REGIMECOUNCIL: PAIRED-EVIDENCE ROUTING FOR MEMORY CONFLICT GOVERNANCE IN LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language agents increasingly rely on long-term memory to sustain extended interactions with their users, yet stored facts become stale as the world changes, and these agents often learn that a fact was wrong only through delayed feedback, after already acting on that fact. A central question is how far a single correction should propagate: should it govern only later uses of the same fact, or also how much the agent trusts memory on other facts? Existing systems typically adopt one of these two policies at design time. We formalize this choice as a sequential decision problem—*memory conflict governance*—in which memory validity is hidden at decision time and outcomes are revealed with delay, and prove that neither policy is uniformly sufficient. We then propose **RegimeCouncil**, which retains both policies and selects between them using delayed paired outcomes, without retraining or assuming a fixed dominance order. Two advisors score the advantage of using memory over a no-memory anchor: Global pools evidence across stored facts, while Local uses only the queried fact. A Meta Governor defaults to Local and switches to Global only when Global sustains a positive recent margin; under delayed feedback, this switch is provably reached within a bounded number of decisions. On delayed-feedback streams, RegimeCouncil outperforms the better fixed-scale baseline by 5.2–7.3 points on disagreement episodes. The same frozen router transfers to naturally occurring drift on TempLAMA and TAQA. Ablations confirm that every component is important: removing each of Global, Local, and the Meta Governor lowers accuracy by 2–7 points.

1 INTRODUCTION

Language agents increasingly rely on long-term memory to sustain extended interactions with their users. Yet the world keeps changing, and stored facts become stale—often while the agent still treats them as valid. Suppose an agent records "the user lives in Beijing" and answers many queries from that record. If the user later mentions, perhaps in passing, that they have moved to Shanghai, the correction arrives after those answers and is not written into the next prompt. The next question is still "Where do I live?": the stored write says Beijing and still matches the query. Once the correction is known, updating that record itself is straightforward. The open question is how far this correction should propagate: should it affect only later uses of the same fact, or also how memory is used for other facts (Figure 1)? Evidence for an unchanged fact should remain local to that fact; a cluster of corrections across facts may justify adjusting a shared memory-use threshold; and a previously unseen fact with no outcome of its own must rely on pooled evidence for its prior.

This situation arises routinely in deployed agents. Personal assistants store addresses, schedules, preferences, and relationships that change over months, while support or enterprise agents store product and policy facts that are revised without notice. Knowledge updates are accordingly one of the core abilities evaluated by long-term memory benchmarks (Wu et al., 2025). In practice, the correction often arrives later, through a user complaint, a failed task, or an explicit edit, after the outdated answer has already been given. An agent that misjudges how far each correction should reach either keeps serving outdated facts or withholds memory that is still valid.

LLM memory systems have made substantial progress on storage, retrieval, retention, and context management (Packer et al., 2023; Chhikara et al., 2025; Zhou et al., 2026; Zhang et al., 2026; Kang et al., 2026b;a; Maharana et al., 2024; Wu et al., 2025). These systems decide *what* to store and *when*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

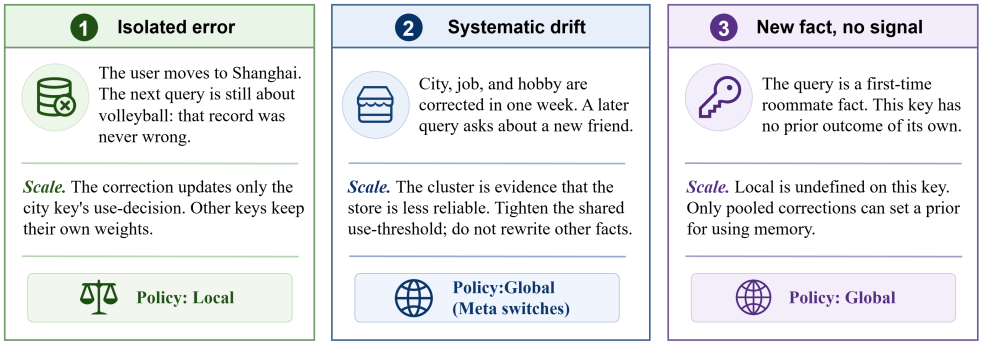

Figure 1: Three regimes of *influence scale*, i.e., the support of a correction's update: (1) a single key; (2) a store-wide use-threshold after a cluster of corrections; and (3) a cold-start prior when the queried key has no history.

to retrieve, but fix in advance whether one key's outcome may change memory use on other keys. We index each stored fact by a *key*—`user_city`, `volleyball`, `job`—so a delayed outcome can stay on the queried record or be pooled across the store.

Existing systems typically fix one of these policies at design time. A shared recency window lets one key's outcome change whether memory is used for other keys (a global policy); a per-key rule reuses only the latest memory-use outcome on the queried key (a local policy). The global policy mixes keys that did not change together; the local policy provides no historical signal for records that are new or infrequently queried. Neither scale is uniformly better over time. We term the online choice between these two policies *memory conflict governance* and formalize it as a sequential decision problem in which memory validity is hidden at decision time and outcomes are revealed after a bounded delay.

We propose **RegimeCouncil**, a paired-evidence router that treats global and local decision rules as complementary advisors. Since neither scale dominates, it maintains both and uses a governor to switch between them based on recent paired outcomes. Concretely, a Global Advisor pools recent advantages across keys, while a Local Advisor retains the latest advantage for the current key; both score the episode-level advantage of using memory over a no-memory anchor. A Meta Governor compares the utility each advisor would have obtained on completed episodes, keeps Local as the conservative default, and grants a global override only after global sharing has accumulated a strictly positive recent surplus over a sliding window. Three theoretical results support this design. First, when validity is hidden, stationary observation-only policies incur a non-zero regret floor, whereas a delayed-replay history policy provably escapes it on revisited keys, with average regret decaying as $O(1/T)$ in the horizon T (Proposition 1). Second, neither a globally pooled window nor a per-key window is uniformly sufficient—each fails on a distinct linear-loss construction (Proposition 2). Third, a windowed surplus test over the two advisors yields finite-time switching under delay (Proposition 3).

Our contributions are as follows. (1) We formalize *memory conflict governance*: when memory validity is hidden at decision time, should delayed feedback update only the touched key or also govern how memory is used store-wide? (2) We propose RegimeCouncil, a paired-evidence router between global and key-local advisors, and prove the three theoretical results above. (3) On the disagreement slice across two held-out schedules and two LLMs (Qwen3-30B-A3B and GPT-4o), RegimeCouncil outperforms the better fixed-scale baseline by 5.2–7.3 points. Ablations confirm that every component is important: removing Global, Local, or the Meta Governor lowers accuracy by 2–7 points. The frozen router also generalizes to naturally occurring drift on TempLAMA and TAQA.

2 Related Work

RegimeCouncil lies at the intersection of long-term agent memory, non-stationary online decision making, and learning under partial observability. Its core focus is the *scale of feedback sharing*: it selects between pooled and key-specific evidence using delayed paired outcomes.

Long-term memory and temporal conflict. Memory-augmented agents use explicit memory tiers, retrieval, and reflection to extend interaction beyond a model's context window (Park et al., 2023; Packer et al., 2023; Chhikara et al., 2025). Learned controllers compress working memory or reweight episodic items from interaction feedback (Zhou et al., 2026; Zhang et al., 2026); constrained retention and budgeted retain/consolidate operators decide what to keep under observability and token limits (Kang et al., 2026b;a). Benchmarks such as LoCoMo and LongMemEval evaluate long-range recall, temporal reasoning, and knowledge updates over multi-session conversations (Maharana et al., 2024; Wu et al., 2025); MemoryAgentBench evaluates four core memory competencies—accurate retrieval, test-time learning, long-range understanding, and conflict resolution—through incremental interactions (Hu et al., 2026). The question they leave fixed is the one we study: after a correction is observed, should it also change memory-use decisions on other keys, or stay on the queried one?

Knowledge-update tests expose a common failure mode of persistent memory. However, they typically present competing statements and their temporal order directly in the decision context (Wu et al., 2025). In that observable setting, a recency-aware retriever can resolve the conflict directly. We instead decouple evidence acquisition from the decision it is meant to supervise by hiding the validity bit until delayed feedback arrives.

Non-stationarity and evidence scale. Concept drift methods adapt predictors as reward distributions change; non-stationary bandits track evolving optimal actions over time (Gama et al., 2014; Besbes et al., 2014; Cheung et al., 2019). Strongly adaptive online learning targets performance on every time interval (Daniely et al., 2015; Jun et al., 2017). Our setting has a structured source of non-stationarity: keys change asynchronously, so an estimator's sharing granularity determines whether observations transfer or interfere. A shared recency window and per-key reuse of the latest use-outcome are the two fixed rules for this choice. RegimeCouncil treats the two rules as counterfactually evaluable advisors and selects between them from paired outcomes, using the delayed record of which scale was better on completed episodes.

Expert aggregation and partial observability. Prediction with expert advice supplies a general framework for combining decision rules; tracking methods allow the best expert to change over time (Herbster & Warmuth, 1998; Daniely et al., 2015). RegimeCouncil differs in how the experts are constructed and evaluated. Global and Local are two granularities of the same paired memory advantage; both are scored counterfactually on every completed episode, and the asymmetric gate switches from Local to Global only after pooled evidence accumulates a strictly positive recent surplus. The design exploits full paired feedback rather than treating advisor choice as an additional bandit problem. Partially Observable Markov Decision Process (POMDP) theory formalizes the related difficulty of acting when latent state is not recoverable from the current observation (Krishnamurthy et al., 2016; Jin et al., 2020). In RegimeCouncil, history enters only as delayed evidence about whether using memory was the better action.

3 METHOD

RegimeCouncil resolves the scale of feedback sharing with one paired-evidence primitive and three modules: two complementary advisors, Global and Local, arbitrated by a Meta Governor. The design is supported by three results, proved in Appendix A: a map of the current observation cannot resolve hidden validity while a delayed-replay history policy provably can on revisited keys; neither a purely global nor a purely local window is uniformly sufficient; and a surplus test on paired replay switches in finite time even under delayed feedback. Figure 2 summarizes the pipeline.

3.1 PROBLEM FORMULATION

At step t the agent observes $z_t \in \mathcal{Z}$, takes a memory action $a_t \in \mathcal{A}$, and after a delay of at most τ receives utility $u_t \in [0, 1]$. A latent validity state r_t (the *regime*) is not included in z_t. The history available when a_t is chosen is $h_t = (z_{1:t}, a_{1:t-1}, \mathcal{F}_t)$, where $\mathcal{F}_t$ is the feedback revealed before step t; it contains the outcome of every step $s \leq t - \tau$ and may contain later ones. The objective is expected cumulative utility $J_T(P) = \mathbb{E}_P[\sum_{t=1}^{T} u_t]$.

As a running example, the stored fact is "the user lives in Beijing," and the user has since moved to Shanghai. At the next decision the retrieved fact still matches the query, while the correct memory

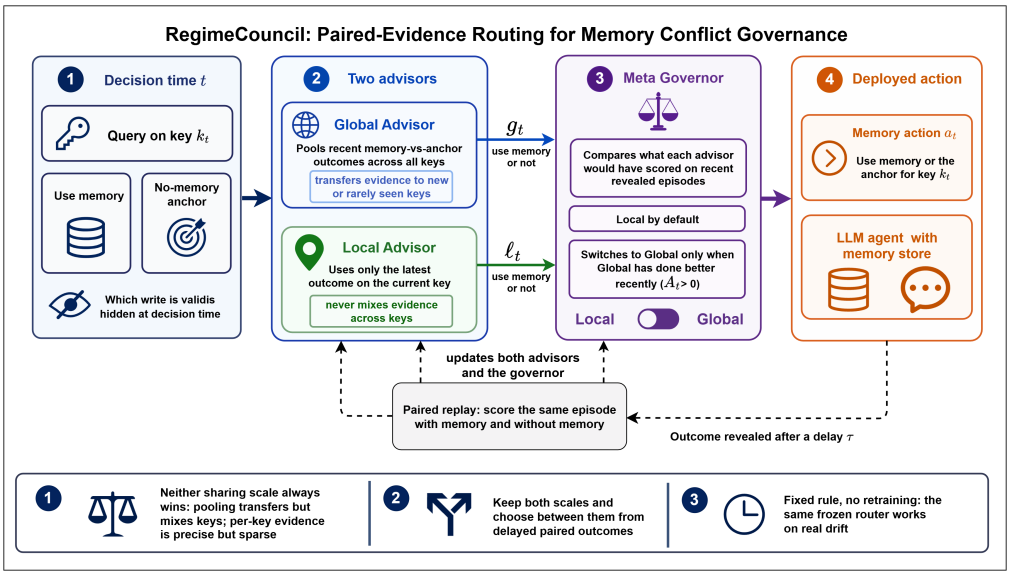

Figure 2: RegimeCouncil. (1) At decision time t the query concerns key k_t; which write is valid is hidden, and the choice is between using memory and the no-memory anchor. (2) The Global Advisor pools recent memory-vs-anchor outcomes across all keys; the Local Advisor uses only the latest outcome on the current key. (3) The Meta Governor compares what each advisor would have scored on recent revealed episodes, keeps Local by default, and switches to Global only when $A_t > 0$. (4) The memory action a_t is deployed. Its outcome is revealed after a delay τ, and paired replay updates both advisors and the governor.

action has reversed; only the subsequent outcome identifies which write was valid. This is a *collision* together with delayed recoverability.

Assumption 1 (Decision-time unobservability)**.** *On colliding contexts, the regime is not a function of* z_t.

Assumption 2 (Hindsight recoverability)**.** *An outcome generated at step* s *is revealed by step* $s + \tau$. *The revealed comparison between using and withholding memory has the sign of the regime-optimal memory action on that episode.*

Assumption 3 (Replayable evaluation)**.** *Completed interactions can be scored under alternative memory actions (paired counterfactuals).*

These conditions describe the information pattern of persistent memory systems. Assumption 1 holds when the environment changes without an explicit update in the observation: a user who moves, changes a preference, or loses a job rarely records it, so the next query matches the stored item while the correct memory action has reversed—the identifying bit is absent from z_t, not merely difficult to extract. Assumption 2 is the supervision already available after the action—a rejected recommendation, a failed tool call, or a later correction—arriving within a bounded delay τ rather than instantaneously, as in cache and retrieval logs. Assumption 3 is the standard offline pairing used to compare memory on or off on a completed query: once the episode exists it can be scored under the alternative action without simulating a new trajectory, which the experiments instantiate by caching the LLM answer and varying only the memory decision. A log that stores only the executed action lacks the other score, so the surplus is defined only for logs that retain both. Assumptions 1–2 thus create the need for history; Assumption 3 makes paired routing measurable on existing logs.

Let $u(a; z, r) = \mathbb{E}[u_t \mid a_t = a, z_t = z, r_t = r]$ denote the mean utility and write $\rho_r(a; z) = \max_{a'} u(a'; z, r) - u(a; z, r) \geq 0$ for its regime-conditional suboptimality gap. Context z is Δ-colliding if it is visited under both regimes and $\min_a[\rho_0(a; z) + \rho_1(a; z)] \geq \Delta$. We treat the context–regime sequence (z_t, r_t) as fixed in advance (oblivious to the policy), so the visit counts and hence the parameters (p, q, Δ) do not depend on π: let p be the colliding-visit frequency and $q \in (0, \frac{1}{2}]$ the visit-weighted minority-regime share. The observation class $\Pi_{\text{obs}} = \{\pi : a_t \sim \pi(\cdot \mid z_t)\}$ contains

stationary maps of the current observation; the history class is $\Pi_{\text{hist}} = \{\pi : a_t \sim \pi(\cdot \mid h_t)\}$. Regret is measured against a regime-aware oracle on (z_t, r_t), in expectation over the policy's randomization.

The first structural fact is that no stationary observation-only policy can resolve collisions: every action that is good under one regime is costly under the other, and a map of z_t cannot condition on the regime.

Proposition 1 (Observation floor and history separation)**.** (i) Floor. *Under Assumption 1 and collisions with parameters (p, q, Δ), every $\pi \in \Pi_{\text{obs}}$ satisfies $\frac{1}{T}\text{Regret}_T(\pi) \geq \delta := pq\Delta$.* (ii) Separation. *If additionally the per-key regime is piecewise constant with at most S switches and the delay is at most τ (Assumption 2), the history policy that replays the last revealed per-key outcome (the Local Advisor, §3.4) satisfies $\frac{1}{T}\text{Regret}_T \leq (S+K)m/T$, with K keys and m the largest per-key visit count in any τ-window; thus Π_{hist} separates from Π_{obs} once $T > (S+K)m/\delta$.*

Corollary 1 (Capacity does not remove missing bits)**.** *No stationary feature map $f(z)$ followed by an arbitrary classifier reduces the floor below δ when the regime bit is absent from z.*

Part (i) restricts *what the decision may condition on*, not model size: any stationary map of z_t, however expressive, still pays δ. Part (ii) shows that replaying the last revealed per-key outcome escapes the floor when that key is revisited after the delay. When a key is not revisited, the bound does not apply; pooled history is the complementary case in Proposition 2 (Appendix A.1).

3.2 Paired evidence

The router does not score raw utilities in isolation. We index each stored fact by a key k; the query at time t concerns k_t. When that interaction has closed and its (possibly delayed) outcome is revealed, Assumption 3 lets us score the same episode twice—once as it ran and once under the alternative memory action—yielding the paired difference

$$d_t \;=\; u_{t,\text{mem}} - u_{t,0} \;\in\; [-1, 1],$$

where $u_{t,\text{mem}}$ and $u_{t,0}$ are the utilities with memory used and with memory withheld—the no-memory anchor—on the identical context. Pairing removes context difficulty from the signal: a positive d_t means memory was the correct action on that episode, independent of query difficulty. Every module below is a different *window* over the stream of d_t values, so the modules can disagree yet remain comparable.

3.3 Global Advisor

The Global Advisor transfers evidence across keys. Early on, and on rarely revisited keys, the current key has no outcome of its own, but other keys may already have revealed whether using memory improved utility. The advisor therefore maintains a rolling window $\mathcal{W}_g(t)$ of the last W_g paired differences *across all keys* and proposes to use memory whenever their sum is positive:

$$g_t = \mathbb{1}\left[\sum_{i \in \mathcal{W}_g(t)} d_i > 0\right],$$

where $\mathbb{1}[\cdot]$ is the indicator function, returning 1 if the condition holds and 0 otherwise. A positive pooled sum transfers to a fresh key immediately. This is the only module that can propose using memory before the current key has ever been seen (Figure 1, scene 3), and it is the right scale when keys share a validity regime.

Pooling is not sufficient by itself. When keys flip asynchronously, opposite-sign advantages occupy the same window: they cancel in the sum, and the pooled sign follows the majority key. The architecture therefore retains a second, unmixed scale.

3.4 Local Advisor

The Local Advisor is the complementary, unmixed statistic. It reads only the last W_ℓ paired differences for the *current* key,

$$\ell_t = \mathbb{1}\left[\sum_{i \in \mathcal{W}_\ell(t; k_t)} d_i > 0\right],$$

and therefore never mixes evidence across keys. When the next outcome for that key arrives, Local follows it: the proposal then matches the regime now in force on the key being decided (Figure 1, scene 1). We use W_ℓ=1: because each key's regime is piecewise constant, the most recently revealed outcome is the one most likely to match the current regime, and a longer window would only dilute it with older, possibly stale evidence.

On a key that has not been revisited, Local has no revealed outcome and defaults to the anchor. Global is the scale that still has a signal in that case.

Proposition 2 (Neither fixed scale is uniformly sufficient)**.** *There exist asynchronous streams on which the Global Advisor suffers linear loss relative to the Local Advisor: opposite-sign per-key advantages are dominated inside the pooled window by the majority key, so the pooled sign follows it and errs on every minority visit. There also exist sparse-key streams on which the Local Advisor suffers linear loss relative to the Global Advisor: a key is not revisited before its local evidence becomes useful, so Local abstains while pooled transfer is already informative.*

The two constructions are complementary rather than competing (Appendix A.2). Pooling is necessary for transfer and fails under asynchrony; isolation is necessary for fidelity and fails under sparsity. This is why RegimeCouncil retains both advisors instead of committing to one scale a priori. Proposition 2 does not specify how to choose between them from z_t; that choice cannot observe the hidden regime bit, so it must be made from delayed paired replay.

3.5 Meta Governor

The two advisors disagree precisely on the decisions where the scale choice is most consequential. The Meta Governor is the arbiter, but it cannot observe the hidden regime bit: it must evaluate the advisors by their record on completed episodes, not by the current observation. A cluster of corrections across keys is exactly this case (Figure 1, scene 2): an isolated correction stays on its key, while a persistent global surplus is the signal to share the *use-threshold*—not to rewrite other facts.

For a completed episode, had advisor $j \in \{g, \ell\}$ been in control, its realized utility would have been

$$v_{j,t} = \begin{cases} u_{t,\mathrm{mem}} & \text{if advisor } j \text{ proposed memory,} \\ u_{t,0} & \text{otherwise.} \end{cases}$$

Each closed episode therefore yields a direct comparison of the two scales without additional labels. The governor accumulates their surplus over the most recent $W_m \geq 1$ episodes whose outcomes have been revealed by time t,

$$A_t \;=\; \sum_{i \in \mathcal{W}_m(t)} \big(v_{g,i} - v_{\ell,i}\big),$$

where $\mathcal{W}_m(t)$ indexes those W_m revealed episodes. Only revealed outcomes enter the sum, so A_t is well defined despite the feedback delay. The governor then applies an asymmetric rule: Local is the default, and Global takes control only when $A_t > 0$, so sharing must accumulate recent measured surplus before overriding key-specific governance. The asymmetry limits unstable switching of the pooled statistic under fast alternation while still letting Global override when pooling is persistently better—the two regimes of Proposition 2.

Proposition 3 (Finite meta-governance response)**.** *If $v_{g,t} - v_{\ell,t} \geq \gamma > 0$ for W_m consecutive revealed outcomes, the governor selects the Global Advisor. If the difference is at most $-\gamma$ for W_m consecutive revealed outcomes, it selects the Local Advisor. With feedback delay at most τ, either response occurs within $W_m + \tau$ decisions of the streak's onset.*

The bound counts decisions until the window has filled (Appendix A.3): W_m revealed outcomes replace the window, and delay adds at most τ decisions. Because pairing makes the two scales comparable on the same episodes, the scalar surplus A_t is enough to select between them.

3.6 The complete router

At decision time RegimeCouncil deploys memory for key k_t according to

$$a_t \;=\; \begin{cases} g_t & \text{if } A_t > 0 \quad \text{(Global override)}, \\ \ell_t & \text{otherwise} \quad \text{(Local default)}. \end{cases}$$

Table 1: Main result on the two primary held-out schedules, disagreement-slice accuracy (%). RC denotes RegimeCouncil; G and L denote the Global and Local Advisors used alone. *Best G/L* is $\max(G, L)$, and *Gap* is RC minus Best G/L (points). *Agree-only* removes the Meta Governor (§4.3).

Schedule	Model	n_{dis}	RC	Best G/L	Agree-only	Gap
A	Qwen	110	**87.3**	80.0	65.5	**+7.3**
	GPT-4o	134	**84.3**	79.1	67.2	**+5.2**
B	Qwen	151	**83.4**	78.1	76.2	**+5.3**
	GPT-4o	146	**82.9**	77.4	76.0	**+5.5**

Once the episode's delayed outcome arrives, the new paired difference d_t is appended to both windows and A_t is updated, closing the loop. All adaptation flows through paired outcomes rather than z_t, which is what lets the router escape the observation-class floor of Proposition 1. The three windows (W_g, W_ℓ, W_m) and the override gate are the entire parameterization; experiments use gate 0, i.e. $A_t > 0$. The finite-time bound of Proposition 3 holds for any $W_m \geq 1$. The advisors are windows over one statistic and the governor is a scalar comparison, so the router adds negligible cost beyond a single memory read.

4 Experiments

The primary test compares paired routing with the better fixed scale on steps where the two advisors disagree, the case identified by Proposition 2. We then test the same frozen router on naturally occurring drift and remove each component in turn. Window lengths and the gate are fixed on disjoint development schedules before the held-out schedules are scored, and LLM answers are cached once so only routing differs. We use GPT-4o and Qwen3-30B-A3B (an MoE model activating about 3B parameters; hereafter Qwen). In the tables, RC denotes RegimeCouncil. Reproduction details, prompts, and worked examples are in Appendices B–D.

4.1 Main: routing where the scales disagree

The stream is built from real LongMemEval knowledge-update content (Wu et al., 2025): we keep the original benchmark questions and their manually audited old and new answers and fabricate no facts. Unmodified items cannot test Assumption 1: the superseded and updated facts appear in z_t with timestamps, so a recency-aware reader recovers the current write—observable conflict, not memory conflict governance. We therefore apply one mechanical, reproducible transformation—stripping the temporal-pointer lexicon—and control only *when* each update becomes valid through a *schedule*, which fixes the order in which keys are visited and when each key's valid answer switches. We use the *block_switch* family: the 16 keys form two cohorts of eight whose valid answers switch in opposite directions near mid-horizon, and each key is visited 22 times, once per pass. Memory serves a snapshot of the latest writes that is committed every 96 episodes, so the stored answer can be stale. Schedule A and Schedule B are two held-out instances of this family with identical parameters. They differ in two hashed choices: the order of keys within each pass, and whether each key switches after its 10th, 11th, or 12th visit. The content is real; only the timing is a controlled abstraction, isolating the mechanism the method targets—validity hidden at decision time under delayed feedback.

Local uses W_ℓ=1 and Meta uses gate 0 by design; the development schedules fix $(W_g, W_m) = (32, 8)$. The primary metric is disagreement-slice accuracy: episodes with $g_t \neq \ell_t$ from the advisor bits alone (no labels), reported as $\text{RC} - \max(G, L)$, where G and L are the accuracies of the Global and Local Advisors used alone. Full-stream accuracy is secondary: when the advisors already agree, Meta has no decision to make.

Table 1 reports the two primary held-out schedules; RC exceeds the better fixed scale by at least 5 points on all four settings. The better fixed scale itself flips between the two schedules: Local is better on Schedule A and Global on Schedule B, on both models. Fixing either scale at design time therefore loses on one schedule, while RC exceeds both without retuning, the empirical counterpart of Proposition 2. A key-cluster bootstrap over the 16 keys gives $P(\widehat{\text{gap}}{>}0) \in [0.93, 0.97]$ (Appendix B.5). Episodes where the advisors agree contribute 0 to the full-stream gap, so that gap equals the disagreement fraction times the disagreement gap (e.g. $0.31 \times 7.3 \approx 2.3$ on Qwen, Schedule A).

Table 2: Disagreement-slice gap on the 54-key pool, per-key snapshot lag matched to the 16-key protocol (refresh=324). All eight key-bootstrap 95% intervals (5000 replicates) lie above zero. Gap is RC minus Best G/L (points). Windows stay frozen.

Schedule family	Model	Schedule	Gap	95% CI
block_switch	Qwen	A	+6.0	[0.6, 9.1]
		B	+8.4	[4.1, 10.0]
	GPT-4o	A	+8.5	[3.0, 11.5]
		B	+9.0	[3.7, 10.6]
independent_switch	Qwen	A	+8.6	[5.1, 10.1]
		B	+9.2	[5.7, 10.7]
	GPT-4o	A	+9.2	[4.6, 11.7]
		B	+9.7	[7.0, 12.4]

Table 3: Ablations on Schedule A (full-stream accuracy, points). The router is frozen: no window, gate, or threshold is retuned. Each row is the change in accuracy from removing that component. Absolute levels and harmful-exposure counts: Table 6.

Variant	Qwen	GPT-4o
without Global Advisor	−2.3	−2.0
without Local Advisor	−3.7	−6.0
without Meta Governor	−6.8	−6.5

Mechanism. Before the first snapshot commit memory falls back to the anchor, so the paired difference is zero and every disagreement, and hence the entire gap, falls after that commit. Both cases of Proposition 2 occur on the slice. On most disagreements (84 of 110 and 99 of 134 on Schedule A; 134 of 151 and 132 of 146 on Schedule B) the key's latest paired outcome is a tie, so Local has no signal and Global transfers evidence from other keys; on the rest, the key's own outcome disagrees with the pool (Appendix B.3). The Meta Governor selects an advisor at least as good as the other on 94–98% of these steps. Local stays the default. A Global override is helpful when Global's action scores above Local's and harmful when it scores below: on Schedule A, 10 helpful against 2 harmful on Qwen and 10 against 3 on GPT-4o; on Schedule B, 10 against 0 and 9 against 0. The remaining overrides do not change the score. On Schedule A, where Local is the better fixed scale, the gap equals helpful minus harmful overrides per disagreement episode: $(10-2)/110 = 7.3$ and $(10-3)/134 = 5.2$ points. Two Schedule A steps on Qwen show why both advisors stay. At visit 7 of "What is my current record in the volleyball league?" the old answer 3-2 is still valid: Local's last outcome on that key is non-positive, so Local skips, while other keys have already shown that the current snapshot is useful, so Global uses memory; the governor overrides, and the step is correct. At visit 14 of "Do I have a spare screwdriver for my laptop?" the old answer No is still valid: Local's last visit to that key was positive, the pooled window is not, and the governor keeps Local, again correctly. Local alone would have skipped the first step and Global alone the second (Appendix D).

Robustness. The same frozen router outperforms the better fixed scale on all 30 held-out settings (33,792 episodes; median +7.3, lowest +0.6 points), which span a third hash (Schedule C), two further schedule families, a 54-key pool, and both real-drift sources (Appendix B.7). On the 54-key pool every key-bootstrap 95% interval lies above zero (gaps +6.0 to +9.7 points; Table 2), and varying $W_g \in \{16, 32, 64\}$ keeps all twelve gaps positive (Appendix B.5). On a separate harm-exposure schedule, RC matches the harmful exposure of the best Local window while exceeding the accuracy of the best Global window on both models (Appendix B.9).

4.2 Naturally occurring drift

The results so far inject non-stationarity with a schedule. We now keep the same harness, strip timestamps as before (Assumption 1), and take both the changing facts and their switch times from real sources. TempLAMA (Dhingra et al., 2022) supplies 120 Wikidata facts that change exactly once over 2010–2020 (a person's club, office, or party; an organization's chair or coach); TAQA (Zhao et al., 2024) supplies 100 Wikipedia-table questions whose answers change over 2000–2023 (a

Table 4: Real drift, disagreement-slice accuracy (%). Gap is RC minus Best G/L (points). TempLAMA has 120 keys and TAQA 100; switch times are the real change years, and the per-key snapshot lag matches the main protocol. Windows stay frozen. All eight key-bootstrap intervals (5000 replicates) lie above zero.

Source	Model	Schedule	n_{dis}	RC	Best G/L	Gap	95% CI
TempLAMA	Qwen	A	1105	85.4	78.2	+7.2	[5.8, 8.9]
		B	1109	86.2	78.8	+7.4	[6.0, 9.1]
	GPT-4o	A	786	84.2	66.2	+18.1	[15.2, 20.5]
		B	793	83.6	66.1	+17.5	[14.6, 19.8]
TAQA	Qwen	A	856	83.4	74.6	+8.8	[6.4, 10.8]
		B	848	84.0	75.0	+9.0	[7.0, 11.1]
	GPT-4o	A	702	81.5	69.8	+11.7	[9.1, 14.7]
		B	682	82.4	70.1	+12.3	[9.5, 15.6]

driver's race win, a constituency's legislator, an NBA scoring leader's team). Each key flips at its true calendar time, so the switch-time distribution is empirical, not chosen. With the switch times fixed by the data, Schedules A and B here vary the order in which keys are visited within each pass. On the disagreement slice the frozen router stays between 81% and 86% on both sources and both schedules, and it outperforms the better fixed scale on every row (Table 4). On TempLAMA the router holds 84% on the GPT-4o slice, while the better fixed scale reaches only 66%. Full-stream gaps are +3 to +5 points. Nothing is retuned: the same router transfers to two independent real-drift sources.

4.3 Every component contributes

On Schedule A, the full router reaches a full-stream accuracy of 70.2 on Qwen and 73.3 on GPT-4o (Table 3), and each component accounts for part of it. Without the Global Advisor, accuracy falls by 2.3 and 2.0 points; Global contributes mainly on keys whose latest paired outcome is a tie, where Local has no signal. The Local Advisor contributes more: its removal costs 3.7 and 6.0 points and raises the number of harmful episodes on GPT-4o, where memory is used and scores below the anchor, from 9 to 19. The Meta Governor (§3.5) contributes most. Without it, memory is used only when the two advisors agree, so it is never used on the disagreement slice. Full-stream accuracy then drops by 6.8 and 6.5 points, and slice accuracy by 21.8 and 17.2 points (RC versus Agree-only in Table 1), which is the value of arbitration. On Schedule B every removal also lowers accuracy, by 2.3 to 3.1 points, and removing Local again roughly doubles harmful episodes, from 8 to 15 on Qwen and from 6 to 14 on GPT-4o (Table 6). The same holds on naturally occurring drift. Across TempLAMA and TAQA (Tables 12 and 13), accuracy falls by 3.9 to 7.0 points without Local, by 3.0 to 5.4 points without Global, and by 3.5 to 5.8 points without the Meta Governor.

5 Conclusion

A stored fact can go stale and be contradicted by later feedback—often only after the agent has already acted on it—and when that correction stays hidden, no design-time sharing scale is uniformly sufficient. We formulated this as *memory conflict governance* and proposed RegimeCouncil, a paired-evidence router that arbitrates between global and local advisors using delayed counterfactual surplus. Per-key replay escapes the observation floor once a key is revisited after the delay; neither fixed scale is uniformly sufficient; and a surplus test switches in finite time under delay. On such revisited streams, RegimeCouncil outperforms the better fixed scale by 5.2–7.3 points where the advisors disagree, and the same frozen router holds on TempLAMA and TAQA drift, with each of its three components contributing. Managing memory staleness therefore means arbitrating complementary evidence granularities as delayed feedback arrives. Two directions remain open: routing from logs that record only the executed action, where the paired difference of Assumption 3 is unavailable, and a live-agent deployment of the same frozen router.

AI USE STATEMENT

We used generative AI tools to assist with polishing the manuscript text and drafting experiment code. All code was reviewed and run by the authors. We did not use these tools to generate or alter evaluation data: all questions and answers come from the original public benchmarks and the stream construction is a fixed, rule-based transformation. The theory, the proofs, and the method were developed by the authors. We have reviewed all AI-assisted content and take full responsibility for the final content of this work, including text, claims, and any artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies memory conflict governance for language-model agents on publicly available benchmarks (LongMemEval, TempLAMA, and TAQA). It does not involve human subjects, private user data, or the release of sensitive datasets. We do not foresee significant dual-use or fairness risks beyond those already associated with general-purpose language-model agents. We will release code and experimental configurations to support transparent evaluation.

REPRODUCIBILITY STATEMENT

We provide the experimental settings in Section 4 and Appendix B, including the frozen windows, the held-out split, scoring, and snapshot refresh. Assumptions and the router are in Section 3; proofs are in Appendix A; prompts are in Appendix C. Upon acceptance, we will release source code, cached completions, and the scripts used to produce the reported tables.

REFERENCES

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/91ba7292e5388b90b58d0b839a7f19ec-Paper.pdf.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Learning to optimize under non-stationarity. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 1079–1087. PMLR, 2019.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In *ECAI 2025*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 2993–3000. IOS Press, 2025. doi: 10.3233/FAIA251160.

Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1405–1411. PMLR, 2015.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273, 2022.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), 2014.

Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2): 151–178, 1998. doi: 10.1023/A:1007424614876.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, 2026.

Chi Jin, Sham Kakade, Akshay Krishnamurthy, and Qinghua Liu. Sample-efficient reinforcement learning of undercomplete POMDPs. In *NeurIPS*, 2020.

Kwang-Sung Jun, Francesco Orabona, Stephen Wright, and Rebecca Willett. Improved strongly adaptive online learning using coin betting. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 943–951. PMLR, 2017.

Qingcan Kang, Mingyang Liu, Shixiong Kai, Kaichao Liang, Zhentao Tang, Yuqi Cui, Tao Zhong, and Mingxuan Yuan. Retain or consolidate? budget-dependent operator selection for language agent memory. *arXiv preprint arXiv:2607.17545*, 2026a.

Qingcan Kang, Mingyang Liu, Shixiong Kai, Kaichao Liang, Tao Zhong, and Mingxuan Yuan. Learning what to remember: Observability-safe memory retention via constrained optimization for long-horizon language agents. *arXiv preprint arXiv:2606.10616*, 2026b.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. In *NeurIPS*, 2016.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. doi: 10.48550/arXiv.2310.08560.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. doi: 10.1145/3586183.3606763.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

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. MemRL: Self-evolving agents via runtime reinforcement learning on episodic memory. *arXiv preprint arXiv:2601.03192*, 2026.

Bowen Zhao, Zander Brumbaugh, Yizhong Wang, Hannaneh Hajishirzi, and Noah A. Smith. Set the clock: Temporal alignment of pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15015–15040, 2024. doi: 10.18653/v1/2024.findings-acl.892.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. In *International Conference on Learning Representations*, volume 2026, pp. 58413–58438, 2026.

A Proofs

The main text uses three results to support the architecture (Proposition 1: an observation floor with a matching history separation; Proposition 2: both evidence scales; Proposition 3: surplus-gated selection under delay). We record complete arguments here.

A.1 Observation floor and history separation (Proposition 1)

Proof of Proposition 1. Fix a colliding context z visited $n(z)$ times, of which a visit-weighted minority-regime share $q(z) = \min\{n_0(z), n_1(z)\}/n(z)$ occurs under the minority regime. For every action a,

$$\rho_0(a; z) + \rho_1(a; z) \geq \Delta$$

by Δ-collision, and both ρ_r are nonnegative, so

$$n_0(z)\,\rho_0(a;z) + n_1(z)\,\rho_1(a;z) \;\geq\; \min\{n_0(z), n_1(z)\}\big[\rho_0(a;z) + \rho_1(a;z)\big] \;\geq\; n(z)\,q(z)\,\Delta.$$

Because $\pi \in \Pi_{\mathrm{obs}}$ conditions only on z, the same action law $\pi(\cdot \mid z)$ governs every visit to z irrespective of the regime (Assumption 1), so the expected total regret accumulated at z equals $\mathbb{E}_{a\sim\pi(\cdot|z)}\big[n_0(z)\rho_0(a;z) + n_1(z)\rho_1(a;z)\big]$. The displayed bound holds for every a, so taking the expectation preserves it: the expected regret at z is at least $n(z)q(z)\Delta$, and randomization cannot undercut it. Summing over colliding contexts (non-colliding visits contribute nonnegative regret) and dividing by T yields average regret at least

$$\frac{\Delta}{T}\sum_z n(z)\,q(z) \;=\; p\,q\,\Delta \;=\; \delta,$$

where p is the colliding-visit frequency and q the visit-weighted minority share. No independence across time is used. This proves part (i).

Part (ii): separation. Consider the history policy that, on each key k, replays the sign of the revealed paired outcome for k with the latest generation time and defaults to the anchor before any outcome for k is revealed—the Local Advisor of Section 3.4 with $W_\ell = 1$, which conditions on h_t and so lies in Π_{hist}. Fix a key k whose regime is constant between switches; the regime-optimal memory action on k is determined by that regime, and by Assumption 2 each revealed comparison carries its sign. Once a single post-switch outcome for k has been revealed, every later-generated outcome carries the same sign, so the policy plays the regime-optimal memory action on every subsequent visit until the next switch and incurs zero regret there. Regret can therefore accrue only on visits that precede the first revealed outcome after a switch, or before the first outcome ever at cold start. An outcome generated at step s is revealed by $s + \tau$, so the stale visits following one switch number at most m, the largest count of visits to a single key within any window of τ steps. With at most S switches and K cold starts, and per-step regret at most 1 (utilities lie in $[0, 1]$), the total regret is at most $(S + K)m$; dividing by T gives the stated bound, which falls below the floor δ of part (i) once $T > (S + K)m/\delta$.

When keys are too sparse for per-key replay (each seen $O(1)$ times), the complementary history policy—the Global Advisor pooling across keys—attains the same separation whenever the keys share a regime: after each of the S switches its pooled window is corrected within $O(W_g + \tau)$ revealed outcomes, giving total regret $O(S(W_g + \tau))$ and average regret below δ for large T. The two cases mirror the constructions of Proposition 2. □

Proof of Corollary 1. Any stationary feature map $f(z)$ composed with a classifier induces a stationary policy in Π_{obs}. Apply Proposition 1. The floor is therefore invariant in the capacity of f: enlarging a class of maps $\mathcal{Z} \to \Delta(\mathcal{A})$ cannot recover a regime bit that is absent from z. □

A.2 Granularity constructions (Proposition 2)

Proof of Proposition 2. Global mixing. Use two keys with *opposite* per-key advantages—$d_t = +1$ whenever the majority key k_1 is queried and $d_t = -1$ whenever the minority key k_2 is queried—and interleave them on a fixed periodic schedule in which k_2 appears a fraction $\beta \in (0, \frac{1}{2})$ of the steps. Once the window $\mathcal{W}_g$ is full, it contains at most $\lceil \beta W_g \rceil$ entries equal to -1, so its pooled sum is at least $W_g(1 - 2\beta) - O(1) > 0$ for W_g large enough; the Global Advisor therefore proposes memory on every step and acts wrongly on every k_2 visit, a fraction β of the T steps. The Local Advisor reads only the current key and, after a single revealed visit, tracks the correct per-key sign on both keys. The per-step utility gap on k_2 visits is bounded below by a positive constant on $\Theta(T)$ steps, so Global suffers linear loss relative to Local.

Local sparsity. Present a sequence of distinct keys, each visited once, all sharing a common positive memory advantage $d_t = +1$. On a fresh key the Local Advisor has an empty per-key window and defaults to the anchor, incurring a constant per-step gap. Once the first outcome is revealed, within τ steps, the Global Advisor's pooled sum is strictly positive and transfers the shared advantage to every subsequent new key. Repeating over T distinct keys yields linear loss for Local relative to Global.

The two streams are disjoint in the failure they induce: the first makes pooling biased, the second makes isolation uninformative. No single fixed scale is therefore uniformly sufficient. □

A.3 FINITE META RESPONSE (PROPOSITION 3)

Proof of Proposition 3. Recall $A_t = \sum_{i \in \mathcal{W}_m(t)} (v_{g,i} - v_{\ell,i})$ sums the surplus of the W_m most recently *revealed* episodes. Suppose first that $v_{g,i} - v_{\ell,i} \geq \gamma > 0$ for the W_m consecutive revealed indices that occupy the window. Then $A_t \geq W_m \gamma > 0$, so the governor selects Global under the rule "override iff $A_t > 0$." If instead $v_{g,i} - v_{\ell,i} \leq -\gamma$ throughout the window, then $A_t \leq -W_m \gamma < 0$ and Local remains (or returns as) the selection.

It remains to account for delay. An outcome generated at time s is revealed by time $s + \tau$. Filling the window with W_m revealed outcomes therefore requires at most W_m generations plus τ waiting steps, i.e. at most $W_m + \tau$ decisions. The bound is deterministic given the persistent-surplus hypothesis; it does not control streams on which the sign of $v_{g,i} - v_{\ell,i}$ flips inside every window of length W_m, nor streams with missing counterfactual utilities. The window is ordered by generation time, so an outcome revealed late does not displace newer ones. On episodes where the two advisors agree the surplus is 0; the same argument shows that Global is selected whenever the surpluses in the window are nonnegative and at least one is positive, and Local whenever they are all nonpositive. □

B EVALUATION PROTOCOL

This appendix records how the delayed-feedback stream is built, how answers are scored, and which artifacts reproduce the main tables.

B.1 ITEMS AND CACHED ANSWERS

The 16 keys are LongMemEval knowledge-update questions (Wu et al., 2025) whose old and new answers we audited manually. Temporal-pointer lexicon is stripped so that z_t does not name which write is current (Assumption 1). The neutralization is a fixed lexical filter (removing cues such as *currently*, *now*, *previously*, *no longer*) applied identically to every item; it changes no facts and injects no distractors, so the questions and both answers remain the original benchmark items, with the old and new answers audited manually. Each key therefore yields a colliding observation: two memory records and a question, with no recency cue.

For each key we cache three completions of the same model: no memory, memory injected with the old answer, and memory injected with the new answer. Routing never calls the LLM again. At episode t, fallback utility is the no-memory score against the currently valid gold, and memory utility is the score of whichever write is in the delayed snapshot (or fallback if the key has not yet been committed). A prediction is scored 1 if the normalized strings are equal or one contains the other, and 0 otherwise. This pairing is Assumption 3: the episode is scored under both actions without simulating a new trajectory.

B.2 HIDDEN SCHEDULE AND DELAY

Each key is visited 22 times ($T = 352$). The latent valid write follows *block_switch*: two key cohorts flip at mid-horizon, staggered by 1–2 visits per key, so Global pooling and Local isolation disagree on a substantial slice. Visit order within each pass is hashed from the schedule identifier. Pending writes are committed to the snapshot only every 96 episodes, so the stored write can lag the latent valid write by up to 96 episodes and memory can be stale when it is used. Each episode is scored under both actions once it closes, and its paired difference enters the windows before the next decision. Feedback therefore arrives after the action, with $\tau = 1$ in Assumption 2, while the snapshot lag governs how long memory stays stale.

Development uses schedules disjoint from evaluation. The held-out schedules never enter that set and change only the hash, hence visit order and per-key phase offsets. In the main text the two primary held-out schedules are named Schedule A and Schedule B, a third is Schedule C, and the accuracy–harm analysis uses a separate harm-exposure schedule.

B.3 REVELATION SLICE

Table 5 splits Schedules A and B at the first snapshot commit. Until episode 96, a key that has not been committed is scored with the no-memory utility (Appendix B.1), so the paired difference is 0, Global and Local propose the same action, and RegimeCouncil matches both. All 110, 134, 151, and 146 disagreement episodes (Schedule A Qwen and GPT-4o, then Schedule B Qwen and GPT-4o), and therefore the entire gap in Table 1, lie after the first commit. In most of them (84, 99, 134, and 132) the key's latest paired outcome is a tie, so Local has no signal and Global transfers evidence from other keys, the second case of Proposition 2. In the rest, the key's own latest outcome disagrees with the pooled window, the first case.

Table 5: Revelation slice on the cached main streams (accuracy, %). Pre is the 96 episodes before any snapshot commit: Global, Local, and RegimeCouncil tie, and n_{dis}=0. Post gap is $\text{RC} - \max(G, L)$ on disagreements after the first commit, equal to Table 1.

Schedule	Model	Pre n	Pre acc.	Post n	Post n_{dis}	Post gap
A	Qwen	96	37.5	256	110	+7.3
	GPT-4o	96	43.8	256	134	+5.2
B	Qwen	96	37.5	256	151	+5.3
	GPT-4o	96	43.8	256	146	+5.5

B.4 MODULE ABLATION

Table 6 is the full module ablation summarized in Section 4.3, on Schedules A and B. On Schedule B every removal lowers accuracy by 2.3 to 3.1 points. All three variants are ablations of the full router. The third removes the Meta Governor (§3.5). Its drop is not the gap versus $\max(G, L)$ in Table 1.

Table 6: Module ablation on Schedules A and B (full-stream accuracy, %). Each row removes one component of the full router. Without Global or without Local, the remaining advisor is used alone. Without the Meta Governor (§3.5) the row falls below either advisor alone in all four settings (on Schedule A, 6.8 points on Qwen and 6.5 on GPT-4o; 21.8 and 17.2 disagreement-slice points). Removing Local roughly doubles harmful exposure in all four settings. Δacc is computed from unrounded accuracies.

Schedule	Model	Variant	Accuracy	Harm exposure	Δacc
A	Qwen	Full RegimeCouncil	**70.2**	9	—
		without Global Advisor	67.9	8	−2.3
		without Local Advisor	66.5	11	−3.7
		without Meta Governor	63.4	8	−6.8
	GPT-4o	Full RegimeCouncil	**73.3**	9	—
		without Global Advisor	71.3	6	−2.0
		without Local Advisor	67.3	19	−6.0
		without Meta Governor	66.8	6	−6.5
B	Qwen	Full RegimeCouncil	**71.6**	8	—
		without Global Advisor	68.8	8	−2.8
		without Local Advisor	69.3	15	−2.3
		without Meta Governor	68.5	6	−3.1
	GPT-4o	Full RegimeCouncil	**73.0**	6	—
		without Global Advisor	70.5	6	−2.6
		without Local Advisor	70.7	14	−2.3
		without Meta Governor	70.2	5	−2.8

B.5 METRICS AND THE W_m/GATE GRID

The primary metric is accuracy on episodes where the Global (W_g=32) and Local (W_ℓ=1) proposal bits differ, using those bits only (no labels), reported as $\text{RC} - \max(G, L)$. Key-cluster bootstrap resamples the 16 keys (5000 replicates, seed 0), not episodes; point estimates use the full sample. On Table 1 the mean gaps are 4.3–6.2 points, with 95% intervals $[0.0, 10.9]$ and $[0.0, 9.6]$ on Schedule A

(Qwen, GPT-4o) and $[-0.6, 8.5]$ and $[-0.8, 8.7]$ on Schedule B, and $P(\widehat{\text{gap}}{>}0) \in [0.93, 0.97]$. Table 7 is the report-only $W_m \times$gate grid on Schedule A; it was not used to select the frozen $(8, 0)$.

Table 7: Report-only disagreement gaps (points) on Schedule A. Larger gates shrink the gain. Not used for selection.

	Qwen, Schedule A			GPT-4o, Schedule A		
W_m	gate 0	gate 1	gate 2	gate 0	gate 1	gate 2
4	+8.2	+6.4	+1.8	+6.0	+3.7	+3.0
8 (frozen)	**+7.3**	+5.5	+5.5	**+5.2**	+4.5	+3.7
16	+7.3	+6.4	+5.5	+4.5	+4.5	+3.0

Table 8 is the companion W_g sweep: it recomputes the disagreement gap at $W_g \in \{16, 32, 64\}$, with Global and Local defined at that W_g and W_ℓ, W_m, and the gate held fixed. All twelve entries are positive; W_g=64 stays positive on all four settings (+3.1 to +5.7), and halving the pooled window to 16 shrinks the gain most on Schedule B (+1.8 on Qwen and +2.0 on GPT-4o), as expected when Global underuses cross-key evidence. The frozen W_g=32 is the most consistent across schedules and is not confined to a single length.

Table 8: Report-only disagreement gaps (points) versus the best advisor at the same W_g. All twelve entries are positive; the smallest is Schedule B, Qwen, at W_g=16 (+1.8). Not used for selection.

W_g	Schedule A, Qwen	Schedule A, GPT-4o	Schedule B, Qwen	Schedule B, GPT-4o
16	+9.9	+4.6	+1.8	+2.0
32 (frozen)	**+7.3**	**+5.2**	**+5.3**	**+5.5**
64	+5.7	+4.9	+3.1	+4.4

B.6 Scale and schedule robustness

These checks reuse the frozen windows $(W_g, W_\ell, W_m, \text{gate}) = (32, 1, 8, 0)$ with new held-out tags; nothing is retuned.

More keys. The audited LongMemEval knowledge-update pool yields 54 keys after the same temporal-pointer neutralization used for the main 16-key set; we cache the no-memory, old-answer, and new-answer completions for both models. To preserve the per-key snapshot lag of the main protocol we scale the refresh with the key count ($\text{refresh} = 6K$, i.e., 96 at K=16 and 324 at K=54), so that between two refreshes each key still receives about six visits regardless of K, keeping the per-key lag constant across pool sizes. Table 2 reports the frozen router at K=54: the disagreement gap stays +6 to +9.7 points and every key-bootstrap 95% interval lies above zero. Holding the refresh at 96 while K grows shortens the per-key lag below the protocol and is not comparable.

A second schedule family. *independent_switch* (one flip per key at a hashed time, keys not cohort-synchronized) and *poisson_flip* (repeated independent flips) were not in the development grid. Table 9 evaluates the frozen router on the original 16-key caches. *independent_switch* is positive on all four settings. *poisson_flip* is also positive on all four, with smaller gaps (+0.6 to +4.6): repeated flips place both regimes inside the governor's W_m window, the surplus stays near zero, and the governor remains on the Local default more often.

Table 9: Disagreement-slice gap (points) on two held-out schedule families, 16-key caches, frozen windows. *independent_switch* is positive on all four settings; *poisson_flip* is positive on all four with smaller gaps.

Schedule family	Schedule A, Qwen	Schedule A, GPT-4o	Schedule B, Qwen	Schedule B, GPT-4o
independent_switch	+6.7	+7.2	+6.9	+7.1
poisson_flip	+0.6	+3.1	+4.6	+3.3

B.7 ADDITIONAL SWITCHING COMPARISONS

Schedules A/B/C, the two held-out schedule families, the 54-key pool, and both real-drift sources, over both models, make 30 settings and 33,792 episodes. Holding $(W_g, W_\ell, \text{gate}) = (32, 1, 0)$, Table 10 varies W_m and adds two switching rules on the same advisor surplus: follow-recent-winner and CUSUM. Every variant updates on the schedule of §3.6.

Table 10: Every surplus-based governor outperforms the better fixed scale on all 30 held-out settings; the choice of governor mainly changes harmful exposure. Gaps are disagreement-slice points over $\max(G, L)$, median and worst over the 30 settings. *Switch rate* is the fraction of consecutive disagreement episodes on which the governor changes advisor. *Excess harm* is the mean number of episodes per setting, beyond the safest fixed scale, in which the policy acts on memory that is worse than the anchor.

Governor	Median gap	Worst gap	Settings < 0	Switch rate	Excess harm
W_m=1	+3.97	+0.62	0/30	0.093	0.13
W_m=2	+5.73	+1.23	0/30	0.071	0.13
W_m=4	+7.09	+1.85	0/30	0.049	0.27
W_m=8 (frozen)	+7.27	+0.62	0/30	0.035	0.90
W_m=16	+7.87	+2.47	0/30	0.027	1.27
W_m=32	+7.20	+1.85	0/30	0.024	2.33
Follow-recent-winner	+8.12	+2.47	0/30	0.025	2.20
CUSUM on surplus	+7.94	+1.23	0/30	0.019	2.20

No row of Table 10 falls below the better fixed scale. Follow-recent-winner, computed on the same streams, is a supplementary comparison.

Table 11: Development comparison, disagreement-slice points and harmful-episode counts, pooled over five tags and both models (refresh 96). The frozen rule has higher accuracy and fewer harmful episodes in every family, including the family Schedule A is drawn from. Per model the accuracy margins are +0.62 (Qwen) and +0.40 (GPT-4o).

Development family	RC−follow	Harm, RC	Harm, follow
v4_default	+0.62	49	65
anticorrelated	+0.47	38	46
long_stable_flip	+0.07	94	105
sparse_need_global	+0.53	17	28
block_switch	+0.84	69	79

On held-out disagreement accuracy the ordering is reversed by a small margin: follow-recent-winner is higher on 19 settings, equal on 4, and lower on 7, with mean difference 0.3 points (sign test p=0.03). Only one key-bootstrap interval excludes zero (54-key *block_switch*, Qwen, -1.2, $[-2.4, -0.2]$). Those streams contain long stretches in which the same advisor remains better. Follow-recent-winner retains that advisor. The Meta Governor returns to Local whenever the recent surplus is not positive, and therefore forgoes a few tenths of a point on those stretches. The largest single deficit is Schedule A, -2.7 and -2.2 points, and W_m=4 remains 1.8 and 1.5 points lower there. Two advantages remain with the frozen rule. On the development families, which were not used to choose the window, its disagreement accuracy is higher in every family, by 0.07 to 0.84 points, and it records fewer harmful episodes in each (Table 11). On the held-out settings it still records fewer harmful episodes on 26 of 30 (707 against 746; the safest fixed scale has 680). A retrospective sweep on the development families, scored by the worse model's mean disagreement gap, ranks W_m=8 first (3.55) and W_m=4 second (3.45), with the same lowest cell (-1.61).

B.8 NATURALLY OCCURRING DRIFT (TEMPLAMA AND TAQA)

All results above inject the non-stationarity with a schedule. To test whether the mechanism survives *real* drift, we replace the schedule with the true calendar chronology of changing facts, on two independent sources. The harness is otherwise unchanged: same paired-utility routing, same frozen

windows, same per-key snapshot lag ($\text{refresh} = 6K$). We report both the disagreement-slice levels (so the gap is interpretable) and the full-stream gap.

Construction and selection. TempLAMA (Dhingra et al., 2022) mines $(\text{subject}, \text{relation}, \text{year}) \rightarrow$ object facts from Wikidata for 2010–2020. Of 5,839 subject–relation groups, 2,191 are single-valued and change exactly once (a single clean transition from an old to a new value); we take a 120-key subset balanced across all nine relations. TAQA (Zhao et al., 2024) supplies Wikipedia-table questions over 2000–2023; most items change several times, so from 1,000 validation items we keep the 284 with a clean adjacent pair of single-valued runs (each ≥ 2 years, distinct non-overlapping answers) and hash-sample 100. Each key becomes a natural-language question with two memory records; years and temporal-pointer lexicon are stripped so the valid write is hidden at decision time (Assumption 1), exactly as for the LongMemEval streams. The regime flips at the fraction of the window corresponding to its *real* change year, so the switch-time distribution is empirical (TempLAMA spread across 2012–2019) rather than designed. Restricting to two regimes is a deliberate simplification that keeps each key a clean flip; multi-change keys are left to future work. The two rows per model (Schedules A and B) share the *same* real switch times and differ only in the hashed visit order.

Table 12: TempLAMA real-drift stream (120 keys, real switch times, matched snapshot lag), disagreement-slice accuracy (%). Gap is RC minus Best G/L (points). The large GPT-4o gap is driven by the fixed Global and Local scales collapsing to $\approx 66\%$ on the slice, not by an unusual RC level. Full-stream gaps are 3.0 and 3.1 points on Qwen and 5.4 and 5.3 points on GPT-4o (Schedules A and B). The difference from follow-recent-winner and CUSUM is within ≈ 0.7 points. All four key-bootstrap 95% CIs (5000 replicates) are above zero ($P(\text{gap}{>}0){=}1.00$); each ablation reduces accuracy (without Local -5.7 to -7.0, without Global -3.0 to -5.4, without Meta Governor -3.9 to -5.8). Holding $\text{refresh} = 96$ while K grows shortens the per-key lag below the protocol and is not comparable.

Model	Schedule	n_{dis}	RC	Best G/L	Gap	95% CI
Qwen	A	1105	85.4	78.2	+7.2	[5.8, 8.9]
	B	1109	86.2	78.8	+7.4	[6.0, 9.1]
GPT-4o	A	786	84.2	66.2	+18.1	[15.2, 20.5]
	B	793	83.6	66.1	+17.5	[14.6, 19.8]

Table 13: TAQA real-drift stream (100 keys, real switch times, matched snapshot lag), disagreement-slice accuracy (%). Gap is RC minus Best G/L (points). Full-stream gaps are 3.4 and 3.5 points on Qwen and 3.7 and 3.8 points on GPT-4o (Schedules A and B). The difference from follow-recent-winner and CUSUM is within ≈ 0.6 points. All four key-bootstrap 95% CIs (5000 replicates) are above zero ($P(\text{gap}{>}0){=}1.00$); each ablation reduces accuracy (without Local -3.9 to -5.3, without Global -3.4 to -3.8, without Meta Governor -3.5 to -5.2).

Model	Schedule	n_{dis}	RC	Best G/L	Gap	95% CI
Qwen	A	856	83.4	74.6	+8.8	[6.4, 10.8]
	B	848	84.0	75.0	+9.0	[7.0, 11.1]
GPT-4o	A	702	81.5	69.8	+11.7	[9.1, 14.7]
	B	682	82.4	70.1	+12.3	[9.5, 15.6]

The same frozen mechanism transfers from constructed schedules to two independent real-drift sources (Wikidata facts and Wikipedia-table questions), with the disagreement-slice levels making the gap interpretable and the full-stream gap a steadier $+3$ to $+5$ points.

B.9 Accuracy–harm frontier

Table 14 is the full accuracy–harm summary on the harm-exposure schedule; Figure 3 visualizes it: RegimeCouncil (red star) matches best-local harmful exposure while exceeding best-global accuracy, placing it on the favorable corner of the frontier. On the disagreement slice of this schedule, RegimeCouncil exceeds $\max(G, L)$ of its frozen advisors by 4.9 points on Qwen ($n_{\text{dis}}{=}143$) and 6.4 points on GPT-4o ($n_{\text{dis}}{=}140$).

Table 14: Complementary Pareto summary on the harm-exposure schedule.

Model	Method	Accuracy	Harm exposure	Drawdown
Qwen	No memory	48.9	0	0
	Always memory	76.7	23	6
	Best global (W=16)	76.7	16	2
	Best local (W=1)	75.9	8	1
	RegimeCouncil	**78.4**	**8**	**1**
GPT-4o	No memory	48.6	0	0
	Always memory	79.0	20	2
	Best global (W=16)	79.0	16	2
	Best local (W=1)	78.4	5	1
	RegimeCouncil	**81.0**	**5**	**1**

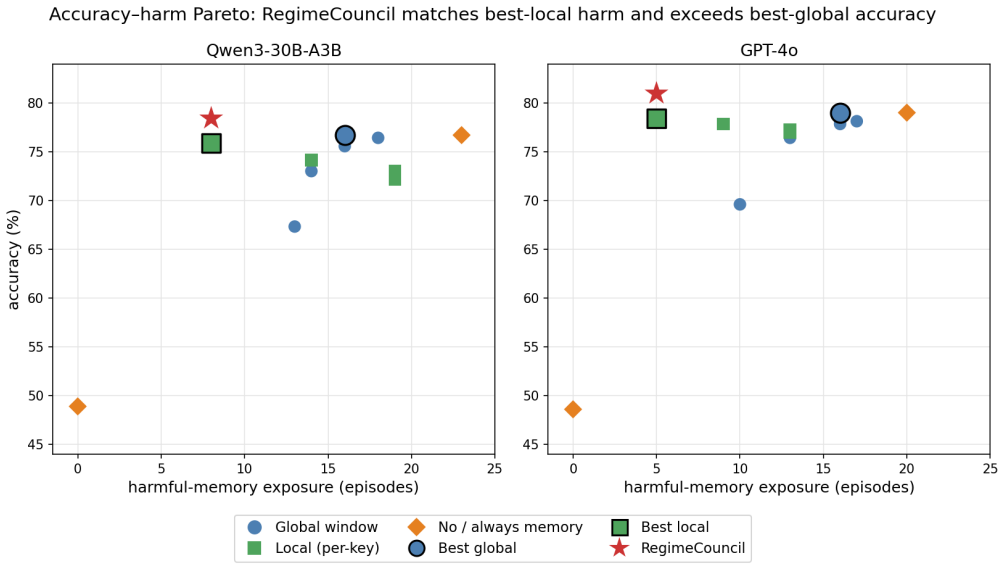

Figure 3: Accuracy vs. harmful-memory exposure on the harm-exposure schedule. Blue circles: Global window. Green squares: Local (per-key). Orange diamonds: no / always memory. Red star: RegimeCouncil. Black outlines mark the best Global and best Local settings from Table 14.

B.10 Observable-conflict selection (not a RegimeCouncil result)

FactConsolidation places both writes in the prompt, so conflict is observable, and the comparison is certified selection between generic retrieval and last-write-wins rather than the Global/Local council under a hidden valid bit. Table 15 records it only so the rule can be inspected outside the main claim. The anchor is generic top-5 retrieval; challengers are generic top-20 and last-write-wins (LWW) at $k \in \{5, 20\}$. Certified selection uses Bonferroni $0.05/3$ on the certification split only (n=132); the reported accuracies are on the held-out split (n=68). Online Meta applies the frozen surplus rule (W_m=8, gate 0) between Gen@5 and LWW@5 after a certification warm-start.

Table 15: Observable-conflict selection on MemoryAgentBench 64K (held-out %, n=68). Not a RegimeCouncil score: both writes are in the prompt, and the choice is between generic retrieval and last-write-wins. Lifts of the selected policy over Gen@5: 35.3 points on GPT-4o and 25.0 on Qwen. Generic k=20 never certifies.

Model	Gen@5	Gen@20	LWW@5	LWW@20	Selected	Guarded	Online Meta
GPT-4o	19.1	20.6	55.9	54.4	LWW@20	**54.4**	54.4
Qwen	17.6	17.6	42.6	39.7	LWW@5	**42.6**	41.2

B.11 ARTIFACT INDEX

Code and evaluation data will be released upon acceptance: the dataset builders (the LongMemEval streams, the 54-key pool, and the TempLAMA and TAQA real-drift sets), the cache fillers and cached model completions, the frozen router, and the evaluation runners that reproduce every table in this paper. The runners accept configuration files rather than hard-coded paths, so every result can be regenerated without editing source.

C PROMPT TEMPLATES

RegimeCouncil does not prompt the LLM to choose Global versus Local. The model only answers the user question; the router is a deterministic function of paired utilities. We therefore cache three completions per key and never vary the prompt across routing methods.

Delayed-feedback QA. The main 16-key caches use the following template. The optional snapshot line is omitted for the no-memory completion and filled with the old or new answer for the two memory completions. Display order of the two records is arbitrary and is not a recency cue.

```
Two memory records conflict.  Their display order is arbitrary and the
currently valid regime is hidden.  Return only the short answer.
{collision_observation}
A deployed feedback snapshot records the correct answer for this key
as:  {snapshot}
[snapshot line included only for memory completions; omitted for no-memory]
Question:  {question}
```

A concrete fill-in for the volleyball-record question (abridged collision text):

```
Two memory records conflict.  Their display order is arbitrary and the
currently valid regime is hidden.  Return only the short answer.
Conflicting memory A: [...]  I've been doing pretty well in the
volleyball league, we're 3-2 so far!
Conflicting memory B: [...]  our volleyball team, the Net Ninjas, is
doing well with a 5-2 record.
Question:  What is my current record in the recreational volleyball
league?
```

Without a snapshot the model must guess from two unordered records. With a snapshot it is given one of $\{3\text{-}2,\ 5\text{-}2\}$ as the last committed write for that key; that write may already be stale relative to the hidden valid bit, which is why using memory is sometimes harmful.

Observable-conflict prompts. On the appendix MemoryAgentBench check the LLM sees retrieved facts and a conflict rule. Generic top-k uses "Answer from the retrieved facts." Last-write-wins additionally states that larger serial numbers are newer. The surplus rule still selects between two cached policies; it does not appear in the prompt.

D WORKED DISAGREEMENT EPISODES

Table 16 walks through two Schedule A episodes on Qwen where Global and Local disagree. Utilities are 1 if the chosen completion matches the hidden gold and 0 otherwise. "Use memory" means deploy the delayed snapshot; "skip" means the no-memory completion.

Helpful override. Visit 7 of the volleyball-record question, while the old write is still valid (gold 3-2). The delayed snapshot still matches the old write, so using memory is correct on this step

Table 16: Two disagreement steps on Schedule A (Qwen). Global pools the last 32 paired advantages; Local reads only the current key. Meta overrides Local only if recent counterfactual surplus is strictly positive.

Episode (hidden gold)	Global	Local	Meta	RC
Volleyball record (3-2)	use	skip	override	use (correct)
Spare screwdriver (No)	skip	use	keep Local	use (correct)

(u=1) and skipping is not (u=0). Local's last outcome on this key was non-positive, so Local skips. The pooled window is already positive—other keys have shown that the current snapshot generation is useful—so Global proposes to use memory. Meta has accumulated $A_t > 0$ and overrides. RegimeCouncil deploys memory and is correct; a local-only rule would have abstained.

Correct local default. Visit 14 of the spare-screwdriver question, while the old write is still valid (gold No). Local's last visit to this key was positive, so Local proposes to use memory. The pooled window is non-positive after mixed flips, so Global skips. Surplus is not positive, Meta keeps the local default, and RegimeCouncil again deploys memory correctly. A global-only rule would have skipped.

These two steps are the mechanism in miniature: the governor reads no part of z_t and only compares the two advisors' counterfactual utilities on completed episodes. On Schedule A, helpful overrides outnumber harmful ones (10 vs. 2 on Qwen).