and every history before it, the probability that the epoch contains a fold of a candidate that is bad in the epoch and whose test segment began in the epoch is at most $\sum_{j>J} \delta_j \leq \delta$, where J is the number of ledger indices used before the epoch; (iii) the segment running at the epoch start, with index J and wealth K_c , folds a candidate that is bad in the epoch before its next restart with probability at most $\min\{1, \delta_J K_c\}$.*

Proof. Restart times are stopping times and every restart starts a fresh mixture of test supermartingales at 1 with a fresh index, so each test segment is a candidate in the sense of Proposition 23. Apply (i) of that proposition to all segments, (ii) to the segments that begin in the epoch, and (iii) to the segment running at its start, with e the earlier of its next restart and the epoch end. $\square$

A restart costs coverage, never safety, because it consumes a fresh index. The carried-over risk of (iii) lasts until the detector's next alarm, so the detector bounds its duration by its detection delay when the change raises the candidate's failure rate. A change that leaves the macro's outcomes unchanged but improves live reasoning is invisible to a detector on macro outcomes; it can make a candidate useless without making it wrong. Appendix D.3.4 measures both effects on KQA-STREAM. In a simulation of discard-on-alarm with the lifetime ledger (Bernoulli likelihood-ratio wealth, $r_1 = 0.95$, bad law at $r_0 = 0.5$, $\delta = 0.05$, $\alpha_i \in \{0.5, 0.9, 0.97\}$), fresh segments fold in at most 0.0003 of bad epochs (mean bound $\sum_{j>J} \delta_j = 0.001$ –0.009), and carried-over segments in 0.009–0.014 (mean bound $\mathbb{E} \min\{1, \delta_J K_c\} = 0.015$ –0.052), as (ii) and (iii) require.

What the restart-only certificate attains. For the j -th test of a template (restarts included) the threshold is $\ln(1/\delta_j) = \ln(1/\delta) + 2 \ln j + \ln(\pi^2/6)$. With i.i.d. audited utilities of law P and the seven bets $\mathcal{B}$ with weights $1/7$, the best bet in $\mathcal{B}$ and Wald’s identity (factors in $[0.1, 1.9]$ because the code clips Z to $[-1, 1]$) give an expected label count of at most

$$\frac{\ln \frac{1}{\delta} + 2 \ln j + \ln \frac{\pi^2}{6} + \ln 7 + \ln 1.9}{g_{\mathcal{B}}(P)}$$

before the fold threshold is reached (ignoring the futility test), where $g_{\mathcal{B}}(P) = \max_{b \in \mathcal{B}} \mathbb{E}_P \ln(1 + bZ)$ is the growth of the best bet, and $g_{\mathcal{B}} \leq \text{KL}_{\text{inf}}(P)$, with equality only if $c^* \in \mathcal{B}$. The restart-only certificate therefore does not attain the limit of Theorem 3 or Corollary 20 as $\delta \rightarrow 0$. It is a finite-candidate, finite-grid version of the idealized certificate, whose extra cost is the additive $2 \ln j + \ln(\pi^2/6) + \ln 7 + \ln 1.9$ and the factor $\text{KL}_{\text{inf}}/g_{\mathcal{B}}$ (1.03–1.07 on the three paired laws we checked).

D Experimental details

Compute and software. All LLM runs used one NVIDIA RTX 4080 SUPER GPU (16 GB). Models were served with vLLM 0.30 (Kwon et al., 2023) (FlashAttention backend) and PyTorch 2.13 under Ubuntu 22.04. The 8B models were served with FP8 weights: Qwen3-8B-FP8 natively (Yang et al., 2025), and Granite-3.3-8B-Instruct (IBM Granite Team, 2025), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) and OLMo-3-7B-Instruct (Team OLMo, 2025) with on-the-fly FP8 quantization. Qwen3-4B ran in BF16. The natural-edit timeline used 4.75 GPU-hours and the code domain 17.2 GPU-hours of wall time. Mistral has no system role, so its system message is prepended to the first user turn. Granite-3.3’s chat template embeds the current date, so its prompts change with the calendar day; all Granite prompts were rendered with the date pinned to 2026-09-25, and prompt hashes confirm that this reproduces the main-run prompts exactly. Qwen3 “thinking” was disabled. Main-table decoding used $T=0.7$, top- p 0.8, top- k 20, with a deterministic per-request seed. Every LLM request carries a content hash. Results are appended to checkpoint shards and skipped on restart, so all runs are resumable. Tables and figures are generated by script from the raw logs.

Mechanism computations. Run lengths of integer-score rules were computed by solving the Markov chain of the reflected walk. The gap of SPRT certification uses Lemma 10 with the exact sample-size distribution of the one-sided SPRT, obtained by dynamic programming on the lattice of log-likelihood-ratio values. Sums over templates run to 10^7 with rigorous tail brackets. The collapse grid crossed $(r_0, r_1) \in \{(0.85, 0.95), (0.6, 0.9), (0.3, 0.7)\}$ plus random pairs, verifier flip rates $\{0, 0.05, 0.1, 0.2\}$, $\delta \in [10^{-32}, 0.03]$, audit rates $p \in [0.03, 1]$ (tabulated settings: $p = 1$, $\Theta \in \{10^{-5}, \dots, 10^{-1}\}$), 10 drift rates for the random settings and $s \in \{1.2, 1.5, 2, 3\}$. Plotted against $x = \nu/p$ alone, the same points spread over two orders of magnitude at fixed x because of the $\ln(1/\delta)/\text{KL}$ factor. The literal-stream check simulated Poisson arrivals, audits with $p \in \{0.2, 0.3, 0.5, 1\}$, Poisson changes and verifier flips, for 500– 10^4 templates, and matched the exact formula within 0.51%.

Natural template streams. KQA Pro templates are the programs of the train and validation sets (106,173 questions) with entity names, strings, numbers and dates abstracted. Function, relation, attribute and concept names are kept, giving 45,544 templates. MuSiQue templates abstract each single-hop sub-question of the answerable training set (19,938 questions; 10,686 templates). We abstract the relation phrase after “ $\gg$ ” when it is present, and otherwise the question with bridge entities replaced by placeholders. Exponents were estimated from rank-frequency data (Sibuya-type estimator on counts ≤ 10), from the Clauset–Shalizi–Newman tail fit, and from Heaps’ law. The Sibuya and Heaps estimators give $s \in [1.15, 1.26]$ for the fine-grained templates. The CSN tail fit gives $\hat{s} \approx 0.85$ – 0.88 with a poor goodness of fit on MuSiQue ($p < 0.01$), so we do not use it. Coverage curves average 100 random stream orders.

KQA-STREAM construction. We index the KQA Pro knowledge base: 16,960 entities, 794 concepts, 363 relations (726 relation–direction types) and 415,702 directed edges. Our path executor agrees with the official KoPL engine on 600 sampled one- and two-hop programs (2,187 steps). A template is an anchor concept plus a relation path of 2–4 hops. It is kept if at least 60 anchors have 1–3 answers of a consistent concept, the path does not immediately backtrack, and no single answer covers more than half of the anchors. From 2,250

qualifying distinct paths we selected 300 templates (120/120/60 with 2/3/4 hops), covering 17 anchor concepts, 51 answer concepts and 114 relations, and sampled 60 anchors each (18,000 instances). A random pattern verbalizes each instance question; 296 of the 300 templates have three LLM-written patterns with a single entity slot (the rest have fewer valid patterns). An edit event replaces the object of a relation for a fraction $f \in \{0.25, 0.6\}$ of the relevant subjects with a random entity of the same concept, keeping inverse edges consistent. Each of three sequences has 24 events. An event changes the gold answer of 223 ($f=0.25$) or 386 ($f=0.6$) instances on average and touches 3.9% of templates, which become 32% or 59% stale respectively. Templates that share an edited relation go stale together. The per-template change rate grows with derivation length (events that hit a 2/3/4-hop template: 2.7/4.2/5.5%; Spearman 0.34), which makes staleness itself a multi-step-reasoning phenomenon.

LLM paths. The *live* agent answers hop by hop. At each step its prompt contains the question, the linked anchor, the path so far, the current frontier (up to 5 entity names) and a numbered list of the (relation, direction) pairs available at the frontier (up to 60, sorted by frequency). It outputs JSON that either takes one hop or answers, with at most 5 hops, and may filter the final frontier by one of its concepts. The answer is the resulting entity set, and it is correct if it equals the gold set of the current knowledge-base version. For Qwen3-8B a live query uses 3.24 calls with 1,486 prompt and 62 completion tokens on average; on the unedited knowledge base its live accuracy is 0.625, 0.305 and 0.190 at 2, 3 and 4 hops (0.410 over all 18,000 instances). *Routing*, the one LLM step of a macro, retrieves the top-5 templates by BM25 over their verbalized patterns and asks the LLM for a template id or NONE. It runs against the full, fixed template catalog, so its outcome does not depend on the policy. Its accuracy is 0.939 for Qwen3-8B, and the correct template is among the top 5 for all but 40 of the 18,000 instances. The *direct* baselines use one call: a parametric answer without the knowledge base, or whole-path generation from the question and the BM25 top-60 relation directions, executed on the current version. For each call we log prompt and completion tokens and latency. After an edit, live traces are re-run only for instances whose earlier trace or gold path

touches an edited edge; all other traces are reused unchanged, which is exact because the agent’s inputs are unchanged. A brute-force re-run of all instances at versions 0, 8, 16, 24 of every sequence matched in all 216,000 comparisons.

Baselines and statistics. The ExpeL-style counter starts at importance 2, moves by ± 1 per audited success or failure and retires the macro at 0. The ACE-style counter keeps helpful and harmful tallies from audited outcomes and retires the macro when it would be flagged (harmful $\geq$ helpful, harmful > 0). *Fold once* folds a verified candidate at formation and never retires it. A semantic cache keyed on question embeddings reduces to this policy behind our router. vCache’s per-request bound assumes i.i.d. prompts and has no retirement; Appendix D.2.4 evaluates a re-implementation, which avoids staleness mainly by rarely answering from cache. No policy was tuned on the evaluation streams. The validation-gated Hebbian rule uses its published defaults ($\eta = 0.1$, $\lambda = 10^{-3}$, $\tau = 0.85$, $c_0 = 0.5$, fusion bonus 0.05). Our certificates use $\delta = \beta = 0.05$ and a likelihood-ratio CUSUM with threshold $h = 7$ and post-change failure rate $\max(1 - \hat{r}^*, \hat{q}_0 + 0.15)$; its in-control rate $\hat{q}_0$ is estimated from the candidate’s shadow labels. The betting certificate mixes seven fixed bets $\lambda \in \{0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 0.9\}$ with equal weights. Any fixed mixture is valid; the grid mixture of Appendix C is needed only for the rate result. All defaults were fixed on synthetic tables before any real-LLM run. *Risk ledger.* The library holds at most one candidate or macro per template. The *lifetime ledger* tests the j -th test of a template, counted over the whole stream, at $\delta_j = 6\delta/(\pi^2 j^2)$. The restart-only certificate adds *restarts*: the replay notifies it of every edit that changes the gold answer of an instance routed to a template, and a candidate under test then discards its wealth and shadow statistics and restarts with the next ledger index (Appendix C.2). By Proposition 23(iv) this satisfies Definition 1 when the certificate’s null changes only at notices (Appendix D.5.1). The recommended deployment (“ours” in the main text) also drops the notified template’s folded macro or candidate and re-forms it under the ledger (invalidate + LR, Appendix D.4.2). Without restarts, the engine never tells a policy that an edit has occurred, and a candidate keeps its betting wealth (or SPRT statistic) across edits. The ledger then still bounds by δ the probability

that a template ever folds a candidate that was at or below break-even at all its evaluations, and, given the history at the start of any epoch, the probability that a candidate formed in that epoch folds while bad. A candidate carried across an edit with wealth w is covered only by the conditional bound $\delta_j w$, so this variant does not satisfy Definition 1 in general. The *per-candidate* variants test each new candidate at a fresh level $\delta = 0.05$. This gives per-candidate conditional validity only, because a template’s total fold risk grows with its number of candidates. Compiled plans were run with the restart-only certificate (restarts), with the ledger without restarts, and per candidate. The SpF columns count every fold event of a candidate whose reliability at the fold’s knowledge-base version is below the break-even r^* of its template at that version ($L = 20\bar{m}$), including candidates carried across edits (Appendix D.3.3). *Same information and tuning.* Every policy sees the same queries, edits, audit coins and verifier labels; all certificates share δ , the bet grid and h , and no threshold was tuned on the evaluation streams. Only our certificates receive edit notices: restart only uses them to restart candidates under test, ours also to invalidate, and Table 2 reports ours with every notice missed; Appendix D.4.2 gives the notices to baselines that invalidate on them, and Appendices D.4.1 and D.4.3 report the answer-only safety replays and the decoding-variance subset. On Qwen3-8B the ledger without restarts lowers accuracy by 0.35 and 0.41 points relative to per-candidate testing and raises J_{20} by about 1% (betting: $0.646 \rightarrow 0.643$, J_{20} $25.6 \rightarrow 25.9$; compiled + betting: $0.677 \rightarrow 0.673$, J_{20} $23.4 \rightarrow 23.7$). On the other four models the change in accuracy ranges from -0.59 to $+0.14$ points. Restarts cost another 0.1 – 0.4 points on every model (Tables 6 and 13). Paired differences use a 95% bootstrap clustered by template (the 300 templates are resampled) over the 10 seeds $\times$ 3 edit sequences. Each instance–version is decoded once per model, so these intervals do not include LLM sampling variance.

Replay setting. The replays fix the following setting. Audits reveal the gold answer of the current knowledge-base version at no LLM cost. Templates are given by the benchmark, and each has up to three verbalization patterns; routing errors enter every metric as measured losses (a misrouted macro answer counts as wrong) and are not mod-

eled by the theory. A macro executes a verified relation path exactly, which is the main source of its accuracy gain over live reasoning; text notes and distilled models would need a costlier, noisier verifier. Each instance–version is decoded once per model, so intervals resample templates, streams and edit sequences, not decoding randomness (measured separately in Appendix D.4.3), and calls and latencies are reconstructed from the recorded requests. The main edits are synthetic but structured: correlated across templates that share a relation, and more frequent for longer derivations; Appendix D.4.4 adds a natural timeline and Appendix D.6 a code domain with an exact verifier. Edit notices list the templates whose gold answers change; changes of live reasoning are not announced (Appendix D.5.1). Two comparisons are deliberately narrow. Compiled plans re-execute a stored path on the current facts; when such re-execution is cheap it can dominate caching the result, and Table 3 reports this baseline. The vCache-style cache stores unverified live answers, so on this benchmark it inherits live accuracy (Appendix D.2.4); a verified exact-repeat cache is evaluated in Appendix D.4.2. The fundamental limit concerns candidates informed by the outcomes of their own template; transfer across correlated templates, or a macro that a change does not affect, lies outside its model.

Completed coverage and validation. The final analysis includes all five models, three edit sequences and ten seeds, with 57,750 distinct configuration–seed checkpoints after deduplicating identical effective settings. Every checkpoint and its trace-derived summary passed completion and version checks. Recomputing the 80 primary model–policy cells reproduces the exported mean accuracy, calls and J_{20} . Paired differences use per-candidate betting + LR-CUSUM as the reference, except the price of restarts (Table 13), which is paired against the same ledger without restarts.

All policies and all models. Table 5 adds to Table 3 one-call path generation, the validation-gated Hebbian rule with time decay, and compiled plans certified by the plug-in SPRT. Hebbian decay almost never folds, because its time-to-live starves all but the most popular templates (Appendix A). Table 6 gives accuracy, calls and J_{20} for all five models.

Where the accuracy comes from. On Qwen3-8B (Table 3), *reuse* without certification (fold once)

Policy (Qwen3-8B)	Acc	Calls	Stale	SpF	J_{20}
direct (path generation)	.186	1.00	0	0	54.1
Hebbian + decay	.444	3.80	0.4	0.3	40.1
compiled plan + SPRT	.650	2.51	10	0.3	25.3
loss-tuned	.631	1.62	659	118	25.7
SPRT + symmetric	.604	2.84	20	0.7	28.7
betting, no restarts	.643	2.55	72	0.1	25.9
betting, per candidate	.646	2.50	76	0.2	25.6
plan, ledger, no restarts	.673	2.42	14	0.0	23.7
plan, per candidate	.677	2.38	12	0.03	23.4
invalidate on notice (no LR)	.651	2.42	0	0.0	25.2

Table 5: Further Qwen3-8B rows for Table 3 (same setting and columns; SpF counts the bad folds of the main text). Loss-tuned: fold and retire thresholds set from a renewal-reward approximation of the loss. *Per candidate*: a fresh level δ for each candidate. The last row drops and re-forms every macro an edit announces, with perfect notices (Appendix D.4.2). Bold: ours.

adds 15.5 points over live reasoning for macros and 26.1 for compiled plans. *Certification* with retirement and restarts adds 10.7 and 2.9 more and cuts stale answers $20\times$ and $6\times$; over the counters, which reuse and retire too, it adds 2.3–2.9. Reuse helps this much because a macro executes a verified relation path symbolically, whereas the live agent re-derives it with a noisy LLM at 38% accuracy. For text or distilled macros this advantage need not hold, so we regard the accuracy gain as specific to this benchmark and the staleness result as the transferable one. On OLMo-3-7B the certificate uses more calls than live reasoning ($2.39 \rightarrow 2.68$; Table 6). Stale counts also depend on how much a policy folds: SPRT + symmetric serves 20 stale answers per 10^4 queries against restart only’s 66 on Qwen3-8B, but answers 43% of queries in one step against 51%. This is why the main text compares retirement rules on a matched population (Appendix D.3.2).

D.1 Natural template streams

We extracted derivation templates from 106k KQA Pro programs (Cao et al., 2022) and 19.9k MuSiQue decompositions (Trivedi et al., 2022), with entities and values abstracted (details above). Both are heavy-tailed: the Sibuya-type estimator gives $\hat{s} = 1.26$ and Heaps’ law $\hat{s} = 1.15$ on both corpora. The static branch of Theorem 14 therefore predicts that coverage improves only as $T^{-(0.13-0.21)}$, as observed (Figure 4): slopes at $n_0 = 1$ are -0.16 (KQA Pro) and -0.18 (MuSiQue), steepening to -0.19 and -0.21 late in the stream. The Good–Turing estimator of

the rare mass (Good, 1953; Orlitsky et al., 2016), $\hat{M}(T) = \sum_{k \leq n_0} kK_k(T)/T$ with K_k the number of templates seen k times, predicts the gap online with a median relative error of 0.4–3.4% (2.1 – 3.4% at $n_0 = 1$), without knowing s . Raising the certification sample size n_0 slows contraction by $C_{s,n_0}/C_{s,1} \approx n_0^{1-1/s}/\Gamma(2 - \frac{1}{s})$. These streams test the demand side of the theory without an LLM in the loop.

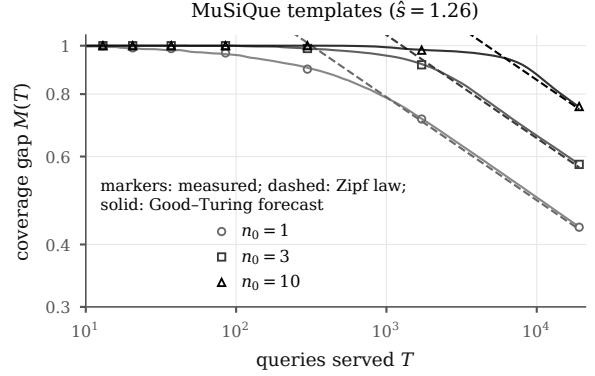


Figure 4: MuSiQue templates: measured coverage gap (markers; mean of 100 stream orders), fitted Zipf law (dashed) and online Good–Turing forecast (solid) for certification sample sizes n_0 .

D.2 Learning rates, counters, noisy verifiers and a semantic cache

These experiments replay the recorded KQA-STREAM outcome tables of the five models; no LLM is called. They use the main setting (200k queries, Zipf $s = 1.2$, Poisson edits at rate $28/T$ so that nearly all 24 edit events of a sequence occur, uniform audits $p = 0.1$, $L = 20$) with three edit sequences and ten seeds, i.e. 30 runs per cell. Intervals are 95% bootstraps that resample the 300 templates (1000 resamples), pooled over sequences and seeds, and paired differences use the same template bootstrap as the main text. Here *ours* is betting + LR-CUSUM at the per-candidate level; it reproduces the per-candidate row of Table 3 exactly (Qwen3-8B: accuracy 0.646, J_{20} 25.58, 76 stale answers per 10^4).

Definitions. A false retirement (FR) is the retirement of a macro whose true reliability is still $\geq r^*$; we count them per 10^4 folded uses of such healthy macros. The detection delay of a stale episode is the number of folded uses from its onset, the version at which a macro folded while healthy drops below r^* , to its retirement. Episodes still open at the end of a stream contribute their uses so far, so

Model	live			ours (recommended)			macros + betting restart only / no restarts / per-cand.				compiled plan + betting restarts / no restarts / per-cand.			
	Acc	Calls	J_{20}	Acc	Calls	J_{20}	Acc	Calls	Stale	J_{20}	Acc	Calls	Stale	J_{20}
Qwen3-8B	.379	3.26	43.8	.648	2.56	25.5	.642/.643/.646	2.57	66/72/76	26.0	.668/.673/.677	2.46	16/14/12	24.1
Qwen3-4B	.226	4.48	74.0	.556	3.10	43.0	.551/.554/.553	3.08	82/85/91	43.5	.591/.596/.596	2.90	10/12/12	39.7
Granite-3.3-8B	.152	4.47	80.3	.439	3.57	53.8	.438/.441/.447	3.52	79/82/84	53.8	.469/.472/.477	3.39	13/15/16	50.9
Mistral-7B	.097	4.96	94.6	.241	5.00	80.5	.245/.247/.250	4.94	38/42/43	80.0	.257/.259/.258	4.88	9/9/10	78.7
OLMo-3-7B	.076	2.39	46.4	.280	2.68	36.7	.276/.280/.283	2.68	51/51/51	36.9	.294/.297/.299	2.64	11/12/12	36.0

Table 6: All five models in the main setting (30 streams of 200k queries each). Stale: stale wrong answers per 10^4 queries. Ours: invalidate on notice + LR-CUSUM, perfect notices (it serves no stale answers). Calls and J_{20} of the other groups are for the first variant listed: the restart-only certificate with restarts (restart only), for macros and for compiled plans (compiled plan, Appendix D.4.2). Entries are descriptive means; calls are reconstructed from recorded requests. Bold: ours.

delays are lower bounds; Appendix D.3.1 gives censoring-aware estimates. At a *matched FR*, each rule’s delay and stale answers are interpolated linearly in log FR along its threshold grid, at the FR^* of our LR-CUSUM at its default $h = 7$. Matching uses the true reliabilities. This makes the comparison of detectors fair, but it is not a deployable tuning procedure.

D.2.1 Learning-rate sweep

Rules. A fixed-ratio rule keeps our betting certificate and retires with the clipped score $+1$ per audited failure and $-1/k$ per audited success, retiring at $S > h$, for $k \in \{0.1, 0.25, 0.5, 0.75, 1, 2, 3, 5, 7, 10, 15, 25\}$ and 15 thresholds $h \in [0.5, 80]$. Ours is the same except that the score is the log-likelihood ratio of the macro’s own estimated $\hat{q}_0$ against $\max(1 - \hat{r}^*, \hat{q}_0 + 0.15)$. The failure rates q_0 and q_1 of Table 7 are the true failure rates ($1 - \text{reliability}$, including routing errors) of the stale episodes of our policy, q_0 at the fold version and q_1 at the on-set version. The prediction $1/\rho(\bar{q}_0, \bar{q}_1)$ uses their means.

Results. Staleness on these streams is nearly total, not partial: 86–97% of stale episodes have $q_1 > 0.5$ (Table 7). The predicted weights $1/\rho$ are therefore 0.67–2.14, not 7. On four of five models the delay at matched FR is smallest below $k = 1$ ($k = 0.1$ –0.75), consistent with the ordering of the prediction, and it grows with k above 1; $k = 7$ is 1.3–2.5 $\times$ slower than the best tested ratio on these models (Figure 5). Below 1 the curves are flat, so the optimum there is not sharply identified. Qwen3-8B is the exception: its curve is flat for $k \geq 5$ and worse at $k \leq 2$, and the bootstrap spreads its optimum over $k = 25, 5$ and 10. The larger effect is not the tuning of k . The per-

macro LR-CUSUM detects 1.3–5.9 $\times$ faster than each model’s best tested fixed ratio (paired intervals exclude 1 on Qwen3-8B, Qwen3-4B and OLMo-3-7B; Appendix D.3.6) and serves 1.7–2.5 $\times$ fewer stale answers than the fixed ratio $k \geq 1$ with the fewest. Every fixed ratio at $k = 7$ is significantly slower than ours, and at $k = 1$ this holds on three of five models (paired intervals exclude 0). Healthy failure rates are heterogeneous: their medians (0–0.21) lie far below their means. We interpret this, without a direct test, as the reason why one global ratio cannot be likelihood-ratio tuned for all macros at once, whereas the per-macro rule adapts $\hat{q}_0$ to each macro. The first-order delay curve follows the rising trend on Granite, Mistral and OLMo but not on Qwen3-8B. At $k = 1$ it underestimates the measured delays by 2.3–17 $\times$, and at larger k it leaves the logarithmic regime on three models. With the episode-median in place of the mean (q_0, q_1) the predicted weights are 0.51–3.01. With only 17–81 stale episodes per model the intervals are wide, and only the large differences are resolved.

D.2.2 Counters in the main setting

ExpeL- and ACE-style counters serve 294–967 stale answers per 10^4 queries (7–13 $\times$ ours), fold 111–595 spurious macros per run (ours: at most 1.7), and have 10–35 false retirements per 10^4 (ours: 1.8–3.4) with naive mean delays of 307–478 uses (ours: 54–84; lower bounds, see Appendix D.3.1). Their J_{20} , however, is not significantly different from ours on any model, and their accuracy is significantly lower only on Qwen3-8B (Table 8). The reason is coverage, as the price of safety of Section 4 suggests: the counters skip certification and answer 28–76% of queries in one step, against 21–53% for the restart-only certificate and 20–53% for ours (recommended), so they

Model	episodes	$\bar{q}_0$	$\bar{q}_1$	$q_1 > \frac{1}{2}$	predicted $1/\rho$ [95% CI]	empirical optimum		delay at matched FR (uses)			stale per 10^4		
						k	$P(\arg \min)$	$k=1$	$k=7$	best k	ours	best k	ours
Qwen3-8B	81	.068	.746	.86	1.84 [1.36, 2.47]	25	.24	853	482	468	81	185	76
Qwen3-4B	65	.044	.746	.88	2.14 [1.52, 3.32]	0.75	.34	330	398	317	54	231	91
Granite-3.3-8B	32	.161	.863	.97	0.93 [0.57, 1.50]	0.25	.40	169	317	126	63	173	84
Mistral-7B	17	.237	.896	.94	0.67 [0.39, 1.11]	0.1	.33	141	252	106	84	78	43
OLMo-3-7B	17	.135	.821	.94	1.14 [0.59, 2.83]	0.25	.23	131	227	115	57	88	51

Table 7: Learning-rate sweep on KQA-STREAM. Episodes: stale episodes over 30 runs. $q_1 > \frac{1}{2}$: share of stale episodes with $q_1 > 0.5$. $P(\arg \min)$: bootstrap frequency of the empirical optimum. Best k : the fixed ratio with the lowest delay at matched FR over all tested k , and with the fewest stale answers over $k \geq 1$. Ours: per-macro LR-CUSUM (per-candidate certificate). Delays are naive means over censored episodes. Bold: ours.

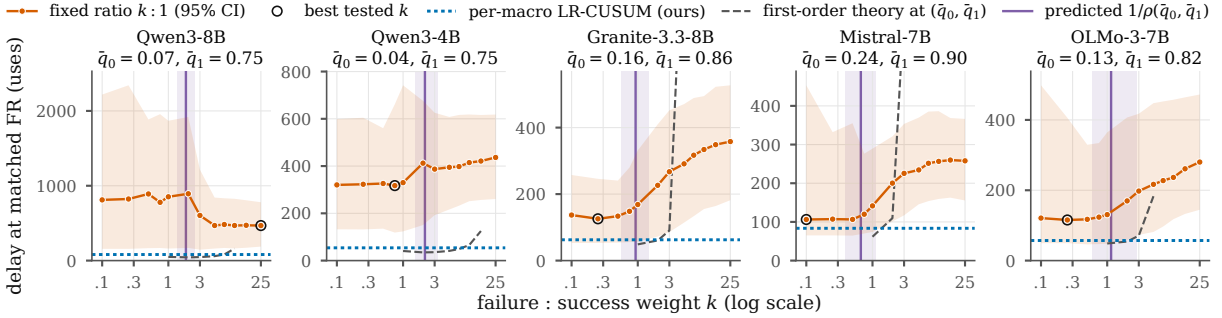


Figure 5: **Per-memory likelihood ratios beat every tested global counter ratio.** Mean uses to detect staleness for a clipped counter that weighs a failure k times a success (orange; circle: best tested k) and for the per-memory LR-CUSUM (blue dotted), at the false-retirement rate of the latter (1.9–3.4 per 10^4 healthy uses). Purple: the weight $1/\rho$ that Corollary 2 predicts from the mean failure rates ($\bar{q}_0, \bar{q}_1$); dashed: its first-order delay, an *optimum*-level rate that the *measured* delays exceed (by 2.3–17 $\times$ at $k = 1$; see the results below). On four of five models the delay is flat or lowest up to the predicted weight and grows beyond it; the per-memory gain is significant on Qwen3-8B, Qwen3-4B and OLMo-3-7B. 30 streams per model, per-candidate certificate; bands: 95% template bootstrap; naive means, which favour the slower rules (Appendix D.2.1).

make fewer LLM calls (e.g. 1.75 and 1.86 against 2.57 on Qwen3-8B, Table 3), and in J_{20} this saving roughly offsets their stale and spurious answers.

D.2.3 A noise-aware certificate

Setup. Noisy verifiers are the next step named in the Limitations; this experiment measures where the current certificate stands. Each audited judgement (of the live answer, the macro and any other evaluated answer) is flipped independently with probability ε . The noisy live verdict also decides which traces count as verified when candidates are formed. The noise-aware variant bets on the paired utility of the observed channel, $X = (1 - 2\varepsilon)(m - 1) + L(\tilde{Y}^M - \tilde{Y}^L)$, and maps the CUSUM’s alternative into the observed channel, $q_1 = \max(\varepsilon + (1 - 2\varepsilon)(1 - r^*), q_0 + 0.15(1 - 2\varepsilon))$, with live accuracy corrected as $a = (a_{\text{obs}} - \varepsilon)/(1 - 2\varepsilon)$. It uses either the true ε or an estimate from a simulated clean calibration subset of 200 items, drawn per sequence and seed (mean estimates 0.054, 0.103 and 0.201). It corrects the certificate and the CUSUM, not the

formation of candidates. The labels-to-fold slowdown is the mean over templates of the number of audited shadow labels a candidate needs before it folds, relative to the uncorrected certificate at $\varepsilon = 0$.

Results (negative). Certification slows faster than $(1 - 2\varepsilon)^{-2}$ (Table 9): the labels-to-fold slowdown is 1.88, 2.56 and 3.67 at $\varepsilon = 0.05, 0.1$ and 0.2, against 1.23, 1.56 and 2.78 predicted, and the slowdown in key-queries is larger still (2.15, 3.20, 5.05). The prediction is for a single-label test; our certificate bets on the difference of two noisy labels. The noise-aware variant, with known or estimated ε , is indistinguishable from the uncorrected certificate on accuracy, stale answers and spurious folds, and nearly every per-model accuracy interval contains 0. Noise does not inflate spurious folds (0.71 per run clean, 0.12–0.27 noisy), because far fewer candidates fold at all. What dominates is the loss of accuracy for every variant (0.436 to 0.279 at $\varepsilon = 0.2$), driven by slower certification

Model	ExpeL-style counter		ACE-style counter	
	ΔAcc	ΔJ_{20}	ΔAcc	ΔJ_{20}
Qwen3-8B	−.033 [−.061, −.004]	+1.38 [−0.56, +3.20]	−.028 [−.055, −.001]	+1.15 [−0.72, +2.95]
Qwen3-4B	+ .006 [−.020, +.035]	−1.18 [−3.93, +1.34]	+ .002 [−.026, +.032]	−0.68 [−3.61, +1.99]
Granite-3.3-8B	−.014 [−.034, +.009]	+0.87 [−1.27, +2.79]	−.014 [−.035, +.009]	+1.05 [−1.14, +3.01]
Mistral-7B	−.008 [−.023, +.006]	+0.30 [−1.17, +1.79]	−.008 [−.022, +.006]	+0.49 [−0.94, +1.93]
OLMo-3-7B	+ .007 [−.019, +.037]	−0.47 [−2.13, +0.90]	−.000 [−.028, +.032]	−0.09 [−1.82, +1.36]

Table 8: Main setting: paired differences of the counters against ours (betting + LR-CUSUM, per-candidate level), with 95% template-bootstrap intervals; positive ΔJ_{20} is worse. Only the counters’ accuracy on Qwen3-8B differs significantly.

ε	certificate	Acc	Stale	SpF	slowdown	$(1-2\varepsilon)^{-2}$
0	uncorrected	.436	69	0.71	1.00	1.00
.05	uncorrected	.370	63	0.21	1.88	1.23
	known ε	.370	61	0.21	2.00	
	estimated ε	.369	61	0.21	1.99	
.1	uncorrected	.326	61	0.17	2.56	1.56
	known ε	.328	59	0.12	2.82	
	estimated ε	.327	59	0.14	2.70	
.2	uncorrected	.279	53	0.27	3.67	2.78
	known ε	.286	50	0.21	3.36	
	estimated ε	.286	49	0.22	3.21	

Table 9: Noisy verifier, per-candidate betting, means over the five models (30 streams each; at $\varepsilon = 0$ the mean of the per-candidate column of Table 6). Stale: per 10^4 queries. SpF: spurious folds per run. Slowdown: audited labels a candidate needs before it folds, relative to $\varepsilon = 0$. Bold: ours (all rows).

and by macros formed from falsely verified live traces, neither of which the correction addresses. Counter rules, which fold nearly everything, keep their accuracy under noise (Qwen3-8B, 15 streams: .592 $\rightarrow$.575 and .597 $\rightarrow$.580 at $\varepsilon = 0.2$) but fold more spurious macros (ExpeL 271 $\rightarrow$ 385, ACE 377 $\rightarrow$ 601 per stream). A certificate built for the noisy *paired* channel is left to future work.

D.2.4 A vCache-style semantic cache

Re-implementation. We re-implemented vCache from Algorithms 1–2 of its paper (Schroeder et al., 2026); the following details are our design choices. A query is looked up against the nearest cached question by cosine similarity (ties go to the newest entry). Each entry fits a sigmoid of similarity by ridge-penalized logistic maximum likelihood ($\lambda = 10^{-3}$), with a delta-method upper confidence bound on its threshold. The exploration probability is the minimum of the paper’s Eq. 11 over a grid of 49 values of its error parameter, and 1 while an entry has no observations. An exploration answers live,

evaluates the cached answer in shadow at no cost, and adds the query to the cache only if the cached answer was judged wrong; an exploitation returns the cached answer at no LLM call, and embedding compute is not charged. Labels are vCache’s own (*agree*: cached answer equals the live answer) or gold correctness on audited queries (*audit*). There is no retirement. The *plug-in* variant uses the point estimate without the confidence band, which is optimistic. The vCache paper used GTE-large, E5-large-v2 and OpenAI embeddings. None was available offline, so we use all-MiniLM-L6-v2 (384 dimensions), with TF-IDF as a sensitivity check; the median nearest-neighbour similarity over the 17,927 unique questions is 0.81 (MiniLM) and 0.66 (TF-IDF). A faithful replication would need a 1024-dimensional embedding and the authors’ code to confirm the confidence band. A cache hit on a different question counts as stale if the cached answer would have been correct for that query when the entry was formed.

Model	vCache, $\delta_c = 0.02$				plug-in, $\delta_c = 0.02$			
	live	Acc	hits	ΔJ_{20}	hits	Acc	Stale	
Qwen3-8B	.379	.124	.377	+17.9	.847	.309	536	
Qwen3-4B	.226	.119	.225	+30.3	.848	.187	241	
Granite-3.3-8B	.152	.110	.151	+26.9	.841	.126	203	
Mistral-7B	.097	.111	.097	+14.7	.847	.082	115	
OLMo-3-7B	.076	.119	.076	+9.5	.835	.063	100	

Table 10: vCache-style cache (MiniLM, *agree* labels). Hits: share of queries served from cache. ΔJ_{20} : against ours (per-candidate betting); all intervals exclude 0. Stale: per 10^4 queries. Our accuracy is .646, .553, .447, .250 and .283.

Results. With its confidence band, vCache serves only 6–26% of queries from cache for $\delta_c = 0.01$ –0.05, and 3% with audited labels. Its accuracy equals live accuracy, and its J_{20} is 9.2–30.7 worse than ours on every model (Table 10). TF-IDF embeddings give similar results. The reason is that this

cache stores earlier live answers without verifying them, so here it inherits live accuracy (0.08–0.38), whereas a certified macro re-executes a verified derivation (our accuracy 0.25–0.65). A cache that stores only verified answers was tested for exact repeats (with notice invalidation; Appendix D.4.2): it serves almost no stale answers but stays at live accuracy, because it answers only 13% of queries in one step; a verified *semantic* cache remains untested. Removing the confidence band raises the hit rate to about 85%, but accuracy falls below live and stale answers rise to 100–536 per 10^4 . vCache’s δ_c bounds disagreement with the LLM, not error against gold: 62–92% of its hits are wrong against gold, mirroring live accuracy. An exact-question answer cache beats vCache on J_{20} but is still 7.7–27.0 worse than ours. Cache hits are charged no LLM call, which favours the caches.

D.3 Censoring, per-epoch safety and per-memory weights

These replays reuse the recorded outcome tables of the five models in the main setting (30 streams per model and policy); no LLM is called. Intervals are 95% template-clustered bootstraps (1000 resamples); pooled rows resample templates jointly across models.

D.3.1 Censoring-aware detection delays

A stale episode starts when a macro that was above break-even at its fold drops below r^* after an edit; its onset is the first query of that version. Retirement is the event, and the end of the 200k-query stream censors the episode. The delay is the number of folded uses from onset to retirement, as in Appendix D.2. We estimate the survival function by Kaplan–Meier. The restricted mean delay within a window w is $\mathbb{E} \min(T, w)$, and “unretired at stream end” is the censored share. This closed-loop analysis covers our LR-CUSUM retirement with the per-candidate certificate and with the SPRT; each rule has its own population of stale episodes. Appendix D.3.2 compares the rules on one matched population, including the restart-only certificate.

On the audited-use clock the restricted means within 10^3 audited uses are 7.6 (betting), 10.4 (SPRT), 299.9 (ExpeL-style) and 257.6 (ACE-style), with medians of 5, 5, 37 and 21 audited uses. The censored episodes of the LR-CUSUM are onsets near the end of a stream (at most 192 folded uses of follow-up), whereas the counters’

Rule (pooled)	episodes	cens.	unretired at end	RM delay, 10^4	median
LR (per-cand. betting)	212	13	6.1 [2.5, 10.8]%	76 [55, 107]	45 [40, 55]
LR (SPRT)	222	14	6.3 [2.4, 10.6]%	103 [64, 176]	50 [44, 64]
ExpeL-style	1,954	1,241	63.5 [57.5, 69.8]%	3,014 [2,204, 3,895]	375 [216, 682]
ACE-style	1,865	999	53.6 [47.6, 59.9]%	2,569 [1,882, 3,332]	211 [122, 373]

Table 11: Censoring-aware detection, pooled over five models (150 streams per rule). RM delay: restricted mean delay within 10^4 folded uses; median: Kaplan–Meier median (folded uses). Bold: ours.

censored episodes include some with 39,467 uses. Per model the counters leave 28–69% of their own stale episodes unretired at stream end (Table 2 gives the matched population). The naive mean delay of Figure 6 counts an open episode with its uses so far. It is therefore a lower bound: 363 and 366 uses for the counters, 7–8 $\times$ below their restricted means.

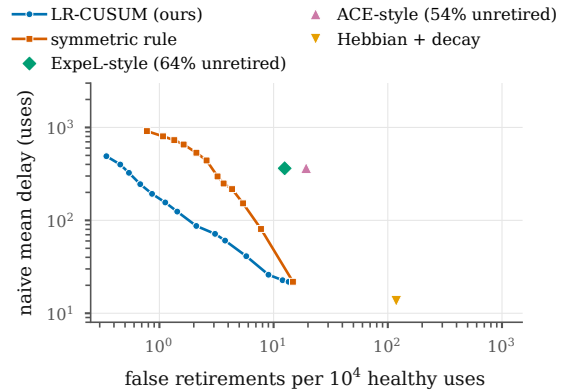


Figure 6: Detection frontier on real streams (five models, 30 streams each; SPRT certification): naive mean delay after an edit against false retirements of healthy macros, as frontiers over each rule’s thresholds. The naive mean counts an episode that is still open at the end of the stream with its uses so far, so it is a lower bound; the counters’ censoring-aware delays are 7–8 $\times$ longer (Table 11). Legend: share of each counter’s stale episodes not retired by stream end.

D.3.2 Detection on a matched population

The comparison above uses different populations: the counters fold every candidate, so their 1,954 and 1,865 stale episodes include many macros that a certificate would never fold, whereas the LR-CUSUM rows contain 212. Here all rules see the same macros and the same audits. Every macro that a certified policy folded (the restart-only certificate, or per-candidate betting) is replayed *open loop* from its fold to the end of the stream, on its own audited label stream (the same audit coins; labels of its build instance withheld), under each retirement rule started at the fold: its own LR-CUSUM (reproducing the closed-loop retirement

of all 7,385 and 9,448 folded macros exactly), the ExpeL-style rule (importance 2 at the fold) and the ACE-style rule. Counters therefore bank no credit from the shadow phase, which favours them. Stale onset is defined without retirement; spurious folds are excluded. *Population*. The restart-only certificate makes 7,385 folds over the 150 streams; the 43 spurious ones (below r^* at the fold version) are not stale episodes. A *matched stale episode* is a fold above r^* whose macro is below r^* at a later version before the stream ends: there are 221. Some rule retired 65 of them before onset (LR-CUSUM 47, ExpeL 27, ACE 41; 24 by the LR-CUSUM alone and 11 by ACE alone), and an episode retired before it began cannot be compared, so the common population is the remaining 156 (49 templates, 25 edited relations, 34 edit events, 3 sequences). Code: `paper_figs/acl3_matched.py`.

Model	unretired at stream end			RM delay within 10^4 uses		
	eps.	ExpeL	ACE	LR	ExpeL	ACE
Qwen3-8B	66	53%	53%	3%	4,501	4,507
Qwen3-4B	53	57%	51%	0%	4,391	4,139
Granite-3.3-8B	18	22%	22%	0%	3,278	3,251
Mistral-7B	8	25%	25%	0%	2,110	2,095
OLMo-3-7B	11	64%	55%	9%	5,071	4,507
pooled	156	50%	47%	2%	4,069	3,949
per-cand., pooled	177	52%	46%	5%	4,115	3,773

Table 12: Matched-population detection: stale episodes of the macros folded by the restart-only certificate (last row: by per-candidate betting), replayed open loop under each retirement rule with identical audits; 30 streams per model. The OLMo LR-CUSUM value comes from one censored episode with long follow-up. Bold: ours.

Results. Pooled, the counters leave 50.0% [38.2, 62.8] and 47.4% [36.3, 60.2] of the episodes unretired at stream end, against 1.9% [0.0, 4.7] for the LR-CUSUM (Table 12, Figure 2). Their restricted mean delays are 4,069 [2,924, 5,448] and 3,949 [2,830, 5,296] uses against 67 [46, 99], ratios 60.8 [44.7, 87.1] and 59.1 [43.7, 83.7] (paired template bootstrap, 2,000 resamples; the ratio is computed in each resample); KM medians are 1,750, 1,303 and 43 uses, and on the audited clock the restricted means within 10^3 audits are 403, 391 and 6.6. Each episode costs 1,359 and 1,346 stale wrong answers under the counters, against 38.5. The ratio interval excludes 1 for Qwen3-8B (49 $\times$), Qwen3-4B (95 $\times$, 90 $\times$) and Granite (75 $\times$, 74 $\times$); Mistral (8 episodes) and OLMo (11) are not resolved. The matched population thus confirms the closed-loop picture of Table 11, with a larger gap in delay.

Robustness of the 60 $\times$. Episodes of one template, relation or edit event are correlated, so we also resample these clusters. The ExpeL/LR ratio is 60.8 with 95% intervals [44.7, 87.1] clustered by template, [44.5, 91.0] by edited relation and [41.1, 90.0] by edit event (ACE/LR: 59.1; [43.7, 83.7], [43.1, 89.3], [40.1, 86.6]). With only three edit sequences, sequence clustering is coarse ([31.2, 93.4]); the ratio is 93, 31 and 89 $\times$ on sequences 101, 202 and 303 alone and 46–92 $\times$ when one sequence is left out. Without the exclusion, i.e. on all 221 matched episodes with pre-onset retirements counted as zero delay, the ratio is 63.5 [46.9, 88.4] (58.9 for ACE), with 40.7%, 37.6% and 3.2% unretired.

D.3.3 Per-epoch safety

A template epoch is a run, a routed template and an interval between edits that change the gold answer of some instance routed to that template. A *false fold* is a fold of a candidate whose true reliability (over all instances routed to the template, at stream popularity) is below r^* at the fold version; this is the SpF event of Table 2. An epoch is *at risk* if a candidate that is below break-even at some version of the epoch is under test in it, and the per-epoch false-fold frequency is the share of at-risk epochs with at least one false fold. A false fold is *carried over* if the candidate formed in an earlier epoch.

This subsection uses the standard scoring; Appendix D.4.1 repeats the analysis with answer-only scoring, which the main text reports (Figure 3).

Results. At $\delta = 0.05$ the per-epoch false-fold frequency is 0.007 [0.005, 0.010] for the SPRT and 0.006 [0.003, 0.010] for betting, both at a fresh level per candidate, 0.005 [0.003, 0.007] for betting with the lifetime ledger and 0.002 [0.001, 0.004] for the restart-only certificate with restarts; all stay below δ for every $\delta \leq 0.5$ (Figure 7). Carried-over folds are 40 of 106 false folds for per-candidate betting (16 by candidates that were good at formation), 32 of 70 with the ledger (11), 62 of 87 for the SPRT (50), and 6 of 43 with restarts (0). Table 13 gives the price of restarts per model.

The scoring convention behind most false folds.

KQA-STREAM scores a macro answer as correct only if the query was also routed to the right template, and a template’s true reliability averages over every query routed to it, including misrouted ones. The certificate instead tests the paired utility of its own shadow evaluations. The two notions of “bad”

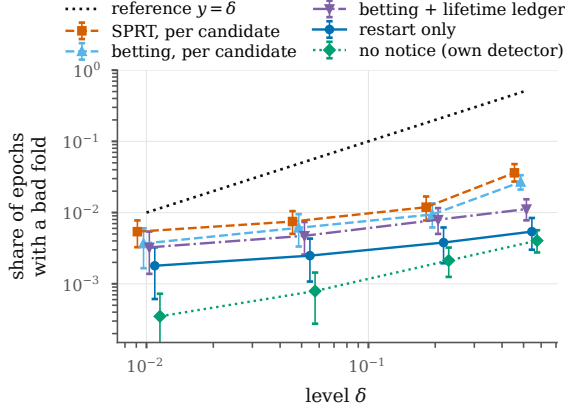


Figure 7: Standard scoring (routing errors count as wrong answers): share of at-risk epochs with a task-level bad fold (reliability below r^*) for five certificates; the dotted line $y = \delta$ is a *reference line*, not a guarantee for these curves. Proposition 23(iv) bounds by δ only folds that violate a certificate’s *own* null (paired utility ≤ 0 ; Figure 3), and per epoch only for the restarted certificate (blue), when notices cover every change of that null; the per-candidate variants are valid per candidate, the ledger without restarts over a template’s lifetime, and the no-notice variant for folds on post-change evidence (Proposition 25). Five models, 150 streams per policy and δ , 95% template-bootstrap intervals.

Model	ΔAcc	ΔJ_{20}	stale	SpF	folds	restarts
Qwen3-8B	-.001 [-.002, .000]	+0.08 [-0.02, 0.18]	72→66	0.10→0.00	60.0→53.8	149
Qwen3-4B	-.003 [-.007, -.001]	+0.30 [0.07, 0.52]	85→82	0.20→0.07	72.9→66.5	110
Granite-3.3-8B	-.003 [-.006, -.002]	+0.33 [0.17, 0.57]	82→79	0.67→0.47	69.4→62.3	100
Mistral-7B	-.002 [-.004, .000]	+0.27 [0.05, 0.48]	42→38	0.97→0.60	36.1→28.1	138
OLMo-3-7B	-.004 [-.006, -.002]	+0.20 [0.12, 0.32]	51→51	0.40→0.30	41.1→35.5	79

Table 13: The price of restarts: the restart-only certificate against the same lifetime ledger without restarts (paired differences with 95% template-bootstrap intervals; stale answers per 10^4 queries, SpF, folds and restarts per stream).

disagree for some candidates: at $\delta = 0.05$, 91 of the 106 false folds of per-candidate betting and all 43 of the restart-only certificate have a positive exact paired-utility drift, so by the test’s own criterion these candidates were useful. Among candidates that are bad under the test’s own null at every version after formation, none folds (0 of 8,434 for per-candidate betting and 0 of 8,084 for the restart-only certificate). Scoring macro answers by the answer alone removes this convention. With answer-only scoring, every remaining false fold of per-candidate betting (15) is carried over by a candidate that was good at formation, which is exactly the event that restarts remove.

D.3.4 Restarts without edit notices

The restart-only certificate restarts a candidate when the replay reports an answer-changing edit of its template. The no-notice certificate (*discard on alarm*; “no notice” in Tables 1 and 3) receives no edit information. It runs, on every candidate under test, the retirement detector of its folded macros: a Bernoulli LR-CUSUM on the candidate’s audited shadow outcomes with $\hat{q}_0$ from the current test segment (estimated before each label), $q_1 = \max(1 - \hat{r}^*, \hat{q}_0 + 0.15)$ and $h = 7$. On an alarm it discards the candidate, and the template’s next candidate forms from a later verified live trace. A retired macro is removed, so its successor starts a fresh test. Proposition 25 covers this rule; a fold is *spanning* if its test segment began before the edit that opened the fold’s epoch, the event that (iii) of that proposition bounds.

Results. Discard-on-alarm has bad folds in 0.0008 [0.0003, 0.0014] of at-risk epochs at $\delta = 0.05$ (9 of 11,375), against 0.0025 [0.0011, 0.0043] for the restart-only certificate (29 of 11,670), and raises accuracy by 1.3 points [0.8, 2.0] and lowers J_{20} by 1.06 [0.63, 1.60] relative to it (Table 14). All 9 of its bad folds are spanning folds of candidates that were good at formation, and all 9 are bad under the test’s own null, so they are exactly the residual risk of Proposition 25(iii); no fold rests on post-change evidence of a bad candidate (0 of 13,935 candidates that are bad under the test’s own null at every version after formation fold). The restart-only certificate’s 29 epochs come from the scoring convention above (all with a positive test drift). Spanning folds are frequent (1,174 of 7,766 folds), but the edit left all but 9 of these candidates good. Retirement is unchanged: 195 stale episodes with 5.1% censored and a mean delay of 68.0 uses, against 174, 4.0% and 68.0. The accuracy gain is positive on every model, because an alarm replaces a candidate whose cached answers broke by one formed on the current facts, while a restarted candidate keeps its stale answers. At every tested level $\delta \in \{0.01, 0.05, 0.2, 0.5\}$, discard-on-alarm has fewer bad-fold epochs than the restart-only certificate (0.0003–0.0040 against 0.0018–0.0054; Figure 7) and 1.1–1.3 more accuracy points.

D.3.5 Partial staleness on real streams

Half of the edits of each sequence change 25% of the affected subjects ($f = 0.25$), and some of

Model	ΔAcc	ΔJ_{20}	bad-fold epochs	alarms	stale
Qwen3-8B	+0.009 [.004, .015]	-0.62 [-1.09, -0.26]	.0009 / .0000	68.3	70←66
Qwen3-4B	+0.016 [.003, .039]	-1.62 [-3.86, -0.26]	.0023 / .0011	61.0	83←82
Granite-3.3-8B	+0.009 [.004, .017]	-0.86 [-1.60, -0.33]	.0006 / .0064	91.8	81←79
Mistral-7B	+0.012 [.006, .022]	-1.32 [-2.27, -0.62]	.0003 / .0026	224.2	42←38
OLMo-3-7B	+0.017 [.004, .042]	-0.89 [-2.22, -0.18]	.0000 / .0053	99.7	49←51

Table 14: Discard-on-alarm (no edit notices) against the restart-only certificate (edit-notified restarts), $\delta = 0.05$: paired differences with 95% template-bootstrap intervals; share of at-risk epochs with a bad fold (no notice / restart only); detector alarms per stream; stale answers per 10^4 queries (no notice $\leftarrow$ restart only).

them leave a macro partially stale. We test the phase diagram on these episodes open loop: every macro that per-candidate betting folds is re-scored on its full audited label stream from its fold to the end of the stream (9,448 macros over the five models). Ground truth is the macro’s exact failure rate $q_v = 1 - R_v$ at each knowledge-base version. An *episode* starts at the first version after the fold at which q rises by at least 0.1 over its value q_0 at the fold, has failure rate q_1 there, and ends at the next version at which q moves by more than 0.02, or at the stream end. We keep episodes with $q_0 \in [0.02, 0.15]$, since with $q_0 = 0$ every rule can alarm at the first failure without false alarms, and call them *partial* if $q_1 \in [0.15, 0.5]$ and *near-total* if $q_1 > 0.5$. Each rule restarts at the change (Lorden’s worst case for a CUSUM); its delay is the number of folded uses from the change to its alarm, censored at the episode end, summarized by the Kaplan–Meier mean restricted to 1,000 uses. Thresholds are matched to 1, 3 and 10 false alarms per 10^4 healthy uses, measured on the uses between fold and change of all folded macros with q_0 in the same band. Intervals are 95% template-clustered bootstraps with thresholds re-matched in every re-sample.

Results. Of the 2,076 single-change episodes, 217 are partial episodes in the band (63 templates; 32–53 per model) with mean $q_0 = 0.067$ and $q_1 = 0.331$, for which Corollary 2 predicts the failure weight $1/\rho = 4.8$. At 1 false alarm per 10^4 healthy uses the best tested weight is 7 (195 uses [152, 246]); the delay is U-shaped in the weight, as the efficiency (3) predicts: the symmetric rule is $1.49\times [1.18, 1.82]$ slower and detects 90% of the episodes within 1,000 uses (weight 7: 99.8%), and weight 25 is $2.09\times [1.71, 2.65]$ slower (Figure 8). At 3 and 10 false alarms the best weight is 5, and the symmetric rule is $1.22\times [1.03, 1.50]$ and $1.29\times [1.04, 1.45]$ slower. In the bootstrap the best

weight is 5 or 7 in 98.6%, 93.2% and 46.3% of re-samples at the three rates (3: 52.5% at the highest rate). On the 170 near-total episodes of the same band (mean $q_1 = 0.690$, predicted $1/\rho = 2.0$) the optimum moves down to 3–5 and the symmetric rule is within 1.04 – $1.19\times$ of it (intervals contain 1), so the advantage of failure-heavy weights is specific to partial staleness, as Theorem 1 predicts. The per-memory LR-CUSUM tuned at the low end $q_1 = \hat{q}_0 + 0.15$ (Proposition 8) is within $1.21\times [0.90, 1.64]$ of the best fixed weight; the deployed LR-CUSUM, tuned for a drop below break-even ($q_1 = \max(1 - \hat{r}^*, \hat{q}_0 + 0.15)$), is $1.59\times [1.20, 2.16]$ slower on these smaller changes, for which it is not tuned.

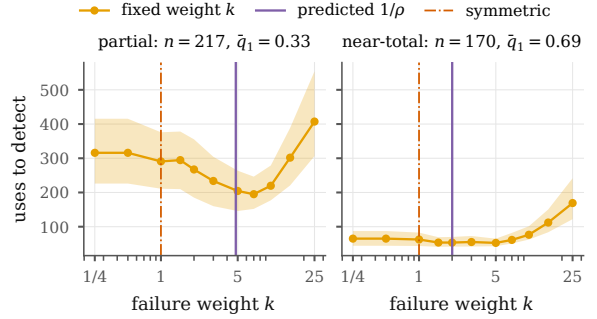


Figure 8: Partial staleness on real streams: delay (Kaplan–Meier mean within 1,000 uses) of fixed-weight counters against the failure weight k at 1 false alarm per 10^4 healthy uses (95% template-bootstrap bands), with the weight $1/\rho$ predicted by Corollary 2 from the mean failure rates. Left: 217 partial episodes ($q_1 \in [0.15, 0.5]$); right: 170 near-total episodes; both with $q_0 \in [0.02, 0.15]$.

D.3.6 Per-memory weights against the best fixed ratio

The learning-rate sweep of Appendix D.2.1 compares the per-memory LR-CUSUM with fixed-ratio counters on the same certificate, closed loop, at matched false retirements (Table 7). Delays here are naive means that count open episodes at their length so far (as in Appendix D.2); slower rules are censored more often, so this bias favours the fixed-ratio rules over the LR-CUSUM. With k down to 0.1, the closed-loop optimum lies below 1 on four of five models ($k = 0.75, 0.25, 0.1$ and 0.05 ; 25 for Qwen3-8B), consistent with the predicted $1/\rho$ of 0.67–2.14 under near-total staleness. The LR-CUSUM remains 5.8, 5.9, 2.0, 1.3 and $2.0\times$ faster than the best tested ratio. Paired intervals of this ratio at each model’s best k exclude 1 for

Qwen3-8B (5.8 [3.2, 9.4]), Qwen3-4B (5.9 [2.3, 11.2]) and OLMo-3-7B (2.0 [1.1, 4.6]), but not for Granite (2.0 [0.9, 3.4]) or Mistral (1.3 [0.9, 3.4]) (paper_figs/ac13_bestk.py). The weights the LR-CUSUM assigns differ across memories and models: the median per-memory weight ratio is 1.90, 1.33, 0.92, 0.50 and 0.67 (Qwen3-8B to OLMo-3-7B).

D.4 Aligned safety, same-information baselines, decoding variance and natural edits

The replays of Appendices D.4.1 and D.4.2 reuse the recorded outcome tables of the five models in the main setting (30 streams per model and policy; no shard is missing). Intervals are 95% template-clustered bootstraps; pooled rows resample templates jointly across models.

D.4.1 Safety scored as the certificate tests it

Scoring. Under the standard scoring of KQA-STREAM, a macro answer counts as correct only if the query was also routed to the right template, so a template’s reliability averages over misrouted queries, which the certificate’s paired test does not see (Appendix D.3.3). Here a one-step answer is correct if and only if it equals the gold answer (*answer-only* scoring). A candidate is then *bad for the task* if its answer-scored reliability is below $r^* = a - (m - 1)/L$ ($L = 20\bar{m}$), and *bad under its own null* if the exact drift $\mathbb{E}Z$ of its paired utility on its routed queries is ≤ 0 (with the certificate’s own L , B and clipping; build instance excluded). An epoch is an interval between answer-changing edits of a template, and it is at risk if a candidate that is bad at some version of it is under test. We count three events: (i) a *false certification*, a fold of a candidate that is bad under its own null at the fold; it is *on null evidence* if the null held at every version of the fold’s test segment (the event bounded by Proposition 23(i)–(ii) and Proposition 25(ii)); (ii) a task-level bad fold, below break-even; (iii) a wrong answer among the one-step answers served.

Results. The restart-only certificate makes no false certification at $\delta \leq 0.1$ on any model (0 of 6,657 at-risk epochs at $\delta = 0.05$; Table 15, Figure 3); at $\delta = 0.2$ and 0.5 the shares are 0.0003 [0, 0.0009] and 0.0019 [0.0006, 0.0036], all on null evidence and in fresh test segments, as Proposition 23(ii) allows. Intervals clustered by relation or by edit sequence are comparable. The two notions

	δ	(i) own null	on null ev.	(ii) task	(iii) wrong	stale
restart only	.01	0/6704	0	0/6693	11.2%	69
	.05	0/6657	0	0/6645	11.6%	75
	.1	0/6620	0	0/6611	11.7%	78
	.2	2/6554	2	2/6546	11.8%	82
	.5	12/6401	12	10/6396	12.8%	103
no notice	.01	6/6471	0	5/6458	10.7%	70
	.05	12/6411	0	13/6396	11.2%	80
	.1	13/6379	0	15/6365	11.4%	85
	.2	26/6319	2	29/6304	11.7%	92
	.5	58/6122	19	59/6109	12.5%	116

Table 15: Safety under answer-only scoring (pooled, 150 streams per row). (i), (ii): epochs with the event / at-risk epochs; on null ev.: epochs with a false certification on null evidence; (iii): wrong answers among served one-step answers; stale: stale answers per 10^4 queries. Restart only: the restart-only certificate; no notice: discard on own alarm; ours (the recommended deployment): Appendix D.5.2. Bold: ours (both rows).

of “bad” nearly coincide: at $\delta = 0.05$, of 27,339 candidates, 3,611 are bad for the task, 3,619 under their own null and 3,571 under both; the task-level counts in Table 15 differ from the own-null counts by at most three epochs. Without notices, no false certification rests on null evidence at $\delta \leq 0.1$; every task-level bad fold at these levels (5, 13 and 15) spans an edit, i.e. its test segment began before the change, which is the residual risk that Proposition 25(iii) bounds. Every point estimate of both certificates is below δ .

False certification is not the error rate. At every δ , 11.2–12.8% of the one-step answers that the restart-only certificate serves are wrong (7.6–8.7% on Qwen3-8B, 13.2–17.1% on Mistral-7B). Stale answers are only 14–17% of them; the rest are errors of macros that were above break-even when folded, which a fold accepts by design ($r > r^*$ allows $r < 1$). δ therefore controls whether a bad memory is trusted, not how often a trusted memory errs; the second is set by r^* and by the detection delay of Section 3.

Standard scoring. With routing errors counted as wrong answers, the restart-only certificate has bad folds in 0.0025 [0.0011, 0.0043] of at-risk epochs at $\delta = 0.05$ (29 of 11,670), all of them candidates with a positive paired-utility drift, and discard-on-alarm has 0.0008 [0.0003, 0.0014] (9 of 11,375; Appendices D.3.3 and D.3.4, Figure 7). The order of the two certificates flips between the scorings because the routing convention creates most of the restart-only certificate’s standard-scoring bad folds, whereas the no-notice certifi-

cate’s bad folds are carried across edits under both. Both scorings stay below δ at every level.

D.4.2 Same-information baselines

The restart-only certificate receives an edit notice: the templates whose gold answer an edit changes. It uses the notice only to restart candidates under test; folded macros are retired by the LR-CUSUM alone. The baselines below receive the same notices, on the same streams, audits and edits, with the same certificate parameters ($\delta = 0.05$, seven bets, $h = 7$) where they certify. *Invalidate on notice* drops the folded macro or the candidate under test of a notified template; a later verified live trace forms a fresh candidate, tested under the lifetime ledger, and folded macros are retired by notices only or also by their LR-CUSUM. Because carried-over candidates are discarded at every announced change, Proposition 23(iv) gives per-epoch safety for those epochs. *Drop only* sends the template back to live reasoning for the rest of the stream. The answer caches evict notified entries: a verified-answer cache, which stores a question’s live answer only when that run was audited and verified correct and returns it on an exact repeat, and the vCache-style cache of Appendix D.2.4. To model missed notices, each answer-changing edit is announced independently with probability f (the same draws for all policies). The *compiled plan* row is the restart-only certificate on compiled plans (with restarts).

Policy ($f = 1$)	Acc	Calls	Stale	Δ Acc vs restart only (points)
restart only (restart-only certificate)	.430	3.36	63.3	–
invalidate	.430	3.20	0.0	–0.06 [–0.94, +0.80]
ours: invalidate + LR	.433	3.38	0.0	+0.24 [–0.24, +0.81]
drop only	.332	3.84	0.0	–9.8 [–12.9, –7.0]
verified-answer cache	.188	3.43	0.0	–24.3 [–28.1, –20.3]
vCache-style	.186	3.50	0.0	–24.5 [–28.3, –20.5]
compiled plan, no restarts	.461	3.20	12.3	+3.1 [+2.5, +3.8]
compiled plan	.456	3.25	11.8	+2.5 [+2.0, +3.2]

Stale per 10^4 queries at notice fraction f	1	0.8	0.5	0
restart only (restart-only certificate)	63.3	65.3	67.8	66.3
invalidate	0.0	124.5	324.5	809.6
ours: invalidate + LR (recommended)	0.0	13.3	28.2	66.3
compiled plan	11.8	12.0	12.4	12.3

Table 16: Same-information baselines, pooled over five models (150 streams per row; standard scoring). Top: perfect notices, paired differences with 95% template-bootstrap intervals. Bottom: missed notices. Bold: ours.

Results. With perfect notices, invalidation matches the restart-only certificate’s accuracy and serves no stale answers (Table 16); its J_{20} differs by -0.12 [–0.82, +0.51]. Dropping without reforming loses 9.8 points, and the answer caches stay at live accuracy (0.19, the pooled live accu-

racy) because they answer only 10–13% of queries in one step. With missed notices, invalidation alone degrades fast: at $f = 0$ every one of its 218 stale episodes is still in service at stream end, and its accuracy falls by 4.4 points [2.2, 6.9]. Adding the LR-CUSUM keeps stale answers at or below the restart-only certificate’s at every f , and at $f = 0$ the two policies coincide. The share of at-risk epochs with a bad fold rises as notices are missed, because carried-over evidence is no longer discarded (Proposition 23(iv)): 0.0026, 0.0030, 0.0036 and 0.0047 for invalidate + LR at $f = 1, 0.8, 0.5$ and 0 (restart-only certificate: 0.0025–0.0047), all below δ . On Qwen3-8B, invalidate + LR adds 0.6 points [0.15, 1.1] over the restart-only certificate (Table 3).

Compiled plans with restarts. With restarts, compiled plans keep +2.5 points over macros and 11.8 stale answers per 10^4 queries, and they halve the bad folds of plans without restarts: 0.0032 against 0.0064 of at-risk epochs (29 of 9,179 against 59 of 9,172) and 0.29 against 0.63 bad folds per stream, for 0.6 accuracy points. Per model the gain over macros is 1.2–4.0 points (Table 6).

D.4.3 Decoding variance

Each instance–version of the main runs is decoded once, so the intervals elsewhere resample templates, streams and edit sequences, not decoding randomness. To measure the latter we re-decoded a Qwen3-8B subset (60 templates $\times$ 10 instances, edit sequence 101, knowledge-base versions 0, 8, 16 and 24) with three decoding seeds. Live accuracy per version is 0.368/0.343/0.322/0.312, 0.367/0.338/0.320/0.312 and 0.368/0.335/0.313/0.305 for the three seeds: the standard deviation across seeds is at most 0.0044 per version, against a template-bootstrap 95% half-width of 0.09–0.10 for one seed. Two seeds give the same final answer on 91.9% of instance–versions and the same correctness on 97.6%. Replays of each seed’s tables (20 streams each) rank the six policies (live, fold once, ExpeL- and ACE-style counters, the restart-only certificate and its no-notice variant) in the same order of accuracy for every seed. The restart-only certificate’s accuracy is 0.447–0.452 and its stale answers 32–33 per 10^4 queries, against 698–771 for fold once and the counters. On the matched population of its folded macros, the LR-CUSUM leaves 0% unretired and the counters 57–75%, a delay ratio of

131–155× on this small subset (32–35 episodes). Decoding noise is thus two orders of magnitude below the template variation that our intervals already include, and no conclusion changes on the subset.

D.4.4 Natural edits from Wikidata temporal qualifiers

Timeline. Of the 207,851 facts of the knowledge base, 15,290 (7.4%) carry time information (7,017 start or end dates and 8,273 dated events). The 300 main templates are too sparse for a natural timeline: only 321 of their instances change answer between 1992 and 2016 while staying answerable. We therefore built a natural subset with the construction rules of Appendix D: of 2,250 candidate paths, 1,769 are valid at every date, 88 change at least 5% of their answers, and 72 templates (68 new, 4 from the main set; 4,304 instances) remain after the diversity caps. The stream applies the 1992 state at query 0 and one knowledge-base version per year, 1993–2016, at uniform stream times (8,000 queries per year). Real changes are small and partial: 43.6 answer changes and 8.8 templates touched per year; in the median template-year only 3.3% of the template’s answers change, and no change is total (main synthetic edits: median 40%, 3% total; Figure 9b). We ran four models (Qwen3-8B, Qwen3-4B, Granite-3.3-8B, Mistral-7B); OLMo-3-7B was not run on this timeline. The offline exact check reproduced all 111,904 traces at all 26 versions. Policies and settings are those of the main replays (30 streams of 200k queries per model, $\delta = 0.05$).

Policy (pooled)	Acc	Calls	Stale	Bad/stream	J_{20}
live always	.175	4.47	0	0	79.3
fold once	.340	2.60	231	8.5	62.6
ExpeL-style counter	.391	3.39	241	149	59.0
ACE-style counter	.391	3.52	238	207	59.2
restart only (restart-only certificate)	.356	4.03	80.5	0.10	62.8
ours (recommended)	.337	4.22	0	0.11	64.8
no notice (own detector)	.384	3.84	99.5	0.03	60.2
compiled plan	.374	3.95	2.8	0.10	61.2

Table 17: Natural edits, pooled over four models (30 streams of 200k queries each; standard scoring). Stale: per 10^4 queries. Bold: ours.

Results. Counters. Counters serve $3.0\times$ [2.2, 4.0] more stale answers than the restart-only certificate (4.0 , 2.8 , 2.2 and $3.2\times$ per model) and make 149–207 bad folds per stream, against 0.10. Replayed on the same macros as in Appendix D.3.2, 70 of the 142 stale episodes remain after removing those that some rule retired before the change;

the counters leave 84.3% [46.3, 100] in service at stream end and the LR-CUSUM 37.1% [4.5, 55.3], with a restricted-mean delay ratio of $20.5\times$ [13.2, 72.9] and 405 against 30 stale answers per episode. The LR-CUSUM leaves more episodes open here than on synthetic edits because real staleness is mostly partial and 26 of its 70 episodes are cut off by the end of the stream; Appendix D.5.3 reports the sensitivity to the event set and to leaving out one year or one relation. **Weights.** On 1,322 single-change episodes (35 templates; mean $q_0 = 0.121 \rightarrow q_1 = 0.330$, optimal weight $1/\rho = 3.7$), the per-memory LR-CUSUM is faster than the best fixed weight at every matched false-retirement rate: $1.08\times$ [1.01, 1.32], $1.56\times$ [1.34, 1.85] and $1.14\times$ [0.93, 1.56] at 1, 3 and 10 false alarms per 10^4 healthy uses ($1.72\times$ on the 907 partial episodes at 3). The best fixed weight is about 2, not the 5–7 of the synthetic partial episodes (Appendix D.3.5), and on the 170 near-total episodes the symmetric rule is best, as Corollary 2 predicts ($1/\rho = 0.73$). **Safety.** Bad-fold epochs of the restart-only certificate are 0.0008, 0.0013, 0.0021 and 0.0062 of at-risk epochs at $\delta = 0.01, 0.05, 0.2$ and 0.5 (4/4,790, 6/4,770, 10/4,744, 29/4,686), and 0 up to $\delta = 0.2$ under answer-only scoring. Every point estimate is below δ ; one per-model upper bound exceeds it (Qwen3-4B at $\delta = 0.01$, 2 of 690 epochs). **Accuracy and cost.** The restart-only certificate adds 18.1 points [14.7, 21.7] over live reasoning with 10% fewer calls (on Mistral-7B it uses slightly more, 5.31 against 5.04), and certified compiled plans serve 2.8 stale answers per 10^4 queries. The counters are 3.5 points more accurate than the restart-only certificate (5.4 than the recommended deployment, Appendix D.5.3), with lower loss, because when real staleness is this mild, folding everything without certification pays; this is the price of safety of Section 4, bought here with 149–207 bad folds per stream. The synthetic ratios of the main text (7.0 – $13.5\times$, $60\times$, a best weight of 5–7) are not claimed for natural edits.

D.5 Silent changes of the null, the recommended deployment, natural-edit sensitivity

D.5.1 When the null changes without a notice

What triggers a notice. In the replays, the notice of version v lists the templates k for which the gold answer of some instance routed to k differs between versions $v - 1$ and v (notice_tables);

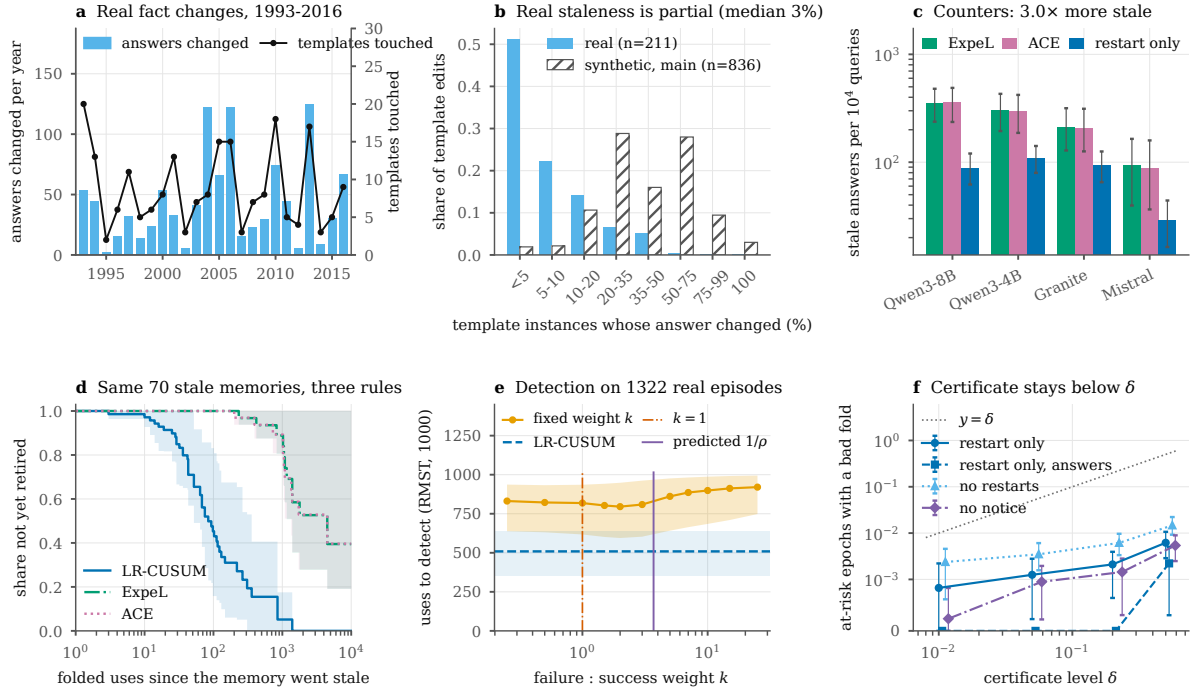


Figure 9: Natural-edit transfer (Wikidata temporal qualifiers, 1992–2016; four models, 30 streams each). **(a)** Real fact changes per year. **(b)** Share of a template’s instances whose answer changes in one edit: half of the real changes touch under 5% of the template, most synthetic ones 20–75%. **(c)** Stale answers per 10^4 queries: counters serve $3.0\times$ more than the restart-only certificate. **(d)** The same 70 stale memories under three retirement rules (Kaplan–Meier). **(e)** Delay to detect at 3 false alarms per 10^4 healthy uses on 1,322 real episodes: fixed failure weights against the per-memory LR-CUSUM. **(f)** Share of at-risk epochs with a bad fold against δ (“answers”: answer-only scoring; no notice: discard at the own alarm). Bands and bars: 95% template bootstrap.

nothing else restarts a test or invalidates a memory. Epochs are the intervals between such notices. The certificate’s null, however, is the paired-utility drift $\mathbb{E}Z \leq 0$, which depends on live reasoning as well: with $C^L - C^M = 2$, $L = 50$ and a macro of reliability 0.9, a live accuracy of 0.8 gives $\mathbb{E}U = +7$, and 0.95 gives $\mathbb{E}U = -0.5$. Live reasoning can thus make a memory useless without any change of its answers or of the gold answers. Proposition 23(iv) therefore gives per-epoch safety only if notices cover every change of the null, including changes of live reasoning; across a change without a notice, the evidence banked before it is covered only by the carried-over bound $\min\{1, \delta_j K_c\}$ of Proposition 23(iii). Two events must be kept apart: a candidate that is null *at* the fold, and a test segment that was null *throughout* (“on null evidence”). Only the second is bounded by δ without further assumptions.

How often it happens. Over the three edit sequences and five models, 16,827 of 107,496 template-versions (15.7%) change some routed

instance’s live trace (answer, correctness or call count) without a notice, 1,828 change a live correctness and 1,715 the routed live accuracy; 5,179 carry a notice (Table 18, top). Within the replays ($\delta = 0.05$, 150 streams per row), we tracked the exact drift of every candidate’s own test at every version of its life. Table 18 (bottom) counts the candidates whose null flips without a notice while they are under test, and the folds whose test segment spans such a flip. Flips are rare (0.3–0.7% of candidates), and of 7,385–9,835 folds per row at most 2 span one, all of candidates that are useful at the fold; no fold of a candidate that is null at the fold occurs at $\delta = 0.05$. So on these streams the per-epoch statement holds as measured, but the assumption it needs is not automatic.

D.5.2 The recommended deployment

The main tables and Figure 3 report the recommended deployment (“ours”: the restart-only certificate plus invalidation on notice, invalidate + LR of Appendix D.4.2); restart only (the restart-only certificate) is the comparison row. Under answer-

Model	template-versions	notice	silent live-trace	change	silent live-accuracy	change	
Qwen3-8B	21,528	884		1,392		286	
Qwen3-4B	21,600	867		3,124		363	
Granite-3.3-8B	21,456	966		4,719		448	
Mistral-7B	21,384	1,333		5,328		385	
OLMo-3-7B	21,528	1,129		2,264		233	
Policy	scoring	candidates	with flip	to null / useful	folds	span a flip	null at fold
restart only	standard	31,593	226	97 / 155	7,385	2	0
restart only	answer	27,339	192	82 / 133	8,632	0	0
ours	standard	42,306	132	68 / 80	8,294	2	0
ours	answer	37,928	114	58 / 75	9,835	0	0

Table 18: Silent changes of the null. Top: template-versions (three sequences, routed templates) with a notice and without one but with a change of live reasoning. Bottom: candidates whose own null flips without a notice while under test (flip events by direction), and folds whose test segment spans such a flip; $\delta = 0.05$, 150 streams per row, pooled over five models. Bold: ours (all policies).

only scoring, ours makes 0 own-null false certifications at $\delta \leq 0.1$ (0/5,814, 0/5,779, 0/5,758 at-risk epochs at $\delta = 0.01, 0.05, 0.1$), 1/5,718 at $\delta = 0.2$ and 11/5,614 (0.0020 [0.0006, 0.0038]) at $\delta = 0.5$, all on null evidence; task-level bad folds are 0, 0, 0, 1 and 8. Of the one-step answers it serves, 9.6–10.6% are wrong (restart only: 11.2–12.8%), and none is stale. In Table 2 its stale answers are reported with every notice missed (the counters’ information; then it coincides with the ledger certificate without restarts); its bad folds per stream are those with perfect notices (0.00, 0.07, 0.43, 0.63, 0.30). The matched retirement comparison (Figure 2) uses the macros that restart only folds, because their folded macros, like ours after a missed notice, are retired by detection alone.

D.5.3 Natural edits: sensitivity

All numbers below reuse the natural replays of Appendix D.4.4; the recommended deployment was replayed in addition (same streams). *Recommended deployment*. It serves no stale answers, has bad folds in 0.0007, 0.0013, 0.0029 and 0.0060 of at-risk epochs at $\delta = 0.01, 0.05, 0.2, 0.5$ (4/5,493, 7/5,498, 16/5,492, 33/5,487), and 0.11 bad folds per stream, but it is 1.9 points [1.0, 2.8] less accurate than restart only (0.337 against 0.356): when an edit changes 3% of a template’s answers, invalidation discards a memory that is still mostly right.

Event set. The matched retirement comparison uses the 70 of 142 stale episodes that no rule retired before onset. On all 142, counting a pre-onset retirement as detection at 0 uses, the counters leave 70.4% (ExpeL) and 62.7% (ACE) in service at stream end against 26.8% for the LR-CUSUM,

with a delay ratio of $13.6 \times [5.4, 88.7]$; each rule on its own episodes gives 90.1%, 89.0% and 45.2% unretired and a ratio of $10.3 \times$.

Leave one out. Leaving out the episodes whose onset falls in one year, the counters leave 57.7–100% and the LR-CUSUM 3.8–45.6% in service, and the delay ratio is 18.5 – $32.0 \times$; leaving out the templates that use one changing relation (21 relations), 56.0–100% against 21.2–45.6% and 17.2 – $38.1 \times$. Most common episodes (44 of 70) start in the last year, 2016, and are censored early; without them the shares are 58% against 4%, so the unretired shares depend on censoring, while the delay ratio is stable. Leaving out one relation, the counters serve 2.5 – $3.2 \times$ more stale answers than restart only (2.3 – $2.9 \times$ than the ledger without restarts, i.e. ours with every notice missed), every certificate’s bad-fold share stays below δ at every level (largest: 0.0080 at $\delta = 0.5$), and the per-memory LR-CUSUM stays faster than the best fixed weight at 1, 3 and 10 false alarms per 10^4 healthy uses (1.01 – $1.17 \times$, 1.33 – $1.74 \times$, 1.04 – $1.29 \times$). The weight comparison uses the 1,322 single-change episodes; episodes with several changes do not have one change point and are not used.

D.5.4 Which variant for which goal

Variant	Use when	Evidence
Recommended: invalidate on notice + certify with restarts + LR-CUSUM	false deployments must be bounded and notices may be incomplete	0 stale with notices; 13–66 per 10^4 when 20–100% are missed (Table 16); no false certification at $\delta \leq 0.1$; code: 2.0 stale per 10^4 with only requirement notices, bad folds in $1.3 \cdot 10^{-5}$ of epochs under test at $\delta = 0.05$ (Appendix D.6)
Restart only	changes are mild and partial, notices reliable	+1.9 points over invalidation on natural edits (Appendix D.5.3)
Compiled plans + certificate	the derivation can be re-executed cheaply on current facts	+2.5 points over restart-only macros, 12 stale per 10^4 (Table 16)
Verified answer cache	exact repeats, reliable notices	live accuracy; 13% of queries in one step
Counters (ExpeL/ACE)	staleness is mild and folding everything pays; never when stale answers are costly	natural edits: +3.5 points, 149–207 bad folds per stream

Table 19: Decision guide (pooled evidence; main text: Section 6). Bold: ours.

D.6 A code domain with an exact verifier: CodeStream

CodeStream tests the theory on a second real domain, chosen so that the verifier is exact and mem-

ories are what tool-building agents store: a verified program, reused as a tool (Wang et al., 2024a; Cai et al., 2024; Wang et al., 2024c; Yuan et al., 2024). It writes the same potential-outcome tables as KQA-STREAM, so every policy, metric and replay setting of Appendix D runs on it unchanged; only the domain, the stream length and the cost unit differ. The setting was fixed after a two-model pilot, before the full run.

Tasks, queries and the verifier. We use 521 EvalPlus tasks (Liu et al., 2023): 359 from MBPP+ (v0.2.0; Austin et al., 2021) and 162 from HumanEval+ (v0.1.10; Chen et al., 2021). We exclude 21 tasks: 17 with a special oracle or a failing canonical solution, and 4 with fewer than 10 usable hidden inputs. A query is a task plus one hidden plus-test input, 20 per task (10,420 queries); its answer is the function’s output. The verifier is EvalPlus’s own check against the reference output of the task version in force: exact equality, or closeness within the task’s tolerance for floating-point outputs. Each (program, version, input set) runs once and is cached, so every policy sees the same verdicts; re-executing the 6,330 sandbox runs of the table build in fresh processes changes 57 of 126,600 verdicts (0.05%), from programs whose output is not deterministic (e.g. set order).

Live reasoning and memories. The live reasoner is a solve–test–repair agent. It writes the function from the task prompt (temperature 0.7, at most 768 new tokens; Qwen3 with thinking disabled), runs it on the visible tests (the EvalPlus base inputs) in a sandbox of the current version, and, if a test fails, gets the failing call, the expected output and the output or exception it produced. It makes at most three attempts; the final program is run on the query input. Each task has 20 samples, one per query, so live outcomes of different queries are independent samples. A *memory* is the final program of a live trace whose answer was audited correct; folded, it is executed directly on later queries of the same task, in the current environment. Memories are thus compiled plans in the sense of Section 6: an environment change can break them, and a requirement revision changes the answers they are checked against. Some programs pass the visible tests but fail other hidden inputs, EvalPlus’s own finding that base tests under-specify tasks; they are the domain’s candidates that are bad at formation, so certification has real work to do.

Edit catalogue. Each timeline has 14 edits (15 versions) at uniform stream times, in the order A1, S1, A2, A3, S2, A4, S3, A5, A6, S4, A7, S5, A8, S6. *Requirement revisions* S1–S6 are the main source of staleness. Each batch revises 22 tasks (6% of the revisable tasks; 132 per timeline, each at most once) by appending one natural post-processing change to the prompt as “Requirement update: ...”: sort, deduplicate, lowercase, reduce modulo 10^9+7 , clamp negatives to 0, round to two decimals, return ‘Yes’/‘No’, or return a list instead of a tuple. The reference output changes exactly on the inputs whose output the change affects, so staleness is partial or total by construction from the real input distribution (in the design-time catalogue the share of a revised task’s queries whose answer changes ranged from 0.05 to 1, median about 0.6). The three timelines (seeds 101, 202, 303) revise different tasks. *API edits* A1–A8 re-enact documented Python breaking changes in the sandbox (Table 20); they change no reference output, and a memory goes stale only if its own program changes behavior. Error messages carry the migration hint, as real deprecations do, so live reasoning can repair from test feedback.

Id	Edit (re-enacted)	Precedent	Programs broken
A1	typing.List etc. removed	PEP 585 deprecation	55 / 68, total
A2	re.sub/split: count, flags keyword-only	deprecated in 3.13	0 / 0
A3	str patterns ASCII-only by default	Python 2/3 default	0 / 0
A4	int-str conversion capped at 640 digits	3.11 (CVE-2020-10735)	1 / 1, partial
A5	round half away from zero	Python 2	2 / 2, partial
A6	math.floor/ceil return float	Python 2	1 / 1, total
A7	Counter.most_common tie order	dict order before 3.7	0 / 0
A8	zip strict by default	PEP 618 option	3 / 3, partial

Table 20: API edits of CodeStream. Programs broken: distinct verified programs whose behavior the edit changes, out of 1,494 (Qwen3-4B) / 1,380 (Qwen3-8B) in the two-model pilot.

Notices. A requirement revision bumps the task’s version: this is the answer-changing notice of the main experiments. API edits are unannounced: they are the domain’s unannounced changes, and for them the LR-CUSUM is the only safety net. They also change live reasoning (in the pilot, after A1 the mean number of attempts on the 210 affected Qwen3-4B traces rose from 1.21 to 2.13) without changing any answer, a silent change of the certificate’s null in the sense of Appendix D.5.1. The *changelog* variant additionally invalidates, at each API edit, every memory whose source mentions a changed symbol, a static match that real maintainers would make: it is exact for removals but flags many unaffected programs for semantic

edits.

Streams, policies and statistics. Streams have $T = 10^6$ queries, Zipf demand over tasks ($s = 1.2$), audits with $p = 0.1$, $L = 20\bar{m}$ and $\delta = 0.05$; the certificates, the bet grid and the LR-CUSUM threshold $h = 7$ are those of the main experiments. Each model is replayed on 10 stream seeds per timeline. All memory policies store programs. Labels: *fold once*; ExpeL- and ACE-style counters; *per candidate*, the per-candidate certificate of Table 1, without notices or restarts, i.e. the counters’ information; *restart only*, the restart-only certificate, which restarts at requirement notices; *ours*, the recommended deployment (invalidate on notice and re-form, ledger, restarts, LR-CUSUM), as in the main text; and *ours+changelog*. A *stale answer* is a folded answer that is wrong although the same memory answered this query correctly when it was formed. 95% intervals resample templates (clusters) with 2,000 bootstrap draws, pooled over seeds and timelines; pooled intervals resample (model, template) clusters.

Models and compute. All five models ran all three timelines, so pooled numbers rest on 150 streams, 30 per model; every stream has the same weight.

Two cost accountings. Writing, testing and repairing a program costs more than running a stored one, and the domain admits two units. *Step accounting*, our main setting and faithful for tool libraries, charges a live query its attempts (LLM calls) plus one execution step, and a folded memory one execution step; live reasoning costs 2.68 steps per query. *Call accounting*, the unit of KQA-STREAM, charges live reasoning its LLM calls (1.68 per query) and a folded memory one call. J_{20} is computed in each unit with $L = 20\bar{m}$, as in the main text. Every cost statement below names its unit.

(a) Counters leave stale memories in service. The counters serve 518 [471, 567] (ExpeL) and 510 [463, 561] (ACE) stale answers per 10^4 queries, against 10.1 [8.8, 11.4] for the certificate with the counters’ information (per candidate): $51 \times$ pooled (95% intervals within [43, 60]) and $35\text{--}73 \times$ on every model, with every per-model lower confidence bound at least 25. Per folded answer, i.e. after dividing each policy’s stale answers by its one-step coverage, the ratio is still $36 \times$ pooled ($27\text{--}48 \times$ per model), so the gap is not an artifact of the counters’

higher coverage. Against restart only (11.0 [9.0, 13.2]) the ratio is $47 \times$ [39, 58] (ExpeL) and $47 \times$ [38, 57] (ACE); against ours (2.0 [1.4, 2.7]) it is $258 \times$ [191, 363] (ExpeL). Ours with changelog notices serves 0.01 per 10^4 (all on Mistral-7B). Each policy’s own stale memories remain in service at stream end in 59% [56, 61] (ExpeL) and 57% [54, 59] (ACE) of cases, against 10% [7, 12] for per candidate; fold once retires none. These shares compare each rule on its own folded memories, not on a matched population as in Figure 2; the matched comparison is claim (b). In call accounting the stale answers of the certified policies are 9.2 (per candidate), 12.3 (restart only) and 2.0 (ours) per 10^4 ; the counters serve $55\text{--}56 \times$ more than per candidate pooled ($38\text{--}86 \times$ per model), and the counter/restart-only ratio is $42 \times$ [32, 56] (ExpeL).

(b) Per-memory weights at matched false retirements. All rules run on the same per-candidate certificate and differ only in how they retire a folded memory. The per-memory LR-CUSUM uses its default threshold $h = 7$. Each fixed failure:success weight $k:1$ (score $+1$ per audited failure, $-1/k$ per success), for $k \in \{1, 2, 7, 25, 50, 100, 200, 500, 1000, 5000\}$, gets its best threshold (between 1 and 2,584 failure units; a finer grid from 34 to 89 for 7:1 and 5000:1) among those with no more false retirements per stream than the LR-CUSUM (a retirement of a memory at or above break-even), a choice that favors the fixed rules. We measure *stale exposure*: folded uses of a memory while it is stale, per stream. Pooled, the LR-CUSUM has 702 stale uses per stream at 33.5 false retirements; the best fixed weight (5000:1, threshold 78, 31.7 false retirements) has 7,300, i.e. $10.4 \times$ [7.7, 13.7] more, and the symmetric 1:1 rule $29.1 \times$ [14.6, 51.7] more (Table 22). On every model the best fixed weight is $7.6\text{--}11.4 \times$ worse, with every lower confidence bound at least 3.5. On the shared threshold grid, exposure falls from 25:1 to 100:1 and is flat from 100:1 to 1000:1 (within 3%): there a success barely discounts a failure, and all weights converge to one rule, a failure count that never forgets; 5000:1, this limit tuned on a finer threshold grid, is lower still. The optimum is interior on Mistral-7B (7:1) and this limit on the other four models.

The theory predicts both the limit and the gap. In 86% of the 1,456 stale episodes the memory was perfect before the change ($q_0 = 0$): a failure can-

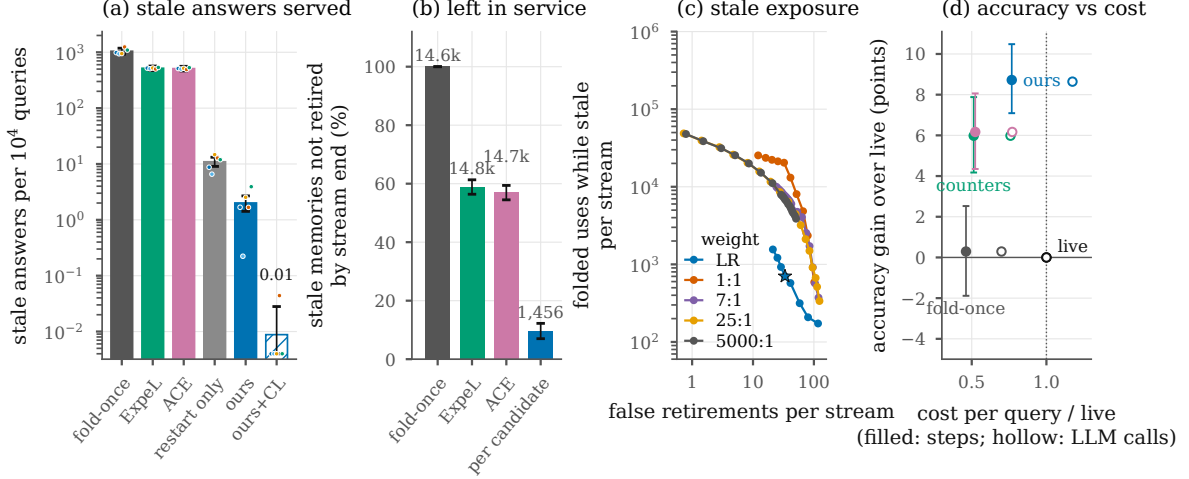


Figure 10: CodeStream (EvalPlus; five models, 150 streams of 10^6 queries). **(a)** Stale answers per 10^4 queries (step accounting; dots: models); ours+CL: ours with changelog invalidation. **(b)** Share of each policy’s own stale memories not retired by stream end (numbers: stale episodes). **(c)** Stale exposure at matched false retirements: folded uses while stale against false retirements, both per stream, for the per-memory LR-CUSUM (threshold sweep; star: $h = 7$) and fixed failure:success weights, all on the same certificate (weights 1:1–5000:1 were run; above 25:1 the curves coincide with 5000:1, the no-forgetting limit). **(d)** Accuracy gain over live reasoning (points) against cost per query relative to live reasoning, in steps (filled) and in LLM calls (hollow). Bars: 95% template-clustered bootstrap; in (d) on the filled markers only, since Δacc in LLM calls is within 0.1 points of it.

not occur before the change, so the optimal weight $1/\rho(q_0, q_1)$ of Equation (4) is infinite, and the best global weight moves to the no-forgetting limit. The other episodes want the opposite: $1/\rho = 0$ when $q_1 = 1$ (9%) and a median of 1.0 otherwise (5%). No single weight fits both groups, and the limit is still $10\times$ worse because a global rule also needs a global threshold. To spare the healthy memories that do fail now and then, it must wait for about 78 failures. The per-memory LR-CUSUM instead charges each failure $\ln(q_1/\hat{q}_0)$ of its own memory, which is large when the memory never failed on its shadow labels. Among retired episodes of perfect memories, it retires after a median of 30 stale uses, about three audits; the best fixed rule needs 874. This is also why the gap is far larger than on KQA-STREAM ($1.3\text{--}5.9\times$ in detection delay) and on natural edits ($1.1\text{--}1.6\times$), where few memories are perfect before the change. 42% of the episodes are partial ($q_1 < 0.95$; median $q_1 = 1$).

(c) Per-epoch false certification. An *epoch under test* is a (task, version) pair in which a candidate is under test, from its formation to its fold or discard; a *bad fold* folds a candidate whose reliability at that version is below break-even, the task-level event of Figure 3b. Ours has bad folds in 6 of 965,730 epochs at $\delta = 0.01$ ($6.2 \cdot 10^{-6}$),

12 of 946,678 at $\delta = 0.05$ ($1.3 \cdot 10^{-5}$; Clopper–Pearson upper bound $2.2 \cdot 10^{-5}$), 16 of 934,466 at $\delta = 0.1$ ($1.7 \cdot 10^{-5}$) and 21 of 917,079 at $\delta = 0.2$ ($2.3 \cdot 10^{-5}$): the rate rises with δ and stays more than $1,000\times$ below it. Per model, every rate is at least $250\times$ below δ ; the closest to δ is Mistral-7B’s at $\delta = 0.01$ (6 of 161,262). With changelog notices it is 0 of 941,085 at $\delta = 0.05$, and in call accounting 10 of 968,421. The counters fold a below-break-even candidate in 4.8% (ExpeL, 4,625/97,172) and 5.4% (ACE, 5,733/106,424) of their epochs under test, fold once in 3.7%; these policies fold at formation, so their epochs under test are formation epochs. The guarantee of Proposition 23(iv) does not cover the API edits, which change live reasoning silently; the measured rates show that, here too, this boundary is not where certification fails.

(d) Accuracy and cost. Live reasoning is right on 76.2% of queries (57.5–83.1% per model). Ours gains 8.7 points [7.1, 10.5]: 2.4 (Qwen3-4B), 3.1 (Qwen3-8B), 7.2 (OLMo-3-7B), 10.0 (Granite-3.3-8B) and 21.0 (Mistral-7B) per model, because a certified program has passed hidden inputs that the live agent’s visible tests do not cover. The cost depends on the unit. In *steps* ours costs 0.62 fewer per query [0.52, 0.71] (−23%) and lowers J_{20} by

Policy	Acc	Δacc (points)	step accounting			call accounting			Stale	Unretired
			steps/q	J_{20}	one-step	calls/q	J_{20}	one-step		
live always	.762	0	2.68	16.0	0	1.68	10.21	0	0	–
fold once	.765	+0.3 [−1.9, 2.5]	1.24	14.1	.94	1.17	9.35	.94	1,043	100%
ExpeL-style counter	.822	+6.0 [4.2, 7.9]	1.37	11.1	.90	1.27	7.44	.90	518	59%
ACE-style counter	.823	+6.2 [4.3, 8.1]	1.40	11.0	.89	1.29	7.40	.89	510	57%
per candidate (no notices)	.850	+8.9 [7.2, 10.6]	2.01	10.2	.63	1.95	7.19	.33	10.1	10%
restart only	.849	+8.7 [7.1, 10.4]	2.06	10.4	.61	1.97	7.26	.32	11.0	9%
ours (recommended)	.849	+8.7 [7.1, 10.5]	2.06	10.3	.61	1.97	7.27	.32	2.0	6%
ours + changelog	.849	+8.7 [7.0, 10.4]	2.08	10.4	.60	1.98	7.30	.32	0.01	–

Table 21: CodeStream, pooled over five models (150 streams of 10^6 queries). Accuracy and Δacc (paired, template-clustered 95% intervals) in step accounting; in call accounting every Δacc is within 0.1 points of these. One-step: share of queries answered by a folded memory. Stale: per 10^4 queries (step accounting). Unretired: share of the policy’s own stale episodes still in service at stream end (step accounting; 14,826, 14,704, 1,456, 1,328 and 96 episodes for the counters, per candidate, restart only and ours; fold once never retires). Bold: ours.

Model	LR-CUSUM ($h = 7$)			best fixed weight (1:1–5000:1)	
	FR	stale uses	episodes	weight (h)	$\times$ LR [95% CI]
Qwen3-8B	38.8	968	196	5000:1 (84)	7.6 [3.5, 21.7]
Qwen3-4B	29.8	656	234	5000:1 (84)	9.8 [5.8, 16.7]
Granite-3.3-8B	47.5	526	281	5000:1 (42)	8.2 [6.4, 10.9]
Mistral-7B	29.7	813	468	7:1 (84)	11.4 [9.3, 13.9]
OLMo-3-7B	21.8	550	277	5000:1 (46)	9.6 [7.5, 12.0]
pooled	33.5	702	1,456	5000:1 (78)	10.4 [7.7, 13.7]

Table 22: CodeStream claim (b), step accounting. FR: false retirements per stream; stale uses: folded uses while stale per stream; $\times$ LR: the fixed rule’s stale uses over the LR-CUSUM’s at no more false retirements. 5000:1 is the no-forgetting limit, tuned on a finer threshold grid; on the shared grid exposure is flat from 100:1 to 1000:1. Bold: ours.

5.3 [4.4, 6.3]; in *LLM calls* it costs 0.29 more per query [0.20, 0.39] (+17%), because shadow mode pays for live reasoning while a candidate is under test, and still lowers J_{20} by 2.6 [2.0, 3.3]. The certified policies have the highest accuracy and the lowest J_{20} in both units. Counters gain 6.0–6.2 points at a lower cost in both units (on Qwen3-8B and Qwen3-4B they are 0.4–0.8 points *below* live reasoning) while serving $47\times$ more stale answers than restart only; fold once is within noise of live reasoning (+0.3 points [−1.9, 2.5]).

Ties: the certification load on a real domain. The cost unit changes how much certified reuse the stream allows, and Theorem 6(ii) says why. Audited memory–live pairs of the certified policies agree on the outcome in 94–97% of cases (84–99% per model), and live reasoning re-derives the memory’s exact program in 49–59% (21–79% per model; 21–33% for Mistral-7B and OLMo-3-7B). In call accounting, a tie in which live reasoning succeeds at its first attempt has paired utility ex-

actly 0; this holds for 84–86% of audited pairs (67–89% per model). The utility law then has a large atom at zero, so $\text{KL}_{\text{inf}}(P_Z)$ is small and certification is slow, and one-step coverage drops to 0.32, against 0.61 in step accounting, where every tie is positive evidence because live reasoning costs at least one step more than the memory. For ours, the drop tracks the share of zero-utility pairs across models: Qwen3-8B and Qwen3-4B (88–89%) keep 34–36% of their step-accounting coverage, Granite-3.3-8B and OLMo-3-7B (86–87%) 52–53%, and Mistral-7B (68%) 81%. This is the certification load $\Theta \propto 1/(p\text{KL}_{\text{inf}})$ of Sections 4 and 5 observed on a real domain: an exact verifier and a strong live reasoner make audits nearly uninformative, and safe contraction must wait for the queries on which the two differ.

Scope. API edits break few programs: in the pilot, A1 breaks 3.7–4.9% of the distinct verified programs and the other seven edits together at most 7 programs per model, because MBPP and HumanEval programs mostly use builtins. They test the unannounced channel and changelog notices, but we claim no staleness magnitudes for real library drift; a benchmark with version-specific APIs would measure them. Requirement revisions are controlled changes of real tasks, as the fact edits of KQA-STREAM are.

E Worked examples from KQA-STREAM

This appendix follows real Qwen3-8B macros through the replays of Table 3 (main setting). Questions, unedited facts, LLM outputs, routing decisions and audit outcomes come from the logs; post-edit facts are counterfactual (Appendix D). Cases are named by edit sequence, stream seed and pol-

icy; t is the query index in the stream. Every replay used here was re-run with an instrumented recorder and matched the stored run exactly. The values r , a , m and r^* are exact popularity-weighted values over the instances routed to the template, with $L = 20\bar{m}$ ($\bar{m}$: the stream’s mean live calls). A *stale* answer is a folded answer that was correct at formation and is wrong now. “Betting” and “SPRT” denote betting + LR-CUSUM (per-candidate level δ , fold threshold $\ln 20$, no ledger; the “per candidate” row of Table 5) and SPRT + LR-CUSUM. These are mechanism examples: the main-text results use the restart-only ledger certificate and the recommended deployment, whose first threshold is $\ln 20 + \ln(\pi^2/6) = 3.49$ (so the 33-label fold of Table 23, at log-wealth 3.43, is a per-candidate fold).

E.1 One macro from formation to re-formation

One real macro passes through every stage of the lifecycle and illustrates Equation (1), Theorem 6, Corollary 2, and Proposition 9. Template T118 maps a university to the people living in its state (edit sequence 303, seed 6). It is the fifth most popular template in this stream ($\pi = 0.036$).

Formation and certification. The macro forms at $t = 44$ from the first audited T118 query (Table 23). At its second step the live agent outputs `{"action": "hop", "relation": "residence (backward)", "direction": "backward"}`. Both one-call baselines fail; the parametric one outputs `{"answers": ["State of Louisiana"]}`. At v_0 the macro is always correct, while live has $a = 0.800$ at $m = 2.85$ calls, so $r^* = 0.772$ with $L = 66.98$. The 26 concordant pairs add 0.010–0.019 each to $\ln W^+$; the 7 pairs where only live fails add 0.33–0.52 each. The SPRT plugs in $r_0 = 0.756$ and $r_1 = 0.788$ (true: 0.772 and 0.800) and needs 72 labels instead of 33.

Retirement and re-formation. Edit v_{16} lowers the macro’s reliability to $0.367 < r^* = 0.491$, while live re-derives answers on the edited facts. The LR-CUSUM was set at the fold to $q_0 = 0.0147$ and $q_1 = 1 - \hat{r}^* = 0.294$. It scores +2.995 per failure and −0.333 per success. One failure thus weighs about nine successes ($\rho = 0.111$ in Equation (4)), and three failures in four labels retire the macro. The SPRT retires it on the same labels at the same t . The re-formed candidate needs only 12

labels, 7 of them discordant, because live accuracy has fallen to 0.517.

Counters and plans. The ExpeL-style, ACE-style and fold-once policies do not retire the macro before the stream ends (last row of Table 23). Its 331 successful audits before the edit bank an ExpeL importance of 333, and Proposition 9(a) predicts a delay of $333/(2 \cdot 0.633 - 1) = 1,249$ labels; only 346 remain. The compiled plan, which re-executes the path on current facts, has no failure in 677 audited labels.

How typical is this case? Among betting’s 81 stale-after-edit onsets in the 30 streams, this one is at the median in folded uses to retirement (43); their median stale count is 30 (IQR 21–47). The counter side is extreme (98.5th percentile of ExpeL onsets): T118 is popular, and 106,806 queries remain. In seeds 1 and 2 the counters serve 358 and 273 stale answers after this edit, and betting 52 and 40.

Takeaway. Certification took 33 labels, and a likelihood-ratio detector limited a real edit to 29 stale answers. Counters with banked credit kept the stale macro to the end of the stream.

E.2 Counters under partial staleness

On one real sequence of audit outcomes, this example compares all retention rules and checks Proposition 9, Theorem 1, and Corollary 2. The T048 macro maps a film to Academy Awards via *producer* and *nominated for* (edit sequence 303, seed 0). It forms at $t = 738$ from the trace for “What Academy Awards did the producer of Man of Steel get nominated for?” and passes all 90 audits before the edit. Edit v_5 gives a new producer to 25% of the relevant subjects. Goodfellas’ producer Irwin Winkler, for example, becomes Eminem, yet the macro still answers Best Picture. Its reliability falls to $0.733 < r^* = 0.768$, so $q_1 = 0.267$: partial staleness. The 333 post-edit labels contain 93 failures.

Table 24 replays every rule on these labels. By Proposition 9(b), ExpeL (credit 92) and ACE (90/0) are ever retired with probability at most $10^{-40.4}$ and $10^{-39.5}$. The symmetric rule is in the negative-drift regime of Theorem 1 ($\mu_1 = -0.467$, $\gamma = 0.344$). At a similar A , the +7/−1 rule of Corollary 2 retires at label 44 instead of 152. Every realized delay lies between the 57th and 81st percentile of its exact law. On the longest real onset (T164, sequence 101, seed 5; 4,004 labels), the

t	Stage	Real evidence (Qwen3-8B, template T118, edit sequence 303, stream seed 6)	Statistic
44	form	Audited live trace of “What individuals have their residence in the state where Louisiana Tech University is situated?” (3 calls, 1,300/59 tokens): hop located in the administrative territorial entity (fwd) → Louisiana; hop residence (bwd) → Elmore Leonard; answer type human. Correct.	$r=1.00, r^*=0.772$
44–10,769	certify	33 audited shadow labels: 26 both correct ($U=+2$), 7 live wrong and macro correct ($U\approx+70$), 0 macro failures	$\ln W^+ : 0 \rightarrow 3.43 \geq \ln 20$
10,769	fold	routing call {“type”: 1} + lookup in a 272-anchor table frozen at v_0	LR-CUSUM $q_0=.015, q_1=.294, h=7$
→93,194	reuse	2,982 one-call answers before the edit, all correct (live: .798 on the same queries); 5,566 LLM calls (64.8%) saved over the macro’s 3,026 folded uses	
93,194	edit v_{16}	<i>located in the administrative territorial entity</i> , $f=0.6$ (451 facts); e.g. Cardiff University: Cardiff ⇒ Hong Kong, gold John Williams ⇒ Bruce Lee, Ip Man, Stephen Chow; the macro still answers John Williams	$r: 1.00 \rightarrow 0.367 < r^*=0.491$
93,227–94,247	retire	audited folded labels 0,1,0,0 (Univ. of Basel, Univ. of Windsor, East Tennessee State Univ., Cardiff Univ.); 1,053 queries, 43 folded uses and 29 stale answers after the edit	$S: 2.99, 2.66, 5.66, 8.65 > 7$
94,378–98,104	re-form	trace (valid at v_{16}) of “Who lives in the state where University of Freiburg is located?”; 12 labels (7 discordant); then 3,694 folded uses to the end, 0 wrong (0 of 334 audited labels fail)	$\ln W^+ = 3.00$
93,194–200k	other rules	ExpeL-style, ACE-style and fold once do not retire the macro before the stream ends: 3,863 folded uses and 2,427 stale answers after the edit. SPRT: fold at 20,978, then the same retirement (29 stale answers). Compiled plan + betting: 0 stale answers, 0 of 346 post-edit labels fail	ExpeL: 333 → 249 ACE: 331/0 → 462/215

Table 23: Lifecycle of one real macro under betting + LR-CUSUM, and the same macro under other rules (last row). t : query index in the 200k-query stream. Stale answers include the retiring query. Post-edit facts are counterfactual by construction.

counters serve 9,997 stale answers. There the symmetric rule with $H = 4$ needs 610 labels and the LR-CUSUM 8. Over the 30 streams, ExpeL has not retired 95.5% of its 287 partial-staleness onsets by the end of the stream, and ACE 80.8% of its 312. The bounds of Proposition 9(b), computed from each onset’s real credit, average 0.89 and 0.76. The LR-CUSUM detectors leave 1 of 24 (SPRT) and 0 of 11 (betting) unretired.

Takeaway. Under partial staleness, ± 1 counters with banked credit are practically never retired: the probability of ever retiring them is small (Proposition 9(b)), and on real streams they stay in service until the stream ends. A failure must weigh $1/\rho(q_0, q_1)$ successes, about 6.5 here.

E.3 Certifying utility instead of reliability

Two real candidates show why Theorem 6 bets on the paired utility and how a plug-in break-even misleads the SPRT (Proposition 5). The replays bet on $Z_t = U_t/(L + 7)$, clipped to $[-1, 1]$, with

$U_t = (m_t - 1) + L(Y_t^M - Y_t^L)$ and a uniform mixture of seven fixed bets. This differs from the grid mixture of Appendix C but keeps validity; mirrored bets (W^-) decide discards.

(a) Less accurate than live, yet worth folding.

Template T096 maps a record label to its time zone and has rank 2 in seed 9 of sequence 101 ($\pi = 0.109$). Its macro forms at $t = 100$ from the trace for “Which time zone is associated with Imperial Records?”. It follows the gold path, and no edit changes its answers. Still $r = 0.997 < a = 0.998$, because 21 misrouted instances (mass 0.003) count as macro failures. For example, “What time zone does the headquarters of Fox Searchlight Pictures observe?” belongs to T281, but the router outputs {“type”: 3} and picks T096. Live, which does not route, answers correctly, so the pair enters with $U_t = -65.3$. The macro is still above break-even ($r^* = 0.968$) and worth +1.93 steps per query. Betting folds it at $t = 22,459$ after 249 labels: 247

Rule	at edit	retires at label	uses / stale	$\mathbb{E}\tau$	A at in-control rate	
					$q_0=.05$	$q_0=.013$
ExpeL-style importance (± 1)	92	not by end (239)	3,283 / 894	∞	∞	∞
ACE-style helpful/harmful	90/0	not by end (330/93)	3,283 / 894	∞	∞	∞
symmetric, $H=2$ (SPRT + symmetric) [†]	0	19	169 / 47	17.8	420	6.2×10^3
symmetric, $H=4$	0	152	1,460 / 374	180.6	1.5×10^5	3.7×10^7
LR +7/-1, $H=43$	0	44	389 / 103	37.1	2.0×10^5	9.7×10^9
LR-CUSUM as run (SPRT), $h=7$ [†]	0	19	169 / 47	13.7	249	2.0×10^4

Table 24: Retention rules on the same 333 real post-edit labels of the T048 macro (edit sequence 303, seed 0). Uses / stale: folded uses before retirement / stale answers incl. the retiring query (to the stream’s end for “not by end”). $\mathbb{E}\tau$: exact mean delay (labels) at $q_1 = 0.267$. A : exact mean labels to a false retirement; .013 is the deployed LR-CUSUM’s in-control estimate. [†]A real replay retired the macro at label 19. Bold: ours.

#	t	instance (T010)	(Y^M, Y^L)	U_t	Z_t	$\ln W^+$	$\ln W^-$	plug-in LLR
1	3179	Q2656	(1,1)	+2	+0.027	.011	-.011	.096
11	27863	Q30002 Chengdu	(0,0)	+2	+0.027	.121	-.115	.910
17	34168	Q25416	(1,1)	+2	+0.027	.190	-.174	1.342
18	35098	Q48320 Kōbe (stale)	(0,1)	-64.4	-.877	-.290	.096	1.295
19	35271	Q1467 Cebu City (stale)	(0,1)	-64.4	-.877	-.599	.402	1.248
35	59956	Q1467	(0,1)	-64.4	-.877	-1.573	3.226 discard	1.781
94	187133	Q123709	(1,1)	-	-	-	-	3.007 fold

Table 25: Real certificates for the T010 candidate (edit sequence 202, seed 1). #: audited label, shared by betting and the SPRT; labels 1–11 precede edit v_7 . Threshold $\ln 20 = 2.996$ for $\ln W^+$ (fold), $\ln W^-$ (discard) and the plug-in LLR (fold).

pairs with both correct, one routing loss and one joint failure. No discordant pair favors the macro, so a sign test on discordant pairs would never fold it. Betting folds on the concordant gain $\eta/(1-\eta)$ of Appendix C. All three numbers below are for the *per-candidate* certificate at level δ (fold threshold $\ln(1/\delta) = \ln 20$, no ledger term), the version this example ran. Lemma 16 gives $\text{KL}_{\inf}(P_U) = 0.0244$, i.e. $\ln 20/\text{KL}_{\inf} = 123$ labels to first order (the *optimum* level of Section 5); the implemented seven-bet mixture adds $\ln 7$ nats to the threshold and grows at the rate $g_B = 0.0223$ of its best bet (0.9), which predicts $(\ln 20 + \ln 7)/g_B = 221$ (*as run*: the implementation loss; the overshoot term $\ln 1.9$ of Appendix C.2 is not included); the 249 labels it took are the *measured* level, 13% above the *as-run* prediction through finite-sample effects. The deployed ledger’s j -th test adds $2 \ln j + \ln(\pi^2/6)$ nats; for $j = 1$ the first two levels become $(\ln 20 + \ln \frac{\pi^2}{6})/\text{KL}_{\inf} = 143$ and $(\ln 20 + \ln 7 + \ln \frac{\pi^2}{6})/g_B = 244$ labels. The SPRT folds earlier ($t = 6,537$, 67 labels); its plug-in error is harmless here because the macro is good.

(b) A plug-in break-even folds a stale macro. The T010 candidate (sequence 202, seed 1) forms at $t = 651$ from “Which time zone is associated

with the administrative twin of Nottingham?”. Before the edit, $r = 0.965 > r^* = 0.947$: a thin margin that needs about 686 labels, while only 11 arrive. Edit v_7 then gives a new time zone to 25% of the relevant subjects. Seattle, for example, moves from Pacific Time Zone to East Africa Time. The cached macro drops to $r = 0.597$, while live stays at $a = 0.977$. Betting sees 9 stale pairs with $U_t = -64.4$ and discards the candidate at $t = 59,956$ (Table 25). The SPRT froze $\hat{a} = 0.345$ from one shrunk key label, so $r_0 = 0.313$ and $r_1 = 0.345$. Its zero-drift point, 0.329, lies below the true r in every phase. As in the proof of Proposition 18, it therefore folds almost surely. It folds at $t = 187,133$, 0.51 below break-even, and serves 45 stale answers. Across the 30 streams, the SPRT makes 21 below-break-even folds and betting 6 (SpF in Tables 3 and 5). In all 21 SPRT cases the plug-in r_0 lay below the true r^* , and betting folded none of those candidates.

Takeaway. Pair the macro with live on the same instance. Routing errors and live mistakes then enter the certificate directly, and a need not be estimated.

Template (rank)	π_i	audits in stream	n_i^*	$n_i^*/(p\pi_i)$ (queries)	epoch (queries)	$e^{-\Theta_i/\pi_i}$ predicted	one-step observed	labels to fold SPRT / betting
T132 (1)	.258	5,042	642	24.9k	200k	.883	.994	19 / 9
T048 (8)	.0214	430	210	98.2k	100k ^a	.375	.738	38 / 106
T027 (10)	.0165	323	213	129k	200k	.524	.906	38 / 106
T110 (73)	.00154	31	213	1.38M	100k ^b	$< 10^{-5}$	0	not in stream (8 of 39 ^b) / not in stream
T115 (204)	.00043	6	208	4.87M	200k	$< 10^{-10}$	0	not in stream (5 of 38) / not in stream
T014 (294)	.00025	3	171	6.92M	200k	$< 10^{-15}$	0	not in stream (2 of 35) / not in stream

Table 26: Six templates of one stream (edit sequence 101, seed 0). π_i : share of foldable queries. Epoch: $1/(\nu s_i + 1/T)$, s_i : fraction of harmful edits. One-step: share of the template’s queries answered by a folded macro, stale ones included. Bold: ours. ^aEdit v_{19} ($t = 149,426$); retired at $t = 161,145$, re-certified after 77 labels at $t = 194,254$. ^bEdit v_7 ($t = 55,476$) drops the candidate’s reliability from 1.0 to .733 after 8 of 39 labels.

E.4 Why rare templates stay uncovered

Six templates of one stream show the mechanism of Theorem 3: a template is covered only if its epoch outlasts $n_i^*/(p\pi_i)$. We use edit sequence 101 and seed 0, with SPRT certification. T048, T027, T110 and T115 have matched difficulty: live solves 59 of 60 instances, the gold-path macro has $r = 1.000$, and n_i^* is 208–213. Only their popularity ranks differ (Table 26).

The head certifies within 38 labels; the tail never does. T110 forms at $t = 5,301$ from a correct trace for “What record labels are connected to graduates of Radcliffe College?”. It gets 8 of the 39 labels it needs before edit v_7 lowers its reliability to $0.733 < r^* = 0.952$. T115 and T014 get 5 and 2 labels in the whole stream.

In the 30 main-setting streams, the SPRT needs a median of 39 labels per fold (all 1,201 folds), fewer than the first-order n_i^* (208–213 for the templates above), which assumes the least favorable candidate. Ranks 101–300 get 7.5–14.5 audits per template-stream on average, and none of their 5,771 template-streams was ever folded.

Takeaway. With a fixed audit budget, templates rarer than the load are not certified between changes; coverage comes from popular templates.

F Extended related work

Reasoning memory and skill libraries. LLM systems increasingly store reusable reasoning: thought templates (Yang et al., 2024a), strategies and insights (Suzgun et al., 2026; Zhao et al., 2024; Ouyang et al., 2026; Zhang et al., 2026b; Didolkar et al., 2025), workflows and programmatic skills (Wang et al., 2025b,a, 2024a,c), verified lemmas (Wang et al., 2024b), and long-term memories (Zhong et al., 2024; Park et al., 2023; Chhikara et al., 2025; Li et al., 2025). Many never

retire an item (Yang et al., 2024a; Ouyang et al., 2026; Wang et al., 2025b). The rest use hand-set rules. ExpeL (Zhao et al., 2024) keeps an integer importance count that starts at 2, moves by ± 1 on up- and down-votes and deletes the item at 0. ACE (Zhang et al., 2026b) keeps unbounded helpful/harmful tallies. Other rules include clipped Hebbian strengths with time decay (Singh and Yu, 2025; Zhu et al., 2026), lifetime failure rates thresholded at $1/2$ (Chen et al., 2026), EWMA utilities (Zhang et al., 2026c), Beta counts (Forouzandeh et al., 2026), usage-frequency pruning (Wang et al., 2024c), time-to-live (Li et al., 2025), sliding-window utilities (Pu et al., 2026) and paired with/without-memory audits (Zhang et al., 2026e; Gao et al., 2026). MQuAKE (Zhong et al., 2023), StreamingQA and RealTime QA (Liška et al., 2022; Kasai et al., 2023) evaluate a model after knowledge edits; KQA-STREAM evaluates a memory lifecycle over a stream with recurring templates and interleaved edits. The statistical analyses we know of either prove stationary consistency of a two-counter “memory worth” (Simsek, 2026) or prove that an EMA converges (Zhang et al., 2026c). Sun et al. (2025) recast memory management in decision-theoretic terms but give no tests. Asadolahi et al. (2026) show that self-graded memory scores are inflated unless verifier errors are independent. Sequential tests enter LLM pipelines only to audit trajectories (Sadhuka et al., 2025) or to stop sampling within one query (Lee et al., 2025). *We show that these maintenance rules are change detectors: clipped counters are Page CUSUMs with mis-specified scores, and unclipped tallies have no renewal. We quantify their detection delay and replace them with certified folding and likelihood-ratio retirement.*

Step skipping versus system-level contraction.

Model-level methods train an LLM to skip or internalize steps (Liu et al., 2024; Yu et al., 2024; Hao et al., 2025), to compress chains of thought (Xia et al., 2025), or to decide when to think (Zhang et al., 2025b); see Sui et al. (2025) for a survey. These methods change what one forward pass can compute. They choose training data offline. At inference time nothing decides whether a template’s one-step answer can be trusted, or notices when it has gone stale. The problem is real. Latent multi-hop composition is unreliable (Yang et al., 2024b), and consensus-based early exit is unsafe (Mo et al., 2026). *Contraction is complementary. It treats any distilled routine as a macro and decides per template, over time and with explicit error control, when to serve that macro and when to retire it.*

Caching, routing and cascades with guarantees.

Semantic caches reuse responses for similar prompts (Bang, 2023), and agent systems cache plan templates (Zhang et al., 2025c). Caching theory gives optimal content under stationary demand (Zhu et al., 2023; Liu et al., 2026). Cascades and routers decide per query (Chen et al., 2024a; Aggarwal et al., 2024). Where these systems carry guarantees, the guarantees are fixed-sample risk control under exchangeability. Examples are CALM (Schuster et al., 2022), which uses Learn-then-Test (Angelopoulos et al., 2025), PAC and group-conditional reasoning (Zeng et al., 2025; Huang et al., 2026), and conformal early stopping (Wang et al., 2026). Two works are closer.

- B-PAC (Yu et al., 2026b) certifies a global thinking/non-thinking router anytime, with inverse-propensity-weighted test supermartingales. Our certificate uses the same tools (betting, inverse-propensity weights, Ville’s inequality); what differs is the object: one certificate per template, restarted at changes and followed by retirement, over a memory’s whole lifecycle.
- vCache (Schroeder et al., 2026) bounds the per-request error of cache hits using randomized LLM verification. Its bound assumes i.i.d. prompts and a correct similarity model, and exploration can stop entirely.

Staleness has been studied only empirically (Mansoor et al., 2026; Wu and Canedo, 2026), although model behavior does drift (Chen et al., 2024b).

Anytime-valid monitors watch whole models, not reusable items (Podkopaev and Ramdas, 2022; Richter et al., 2025; Khosravi and Huo, 2026). *We certify a reusable derivation with a test that stays valid at all times, under adaptive shadowing and arbitrary drift below the null. The indifference zone comes from the loss. Certification is followed by quickest-detection retirement, and the result yields population-level coverage laws.*

Sequential testing and change detection. Certified folding rests on the SPRT (Wald, 1945; Wald and Wolfowitz, 1948), Ville’s inequality (Ville, 1939), and testing by betting (Shafer, 2021; Ramdas et al., 2023; Howard et al., 2021; Waudby-Smith and Ramdas, 2024). Retirement rests on Page’s CUSUM (Page, 1954) and its optimality (Lorden, 1971; Moustakides, 1986; Lai, 1998). Two of our results are instances of older work. The counter identity matches cumulative-score and integer CUSUM charts (Munford, 1980; Brook and Evans, 1972). Our likelihood-ratio learning rates are the Bernoulli CUSUM (Reynolds and Stoumbos, 1999). Analyses of mis-specified CUSUM assume positive post-change drift (Molloy and Ford, 2017; Cooper and Meyn, 2024), or note only qualitatively that delay can exceed the false-alarm time (Molloy, 2021). Wu and Yang (2016, Thm. 2) state that the delay is infinite in the negative-drift regime; their proof exchanges an expectation and a minimum over the CUSUM’s restarts. Observation-controlled detection (Banerjee and Veeravalli, 2013; Xu et al., 2021) and e-detectors (Shin et al., 2024) are the natural comparisons for audited and composite settings. *Using Cramér–Lundberg asymptotics (Iglehart, 1972; Siegmund, 1985; Asmussen, 2003), we make explicit the full delay–ARL phase diagram for additive-score detectors. It includes the power law $D \asymp A^{\theta_1/\theta_0}$ under negative drift, where Wu and Yang (2016) state an infinite delay; Appendix A locates the step of their proof that fails and gives a counterexample.*

Occupancy, freshness and scaling laws. Our static amortization law is a corollary of occupancy asymptotics (Karlin, 1967; Gnedin et al., 2007). Its constant follows from rare-mass results (Ohannesian and Dahleh, 2012), and Good–Turing theory estimates and forecasts it (Good, 1953; Good and Toulmin, 1956; McAllester and Schapire, 2000; Orlitsky et al., 2016). For $n_0 = 1$ it is the cold-start or characteristic-time miss ratio of caching (Fagin, 1977; Che et al., 2002; Jelenković, 1999).

Admission after k hits generalizes this case (Martina et al., 2014), and memorization learning curves share the exponent (Hutter, 2021; Michaud et al., 2023). Under drift, the $n_0 = 1$ steady state is the freshness objective of crawling (Azar et al., 2018) and the hit law of TTL and random-replacement caches (Choungmo Fofack et al., 2014; Gallo et al., 2014). Square-root refresh and replication rules are classical (Yates et al., 2017; Cohen and Shenker, 2002; Kolobov et al., 2019), and so is abandoning fast-changing items (Cho and Garcia-Molina, 2003; Bastopcu and Ulukus, 2021). Square-root allocations also appear in label-efficient inference (Xu et al., 2024; Zrnic and Candès, 2024). The law of practice has been explained by chunk recurrence and by averaging over items (Newell and Rosenbloom, 1981; Logan, 1988; Heathcote et al., 2000). *We couple sequential-test sample complexity with occupancy. This gives the fundamental limit of safe one-step coverage under drift, $M^* \sim \kappa_s \Theta^{1-1/s}$, with a first-order matching converse for every δ -safe policy.*

Safe self-improvement and multi-agent memory. Self-modifying agents accept a change when a benchmark score rises (Zhang et al., 2026a). Per-decision betting gates (Shawn, 2026) and trajectory-level error spending (Wu et al., 2025) control false commits. Earlier antecedents are high-confidence and Seldonian improvement (Thomas et al., 2015, 2019; Chandak et al., 2020) and the “bio-creep” study of approval policies (Feng et al., 2021). Recent work proves limits for classifier-style gates (Scrivens, 2026) and near the reward ceiling (Cai et al., 2026). Shared multi-agent memories admit items through judges, one-shot A/B replay or count-and-mean rules (Zhang et al., 2025a; Xu et al., 2025; Yu et al., 2026a). Yet LLM errors are correlated (Kim et al., 2025), judges are noisy (Zhang et al., 2026d), and shared memory is a poisoning channel (Dong et al., 2025). *These gates decide one modification or one shared item at a time. We study a single memory whose items are certified per template from verifier feedback, and leave shared and self-modifying memories to future work.*

Speed-up learning and the utility problem. Symbolic AI compiled multi-step reasoning into macro-operators (Fikes et al., 1972; Korf, 1985), explanation-based rules (Mitchell et al., 1986), chunks (Laird et al., 1986) and compiled productions (Anderson et al., 2004; Taatgen and Lee, 2003). Program synthesis grows libraries in the

same way (Ellis et al., 2021). The utility problem (Minton, 1990) led to statistically gated adoption (Gratch and DeJong, 1992; Greiner, 1996) and PAC analyses of speed-up learning (Tadepalli and Natarajan, 1996). These analyses assume a fixed problem distribution and never un-adopt a rule. *We add four things: validity at all times under adaptive schedules and drift, an indifference zone derived from the loss, retirement with optimal detection delay, and closed-form population laws for how much reasoning can be contracted.*