DETECTING PREFERENCE CHANGE IS NOT ENOUGH: RETHINKING REWARD MODEL PERSONALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalized Reward Models (PRMs) train on users' past choices to predict future preferences. Temporal personalization often assumes that once preference change is detected, the model should adapt its use of that history. We argue that this collapses three distinct questions: *is there enough evidence to adapt this particular user? If so, how should memory change? And is the resulting adaptation genuinely driven by that user rather than by population-level regularities?* We study these questions prospectively, using only information available before evaluation. We fix a pairwise PRM with one component learned across training users and vary only how a second, user-memory component constructs the specific user's history. We compare four memory modes: *retain* keeps all past choices, *discount* progressively downweights older choices, *reset* rebuilds recent memory after a detected change, and a multiscale LETHE mode learns a mixture of causal memories operating at different timescales. We evaluate these modes on controlled synthetic histories, PENS, PRISM, and Chatbot Arena, while testing whether temporal structure is real, individually actionable, and prospectively useful. We uncover an *information paradox*: preference change can be clearly measurable across users, while an individual user's earlier history does not reliably identify, before seeing the future, which evaluated memory rule will work best. Choosing the best rule after seeing the outcomes appears to gain 4 to 6 percentage points, but about half of that is selection optimism, and the gain disappears when the rule must be chosen earlier. We then uncover an *intervention paradox*. A detected user-specific change does not determine the right memory response. Under abrupt change and identical alarms, event-triggered reset beats a temporary fast discount by about 1 percentage point, but which response wins depends on how late the alarm arrives and how far back reset rebuilds memory. Finally, exploratory analysis reveals an *illusion of personalization*. We find that the PRM can appear to adapt to a particular user while partly acting on population-level regularities rather than reliable evidence about that user's temporal state, helping some users and harming others without reliably identifying who has actually changed. Even the same held-out stable users receive much shorter assigned memory when the PRM is trained in a population where changes are common. Removing history length from the controller greatly weakens this behavior. Our findings show that *detecting change is not knowing whom to adapt, knowing whom to adapt is not knowing how to adapt, and adapting differently is not proof of genuine personalization*.

1 INTRODUCTION

A personalized reward model (PRM) predicts a user's next preference from earlier choices. Personalized representations capture differences that a population reward misses (Poddar et al., 2024; Chen et al., 2025; Nam et al., 2026). When user preferences evolve (i.e., *temporal change*), it creates an additional ambiguity. An older choice may still help, become less informative, or become actively misleading. In this paper, we argue that although recent feedback may deserve greater weight (Son et al., 2025), evidence that preferences have changed does not, by itself, tell us which user should adapt, which memory response should follow, or what an adaptive state says about that individual. *Temporal personalization can therefore obscure three distinct claims: that change occurred, that a particular memory intervention is warranted, and that the resulting adaptive state reflects change in*

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.

this user. To illustrate this, suppose a user's recent choices favor B after a long history of favoring A. In such a case, a recent-only prediction may outperform an old-only prediction, indicating that preference has changed. However, it may happen that the complete history can still beat the recent-only prediction. This shows that the old choices remain useful. In other words, *although change has been detected, forgetting is not yet justified.* That is, recency does not imply staleness.

This possibility motivates three questions: *Does the available history support a prospective memory decision for a particular user? Which response should follow a detected event? Does an adaptive memory state establish that the model identified change in that individual?* In this paper, we isolate these questions in a deliberately simple predictive setting. A linear Bradley–Terry predictor combines population preference with the user's earlier choices, while memory policies differ only in whether that history is retained, discounted, mixed across several timescales, or reset after an alarm. Our multiscale policy, LETHE, is one such response rather than the object of evaluation itself. Controlled synthetic histories expose the latent preference changes, letting us distinguish population temporal signal from prospective per-user choice and compare different responses under shared detected events (Figure 1). We then test how far these distinctions carry to natural histories from PRISM, Arena, and PENS (Kirk et al., 2024; Zheng et al., 2023; Ao et al., 2021); source-user-disjoint Arena constructions probe intervention behavior, and Skywork-Reward-V2 checkpoints test sensitivity to the reward representation (Liu et al., 2026).

The experiments expose three paradoxes. The *Information Paradox* is that temporal signal exists without becoming reliably actionable for one user: hindsight memory selection gains 5.66 points on the main mixture, while selection from an earlier window gains only 0.01; in the change-point-heavy mixture, the corresponding gains are 3.85 and -1.86 points. *Population temporal signal does not guarantee that one user's available history identifies a better rule for later predictions.* The *Intervention Paradox* is that detecting change does not determine the response. On the broad mixture, recent evidence beats old evidence by 5.35 points. However, adding the older evidence still improves over recent-only prediction by 2.33 points. When abrupt changes are common, adding old evidence instead hurts by 5.22 points. Although reset consequently helps by 1.2 points in the targeted abrupt-change population, it can hurt in the source-user-disjoint Arena construction when the history returns to its first source. Hence, *recency does not establish staleness, and detection does not prescribe the response.* Finally, an *Illusion of Personalization* appears even among stable users. On identical held-out histories, their median assigned memory falls from 29 to 16 steps when the controller is trained in a change-point-heavy population, costing them 2.30 points, while a Bayesian change-point learner given the true hazard keeps about 30. *A user-specific adaptive state is therefore not, by itself, evidence that this particular user changed.* Our contribution is the controlled separation of these questions. *Detecting change is not knowing whom to adapt, knowing whom to adapt is not knowing how to adapt, and adapting differently is not proof of genuine personalization.*

2 RELATED WORK

Personalized Preference Models. Variational Preference Learning and PAL infer user-specific preference structure (Poddar et al., 2024; Chen et al., 2025). MiCRo adds context-aware online routing among preference components (Shen et al., 2025), while PLUS co-adapts user summaries and reward prediction (Nam et al., 2026). Rubric-based and in-context approaches are compared in Appendix O. These methods condition rewards or generation on user information. We instead compare which parts of an ordered history remain useful within a common predictor. A personalized representation need not identify a better temporal memory rule from an earlier user window.

Temporal Adaptation And State Revision. Forgetting discounts older evidence when model parameters may drift (Kulhavý & Zarrop, 1993); shorter effective memories track change faster but estimate less precisely. Expert tracking and nonstationary optimization address changing environments (Herbster & Warmuth, 1998; Besbes et al., 2015). NS-DPO weights preferences exponentially by age (Son et al., 2025), and shifted dueling bandits characterize preference tracking under structural assumptions (Suk & Agarwal, 2023). Cumulative-sum detection and Bayesian run-length inference supply established event mechanisms (Page, 1954; Adams & MacKay, 2007). CORE is closer to the update decision, separating local dialogue evidence from persistent persona revision (Zhang et al., 2026b). Our predictive-memory comparisons distinguish whether older observations hurt accuracy

from which calibrated response helps under the same alarms. They do not claim forgetting, detection, or state revision as new operators.

Selection And Evaluation. Cawley and Talbot establish optimism from optimizing a noisy model-selection criterion (Cawley & Talbot, 2010). Prequential evaluation separates prediction from the feedback that updates it (Dawid, 1984). Reward-model evaluations also test whether proxy scores predict downstream generation or policy quality (Frick et al., 2025; Ma et al., 2026). Our contribution concerns individual memory decisions, predictive staleness, and attribution of adaptive state, not hindsight optimism itself. Appendix O develops the comparison.

3 METHODOLOGY

Setup. A temporal Personalized Reward Model (PRM) encodes a user's earlier choices to predict future preferences. Hence, it must decide how to use that history when preferences may evolve. At each interaction, a user chooses between two candidate responses. For example, the user may initially prefer a concise response a over a more detailed response b, while later comparisons may suggest a preference for greater detail. These earlier pairwise choices form the history available for personalization. The usual intuition is that a detected preference change should trigger memory adaptation. This hides three distinct decisions. First, a population-level pattern, such as many users gradually preferring more detailed responses, may be real without providing enough evidence that a particular user has changed. Second, even if this user's recent choices differ from older ones, that does not determine whether the older choices should be retained, gradually downweighted, or discarded after a detected event. Finally, the model may change this user's memory because users in similar populations often change, rather than because this user's own history provides convincing evidence of change. In this paper, we therefore test: *Does the available history justify changing memory for this user? If an event is detected, how should memory respond? When memory adapts, does that adaptation provide evidence of change in the individual user?* We hold the pairwise prediction problem, candidate representations, and population preference component fixed, and vary only the *user-memory component*. For any current comparison, every policy receives the same candidates and representations; only its use of earlier choices changes.

Controlled Preference Prediction. We use a linear Bradley-Terry model (Bradley & Terry, 1952) for pairwise preference prediction. At step t, user u is shown candidates $a_{u,t}$ and $b_{u,t}$, and the model predicts which response the user will prefer before observing the choice. A frozen encoder $e(\cdot)$ gives the comparison vector $x_{u,t} = e(a_{u,t}) - e(b_{u,t}) \in \mathbb{R}^p$. The label $y_{u,t} \in \{-1, +1\}$ is observed afterward, with $y_{u,t} = +1$ indicating preference for $a_{u,t}$. Once observed, the choice defines the signed preference direction $d_{u,t} = y_{u,t}x_{u,t}$, which can affect only later predictions. We decompose the Bradley-Terry score into a population preference and a user-specific memory correction. A fixed vector $w \in \mathbb{R}^p$ captures preference structure learned across users, while $s^m(H_{u,<t}) \in \mathbb{R}^p$ summarizes user u's earlier choices under memory rule m:

$$\ell^m_{u,t} = w^\top x_{u,t} + \alpha_m s^m(H_{u,<t})^\top x_{u,t}, \qquad \Pr_m(y_{u,t} = +1 \mid x_{u,t}, H_{u,<t}) = \sigma(\ell^m_{u,t}), \tag{1}$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$. The first term captures the population preference, the second supplies the user-specific correction, and α_m controls its fitted strength. The memory policies differ only in how $s^m(H_{u,<t})$ is constructed.

3.1 MEMORY CONSTRUCTIONS

***Retain* and *Discount*.** *Retain* treats every earlier choice as equally relevant. It sums the signed preference directions and normalizes the result: $s^{\text{ret}}(H_{u,<t}) = \frac{\sum_{i<t} d_{u,i}}{\left\|\sum_{i<t} d_{u,i}\right\|_2}$ when the sum is nonzero, and 0 otherwise. *Discount* instead lets older choices gradually lose influence through $S_{u,\gamma,t} = \gamma S_{u,\gamma,t-1} + d_{u,t}$, where $\gamma \in [0, 1]$. Prediction uses $S_{u,\gamma,t-1}/\|S_{u,\gamma,t-1}\|_2$ when nonzero, and 0 otherwise. With $\gamma = 0.5$, for example, a choice five steps old has raw weight $0.5^5 = 1/32$ relative to the newest choice; $\gamma = 1$ reduces to retain. Discount therefore permits recency but imposes one decay rate on the history.

LETHE: *Multiscale Mixture Memory*. A single decay rate can be restrictive when recent and older choices are both informative. We propose LETHE (*LE*arned *T*imescales for *HE*terogeneous

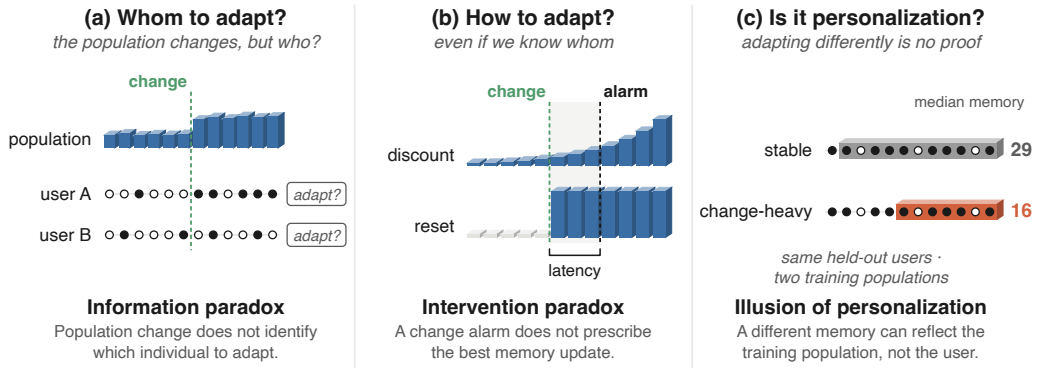


Figure 1: **Three paradoxes in temporal personalization. (a) Information paradox:** the population's choices shift at a change point, but individual histories (users A and B) are too noisy to reveal whose memory the reward model should adapt. **(b) Intervention paradox:** even once a change is flagged, the alarm does not prescribe the update; a reset (back-dated to the change) and a fast discount weight past evidence differently, and which is better depends on alarm latency. **(c) Illusion of personalization:** the same held-out stable users are assigned a median memory of 29 or 16 steps depending on the training population (Figure 7), so adapting differently is no proof of genuine personalization. Alarms are detector outputs, not oracle boundaries. Panel (c) is exploratory.

preference evolution), which maintains six causal traces with $\gamma_k \in \{0.5, 0.8, 0.92, 0.969, 0.9886, 1\}$. Each trace follows $S_{u,k,t} = \gamma_k S_{u,k,t-1} + d_{u,t}$, so fast traces emphasize recent choices and slow traces preserve longer-term evidence. LETHE combines them as $s^{\text{mix}}(H_{u,<t}) = \sum_{k=1}^{6} \pi_{k,t} \frac{S_{u,k,t-1}}{\|S_{u,k,t-1}\|_2}$, where $\pi_{k,t} \geq 0$, $\sum_{k=1}^{6} \pi_{k,t} = 1$, and a zero trace contributes zero. A controller selects the mixture weights from prefix information including history length, directional changes, and earlier prediction errors. For example, a $0.7/0.3$ mixture of slow and fast traces can preserve a longer-term preference while still responding to recent choices. LETHE therefore adapts memory continuously across several timescales rather than selecting one fixed decay rate.

Event-Triggered *Reset*. Continuous adaptation may be insufficient when recent choices indicate a sharper break. We therefore maintain a separate detection memory $M_{u,t}$. For a newly observed direction $d_{u,t}$, its alignment with prior detection memory is $c_{u,t} = d_{u,t}^{\top} M_{u,t-1} / \|M_{u,t-1}\|_2$ when $M_{u,t-1} \neq 0$, and $c_{u,t} = 0$ otherwise. We accumulate disagreement through $C_{u,t} = \max\{0, C_{u,t-1} + b - c_{u,t}\}$ and raise an alarm when $C_{u,t}$ exceeds a fitted threshold h. After an alarm, *reset* reconstructs the user-memory state from the most recent k_{back} choices together with the current choice, then resumes the fitted population decay rule. Because the alarm uses the choice observed at step t, it can affect prediction only from $t+1$. An alarm is evidence produced by the detector, not an oracle indicator of true preference change.

3.2 PROSPECTIVE TESTS

The following tests separate temporal signal, individual actionability, predictive staleness, intervention choice, and interpretation of adaptive state.

Does the Order of the History Carry Predictive Information? We first ask whether chronology contains information beyond the unordered collection of choices. For memory policy m and order perturbation π, order-carried information (OCI) is $\text{OCI}_{\pi}(m) = \text{Acc}(m \mid H) - \mathbb{E}_{\pi}[\text{Acc}(m \mid \pi(H))]$. *Full shuffle* removes chronology, *reversal* reverses temporal direction, *block shuffle* preserves local sequences while disrupting global order, and *freeze-recent-k* preserves the most recent $k \in \{4, 8, 16\}$ choices while shuffling the earlier prefix. An order-agnostic policy such as retain must have zero OCI. Positive OCI establishes predictive information in chronology, not that this information is sufficient for an individual memory decision.

Can Apparent Preference Change Be Distinguished from Noise? Order can matter even when a particular user's apparent change is largely sampling noise. Let $\bar{d}_u^{(1)}$ and $\bar{d}_u^{(2)}$ be the mean signed preference directions in the first and second halves of user u's history. Their directional difference is $D_u = (1 - \cos(\bar{d}_u^{(1)}, \bar{d}_u^{(2)}))/2$. Short noisy histories can produce large D_u even for stable users, so we compare the observed value with values obtained after permuting that user's history. Across users, the corrected population statistic is $\Delta = |U|^{-1} \sum_{u \in U} [D_u - \mathbb{E}_\pi D_u^\pi]$, where D_u^π is the same directional difference after permutation. Under exchangeability, $\mathbb{E}\Delta = 0$. This correction removes the average finite-sample floor; it does not make one user's drift estimate precise.

Does Measured Change Predict Benefit from Adaptation? A measured change matters operationally only if it predicts improvement from changing memory. Using the same noisy outcomes to measure both drift and benefit can create a spurious association. We therefore estimate drift from odd-indexed outcomes and evaluate policy benefit on even-indexed prediction steps, then swap the roles and average. This *outcome-disjoint* comparison keeps the scored outcomes for the two statistics separate. We then test whether users with larger measured drift obtain larger gains from order-aware memory. A population-level association does not imply that drift is precise enough to select an operation for one user.

Can Earlier Evidence Select a Better Memory Rule? Individual memory choice must be made before the future outcomes used to evaluate it are known. We therefore use an earlier selection window A and a later disjoint evaluation window B. The selector bank $\mathcal{M}$ contains retain, exponential discount, and a causal recurrent expert. Let $A_u(m)$ and $B_u(m)$ denote user u's predict-before-update accuracy in the two windows. We choose $\hat{m}_u = \arg\max_{m \in \mathcal{M}} A_u(m)$ and compare its accuracy on B with the best single expert applied to all users on B. The selected expert remains fixed during B, although its memory continues to update after each scored choice. Hindsight selection using B is reported only as a retrospective reference. The exact windows are given in Appendix F.1.

Can Older Evidence Be Harmful? Recency and staleness are different. We predict the same later outcomes from an *old* history block, a *recent* block, and the *full* history, using only the corresponding history direction and no fitted population component. Let their accuracies be A_{old}, A_{recent}, and A_{full}. A positive $A_{\text{recent}} - A_{\text{old}}$ means recent evidence predicts better. We call older evidence predictively *stale* only when $A_{\text{recent}} - A_{\text{full}} > 0$, because adding the older evidence then makes prediction worse. Thus, older choices can be less informative than recent ones without being harmful to retain. Appendix F.3 gives the fixed-prefix construction.

Does a Detected Event Prescribe a Memory Response? Detection and intervention are separate decisions. In a fixed-detector comparison, the same alarm stream is supplied to two responses. Reset reconstructs memory from the recent backdated window, whereas fast discount retains the state but temporarily applies a faster decay γ_{fast} before returning to the population rate. Because the alarm times are identical, their difference isolates the response to a detected event. The response policies remain separately calibrated, so this comparison holds event selection fixed rather than forcing the two interventions to share parameters.

Does Adaptive Memory Identify Individual Change? An adaptive state may reflect evidence from one user or regularities learned across the population. We examine this distinction exploratorily on stable users. At each prediction step, we summarize LETHE's assigned memory by the median lag of its induced weighting over a fixed 64-step horizon. We compare stable-user cohorts under controllers fitted on a stable-only population, the main mixture, and a mixture containing many change-point users, then repeat the comparison without history length among the controller inputs. If stable cohorts receive systematically shorter assigned memory as change becomes more common in the training population, and this dependence weakens when history length is removed, part of the adaptive state reflects a population regularity rather than direct evidence of individual change. The cohorts are not identical histories replayed across populations, so this test supports an attribution diagnostic rather than a counterfactual claim about the same user; a follow-up scores all controllers on identical held-out histories. Appendix F.4 gives the full construction.

Evaluation Protocol. For all comparisons, information may affect a prediction only if it was available beforehand. Users are divided into disjoint fit and test populations, and shared parameters and representation transforms are estimated from fit-user prefixes. Test users are evaluated prequentially: at step t, prediction uses only $H_{u,<t}$, and the observed choice at t enters memory only after that prediction has been scored. When a method selects a memory rule for an individual user, an earlier

window determines the choice and a later disjoint window evaluates it. Section 4 specifies scoring, uncertainty estimation, and the evidence status of each analysis.

4 Experimental Setup

Natural & Controlled Histories. We first evaluate natural preference histories from PRISM response judgments, Arena votes, and PENS clicks (Kirk et al., 2024; Zheng et al., 2023; Ao et al., 2021). PRISM pairs chosen and unchosen responses from a turn, Arena uses timestamped non-tied comparisons, and PENS contrasts clicked headlines with a shared row-level unclicked pool rather than impression-specific negatives. The text encoder is the frozen `all-mpnet-base-v2`, and the Arena and PENS representation transforms are fitted from fit-user prefixes. These datasets reveal what users chose over time, but not whether an underlying preference truly changed, when such a change occurred, or whether older choices became harmful afterward. We therefore complement them with controlled synthetic histories in which the hidden preference trajectory is known to us but never revealed to the evaluated policy. Stable users do not change, ramp users move gradually, change-point users switch once, and bursty users switch repeatedly. The main population mixes all four regimes, while the change-point-heavy population deliberately places greater weight on abrupt changes. Stable and noise controls provide no true temporal change and isolate apparent drift caused by finite, noisy histories. Appendix B gives the complete generator and noise settings.

Constructed Boundaries & Reward Signals. Natural Arena histories do not reveal when user preferences change. We therefore construct controlled transitions from real Arena comparisons by joining segments from source-user-disjoint users. The transition points are known to the analysis but are not supplied to the evaluated policies as oracle alarms. These constructions test whether an observed transition necessarily favors forgetting while retaining natural response comparisons. To test sensitivity to the underlying reward representation, we additionally repeat the memory-policy comparison with five Skywork-Reward-V2 checkpoints spanning Llama and Qwen models from 1B to 8B parameters (Liu et al., 2026). These checkpoint experiments vary the reward signal while keeping the temporal-memory comparison fixed; they are sensitivity analyses rather than native personalized reward-model replications. Appendix G.2 gives the full evaluations.

Scoring & Evidence Status. Following the prospective protocol in Section 3.2, accuracy is computed predict-before-update, averaged first across scored choices within each user and then equally across users. Policy contrasts are paired. Intervals are paired user-bootstrap percentile intervals; follow-up analyses in the appendix report 95% t-intervals over seeds. Five-seed synthetic summaries additionally resample seeds because users within one generated population share fitted parameters. Arena splice intervals cluster by primary source user. The prespecified reset gate (Appendix D.3) requires a one-point gain, positive user- and seed-resampled intervals, and positive gain in at least four of five seeds; a separate target requires recovery of at least 50% of oracle headroom. Fixed-alarm actuator comparisons, recovery-horizon analyses, and adaptive-state attribution tests are exploratory (Appendix D).

5 Results

We organize the results around the three decisions introduced in Section 3: whether temporal evidence is actionable for an individual user, what memory should do after a detected event, and what an adaptive memory state actually establishes about that user.

Information Paradox. We first ask whether temporal structure exists and whether it supports a better memory decision for one user. Our synthetic tests use three complementary populations: the *main mixture* combines stable users with users whose preferences change gradually or abruptly; the *change-point-heavy mixture* puts more weight on abrupt preference shifts; and the stable/noise controls contain no true preference change, letting us measure how much apparent drift arises from finite, noisy histories alone. Shuffling earlier choices removes order-carried information (OCI) worth 0.6 to 1.7 percentage points on the main mixture and 3.7 to 4.1 points on the change-point-heavy mixture; retain is unchanged, and preserving only the latest eight choices retains 55 to 87% of the measured order benefit (Figure 2(a); Appendix F.3). This suggests that *temporal order carries predictive information, especially when abrupt change is common, and much of this information*

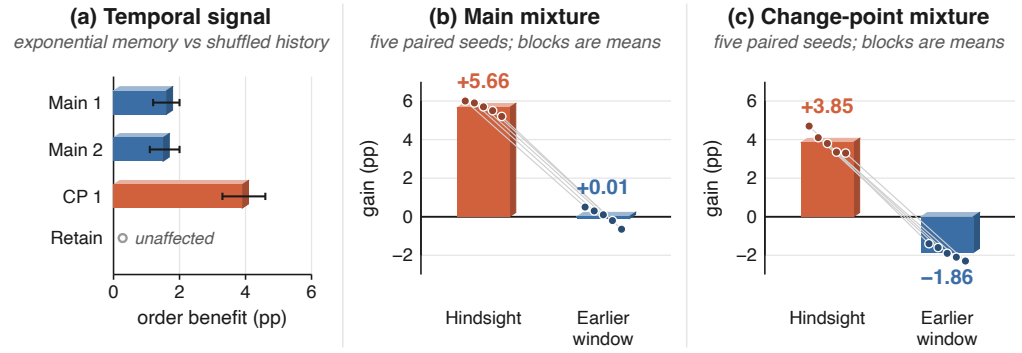


Figure 2: **Information paradox.** Temporal structure is visible at the population level but does not yield a reliable prospective memory choice for one user. **(a)** Shuffling history reduces predictive accuracy for order-aware memory, especially when abrupt changes are common, while retain is unaffected. **(b,c)** Hindsight selection finds substantially better memory rules, but this advantage largely disappears when the rule must be chosen from an earlier window. Whiskers denote 95% intervals; blocks are means over five paired seeds (dots, connected by lines).

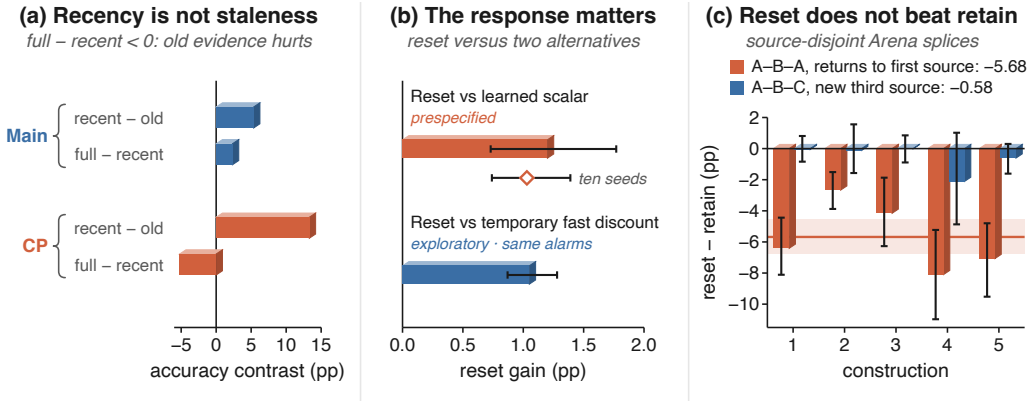


Figure 3: **Intervention paradox.** A detected change does not fix the memory response. **(a)** Recent evidence predicts better in both mixtures, but old evidence hurts only when abrupt changes are common (full − recent < 0). **(b)** In that stale regime, reset beats the learned scalar (prespecified comparison) and temporary fast discounting under identical alarms (exploratory); whiskers are seed-resampled 95% intervals. **(c)** Source-disjoint Arena splices, detected reset minus retain per construction with primary-source 95% intervals. When the final segment returns to the first user (A–B–A, 300 histories; band: pooled interval), reset hurts; when it comes from a new user (A–B–C, 500 histories, exploratory), reset does not help. Splice points are not given to the policies.

lies in recent choices. However, we also see that population-level change is also detectable without being precise for one user. Permutation correction removes the average finite-sample drift floor, but at $T = 64$ and noise scale 1 the raw statistic has a stationary floor of about 0.30, while a true rotation of magnitude 0.25 adds only about 0.01 and the between-user standard deviation is about 0.06. Outcome-disjoint analysis nevertheless links larger drift to greater adaptation benefit: in the change-point-heavy population, adaptation gains about 5.7 points for users who undergo an abrupt change but loses 2.0 points for stable users. *The population tells us that change matters, but one user's short history does not tell us precisely enough how that user changed.* The prospective selection test makes this gap explicit. Across five seeds, choosing the best memory rule after seeing the outcomes gains 5.66 points on the main mixture and 3.85 on the change-point-heavy mixture, whereas choosing from an earlier window yields only 0.01 and -1.86 points (Figure 2(b,c)). The best expert agrees across adjacent windows for only about 42 to 50% of users, and the broader adaptive comparisons

do not establish a reliable prospective per-user improvement (Appendix F.2). Scored against the generator's expected accuracy, about half of the hindsight gain is selection optimism, and a reliable per-user choice would need a median of 130 and 258 outcomes against selection windows of 16 and 24 (Table 14). Hence, *knowing that temporal change exists in the population is not enough to choose the right memory rule for a particular user before future outcomes arrive.*

Takeaway. These results reveal an *Information Paradox*: population-level temporal signal does not translate into a reliable earlier-window choice of the memory rule that performs better later. This is an estimability result for the evaluated histories and methods, not an impossibility claim.

Table 1: Memory policies under individual temporal regimes. Each population contains only the stated pattern. Entries are mean accuracies (%) over three seeds; the comparisons are descriptive.

Temporal pattern	Population discount	Learned scalar	LETHE	Event reset
Stable (no change)	76.8	76.9	76.8	76.8
Ramp (gradual change)	71.3	71.2	69.6	71.3
Change-point (one switch)	67.5	67.8	66.2	69.4
Bursty (repeated switches)	60.4	60.7	61.2	60.4

Intervention Paradox. We next ask whether detecting change determines what memory should do. Figure 3(a) separates *recency* from *staleness*. On the main mixture, recent evidence exceeds old evidence by 5.35 points, yet the complete history still exceeds recent-only prediction by 2.33 points. On the change-point-heavy mixture, recent exceeds old by 13.31 points and recent-only exceeds the complete history by 5.22 points. *Newer evidence can be better without older evidence being harmful; forgetting is justified only in the latter case.*

In this targeted stale regime, the prespecified event-reset comparison improves over the best non-reset policy by 1.2 points, with seed-resampled 95% CI $[0.73, 1.77]$ and positive effects in all five seeds, although it recovers only 34% of the specified oracle headroom; ten seeds give 1.03 $[0.74, 1.39]$, all positive, Holm-adjusted $p = 0.001$. On the broader main mixture, the gain is only 0.14 points at five seeds, with a user interval that includes zero, and 0.30 $[0.14, 0.46]$ at ten. Under the same detected alarms, the exploratory fixed-detector comparison further shows reset exceeding temporary fast discount by 1.05 points, again positive in all five seeds (Figure 3(b)). The better response depends on alarm delay: with oracle alarms, reset minus fast discount is -0.28, $+0.53$ and $+1.04$ points at delays of 0, 4 and 8 steps (Table 9). *When old evidence is stale, the better response depends on how late the alarm arrives.*

The boundary is equally important. Reset also needs enough post-change evidence to rebuild from (Appendix E.3). On the source-user-disjoint Arena construction, detected reset is 5.68 points below retain, with 95% CI $[-6.79, -4.54]$ and negative effects in all five constructions, and 3.01 points below continued population discount (Figure 3(c)). The constructed boundaries are not supplied as oracle alarms. When the final segment comes from a third source user instead, reset is not worse than retain (-0.58 over all histories and $+0.21$ $[-0.76, 1.19]$ on spliced histories alone; Appendix G.1). *A detected event can therefore favor forgetting in one regime and preserving history in another.* We also observe that *reset has higher mean accuracy than* LETHE *under a single switch, but lower mean accuracy under repeated switches* (Table 1).

Takeaway. These results reveal an *Intervention Paradox*: detecting change does not prescribe the memory response. Change is not equivalent to staleness, and evidence for an event is not itself evidence for forgetting.

Illusion Of Personalization. We finally test whether an adaptive user-specific state is itself evidence that the individual changed. In this exploratory analysis, even stable users receive substantially shorter assigned memory when abrupt change is more common in the training population (Table 17). Scored on identical held-out histories, stable users receive a median memory of 29, 24 and 16 steps from controllers trained on the stable-only, main and change-point-heavy populations, whereas a Bayesian change-point learner given each true hazard keeps 30 to 31; stable users lose 0.70 and 2.30 points while change-point users gain. Removing history length from the controller then improves stable-user accuracy while reducing change-point-user accuracy, with almost no aggregate change. *Part of*

the individualized adaptation therefore behaves like a population-level expectation of change that redistributes accuracy between stable and changing users.

Takeaway. These exploratory results reveal an *Illusion of Personalization*: user-specific adaptive states can partly encode population regularities rather than reliable evidence that the corresponding user changed.

Coverage Across Natural Histories And Reward Signals. The natural histories reinforce the same boundaries. PRISM provides little separation among memory rules. In causal Arena, the intervals for both comparisons in Table 2 include zero. *These short histories do not establish a preferred temporal memory rule.* In the PENS evaluation, LETHE improves over retain and fixed

Table 2: Memory-policy comparisons on natural histories. Entries are accuracy differences in percentage points with reported paired intervals; positive values favor the first policy. The learned scalar uses one learned discount timescale. Causal Arena: 53 test users; PENS: 1,800 test users.

Dataset	Comparison	Difference	Reported interval
Causal Arena	LETHE − online discount	$+0.16$	$[-2.37, 2.49]$
	Event reset − learned scalar	-1.49	$[-4.1, 1.32]$
PENS	LETHE − retain	$+1.71$	$[1.55, 1.88]$
	LETHE − fixed population discount	$+0.15$	$[0.06, 0.23]$
	LETHE − learned scalar	-0.23	$[-0.31, -0.16]$
	Event reset − learned scalar	-0.34	$[-0.4, -0.27]$

population discount, but a single learned timescale outperforms both LETHE and event reset. On PENS the fitted reset effectively never triggers, so it behaves like continued discounting, and per-user selection also fails: hindsight gains 1.86 points, below an exchangeable-label null of 3.17, and earlier-window choice gains -0.11 to $+0.01$ (Appendix F.1). *PENS supports downweighting older evidence, but not an advantage for multiscale mixing or reset over a single learned timescale.* Across five Skywork-Reward-V2 checkpoints spanning Llama and Qwen models from 1B to 8B on PRISM and Arena, every LETHE-versus-retain interval includes or touches zero. These are checkpoint-sensitivity tests, not native personalized reward-model replications. *The checkpoint comparisons do not establish a consistent gain from* LETHE *over retention.*

6 DISCUSSION & LIMITATIONS

Across the three paradoxes, the central message is that temporal signal, individual action, and interpretation are different questions.

Change detection, memory intervention, and state interpretation should therefore be evaluated separately. Note that this conclusion is scoped. The strongest identification comes from controlled synthetic histories, natural histories often provide weaker separation, the tested predictor and policies cover only a limited design space, and the Illusion of Personalization analysis is primarily exploratory. We do not claim that temporal personalization is impossible or that any memory policy is universally preferable. Instead, we stress that *each adaptive decision requires evidence beyond the mere presence of temporal change*.

7 CONCLUSION

Temporal personalization is usually framed as detecting when preferences change. Our results show this is only the first step: population signal need not support a prospective memory choice for one user, a detected change does not fix the response, and an adaptive state need not mean the user changed. *Detecting preference change is not enough; a personalized system must also know what that change justifies.*

AI USE STATEMENT

We used generative AI tools for language editing, LaTeX and figure preparation, auditing and writing analysis code, and discussing hypotheses and the interpretation of results. The authors ran the experiments, checked every reported number against the saved records, and take full responsibility for the claims, code, and submission.

ETHICS STATEMENT

The study evaluates offline preference prediction from existing datasets. A false change alarm can erase useful evidence or misrepresent a stable preference. A deployed system would additionally require user control, privacy protections, and subgroup assessment, none of which is established here. Raw user text and identifiers are not included in the paper's figure-reproduction bundle. Dataset reuse and redistribution remain subject to the source terms.

REPRODUCIBILITY STATEMENT

Code, configurations, and processed evaluation records accompany the submission. Appendices A–I specify the prediction protocol, data construction, memory policies, fitting procedures, and evaluation methods; Appendix L lists the released artifacts. Reproduction of natural-history and checkpoint experiments additionally requires access to the corresponding third-party datasets and model checkpoints under their original distribution terms.

REFERENCES

Ryan P. Adams and David J. C. MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. PENS: A dataset and generic framework for personalized news headline generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(70):2079–2107, 2010.

Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Korlakai Vinayak. PAL: Sample-efficient personalized reward modeling for pluralistic alignment. In *International Conference on Learning Representations (ICLR)*, 2025.

A. Philip Dawid. Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society: Series A*, 147(2): 278–290, 1984.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. How to evaluate reward models for RLHF. In *International Conference on Learning Representations*, 2025.

Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32:151–178, 1998.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Rudolf Kulhavý and Martin B. Zarrop. On a general concept of forgetting. *International Journal of Control*, 58(4):905–924, 1993.

Allison Lau, Younwoo Choi, Vahid Balazadeh, Keertana Chidambaram, Vasilis Syrgkanis, and Rahul G. Krishnan. Personalized adaptation via in-context preference learning. *arXiv preprint arXiv:2410.14001*, 2024.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, and Yang Liu. Skywork-Reward-V2: Scaling preference data curation via human-AI synergy. In *International Conference on Learning Representations (ICLR)*, 2026.

Qiyao Ma, Dechen Gao, Rui Cai, Boqi Zhao, Hanchu Zhou, Junshan Zhang, and Zhe Zhao. Personalized RewardBench: Evaluating reward models with human aligned personalization. *arXiv preprint arXiv:2604.07343*, 2026.

Hyunji Nam, Yanming Wan, Mickel Liu, Peter Ahnn, Jianxun Lian, and Natasha Jaques. Learning to summarize user information for personalized reinforcement learning from human feedback. In *International Conference on Learning Representations (ICLR)*, 2026.

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

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing reinforcement learning from human feedback with variational preference learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Kwangwook Seo and Dongha Lee. P-Check: Advancing personalized reward model via learning to generate dynamic checklist. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 43447–43471, 2026.

Jingyan Shen, Jiarui Yao, Rui Yang, Yifan Sun, Feng Luo, Rui Pan, Tong Zhang, and Han Zhao. MiCRo: Mixture modeling and context-aware routing for personalized preference learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

Seongho Son, William Bankes, Sayak Ray Chowdhury, Brooks Paige, and Ilija Bogunovic. Right now, wrong then: Non-stationary direct preference optimization under preference drift. In *International Conference on Machine Learning (ICML)*, 2025.

Joe Suk and Arpit Agarwal. When can we track significant preference shifts in dueling bandits? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Robert C. Wilson, Matthew R. Nassar, and Joshua I. Gold. Bayesian online learning of the hazard rate in change-point problems. *Neural Computation*, 22(9):2452–2476, 2010.

Pinyi Zhang, Ting-En Lin, Yuchuan Wu, Jingyang Chen, Zongqi Wang, Hua Yang, Bing Zhao, Fei Huang, Yongbin Li, and Kai Zhang. P-GenRM: Personalized generative reward model with test-time user-based scaling. In *International Conference on Learning Representations*, 2026a.

Youyuan Zhang, Siyuan Li, Fangming Liu, and Jing Li. Toward robust personalized alignment for LLMs: Mitigating persona drift in multi-turn dialogue. *arXiv preprint arXiv:2609.12373*, 2026b.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

SUPPLEMENTARY MATERIAL CONTENTS

A Prediction Task And Causal Protocol **13**
A.1 Notation And Scoring . . . 13
A.2 Temporal Provenance . . . 13
A.3 Population And Within-User Splits . . . 13

B Synthetic Data Generator **13**

C Policies And Memory Operations **14**
C.1 Baseline Ladder . . . 14
C.2 LETHE Trace Bank And Controller . . . 15
C.3 Event Detector And Actuators . . . 16

D Diagnostics And Statistical Inference **17**
D.1 Order Sensitivity And Its Null Control . . . 17
D.2 Permutation-Centered Change And Outcome Separation . . . 17
D.3 Resampling Units And Decision Criteria . . . 18

E Complete Synthetic Results **18**
E.1 Fixed-Detector Actuator Comparison . . . 20
E.2 Regime Boundaries And Ablations . . . 21
E.3 Recovery After A Change . . . 22

F Additional Diagnostics And Attribution Analysis **22**
F.1 Prospective Selection Records . . . 22
F.2 Completed Adaptive Comparisons . . . 24
F.3 Noise, Locality, And Staleness . . . 25
F.4 Exploratory Illusion Of Personalization . . . 25

G Natural-History And Checkpoint Results **27**
G.1 Source-User-Disjoint Arena Constructions . . . 28
G.2 Checkpoint-Based Memory Comparisons . . . 29

H Dataset Construction And Preprocessing **30**
H.1 Corpus-Specific Pair Construction . . . 30
H.2 Representation Fitting And Split Alignment . . . 30
H.3 Checkpoint Interface . . . 31

I Training, Hyperparameters, And Compute **31**
I.1 Baseline Fitting And Model Selection . . . 31
I.2 Execution Environment And Recorded Compute . . . 31

J Evaluation Metrics And Reporting Conventions **31**

K Kernel Representation Analysis And Scope **32**
K.1 A Separate Two-Timescale Model . . . 32
K.2 From A Predictor To Its Lag Kernel . . . 33
K.3 Numerical Coverage And Its Limits . . . 33

L Reproducibility And Artifact Coverage **34**

M Asset Usage, Ethics, And Intended Scope **34**

N Scope Of The Empirical Claims **35**

O Extended Related Work And Evaluation Map **35**

A Prediction Task And Causal Protocol

A.1 Notation And Scoring

Each prediction compares two known candidates before the user's choice is available. Let $x_{u,t} = e(a_{u,t}) - e(b_{u,t}) \in \mathbb{R}^p$ be their representation difference and let $y_{u,t} \in \{-1, +1\}$ record which candidate the user later chooses. After that choice is observed, $d_{u,t} = y_{u,t}x_{u,t}$ points in the chosen direction. Concretely, $y = +1$ leaves x unchanged and $y = -1$ reverses it. This direction can update later memory, but it is unavailable when the current prediction is made.

Equation 1 defines the current logit $\ell^m_{u,t}$ from the strictly earlier history. After the label arrives, the signed margin is $y_{u,t}\ell^m_{u,t} = d^\top_{u,t}w + \alpha_m d^\top_{u,t}s^m(H_{u,<t})$. A positive margin means that the earlier prediction agreed with the observed choice. This multiplication is used to score a prediction already made; it does not feed the current label back into that prediction.

For a nonempty set of scored steps E_u, $\mathrm{Acc}_u(m; E_u)$ is the fraction of those steps with positive signed margin. For example, seven correct choices out of ten scored steps give user accuracy 0.7. Population accuracy first computes this fraction for each user and then averages across users, so a user with a longer scored history does not automatically receive more evaluation weight. The selector notation $A_u(m)$ and $B_u(m)$ uses the same accuracy on its selection and evaluation windows. Zero-margin ties count as incorrect. Likelihood fits parameters, whereas accuracy evaluates the improvement gates. Candidate-specific scales α_m are calibrated on permitted fit data.

A.2 Temporal Provenance

The timing contract in Table 3 distinguishes the state used to predict at step t from the state formed after observing $y_{u,t}$. An alarm that uses the newly observed choice can therefore first affect prediction at $t + 1$. Representation transforms, population parameters, and shared normalization are fitted using permitted fit-user prefixes and then applied to held-out users. Corpus-specific preprocessing and representation fitting are described in Appendix H.2.

A.3 Population And Within-User Splits

The main evaluations allocate 40% of eligible users to fitting and the remainder to testing. Within a user, evaluation begins at

$$t_u^{\mathrm{eval}} = \max\{8, \lfloor T_u/2 \rfloor\}.$$

Thus, a length-64 history uses its first 32 observations as a prefix and scores observations 33 to 64. Population parameters use fit-user prefixes, while per-user rate selection uses only the corresponding user's permitted earlier prefix. During evaluation, each label enters memory only after its prediction is scored. The three-expert selector uses a separate earlier-to-later arrangement. Experts are fitted using fit-user prefixes, an earlier window A provides individual selection evidence, and a later disjoint window B evaluates that choice. In the main mixture, the one-based blocks are 33 to 48 and 49 to 64; in the change-point mixture, they are 49 to 72 and 73 to 96. The best constant expert on B is a retrospective diagnostic reference rather than a deployable selector. Appendix F.1 reports the corresponding seed-level results. The odd/even drift-benefit diagnostic is not this earlier-to-later selector. It computes drift from one parity of outcomes and benefit from the other, reverses the parities, and averages the two associations. The policy states can still include earlier observations of either parity. This construction separates the scored outcomes used for the two statistics without establishing complete source independence. In the single-change generator, a change can also fall inside A while B is entirely post-change. Consequently, failure of an A-selected expert to transfer to B need not be caused by sampling noise alone.

B Synthetic Data Generator

Synthetic histories let us know the user's latent preference and true change time for evaluation, while keeping both hidden from every deployable policy. Each response has 48 attributes. A shared unit direction q contributes population-wide quality, while an eight-dimensional subspace contains the

Table 3: Prediction-time provenance contract. A maximum label time of $t-1$ excludes the current and future labels. The post-label statistic $C_{u,t}$ may trigger an action for $t+1$, not revise the prediction at t. Projection conformance is assessed separately in Appendix H.2.

Quantity	**Permitted source**	**Maximum label time**
Current comparison $x_{u,t}$	current candidate pair	none
Retain, discount, and trace states	signed prefix directions	$t-1$
Trace residual features	earlier scored trace predictions	$t-1$
Drift and change-point features	prefix summaries	$t-1$
Pre-prediction statistic $C_{u,t-1}$	earlier detection-memory innovations	$t-1$
Reset-aware prediction state	alarms declared before prediction	$t-1$
Controller normalization	training rows from fit users	fit data only
Text projection (required contract)	aligned fit-user prefix rows	fit data only
Population hyperparameters	fit-user selection windows	fit data only

user-specific preference. The linear reward is $r_{u,t}(a) = \theta_{u,t}^{\top} a + 0.3 q^{\top} a$. The nonlinear stress test additionally uses $0.5 a^{\top} Q_u a$, with Q_u restricted to the preference subspace.

The generator separates two kinds of uncertainty. First, even if one candidate has higher latent reward, the user need not choose it every time. For candidates a and b,

$$\Pr(a \succ b \mid u, t) = \sigma\left(\frac{r_{u,t}(a) - r_{u,t}(b)}{\tau}\right), \qquad \tau = 0.2. \tag{2}$$

For example, a reward advantage of 0.2 produces logit 1 and choice probability $\sigma(1) \approx 0.73$, not certainty. Second, the policy does not observe the clean 48-dimensional attributes. The synthetic encoder applies a fixed orthogonal map, adds isotropic Gaussian perturbation with coordinate standard deviation $0.2/\sqrt{48}$, and unit-normalizes the result. Before normalization, that perturbation has root-mean-square norm 0.2. Choice noise therefore changes which candidate may be selected; representation noise changes what the policy sees about the candidates.

Each user begins at a unit preference vector $\theta_{u,0}$. A second state is sampled at angle $m_u \pi$, with m_u uniform over the configured magnitude range. Stable users keep the first state. Ramp users interpolate on the unit sphere toward the second. Change-point users switch once at a uniformly sampled integer step in $[T/4, 3T/4)$. Bursty users switch between the two states with probability 0.15 per step. The schedule label, magnitude, true state, and true change time are withheld from deployable policies and supplied only to the corresponding diagnostic or oracle. *Known synthetic change permits a check of an adaptive state's interpretation. It does not give the fitted policy an oracle event signal.*

The generator configurations are recorded in Table 4 in Appendix B. The pool contains 120 prompts and eight responses per prompt. The held-out-pair count is a generator field for separate response-comparison diagnostics, not the number of held-out users or the later scored-window length. The principal mixture ladders use 200 fit and 300 test users per seed. Predict-before-update evaluation begins only after at least eight observations. The order and drift diagnostics use eight permutations per user. The single-regime, noise-null, and nonlinear configurations address specific boundaries rather than redefine the principal comparison.

C Policies And Memory Operations

C.1 Baseline Ladder

The policies in Table 5 in Appendix C.1 test different uses of the same observed preference history. Retain anchors order invariance. The population, per-user, and online rates test whether one timescale should be shared, estimated individually, or updated sequentially. Bayesian online change-point detection (BOCPD) instead averages over inferred run lengths. Learned scalar and mixture controllers test whether prefix features can allocate timescales. Recurrent controls test a different state representation, and reset variants separate event timing from its implemented response. The complete results and their noninterchangeable gates are in Appendices E and F.1. These are implemented comparators, not exhaustive tests of their model classes.

Table 4: Canonical synthetic configurations. All use a 48-dimensional observed space and an eight-dimensional preference subspace, and an eight-dimensional preference subspace.

Population	Users	T	Held-out pairs	Fit frac.	Mix S/R/C/B	Magnitude	Role
Main mixture	500	64	40	.40	.35/.25/.25/.15	$[0, 1]$	broad diagnostic
Change-point mix	500	96	20	.40	.20/.20/.50/.10	$[.5, 1]$	reset test
Pure regimes	500	96	20	.40	one regime	$[.5, 1]$	boundary
Noise null	300	64	20	.40	1/.0/.0/.0	0	falsification
Nonlinear arm	400	64	40	.40	.35/.25/.25/.15	$[0, 1]$	reward check

Table 5: Policy ladder and the question isolated by each rung. Oracle rows are privileged reference policies, not upper bounds on every temporal policy.

Rung	Memory state	Scientific role
Set	Unweighted mean of the prefix	retain and order anchor
Population discount	One fitted exponential rate	stable forgetting
Per-user exponential moving average (EMA)	Prefix-selected rate, raw or shrunk	individual scalar
Bayesian online change-point detection (BOCPD)	Bayesian run-length-weighted state	change adaptation
Online gamma	incremental delta-bar-delta (IDBD) causal rate update	online scalar
Single gamma	Learned scalar in the trace bank	controller reduction
LETHE	Prefix-conditioned mixture of six traces	continuous mix
Causal gated recurrent unit (GRU)	Recurrent state over raw directions	sequence baseline
Same-statistics GRU	Recurrent state over LETHE features	capacity control
Block reset	Reset after block-cosine change	event heuristic
Event reset	cumulative-sum (CUSUM) alarm and backdated reset	fitted reset
LETHE-E	Head over reset-aware traces	reset plus mix
Group oracle	Rate selected with true schedule	group ceiling
Change-point oracles	True boundary with delays 0, 4, 8, or 16	event ceilings
Regime oracle	Best operation for true schedule	broad ceiling

Population discount searches $\{0.5, 0.7, 0.8, 0.85, 0.9, 0.93, 0.95, 0.97, 0.985, 0.999\}$. Raw per-user rates maximize that user's permitted prefix likelihood. Shrinkage adds 20 pseudo-observations carrying the population's mean prefix likelihood. The online-rate candidate updates its decay rate after each observation with an incremental delta-bar-delta (IDBD) meta-gradient step. Learned `single_gamma` is a different candidate from this per-user rate grid. It uses the scalar controller described below, including its user-balanced loss. Appendix I.1 distinguishes these fitting paths.

C.2 LETHE Trace Bank And Controller

The trace bank keeps several summaries of the same past choices, each forgetting at a different speed. Its six rates are

$$\{0.5,\ 0.8,\ 0.92,\ 0.969,\ 0.9886,\ 1\}.$$

Traces start at zero and use the recurrence defined in Section 3.1. A rate of 1 gives every earlier direction the same raw weight. With rate 0.5, a direction from five steps ago receives raw weight $0.5^5 = 1/32$ relative to the newest one. Intermediate rates fill the gap between these behaviors.

At prediction time, each trace contributes its strictly earlier normalized state, with a zero trace contributing zero.

Six residual features track earlier trace-prediction correctness with persistence 0.9 and initial value 0.5. Nine further statistics encode history length, split-half and recent directional differences, permutation correction, odd/even instability, recent-versus-long-history disagreement, the largest previous window jump, and static/recency prediction residuals. The blockwise configuration refreshes these nine statistics every eight observations. Residual updates and feature refreshes use only outcomes already observed. The controller has 15 inputs and a 16-unit hidden layer. Its mixture head produces six softmax weights, while its scalar head interpolates between adjacent traces.

Training uses at most 300 Adam steps, learning rate 0.05, patience 30, output shrinkage 10^{-2}, and weight shrinkage 10^{-4}. Validation holds out 15% of fit users, with at least one user on each side when possible. Feature normalization uses training rows rather than validation or test rows. Inverse per-user row counts give equal fit-user contribution to controller training. The output prior favors the longest trace. For fixed trace states, the score is a weighted sum of the mixture weights. The logistic training loss is nonlinear, so this controller training is not guaranteed to have a single global optimum.

Assigned Memory. The plotted memory summary answers a simple question: where does the controller place half of its nominal weight over the last 64 lags? Let $j = 0$ denote the most recent earlier observation. For trace k, the normalized weight at lag j is $e_k(j) = \gamma_k^j / \sum_{i=0}^{H-1} \gamma_k^i$ with $H = 64$. The controller combines these trace-specific curves as $\kappa_t(j) = \sum_k \pi_{k,t} e_k(j)$. Its median lag is the smallest j for which the cumulative mass reaches $1/2$. For example, if a toy kernel put masses 0.4, 0.35, and 0.25 on lags 0, 1, and 2, its median lag would be 1 because the first two lags already contain 0.75 of the mass.

Nonnegative controller weights summing to one make κ_t a nonnegative distribution over the 64 displayed lags, so this median always exists.

Operative Lag Contributions. The nominal kernel is not identical to the weighting that appears in the predictor because each trace state is normalized before scoring. Collecting the coefficient of an earlier direction therefore gives the operative lag contribution defined below. This distinction matters for interpretation: assigned median lag describes how the controller allocates mass across timescales, not the exact causal influence of every past choice on the current prediction.

For nonzero trace states, collecting the coefficient of direction $d_{u,t-1-j}$ gives

$$a_t(j) = \sum_k \frac{\pi_{k,t}\gamma_k^j}{\|S_{u,k,t-1}\|_2}, \qquad q_t(j) = \frac{a_t(j)}{\sum_{i=0}^{H-1} a_t(i)}, \quad 0 \le j < H. \tag{3}$$

The first expression follows by substituting $S_{u,k,t-1} = \sum_{j=0}^{t-2} \gamma_k^j d_{u,t-1-j}$ into the LETHE mixture s^{mix} of Section 3.1 and collecting the coefficient multiplying each earlier direction. The second simply rescales the displayed 64-lag coefficients to sum to one when their denominator is positive. A trace whose state is zero contributes zero to the score.

We report the median lag of these coefficients averaged over evaluated steps. Its fixed 64-lag horizon may omit still older observations or include lags that were not yet available early in a history. *Assigned half-life therefore summarizes the controller's nominal allocation; it is not a recovered user timescale or a complete measure of how each past choice affects the prediction.* Reset-aware traces also require their actual retained segments. Changing a past observation could change the controller and the trace norms themselves, so these coefficients are not derivatives with respect to that observation.

C.3 Event Detector And Actuators

The detector asks whether a newly observed choice points away from the recent direction of the user's earlier choices. Its detection memory M_t is separate from the prediction state, and C_t stores accumulated disagreement. At an eligible step,

$$c_t = d_t^\top \operatorname{unit}(M_{t-1}), \qquad C_t = \max\{0, C_{t-1} + b - c_t\}.$$

An alarm is declared only when $C_t > h$, after the current choice has been observed, so it can first affect prediction at $t + 1$.

Table 6 summarizes how the fitted detector behaves on the ten-seed ladders: it catches 79% of switches within 24 steps in the change-point-heavy mixture with a median delay of about seven steps, which is the delay regime studied in Appendix E.1.

Table 6: Operating characteristics of the fitted detector on test users of the ten-seed ladders. Detection counts an alarm within 24 steps of a true switch; rates are per 100 steps.

Mixture	Seeds	Detection rate	Median latency	False alarms (pre-change)	False alarms (stable)	Fires (ramp, bursty)
Change-point-heavy	10	0.79	7.1	0.19	0.10	0.52, 2.29
Main	10	0.42	7.3	0.27	0.13	0.44, 1.36

After an alarm, reset reconstructs prediction memory from the preceding k_{back} observations together with the current direction, clears the detector statistic, and later resumes detection according to the fitted eligibility rule.

Temporary fast discount receives the same detected alarms as reset, preserves prediction memory, applies a faster decay for a fitted duration, and then resumes the population rate. Random and periodic resets instead alter intervention timing and therefore serve as timing controls rather than shared-alarm response comparisons.

D Diagnostics And Statistical Inference

D.1 Order Sensitivity And Its Null Control

An order-sensitive predictor should lose information when chronology is disrupted without changing which observations it receives. OCI compares accuracy using each target's available prefix with accuracy after an order perturbation of that prefix. The target itself is fixed and never enters its own history. Full shuffle destroys chronology, reversal reverses it, block shuffle preserves local neighborhoods, and recent-preserving controls leave the final $k \in \{4, 8, 16\}$ observations unchanged. The fitted parameters are held fixed during this diagnostic. OCI is policy- and perturbation-specific, not an intrinsic information measure of the dataset.

Retain Invariance. Retain is insensitive to the order of a fixed set of earlier directions because it depends only on their sum. Therefore, any permutation of the same prefix leaves its memory state, prediction, and correctness unchanged. Retain consequently provides an exact zero-OCI control. Passing this control says nothing about whether order-aware policies can exploit temporal information.

D.2 Permutation-Centered Change And Outcome Separation

A stable user can look different in the first and second half of a short history simply by chance. Imagine eight choices drawn from one unchanged preference. Four happen to point slightly left of the true direction and the next four slightly right. Comparing the two half-means then produces an apparent change even though the latent preference never moved.

For a prefix with at least two observations, let $\bar{d}_u^{(1)}$ and $\bar{d}_u^{(2)}$ be the means before and after its midpoint. We measure their directional disagreement by $D(H_u) = [1 - \mathrm{unit}(\bar{d}_u^{(1)})^\top \mathrm{unit}(\bar{d}_u^{(2)})]/2$, which lies in $[0, 1]$. Prefixes shorter than two observations return zero.

Why Permutation Centering Has Zero Mean Under Exchangeability. If chronology carries no information for a user, randomly reordering the same observed directions should not change the distribution of this split-half statistic. Formally, assume that the directions are exchangeable under the within-user permutations used by the diagnostic. For an independent uniform permutation π, H_u and $\pi(H_u)$ then have the same distribution, so $\mathbb{E}D(H_u) = \mathbb{E}_{H_u,\pi} D(\pi(H_u))$. Subtracting the shuffled reference and averaging across users therefore gives $\mathbb{E}\Delta = 0$. A finite number of random shuffles still has this expectation, although the realized correction has Monte Carlo noise.

This correction removes the average split-half floor created by finite samples. It does not make one user's change estimate noise-free. A positive value can also come from changing prompts or candidate composition, not only from a changing latent preference.

The drift-benefit diagnostic avoids using the same scored outcomes in both quantities. One pass estimates drift from odd-indexed outcomes and evaluates the policy benefit on even-indexed prediction steps. The second reverses these roles, and the reported association averages the passes. Earlier observations from either parity may still enter causal policy states. This is outcome separation, not independent histories or an earlier-to-later forecast. The stationary controls and observed associations are interpreted under that narrower design in Appendix F.3.

D.3 Resampling Units And Decision Criteria

Within a paired policy contrast, the two policies are evaluated on the same users. The bootstrap therefore resamples users as pairs rather than drawing each policy's outcomes separately. If user 17 is selected in a bootstrap draw, both policy outcomes for user 17 travel together. Intervals use 1,000 (order and drift diagnostics) or 2,000 (all other comparisons) paired percentile-bootstrap draws. Five-seed aggregates also resample seeds because users from one seed share a generated population and fitted components. An all-positive result across five independent symmetric signs has probability $1/32$ for a prespecified positive direction and $1/16$ for either direction. Direction consistency is useful alongside effect sizes, but five positive signs alone do not produce a two-sided sign-test result below 0.05.

For the source-user-disjoint Arena constructions, uncertainty is clustered by primary source user so that pseudo-user instances sharing that source remain together. This does not capture every possible dependency through secondary sources, so the resulting intervals are interpreted conditional on the constructed histories. The prespecified reset criterion requires at least one percentage point over the best non-reset policy, positive user intervals, positive intervals obtained by resampling seeds, and at least four positive seed effects. A separate gate requires 50% recovery of the specified change-point oracle headroom. Those are different targets with different recorded outcomes. Holm corrections apply within each prespecified family. *An interval containing zero withholds evidence of a directional difference. It does not establish equivalence or the absence of usable preference information.*

E Complete Synthetic Results

We compare memory policies on 300 test users per seed, with 200 disjoint fit users and five seeds for each mixture. Table 7 reports user-macro predict-before-update accuracy: each choice is scored before its label is allowed to update memory. On the main mixture, population discount improves over retain, although the learned mixture does not exceed online discount. On the change-point mixture, event reset exceeds the learned scalar by 1.201 points. The prespecified user interval is $[0.986, 1.413]$ and the interval obtained by resampling seeds is $[0.728, 1.765]$. Rounded per-seed gains are 0.44, 0.94, 1.42, 0.88, and 2.32 points. The accuracy gate passes, while 34% recovery of the specified change-point oracle headroom misses the separate 50% target. On the main mixture, reset gains only 0.138 points over online discount and its user interval includes zero.

Extending both ladders to ten seeds (seeds 0 to 9; the recurrent rungs omitted) gives, on the change-point mixture, 1.03 points for event reset over the learned scalar (user interval $[0.87, 1.19]$, seed interval $[0.74, 1.39]$, ten of ten seeds positive; sign-flip $p = 0.0005$, Holm-adjusted $p = 0.001$ across the two mixtures) and 32% of change-point-oracle headroom. On the main mixture, event reset beats online discount by 0.30 $[0.14, 0.46]$, with nine of ten seeds positive, which remains below the one-point criterion. LETHE trails the best baseline at ten seeds in both mixtures, by 1.02 $[0.85, 1.20]$ points on the change-point mixture and 0.34 $[0.19, 0.50]$ on the main mixture.

Oracle rows are privileged reference procedures, not upper bounds on every memory policy. In particular, exact boundary access does not make the implemented reset operator optimal, and its mean can be below a deployable policy's mean. *The positive reset result is specific to the targeted population and its prespecified comparator, not a universal ordering of memory operations.*

Table 7: Five-seed user-macro prequential accuracies as proportions, with 300 test users per seed and mixture. The privileged oracle and supervised-detector rows are specified reference procedures, not universal ceilings.

Policy	Main mix	Change-point mix	Role
Set	.692	.630	retain
Population discount	.702	.679	discount
Per-user EMA, raw	.690	.655	individual rate
Per-user EMA, shrunk	.696	.661	individual rate
BOCPD	.694	.661	run-length
Online gamma	.704	.679	online rate
Single learned gamma	.698	.682	scalar controller
LETHE	.699	.671	continuous mix
Block reset heuristic	.685	.677	heuristic event
Supervised detector	.702	.680	labelled reference
Soft event reset	.707	.695	soft ablation
Event reset	.706	.694	prespecified reset
LETHE-E scalar	.702	.691	reset scalar
LETHE-E mixture	.703	.688	reset mixture
Group oracle	.703	.689	group reference
True change oracle	.672	.689	boundary reference
Delayed oracle, 4	.690	.695	delay reference
Delayed oracle, 8	.698	.695	delay reference
Delayed oracle, 16	.700	.687	delay reference
Regime oracle	.711	.714	regime reference

Table 8: Exploratory actuator contrasts on the change-point mixture. Gains are percentage points and intervals are 95% seed-cluster intervals over five seeds. Only reset versus temporary fast discount shares the intervention times. Random and periodic reset rows test timing under their matching rules.

Contrast	Mean gain	95% seed CI	Positive seeds
Reset − temporary fast discount	1.049	[0.872, 1.279]	5/5
Reset − random reset	1.060	[0.660, 1.569]	5/5
Reset − periodic reset	1.225	[0.840, 1.753]	5/5

E.1 Fixed-Detector Actuator Comparison

When alarm times are fixed, do the fitted reset and temporary fast-discount responses differ? The contrast in Table 8 removes detection timing as a difference while retaining each response's separately fitted parameters and scale. Reset gains 1.049 points. Both responses resume population decay. The random and periodic controls change intervention times, giving gains of 1.060 and 1.225 points for detected reset. These latter contrasts are consistent with a benefit from data-dependent timing under their recorded matching rules, not an identical-alarm comparison.

All three contrasts are exploratory, and all five seed effects are positive. With five independent signs, an all-positive result has probability $1/32$ under a symmetric one-sided sign null and $1/16$ for either sign. We therefore report direction consistency and effect sizes rather than imply a two-sided sign-test result below 0.05. On the broader main mixture, reset exceeds temporary fast discount by only 0.213 points and one seed is negative. *Holding detected events fixed leaves a response difference in this synthetic change-point population; it does not establish reset as a generally preferred intervention.*

Alarm delay (exploratory). A separate five-seed run supplies the same alarm stream to four responses (reset with the fitted back-dating window, reset with a window re-fitted to the stream, partial reset, and temporary fast discount) under the fitted detector and under oracle alarms placed 0, 4 or 8 steps after each true change. It reproduces the detector contrast above (1.05 $[0.72, 1.38]$). Table 9 and Figure 4 report the results. With the detector-fitted window, reset does not beat fast discount at zero delay and leads by more as the delay grows, over all users and on change-point users alone. The re-fitted window equals the delay in all 15 change-point runs, and reset then leads at every delay in that mixture; in the main mixture the zero-delay contrast is 0.09 $[-0.07, 0.25]$. On PENS the fitted detector never fires (0.00 alarms per 100 steps), so no response is triggered.

Table 9: Reset minus temporary fast discount under shared alarms (exploratory, five seeds), percentage points with 95% t-intervals over seeds. "Detector-fitted" uses the back-dating window fitted with the detector; "re-fitted" re-selects it for each alarm stream (selected k_{back} per seed in the last column, change-point mixture).

Mixture	Alarm stream	Alarms/100 steps	Detector-fitted, all users	Detector-fitted, change-point users	Re-fitted, all users	k_{back}
Change-point	oracle, delay 0	0.49	-0.28 $[-0.72, 0.16]$	-0.54 $[-1.47, 0.38]$	0.66 [0.44, 0.88]	0,0,0,0,0
	oracle, delay 4	0.60	0.53 [0.11, 0.94]	0.99 [0.25, 1.73]	0.90 [0.43, 1.37]	4,4,4,4,4
	oracle, delay 8	0.70	1.04 [0.75, 1.33]	2.07 [1.53, 2.61]	1.13 [0.87, 1.40]	8,8,8,8,8
	fitted detector	1.10	1.05 [0.72, 1.38]	1.97 [1.42, 2.53]	1.06 [0.80, 1.32]	8,2,4,4,8
Main	oracle, delay 0	0.40	-0.22 $[-0.42, -0.02]$	-0.87 $[-1.65, -0.08]$	0.09 $[-0.07, 0.25]$	
	oracle, delay 4	0.48	0.34 [0.21, 0.47]	1.32 [0.87, 1.77]	0.40 [0.14, 0.67]	
	oracle, delay 8	0.57	0.36 [0.07, 0.65]	1.43 [0.30, 2.55]	0.43 [0.25, 0.61]	
	fitted detector	0.75	0.21 $[-0.03, 0.45]$	1.26 [0.60, 1.92]	0.18 $[-0.06, 0.42]$	

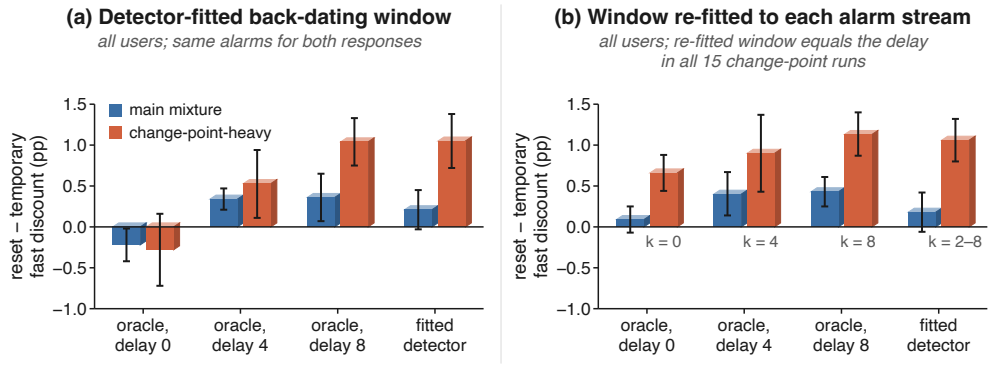


Figure 4: **Alarm delay decides the better response (exploratory).** Reset minus temporary fast discount under identical alarms, over all users, five seeds per mixture; whiskers are 95% t-intervals over seeds. **(a)** Back-dating window fitted with the detector. **(b)** Window re-fitted to each alarm stream; in the change-point mixture it equals the delay. Numbers in Table 9.

E.2 Regime Boundaries And Ablations

Table 10: Three-seed pure-regime results. Accuracies and margins are proportions. The margin is reset minus the best non-reset policy in that regime, which can differ between regimes. The rows isolate different temporal patterns and do not supply a universal ranking.

Regime	Pop. discount	Single gamma	LETHE	Event reset	Margin
Stable	.768	.769	.768	.768	$-.001$
Ramp	.713	.712	.696	.713	$+.000$
Change-point	.675	.678	.662	.694	$+.013$
Bursty	.604	.607	.612	.604	$-.009$

The pure regimes in Table 10 in Appendix E.2 test where the mixed-population result applies. Event reset improves over the strongest prior candidate on the change-point regime, has essentially no advantage under gradual ramp change, and falls behind the mixture under bursty change. The stable regime provides no positive reset margin. *Abrupt change and repeated rapid changes are not interchangeable regimes for evaluating a reset policy.*

Table 11: Event-operator variants on the change-point mixture. Accuracy is a proportion. Differences are percentage points relative to full event reset. The "No refractory" variant lowers the minimum-run gate to one and refits the detector. It can alter both warm-up and renewed eligibility, rather than isolating an eight-step post-alarm pause.

Variant	Accuracy	Difference (pp)
Full event reset	.694	$+0.000$
No backdating	.689	-0.535
Tracking detection memory	.692	-0.201
"No refractory" (gate = 1)	.694	-0.049
CUSUM drift $b = 0$	.693	-0.119
Soft fractional reset	.695	$+0.056$
Block-cosine heuristic	.677	-1.738
Supervised window detector	.680	-1.453
LETHE-E scalar	.691	-0.319
LETHE-E mixture	.688	-0.640

The ablations in Table 11 keep the change-point population fixed while altering the fitted event operator. Removing backdating reduces accuracy by about half a point, although it changes both the retained window and subsequent detector eligibility. Reset-aware scalar and mixture heads do not improve over the fitted event operator. Smaller ablation differences should be interpreted together with their paired uncertainty.

Table 12: Descriptive fit-fraction and history-length sweeps on the change-point mixture. Margins are accuracy proportions and oracle capture is a fraction. Fit-fraction entries use three seeds, and history-length entries use two. The best prior comparator can change with the sweep setting.

Fit fraction	**.05**	**.10**	**.20**	**.40**	**.60**	**Seeds**
Event margin	.016	.010	.011	.004	.009	3
Oracle capture	.305	.399	.469	.202	.302	3
History T	**32**	**48**	**64**	**96**	**128**	**192**
Event margin	.003	.005	.007	.007	.008	.009
Oracle capture	.147	.133	.264	.344	.403	.580

The fit-fraction sweep in Table 12 in Appendix E.2 does not improve monotonically as more users are allocated to fitting. The history-length sweep instead gives an increasing envelope for reset's margin over the strongest prior candidate at each length. That comparator changes with length. The recorded reset-minus-population-discount sequence is non-monotone. The envelope therefore does not establish that increasing history length alone causes a monotone reset benefit. These descriptive sweeps use fewer seeds than the prespecified result, with counts stated in the table.

E.3 Recovery After A Change

Reset removes accumulated evidence as well as bias from the earlier regime. We examine the amount of new evidence available after a change through a retrospective analysis of 3,499 changed histories pooled across five synthetic populations. Histories are grouped by their realized mean post-change segment length, which is unavailable to a deployable policy at prediction time. Figure 5 in Appendix E.3 reports gains over each run's population-discount reference.

Immediate true-boundary reset has little benefit in the shortest bin and a larger positive gain once more post-change observations are available. With an imposed eight-step delay, mean gains remain negative through the 10-to-16-step bin and become positive from the 16-to-24-step bin onward. Binwise uncertainty is unavailable, and the reference rates vary across pooled families. The pattern is therefore consistent with reset needing enough replacement evidence, not with a universal prospective crossover threshold.

F Additional Diagnostics And Attribution Analysis

F.1 Prospective Selection Records

The selector bank contains retain, exponential discount, and a causal recurrent expert in every reported main and change-point run. Selection uses an earlier window A and evaluation uses a later disjoint window B. For $T = 64$, these windows are observations 33-48 and 49-64; for $T = 96$, they are observations 49-72 and 73-96. Every test user has predict-before-update accuracy for each expert on both windows, and each choice is scored before its label updates the expert's memory.

Table 13: Per-seed expert-selection gains in percentage points. Hindsight and earlier-window choice are scored on the same later outcomes. Both use the best constant expert on that scored window as reference. The expert bank is retain, exponential discount, and a causal recurrent expert.

Population	Seed	Hindsight Choice	Earlier-Window Choice
Main	1	+5.73	+0.42
	2	+6.08	+0.65
	3	+5.71	+0.31
	4	+5.63	-0.79
	5	+5.15	-0.54
Main mean		+5.66	+0.01
Change-point	1	+4.65	-1.44
	2	+3.35	-1.69
	3	+3.81	-1.46
	4	+4.14	-2.25
	5	+3.31	-2.46
Change-point mean		+3.85	-1.86

We change the window used to choose the expert while retaining the final window used to score it. Table 13 in Appendix F.1 gives the five seed results, not intervals over seeds. The comparator is the best constant expert on the final scored window, a descriptive reference rather than an independently deployed selector. The prospective mean is nearly zero in the main mixture and negative in the change-point mixture, although three main-mixture seeds are positive.

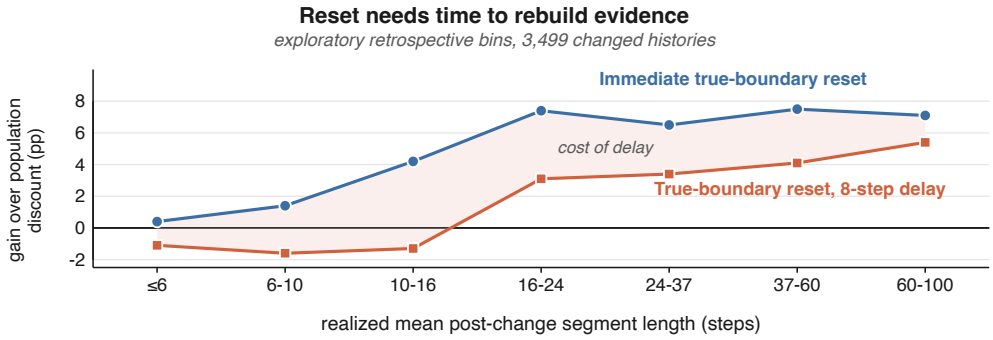


Figure 5: **Recovery after a change, exploratory.** Bin means for 3,499 changed histories pooled across five synthetic populations. Immediate and eight-step-delayed true-boundary resets are compared with each run's population-discount reference; the shaded region is the cost of delay. Bins use realized mean post-change segment length, not information available to a prospective decision. Connected points guide the eye across bins and do not estimate a continuous response law. Bins show means without intervals. Varying reference rates and retrospective grouping preclude a universal crossover threshold.

Maximizing-expert agreement across adjacent windows averages 47.67% on the main mixture and 44.27% on the change-point mixture, with seed values spanning approximately 42 to 50%. Ties and finite-window variability affect these observed winners. The generator also places each single change before the final quarter of the history. For one-switch users, the selection window can span change or recovery while the scoring window is entirely post-change. Neither winner disagreement nor the hindsight gap identifies sampling noise as the sole cause of failed transfer.

Decomposing the hindsight gain (exploratory). A replicate run of the same selector (five seeds per mixture, identical windows and bank) also scores every expert against the generator's expected accuracy, obtained by resampling labels from the known preference trajectory and cross-fitting. Its observed hindsight gains are close to, but not identical with, the main runs: 5.76 $[5.39, 6.13]$ and 4.09 $[3.49, 4.69]$ points against 5.66 and 3.85 (per-seed main-mixture values 5.79, 6.19, 5.87, 5.50, 5.46 against 5.73, 6.08, 5.71, 5.63, 5.15). Intervals here are 95% t-intervals over five seeds. Table 14 reports the decomposition. True per-user headroom is 2.77 and 1.64 points, and the remaining 2.99 and 2.45 points are selection optimism. Scored against expected accuracy, earlier-window argmax gains 0.11 and -1.78, a shrinkage selector 0.09 and -0.05, and fixed-share switching 0.11 and -0.54. An oracle that is told each user's truly best expert on window A gains 1.03 and -0.87 on window B: the true best expert differs from the population-best expert for 44% and 48% of users and changes between A and B for 36% and 42%. For users whose best expert differs, a selector would need a median of 130 (main) and 258 (change-point) outcomes to pick it with 90% probability, far more than the 16- and 24-step selection windows. A permutation null that exchanges expert labels at every step gives 6.28 and 5.27, above the observed gains; because it breaks the correlation between experts it overstates noise, so we use generator truth for synthetic claims and the null only for natural corpora.

Table 14: Decomposition of the per-user selection gain (exploratory replicate, five seeds per mixture). Gains over the best constant expert in percentage points, 95% t-intervals over seeds. "True" quantities are scored against the generator's expected accuracy. Hindsight selection is a diagnostic reference.

Quantity	Main mixture	Change-point-heavy
Observed hindsight gain	$5.76\ [5.39, 6.13]$	$4.09\ [3.49, 4.69]$
True per-user headroom	$2.77\ [2.13, 3.41]$	$1.64\ [1.38, 1.90]$
Selection optimism	$2.99\ [2.53, 3.45]$	$2.45\ [2.02, 2.88]$
Earlier-window argmax (true)	$0.11\ [-0.26, 0.47]$	$-1.78\ [-1.99, -1.56]$
Shrinkage selector (true)	$0.09\ [-0.12, 0.30]$	$-0.05\ [-0.12, 0.02]$
Fixed share (true)	$0.11\ [-0.43, 0.66]$	$-0.54\ [-0.81, -0.28]$
Oracle on true window-A best	$1.03\ [0.52, 1.55]$	$-0.87\ [-1.27, -0.48]$
Users whose true best rule changes $A \to B$	36%	42%
Outcomes needed for a 90% correct choice (median, users whose best rule differs)	130	258
Selection window in the main runs	16	24

Selection-window sweep. With 256-step histories (same generator, $t_0 = 96$, fixed 32-step evaluation window), earlier-window argmax gains 1.01, 1.29, 1.58, 1.75 and 1.77 points on the main mixture for selection windows of 8, 16, 32, 64 and 128 choices, against true headroom of 3.19; on the change-point-heavy mixture the gains are -0.99, -0.17, 0.42, 0.61 and 0.24 against 2.01. The rate at which the selector recovers the true best expert (0.57 to 0.62 and 0.50 to 0.56) stays below what label noise alone would allow (0.63 to 0.82 and 0.60 to 0.79), because the true best expert changes for 28 to 41% of users between windows. The selection limit therefore narrows with history length but does not reduce to sampling noise.

Natural corpora. Without generator truth, only the permutation null is available. On PENS (2,219 users with at least 160 clicks) hindsight selection gains 1.86 points against a null mean of 3.17 (95th percentile 3.29); earlier-window argmax gains -0.10, -0.11, $+0.01$ and -0.07 for windows of 8, 16, 32 and 64 clicks, shrinkage 0.16 to 0.25, and fixed share 0.10. On Arena (88 users, $T = 20$) hindsight gains 12.45 against a null of 14.23, and earlier-window argmax and shrinkage over a 5-choice window gain -3.02 and -2.26.

F.2 Completed Adaptive Comparisons

Table 15: Completed adaptive comparisons across five seeds per mixture. Entries are mean accuracy gains in percentage points. The first two rows use the selector protocol. The others use the separate prequential ladder. For the scalar and mixture, the prespecified reference is online discount on the main mixture and population discount on the change-point mixture. Rounded near-zero values do not denote equivalence.

Rule	**Main**	**Change-point**	**Reference And Interpretation**
Expert router	0.00	-3.59	Final-window best constant. Prefix-statistic router.
Conservative router	-0.85	$+0.01$	Same selector reference. Escalates from exponential to recurrent memory.
Online rate	$+0.18$	-0.03	Population discount. Rate updates after feedback.
Learned scalar	-0.63	$+0.28$	best baseline. Formal accuracy gate fails.
LETHE mixture	-0.55	-0.87	best baseline. Formal accuracy gate fails.
Raw-direction GRU	-5.53	-2.38	Population discount. Single tested configuration.
Same-statistics GRU	-2.78	-2.54	Population discount. Recurrent control over LETHE features.

The candidates in Table 15 answer different adaptation questions and do not form one common ranking. The learned router predicts an expert from prefix statistics, the online-rate rule updates one persistence parameter after feedback, the learned scalar and LETHE allocate weight within the trace bank, and the recurrent controls use alternative state representations. The selector-based rows and prequential-policy rows use different evaluation protocols and references. On the change-point mixture, the learned scalar gains 0.28 points over its prespecified reference, with seed-resampled interval $[-0.59, 1.38]$, while LETHE loses 0.87 points with interval $[-1.62, -0.14]$. Neither passes its

prespecified criterion. The recurrent controls are limited baselines and do not characterize recurrent-model capacity. Overall, the completed comparisons do not establish the targeted prospective adaptive gain.

F.3 Noise, Locality, And Staleness

Table 16: Fixed-prefix staleness diagnostic, 500 histories per mixture. Differences are accuracy percentage points on identical later outcomes. Intervals are the recorded 95% paired user-bootstrap intervals from 2,000 resamples. No prediction state updates during scoring. This diagnostic does not use the fitted population component of the main ladder.

Population	Contrast	Difference	95% User Interval
Main	Recent minus old	$+5.35$	$[+3.84, +7.00]$
Main	Full minus recent	$+2.33$	$[+1.08, +3.57]$
Change-point	Recent minus old	$+13.31$	$[+11.67, +15.07]$
Change-point	Full minus recent	-5.22	$[-6.57, -3.92]$

The staleness diagnostic uses 500 generated histories per mixture and predicts a common later evaluation window from an old block, a recent block, or the complete prefix. The complete-prefix and recent-block probes therefore score exactly the same future outcomes. Table 16 separates the two relevant comparisons. Recent evidence outperforms old evidence in both mixtures, but complete history improves over recent-only prediction in the main mixture and reduces accuracy in the change-point-heavy mixture. Better recent-only evidence therefore does not imply that retaining older evidence is harmful.

The order controls ask a different question. Retaining the latest eight comparisons while permuting the rest preserves 55 to 87% of the measured order benefit, and retaining the latest 16 preserves 73 to 100%. These percentages summarize the policy-specific diagnostic runs, not a universal fraction of information in every user history.

At $T = 64$ and unit observation noise, the raw drift floor is approximately 0.302, the tested small-rotation increment is 0.012, and the stationary cross-user standard deviation is 0.055. The unrounded floor is about 25 times the increment. Same-outcome drift-benefit correlations range from 0.16 to 0.29 on stationary controls, while the outcome-disjoint construction's intervals include zero. Those controls support removal of the corresponding measurement artifact, not statistical equivalence to zero or complete removal of every temporal dependency.

Nonlinear Reward Sensitivity. A separate five-seed nonlinear-reward diagnostic uses 400 generated users per seed, with 160 fit and 240 test users. Exponential memory loses a mean 1.286 accuracy points under full shuffle, with between-seed standard deviation 0.227 points. Retain remains at zero. The mean permutation-corrected population change is 0.049 and the mean outcome-disjoint drift-benefit correlation is 0.119, with between-seed standard deviations 0.005 and 0.043, respectively. These are means and standard deviations across five diagnostic runs, not pooled confidence intervals. The arm extends the temporal-signal check to another reward specification without repeating the complete prospective selector, fixed-alarm, or attribution experiments.

F.4 Exploratory Illusion Of Personalization

The cohort analysis below (Figure 6) compares different stable-user cohorts across training populations; the matched analysis that follows scores every controller on identical held-out histories (Table 17). Assigned memory is the nominal 64-lag kernel median defined in Appendix C.2, not the complete predictive influence of past observations. Endpoints are the median assigned lag of never-changing test users at the first and last evaluation step, pooled over all users and seeds (three seeds for stable-only training, five otherwise): $29 \to 29$ under stable-only training, $23 \to 19$ in the main mixture, and $22 \to 10$ in the change-point-heavy mixture. Without the history-length input, the endpoints are $25 \to 24$ (main) and $25 \to 23$ (change-point-heavy). All five change-point-mixture seeds decline.

In the cohort analysis, the history-length ablation has five paired runs per mixture. Removing length improves stable-user accuracy by 0.94 points in the change-point mixture and 0.43 in the main mixture, with positive effects in all five seeds of each. It reduces change-point-user accuracy by 0.35 points with no positive seeds, while changing the change-point-mixture average by -0.02 points. The main-mixture average increases by 0.2 points. Residual shortening after removal limits the conclusion to a contribution from this input, not a complete explanation of memory allocation.

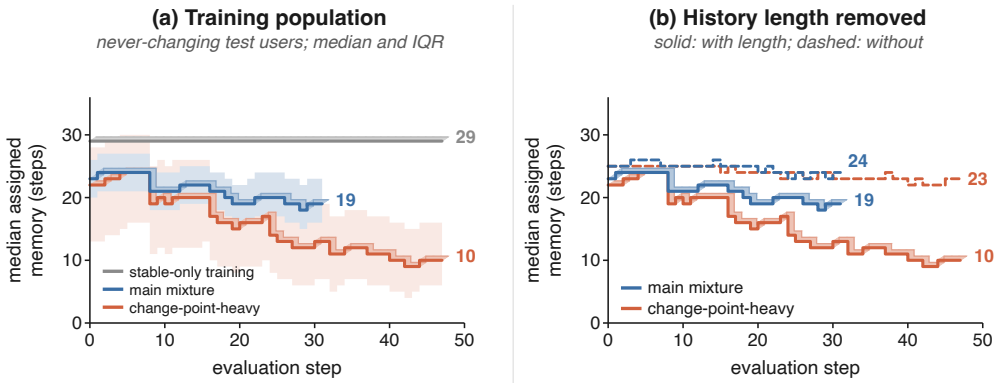


Figure 6: **Cohort analysis (exploratory).** Median assigned memory of never-changing test users across the evaluation window, pooled over seeds; shaded bands are interquartile ranges. **(a)** Controllers trained on the stable-only, main and change-point-heavy populations. **(b)** The same controllers with and without the history-length input. Cohorts differ across training populations; Figure 7 and Table 17 use identical histories.

Matched histories (exploratory). To remove the cohort difference, controllers trained on the stable-only, main and change-point-heavy populations, each with and without history length, are scored on the same held-out stable and change-point histories (five seeds). Table 17 and Figure 7 give the results. On stable users, median assigned lag is 29.2, 23.9 and 15.7 with history length and 29.2, 27.1 and 23.8 without it; its slope over the window is $+0.6$, -9.6 and -19.7 steps per 100 evaluation steps. On change-point users it is 28.6, 16.0 and 6.0. A Bayesian online change-point learner (Adams & MacKay, 2007) given each population's true hazard, summarized by the median lag of its run-length-weighted memory over the same 64 lags, keeps 31.0, 31.0 and 30.0 steps on the stable users, although its prior probability that a switch has already happened rises to 0.25 and 0.50 late in the main and change-point-heavy histories (Wilson et al., 2010). Paired on the same users, the main- and change-point-trained controllers change stable-user accuracy by -0.70 $[-1.09, -0.30]$ and -2.30 $[-3.28, -1.32]$ points relative to the stable-trained controller and raise change-point-user accuracy by 4.09 $[2.28, 5.90]$ and 7.08 $[6.81, 7.35]$. Removing history length from the change-point-trained controller changes stable-user accuracy by $+0.83$ $[0.38, 1.29]$ and change-point-user accuracy by -0.78 $[-1.22, -0.33]$. The controller therefore shortens memory beyond what a calibrated change prior implies, and history length carries part of that population expectation.

Table 17: **Same stable users, different memory.** Controllers trained on three populations are scored on identical held-out histories (five seeds). Lags are median assigned memory in steps. Accuracy differences are in percentage points relative to the stable-trained controller on the same users, with 95% intervals over seeds. The Bayesian reference is a change-point learner given the true hazard of each training population. Exploratory.

	Stable users: median lag			Accuracy vs stable-trained	
Training population	LETHE	LETHE, no length	Bayesian	Stable users	Change-point users
Stable-only	29.2	29.2	31.0	ref.	ref.
Main mixture	23.9	27.1	31.0	-0.70 $[-1.09, -0.30]$	$+4.09$ $[2.28, 5.90]$
Change-point-heavy	15.7	23.8	30.0	-2.30 $[-3.28, -1.32]$	$+7.08$ $[6.81, 7.35]$

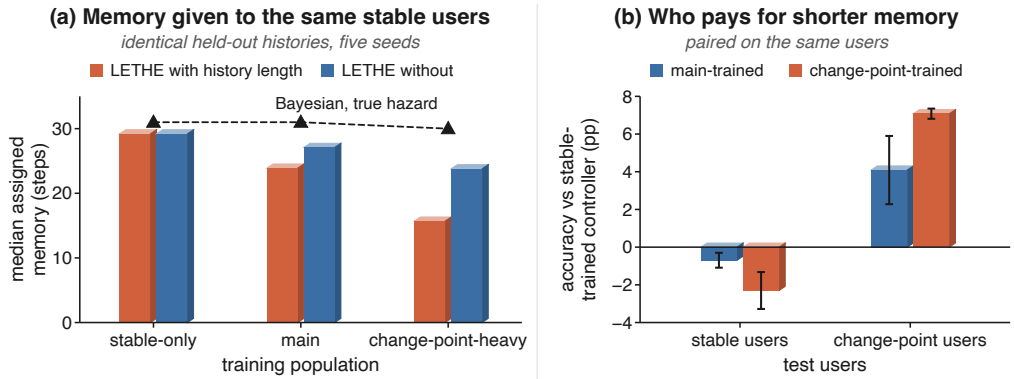


Figure 7: **Same stable users, different memory (exploratory).** Controllers trained on three populations are scored on identical held-out histories, five seeds. **(a)** Median memory assigned to the stable users, with and without the history-length input; triangles are a Bayesian change-point learner given each population's true hazard. **(b)** Accuracy relative to the stable-trained controller on the same users; whiskers are 95% t-intervals over seeds. Numbers in Table 17.

A population-level timing cue remains a candidate explanation. For one-switch users, the generator draws a change uniformly from $[T/4, 3T/4)$. As time passes, the probability that the switch has already happened rises even if the current user gives no new evidence. This cumulative probability is different from the step-by-step probability of a switch at the next moment. Showing that the controller learned either quantity would require a matched intervention on event timing, which the current experiments do not provide.

The attribution evidence concerns how to interpret an adaptive state. It does not show that population information is illegitimate or that the model learned a population rule for when change occurs.

G Natural-History And Checkpoint Results

PRISM and Arena provide relatively short natural preference histories, whereas PENS provides a substantially longer sequence of user interactions. These datasets record observed choices over time but do not reveal verified latent preference-change times.

PRISM provides little separation among temporal memory policies. On Arena, under a projection fitted from fit-user prefixes, LETHE differs from online discount by $+0.16$ points with paired 95% interval $[-2.37, 2.49]$, while event reset differs from the learned scalar by -1.49 points with interval $[-4.10, 1.32]$. Both comparisons remain unresolved. *The available short natural histories therefore do not establish a preferred temporal memory rule.*

PENS. The PENS evaluation uses 3,000 users, with 1,200 fit users and 1,800 test users. The representation projection is fitted only on fit-user selection-prefix rows and then applied to held-out users. Retain reaches 88.52% accuracy, fixed population discount 90.08%, a single learned timescale 90.46%, LETHE 90.23%, and event reset 90.13%. On PENS the fitted reset effectively never triggers, so it behaves like continued discounting.

Relative to retain, LETHE gains 1.71 points with paired 95% interval $[1.55, 1.88]$, and it exceeds fixed population discount by 0.15 points with interval $[0.06, 0.23]$. However, the single learned timescale exceeds LETHE by 0.23 points with interval $[0.16, 0.31]$, while event reset remains 0.34 points below the learned scalar with interval $[-0.40, -0.27]$. PENS accuracy concerns prediction of a clicked headline against its row-level unclicked pool rather than impression-level ranking. *Longer natural histories support downweighting older evidence, but not a specific advantage for multiscale mixing or reset over a single learned timescale.*

G.1 SOURCE-USER-DISJOINT ARENA CONSTRUCTIONS

Table 18: Two source-user-disjoint Arena splice series with five constructions each: Series A (300 test pseudo-users, used in Figure 3(c)) and Series B (360). Both reuse the same 53 test source users, so they are not independent replications. The exploratory A–B–C control (500 histories) uses a third source user in the final segment; all five include zero for reset minus retain.

Series	Contrast	Difference	95% Source Interval
Series A	Reset minus retain	-5.68	$[-6.79, -4.54]$
Series A	Reset minus continued discount	-3.01	$[-3.72, -2.31]$
Series A	Reset minus temporary fast	-1.13	$[-1.66, -0.55]$
Series B	Reset minus retain	-5.57	$[-7.31, -3.80]$
Series B	Reset minus continued discount	-1.96	$[-3.01, -0.99]$
Series B	Reset minus temporary fast	-0.53	$[-1.22, +0.16]$
A–B–C	Reset minus retain	-0.58	construction level
A–B–C	Reset minus continued discount	-0.15	construction level
A–B–C	Reset minus temporary fast	-0.28	construction level

To study intervention behavior while retaining natural response comparisons, we construct longer Arena histories by joining segments from different source users while keeping source users disjoint across fitting and evaluation. The constructed transition points are known to the analysis but are not supplied to the evaluated policies as oracle alarms.

In a second, 360-history construction (Series B in Table 18), detected reset is 5.57 points below retain, with 95% primary-source-cluster interval $[-7.31, -3.80]$, and is lower in all five constructions. Reset is also 1.96 points below continued population discount, with interval $[-3.01, -0.99]$. Its difference from temporary fast discount is smaller at -0.53 points, with interval $[-1.22, 0.16]$, so that comparison remains unresolved.

The 300-history construction shown in Figure 3(c) (Series A) gives the same qualitative conclusion: reset is 5.68 points below retain and 3.01 points below continued population discount, with both primary-source intervals excluding zero. Reset is also below temporary fast discount by 1.13 points in that construction. These constructions share source populations and should not be interpreted as independent replications.

The reported intervals cluster by each constructed history's primary source user. A history may contain additional source users, so this clustering does not capture every secondary-source dependency. More importantly, the constructed switches are not verified within-person preference changes. Each constructed history alternates two source users, so its final segment returns to the first source; for source users with fewer than two segments of votes, the first and final slices can share comparisons. Both properties favor retaining older evidence, so these series measure the cost of resetting when the pre-alarm regime returns, not the cost of resetting after a lasting change. *These results show that a detected transition in natural comparisons can favor preserving older evidence rather than reset, without implying a universal ordering of memory responses.*

No-return control (A–B–C, exploratory). We repeat the construction with a third source user C in the final segment, so no comparison recurs (five constructions of 100 test pseudo-users, 80 of them spliced). Predicting the final segment from post-switch evidence alone is then no better than using the full history (mean difference -0.1 points), whereas in the A–B–A series 39 to 50% of final-segment comparisons already appeared in the first segment. Detected reset is 0.58 points below retain on average, with construction-level means -0.06, -0.12, -0.02, -2.11 and -0.61, and every construction's primary-source interval includes zero. Restricted to the spliced histories, reset minus retain is -5.55 $[-8.05, -3.06]$ for A–B–A and $+0.21$ $[-0.76, 1.19]$ for A–B–C, with 95% t-intervals over the five constructions. Without a return to the first source, the data therefore do not show reset either helping or hurting.

G.2 CHECKPOINT-BASED MEMORY COMPARISONS

The five Skywork-Reward-V2 checkpoints use Llama-3.2-1B, Llama-3.2-3B, Llama-3.1-8B, Qwen3-1.7B, and Qwen3-8B backbones (Liu et al., 2026). Each supplies an alternative reward representation inside the common temporal-memory evaluation. These experiments therefore test sensitivity to the reward signal rather than native personalized reward-model performance.

Table 19: Checkpoint-based memory ladders on PRISM and Arena. Accuracies are percentages. Mix denotes LETHE. The final column is its paired 95% user-bootstrap interval against retain in points. One seed per checkpoint-corpus pair.

Checkpoint	Corpus	Retain	Disc.	Mix	Reset	Mix Minus Retain
Llama-3.2-1B	PRISM	57.22	57.48	56.72	57.18	$[-1.25, 0.23]$
Llama-3.2-1B	Arena	80.72	80.32	80.66	80.29	$[-0.49, 0.38]$
Llama-3.2-3B	PRISM	55.63	55.99	55.59	55.78	$[-0.36, 0.26]$
Llama-3.2-3B	Arena	80.54	80.54	80.62	79.37	$[-0.1, 0.26]$
Llama-3.1-8B	PRISM	55.46	55.35	55.47	55.39	$[-0.2, 0.21]$
Llama-3.1-8B	Arena	80.12	80.12	80.12	79.97	$[0, 0]$
Qwen3-1.7B	PRISM	56.2	56.22	55.85	55.83	$[-0.74, 0.04]$
Qwen3-1.7B	Arena	80.47	77.69	80.42	77.77	$[-0.85, 0.57]$
Qwen3-8B	PRISM	56.89	56.91	57.02	56.97	$[-0.27, 0.49]$
Qwen3-8B	Arena	82.24	82.24	82.69	81.76	$[-0.18, 1.46]$

All ten LETHE-minus-retain intervals in Table 19 include or touch zero, and the point differences have both signs. The Llama-3.1-8B Arena interval is exactly zero for the evaluated users, which does not establish equivalence on future data. *Across these reward representations, the checkpoint comparisons do not establish a consistent gain from* LETHE *over retention.*

The Llama-3.2-1B Arena drift estimate is 0.014 with interval $[0.003, 0.029]$, while its drift-benefit correlation is 0.145 with interval $[-0.100, 0.360]$. The other checkpoint drift-benefit intervals also span zero. Thus, measurable population change in a reward representation need not yield a resolved association with memory benefit.

Table 20: Checkpoint-specific source-user-disjoint splices, five constructions per checkpoint. Effects are detected reset minus retain in percentage points. Intervals cluster by primary source and are conditional on the constructions.

Checkpoint	Pseudo-Users	Sources	Difference	95% Cluster Interval
Llama-3.2-1B	400	53	-0.53	$[-1.19, 0.14]$
Llama-3.2-3B	380	53	-0.26	$[-0.85, 0.24]$
Llama-3.1-8B	280	52	-0.89	$[-1.91, 0.04]$
Qwen3-1.7B	380	53	-1.11	$[-2.53, -0.005]$
Qwen3-8B	300	53	-0.47	$[-1.46, 0.41]$

Across the checkpoint-specific source-user-disjoint constructions, reset-minus-retain point estimates are negative for all five checkpoints (Table 20). Four of the five primary-source intervals include zero; only Qwen3-1.7B excludes zero, with its upper endpoint close to zero at -0.005 points. These checkpoint experiments reuse the same source populations and therefore provide sensitivity evidence rather than independent human-data replications.

Each checkpoint-corpus pair is evaluated as a sensitivity analysis rather than a replication of the complete study. In particular, the prospective expert-selection and stable-user attribution experiments are not repeated for every checkpoint. *Checkpoint coverage therefore tests whether the reported memory contrasts persist under additional reward signals, not whether every component of the three paradoxes independently replicates at every checkpoint.*

H Dataset Construction And Preprocessing

H.1 Corpus-Specific Pair Construction

Table 21: **Evaluation populations and their roles.** Synthetic counts are test users per seed; synthetic results pool five seeds (three for pure regimes). Arena splice counts are constructed test histories, not independent source users. History gives T for synthetic populations and the minimum per-user length for natural data.

Population	Test Users	History	Role
Main mixture	300	$T = 64$	Stable, gradual, abrupt, and bursty users mixed
Change-point-heavy mixture	300	$T = 96$	Abrupt changes at hidden times dominate
Pure regimes	300	$T = 96$	One temporal pattern per population (Table 1)
Noise null	180	$T = 64$	No true change; measures apparent drift from finite histories
PRISM	488	≥ 4 pairs	Natural response judgments; latent changes unknown
Arena	53	≥ 20 votes	Natural ordered votes; latent changes unknown
Arena splice	360	constructed	Transitions between disjoint source users' real comparisons
PENS	1,800	≥ 8 clicks	Long natural click histories; latent changes unknown

Table 22: Corpus construction. "Shared row pool" means that all clicks in one PENS log row use the same unclicked set. It is not an impression-specific negative set.

Corpus	User and order	Chosen versus rejected pair	Filter	Access
PRISM	user ID, conversation time, then turn	chosen model utterance versus highest-scoring unchosen model utterance in the same turn	at least 4 pairs	Hugging Face release
Chatbot Arena	hashed judge, vote timestamp	winning model's final assistant message versus losing model's final assistant message. Ties are dropped	at least 20 votes	Hugging Face release
PENS	user ID, click order	clicked headline versus mean embedding of the row-level unclicked headline pool	at least 8 valid clicks after embedding	manual Microsoft Research download

Each corpus supplies observed behavior rather than a verified latent change label. PRISM contributes within-turn judgments between model utterances (Kirk et al., 2024). We pair the selected response with the highest-scoring remaining response in the same turn and orders conversations by timestamp with deterministic turn offsets. Arena contributes non-tied winner votes (Zheng et al., 2023). We use the final assistant message from each model conversation and no prompt field. These two transformations test preference prediction on naturally observed comparisons, not detection against known natural change times.

PENS contributes ordered news-click evidence (Ao et al., 2021). Each clicked headline is contrasted with the mean embedding of the unclicked headlines shared by that log row. This is a row-level preference surrogate, not an impression-specific negative set or a headline-generation evaluation. The longer histories support a separate memory comparison. The high absolute accuracy is specific to this constructed task. The filters are listed in Table 22 in Appendix H.1.

H.2 Representation Fitting And Split Alignment

The base encoder is frozen `sentence-transformers/all-mpnet-base-v2`. Its 768-dimensional response representations are reduced to 64 dimensions using an uncentered second-moment projection fitted only from permitted fit-user prefix rows. The fitted transform is then applied unchanged to held-out users. For PENS, the projection is fitted on 162,371 selection-prefix rows from 1,200 fit users and then applied to 1,800 test users. Arena likewise uses a projection fitted from its fit-user prefixes. This separation prevents held-out-user evaluation rows from determining the representation transform.

H.3 Checkpoint Interface

The Skywork experiments replace the base reward representation while leaving the temporal-memory comparison unchanged (Liu et al., 2026). For each checkpoint, we obtain response representations and the corresponding reward signal through its tokenizer and scoring interface, then use these features inside the common memory-policy evaluation. These experiments should therefore be interpreted as sensitivity to alternative reward representations rather than as evaluations of the checkpoints' native personalized scoring behavior. The evaluated checkpoint families and results are reported in Appendix G.2.

I Training, Hyperparameters, And Compute

I.1 Baseline Fitting And Model Selection

The fitting objective determines how much a user with a long history influences the fitted parameters. Suppose fit user A contributes 10 eligible steps and fit user B contributes 40. If we pool all 50 observations, B receives four times A's weight. If we first average within each user and then average the two users, A and B each receive half of the weight.

We use both conventions. For eligible fit-prefix steps P_u, let $n_u = |P_u| > 0$ and let $L_{u,t} = -\log \sigma(y_{u,t}\ell_{u,t})$ be logistic loss. Observation-pooled fitting uses

$$L_{\text{obs}} = \frac{\sum_{u \in U_{\text{fit}}} \sum_{t \in P_u} L_{u,t}}{\sum_{u \in U_{\text{fit}}} n_u}, \qquad L_{\text{user}} = \frac{1}{|U_{\text{fit}}|} \sum_{u \in U_{\text{fit}}} \frac{1}{n_u} \sum_{t \in P_u} L_{u,t}. \tag{4}$$

The first expression pools observations; the second gives each fit user equal total weight. Controller penalties and early stopping remain as specified in Appendix C.2. *Equal user weighting at evaluation does not imply equal user weighting during fitting.*

Observation-pooled fitting is used for the population-level decay and event-detector candidates, while the learned scalar and LETHE controllers use user-balanced fitting so that long-history users do not dominate controller training. Individual rate selection uses only the permitted prefix of the corresponding user. Each memory candidate retains its own fitted personalization scale. The recurrent baselines provide alternative state representations but are not an exhaustive search over recurrent architectures. Their performance should therefore be interpreted as baseline evidence for the tested configurations rather than as a statement about recurrent-model capacity.

I.2 Execution Environment And Recorded Compute

Experiments use Python 3.10 with standard scientific Python, PyTorch, Transformers, and Sentence-Transformers tooling. Embedding and checkpoint extraction use NVIDIA A100 GPUs, while the lightweight memory-policy evaluations can run on CPU. Full dependencies, configurations, and run instructions accompany the released code. We make no comparative efficiency, energy, or carbon claim.

J Evaluation Metrics And Reporting Conventions

The metrics in Table 23 separate prediction performance from diagnostics of history and adaptive state. When accuracy rises from 0.70 to 0.71, we report a gain of one percentage point. This is not the same as a one-percent relative increase. Accuracy is first averaged over each user's scored steps and then across users. Pairing applies only when two methods are evaluated on the same observational units. The fixed-prefix diagnostic, different natural cohorts, and the two splice series are therefore not treated as one paired experiment. User-, seed-, and source-cluster intervals also answer different uncertainty questions, and primary-source clustering cannot capture every dependency through secondary sources.

Table 23: Metric dictionary. Within each matched policy contrast, both policies use the same users and scored steps. Distinct experimental protocols are not pooled as one contrast. Interpretation and units are specified in Appendix J.

Quantity	Definition	Reporting unit
Prequential accuracy	mean of $\mathbf{1}\{y_{u,t}\ell_{u,t} > 0\}$ over evaluation steps, then users	accuracy proportion
OCI	ordered-history accuracy minus accuracy after a declared order-destruction control	percentage points
Debiased drift	split-half angular drift minus its within-user permutation mean	unitless $[-1, 1]$ scale
Staleness gap	recent-only accuracy minus complete-history accuracy on the same future outcomes	percentage points
Routing headroom	accuracy of a selected expert minus the declared reference. Hindsight and earlier-window values reported separately	percentage points
Oracle capture	deployable gain divided by true-boundary oracle headroom for the specified population	fraction
Detection rate	fraction of true events followed by an alarm under the matching rule	fraction
Detection latency	event-to-alarm steps among detected events	steps
False-alarm rate	alarms before a true event or on stable histories	alarms per 100 steps

Oracle capture uses the particular numerator, reference, and subgroup specified by its gate. The reported 34% capture concerns change-point users and the corresponding oracle headroom. Hindsight gain likewise uses the best constant expert on the same scored window. The earlier-window rule is an empirical selector, not an upper bound on all learners. Oracle boundary access is privileged information for a specified reset operator, not a guarantee of optimal prediction.

OCI depends on both the policy and the order perturbation. Permutation-centered drift concerns observed nonexchangeability. Predictive staleness concerns the harm of including older evidence under the fixed-prefix comparison in Appendix F.3. Assigned median lag concerns the nominal controller allocation in Appendix C.2, not the complete influence of past observations. *A positive drift statistic, a shorter assigned memory, and a positive prediction gain are different measurements and cannot substitute for one another.*

K Kernel Representation Analysis And Scope

K.1 A Separate Two-Timescale Model

The representation question is easiest to see with a two-speed signal. Imagine a user's observed preference contains a slowly changing baseline and a faster temporary component. A single decay rate must compromise between remembering the slow part and reacting to the fast part. A bank of traces can represent both timescales at once. Appendix K.1 studies that limited representation question in a separate linear-Gaussian model; it is not the main binary-choice experiment.

Let u_t be the slow scalar component and v_t the fast one. Their sum is the latent signal θ_t, and y_t is a noisy real-valued observation:

$$\begin{aligned} u_t &= \rho_u u_{t-1} + \epsilon_{u,t}, \qquad v_t = \rho_v v_{t-1} + \epsilon_{v,t}, \\ \theta_t &= u_t + v_t, \qquad y_t = \theta_t + n_t. \end{aligned} \tag{5}$$

Here $0 < \rho_v < \rho_u \leq 1$, so u_t persists longer than v_t. The independent zero-mean Gaussian noises have positive variances q_u, q_v, r. The unit-root case $\rho_u = 1$ is handled through a stable prediction-error covariance; it does not assume a stationary latent level. This model gives us a numerical target memory curve to approximate. The learned preference controller never receives these latent parameters.

K.2 From A Predictor To Its Lag Kernel

To compare this model with exponential traces, we ask how much each past observation contributes to the next prediction. If yesterday's observation has coefficient $\kappa(0)$ and an observation two steps back has coefficient $\kappa(1)$, the sequence $\kappa(0), \kappa(1), \ldots$ defines the predictor's memory curve. For $x_t = (u_t, v_t)^\top$, $A = \mathrm{diag}(\rho_u, \rho_v)$, $C = (1, 1)$, and $Q = \mathrm{diag}(q_u, q_v)$, we compute the steady-state linear predictor from its prediction-error covariance and obtain the corresponding lag kernel.

$$\begin{aligned} P &= APA^\top + Q - APC^\top (CPC^\top + r)^{-1} CPA^\top, \\ K_p &= APC^\top (CPC^\top + r)^{-1}, \qquad F = A - K_p C, \\ \hat{x}_{t+1|t} &= F\hat{x}_{t|t-1} + K_p y_t. \end{aligned} \tag{6}$$

The last line has a direct reading: F carries the previous estimate forward, while $K_p y_t$ adds the newest observation. The first two lines specify the fixed matrices used by that steady-state predictor.

Why The Two-State Predictor Produces Two Geometric Decays. Repeatedly expanding the last line of Equation 6 gives the coefficient of observation y_{t-j} as $\kappa(j) = CF^j K_p$. A 2×2 matrix with two distinct decay factors λ_1 and λ_2 can be decomposed into two corresponding modes. Therefore

$$\kappa(j) = \sum_{i=1}^{2} w_i \lambda_i^j, \qquad w_i = (CV)_i (V^{-1} K_p)_i, \tag{7}$$

when $F = V\,\mathrm{diag}(\lambda_1, \lambda_2)V^{-1}$. In ordinary terms, this fixed two-state predictor remembers the past as a weighted sum of two geometric decay curves.

The calculation follows directly from repeated substitution: $\hat{x}_{t+1|t} = F^n \hat{x}_{t-n+1|t-n} + \sum_{j=0}^{n-1} F^j K_p y_{t-j}$. Multiplying by C picks out the contribution to the scalar prediction, and substituting the matrix decomposition gives Equation 7. If F is stable, the contribution from a fixed starting state vanishes as the history grows. This identity does not by itself guarantee that the two rates are real, that the weights are positive, or that either rate lies inside our trace bank; the numerical study checks those properties on its recorded grid.

When both rates and weights are positive, the 64-lag target can be normalized into a mixture of two normalized exponentials. With $Z_i = \sum_{j=0}^{63} \lambda_i^j$, the mixture weight is $p_i = w_i Z_i / \sum_l w_l Z_l$. Positive w_i give positive p_i that sum to one. This is the exact condition needed for the trace-bank approximation below; it is not a claim that every admissible process or every trained controller satisfies it.

K.3 Numerical Coverage And Its Limits

Table 24: Numerical kernel approximation on the 90 positive-persistence cells covered by the trace bank. Entries are median normalized-kernel L_1 errors across cells, not preference-prediction accuracies or repeated-seed estimates. The full positive grid has 300 cells. The largest six-trace error on the covered subset is 0.120.

Kernel Class	Median Error	Comparison Set
Single exponential	0.096	90 covered cells
Adjacent trace pair	0.089	Same 90 cells
Six-trace mixture	0.031	Same 90 cells

The comparisons in Table 24 in Appendix K.3 approximate the normalized target with a single exponential, an adjacent pair from the six-trace bank, or a convex mixture over that bank. Let $\bar{\kappa}(j) = \kappa(j) / \sum_{i=0}^{63} \kappa(i)$ denote the normalized target and $\tilde{\kappa}$ its approximation. Error is the L_1 distance $\sum_{j=0}^{63} |\bar{\kappa}(j) - \tilde{\kappa}(j)|$ over 64 lags. It is not a pairwise-accuracy loss. The single-exponential comparator obeys the recorded lower rate bound. It is not an unconstrained search over arbitrarily fast decay.

Across all 300 positive-persistence grid cells, the calculation yields two real decay rates with positive weights. Ninety cells also place the faster decay inside the range covered by the six-trace bank. On those 90 cells, median L_1 error is 0.096 for one constrained exponential, 0.089 for the best adjacent pair, and 0.031 for the full six-trace mixture; the largest six-trace error is 0.120. Thus, the six traces approximate this evaluated two-speed family better than the restricted one-rate comparator, although not perfectly. The negative-persistence extension marks a clear boundary. Among 180 cells, 88.9% of target memory curves are non-monotone and 47.2% become negative at some lag. A nonnegative mixture of decreasing exponentials cannot exactly reproduce a curve that rises again or changes sign. Multiple traces therefore improve approximation on the evaluated model family; this does not imply that LETHE recovers an individual's temporal state, improves binary-choice accuracy, or makes reset optimal.

L REPRODUCIBILITY AND ARTIFACT COVERAGE

The released materials include the experiment code, configurations, and processed evaluation records used for the reported analyses. Appendices A-I specify the prediction timing, data construction, memory policies, fitting procedures, and evaluation protocols needed to interpret these artifacts. Synthetic experiments can be regenerated from their configurations and seeds. Natural-history and checkpoint experiments additionally require access to the corresponding third-party datasets and model checkpoints under their original distribution terms. The release does not redistribute raw user text or third-party model weights. The supplied figure-generation code reconstructs the manuscript figures from the reported numerical records.

M ASSET USAGE, ETHICS, AND INTENDED SCOPE

Assets And Their Scientific Roles. The assets in Table 25 in Appendix M support the experiments defined in Appendix H. PRISM and Arena provide response judgments, PENS supplies click histories, the sentence encoder supplies candidate representations, and the reward checkpoints supply alternative reward-feature spaces.

Table 25: Third-party assets, their roles in the experiments, and primary distribution terms. These assets remain subject to their original licenses and usage conditions.

Asset	Role And Protocol Location	Primary Terms Or Remaining Boundary
PRISM	Human response comparisons, Appendix H.1	Dataset card: human content CC BY 4.0, model-generated content CC BY-NC 4.0, with original provider terms also applicable
Chatbot Arena	Timestamped votes and constructed source switches, Appendix H.1	Dataset card: human content CC BY 4.0 and model-generated content CC BY-NC 4.0, subject to original model terms
PENS	Click-order preference surrogate, Appendix H.1	Official dataset page: research use under the linked Microsoft Research License Terms
`all-mpnet-base-v2`	Frozen sentence representations, Appendix H.2	Model card: Apache 2.0
Skywork Reward-V2 Llama-3.2, 1B and 3B	Checkpoint-feature sensitivity, Appendix H.3	Family model card: Llama 3.2 Community License.
Skywork Reward-V2 Llama-3.1, 8B	Same checkpoint interface	Model card: Llama 3.1 Community License.
Skywork Reward-V2 Qwen3, 1.7B and 8B	Same checkpoint interface	Family model card: Apache 2.0.

Redistribution Boundaries. Third-party histories, generated responses, model weights, and derived assets remain subject to their original distribution terms. The manuscript release, therefore, includes only materials that can be redistributed under the applicable licenses and does not include raw user text, source-user identifiers, or third-party model weights.

Offline Evaluation And User Impact. No new human-subject study or annotation exercise is reported. Using an existing dataset does not itself establish consent for a new deployment, privacy protection, subgroup fairness, or improved user welfare. A false alarm can remove useful evidence, and an adaptive state can misrepresent a stable user. A deployed system would require control over stored history, access restrictions, and assessment of consequential errors across relevant users. Retention as a predictive comparator is not a recommendation to store personal data indefinitely. *The findings concern the evidence needed to justify temporal memory decisions, not permission or safety guarantees for deploying those decisions.*

Author Responsibility. Generative-AI use is described in the AI Use Statement. Explanations proposed with that assistance are not experimental observations. The authors remain responsible for the claims, experimental records, asset permissions, and final submission.

N Scope Of The Empirical Claims

Signal And The Individual Decision. The information paradox separates population evidence from the evaluated prospective expert choices. Its negative result does not prove that individual temporal state is unidentifiable. Sampling noise, different regimes in adjacent windows, representation error, candidate-bank limitations, and learning failure remain possible contributors. The three-expert selector is also distinct from the four memory families and additional fitted candidates. Its hindsight reference measures apparent selection opportunity rather than an attainable gain guaranteed to any deployable learner.

Staleness And The Response To An Event. The fixed-prefix staleness diagnostic tests whether adding older directions improves or harms later prediction under that diagnostic's predictor. It is not a prospective detector of staleness for each individual. The prespecified reset comparison supports its specified change-point population and accuracy criterion, while the separate oracle-capture criterion fails. Shared alarms control event selection in the actuator comparison, although the response policies retain their separately fitted scales. The two Arena splice series (A and B) are distinct constructions, and constructed source switches are not verified natural changes in one person's preferences. None of these results establishes Bayes optimality of reset, a universal response to abrupt change, or the failure of every exact-boundary reset.

Assigned State And Attribution. The exploratory illusion concerns interpreting an individualized memory summary as evidence of individual temporal inference. Stable-user cohorts differ across the compared training populations. Their nominal half-life trajectories are not identical histories subjected to a randomized population-context intervention and do not measure the complete operative influence of the past. Removing history length redistributes accuracy between stable and change-point users without an aggregate improvement on the change-point mixture. Population information can legitimately aid prediction. A learned population-level rule for when change occurs remains a proposed explanation rather than an identified mechanism.

Coverage. The checkpoint experiments provide sensitivity evidence for the reported memory comparisons rather than independent replications of the prospective selector or attribution analyses. Source-user-disjoint Arena histories provide controlled transitions in natural comparisons, but they are not verified within-person preference changes and may retain dependence through secondary sources. The contribution is therefore a set of separately testable distinctions between temporal signal, individual actionability, intervention choice, and state interpretation, not an automatic policy that infers a user's state and selects an optimal memory intervention.

O Extended Related Work And Evaluation Map

Objects Of Comparison. For pairwise personalized reward prediction, our question is whether earlier user evidence supports a memory decision for later choices. Context-aware online reward routing in MiCRo is a close precedent (Shen et al., 2025). The additional comparison here tests transfer of an earlier-window expert choice rather than the success of MiCRo's router. Tables 26 and 27 in this appendix first compare personalized reward and history representations, then broaden

to temporal policy learning, persona revision, and evaluation principles. They describe reported objects and the additional observations used here, not a feature-count ranking or an assumption that prior methods cannot process temporal histories.

User Representations And Online Routing. Variational Preference Learning infers a latent representation from a user's feedback (Poddar et al., 2024). PAL learns shared preference structure so that a new user's location can be estimated from limited comparisons (Chen et al., 2025). MiCRo is not merely a static mixture. It learns preference components and performs context-aware routing, including online updates of mixture weights from additional user feedback (Shen et al., 2025). These are direct precedents for user-conditioned reward prediction and adaptation. Our selector comparison asks a narrower transfer question. Does the expert favored by one user's earlier outcomes improve that user's later outcomes relative to a common expert? The comparison does not establish that MiCRo's online router would fail, because that complete system is not reimplemented here.

Textual User State And In-Context Adaptation. PLUS jointly trains a user-summary generator and a reward model conditioned on the summary (Nam et al., 2026). P-GenRM generates personalized personas and scoring rubrics and uses test-time user-based scaling (Zhang et al., 2026a). P-Check learns to generate a dynamic personalized evaluation checklist from user history (Seo & Lee, 2026). PPT instead adapts a policy directly from in-context preference feedback (Lau et al., 2024). In each case, user information affects the evaluation or generation of a response. Our memory comparison concerns which observations remain active inside a fixed pairwise predictor. A gain from user conditioning does not, by itself, determine whether older observations have become harmful or whether an individualized state identifies temporal change in that user.

Changing Rewards And Persistent-State Revision. NS-DPO models time-dependent preferences through a dynamic Bradley-Terry formulation and discounts older training examples in its objective (Son et al., 2025). This is a temporal weighting parameter, not the individual prospective choice among memory constructions tested here. Shifted dueling bandits characterize when changing pairwise preference distributions can be tracked with dynamic-regret guarantees under structural assumptions (Suk & Agarwal, 2023). Their guarantee concerns sequential choices and regret, rather than the fixed-prefix recent-versus-complete-history accuracy diagnostic. CORE makes revision of a persistent persona an explicit decision, separating turn-local evidence from Commit, Defer, and Ignore updates (Zhang et al., 2026b). This closely related motivation should not be described as an absence of state-update control. The represented object differs from our reward-prediction memory, and our attribution test concerns stable-user cohorts and nominal memory allocation. These are differences in the tested object, not a claim that state revision is new.

Detecting Change And Choosing A Response. Cumulative-sum detection supplies a sequential alarm statistic (Page, 1954), while Bayesian online change-point detection infers a distribution over the current run length (Adams & MacKay, 2007). Forgetting methods address uncertainty about changing parameters (Kulhavý & Zarrop, 1993). Expert tracking compares sequential prediction with a changing expert sequence (Herbster & Warmuth, 1998), and nonstationary optimization relates achievable performance to environmental variation (Besbes et al., 2015). These works supply established operators and guarantees under their own assumptions. Our empirical comparison holds one alarm sequence fixed and measures the consequences of two separately calibrated responses. The additional fixed-prefix diagnostic distinguishes a recency advantage from harm caused by retaining older evidence. Neither comparison asserts a universal new law of forgetting.

Selection Bias And Predictive Evaluation. Cawley and Talbot identify overfitting of a finite-sample model-selection criterion and resulting optimistic evaluation (Cawley & Talbot, 2010). The hindsight expert gain is therefore a diagnostic use of a known phenomenon, not an independent statistical invention. Predict-before-update evaluation, often called prequential evaluation, requires a prediction before incorporating its feedback (Dawid, 1984). Our earlier-window choice uses that ordering while holding the selected expert fixed for later scoring. The policy's causal memory may still update after each newly scored label. The resulting negative comparison concerns the recorded histories and selectors, not an upper bound on all learning procedures.

Table 26: Evaluation map for personalized and temporal objects. The last column identifies the additional question in this study, not an untested capability that the cited method is assumed to lack. The detailed comparisons are in Appendix O.

Work	Reported Object And Adaptation	Additional Comparison In This Study
MiCRo (Shen et al., 2025), EMNLP	Preference-component mixture with context-aware and online routing	Transfer of an observed earlier-window expert winner, distinct from evaluating MiCRo's router
VPL (Poddar et al., 2024), NeurIPS; PAL (Chen et al., 2025), ICLR	Latent user preference and shared structure for sample-efficient user localization	Earlier-window selection of an expert for the same user's later outcomes
PLUS (Nam et al., 2026), ICLR	Learned user summary and co-adapted reward model	Which older observations help or hurt inside a fixed reward predictor
P-GenRM (Zhang et al., 2026a), ICLR; P-Check (Seo & Lee, 2026), ACL	User-conditioned personas, rubrics, and dynamic evaluation criteria	Whether adaptive temporal memory on stable-user cohorts supports a claim of individual change
PPT (Lau et al., 2024), preprint	Policy adaptation from in-context preference feedback	Reward-memory construction rather than complete policy adaptation
NS-DPO (Son et al., 2025), ICML	Time-dependent reward learning with exponential example weighting	Recent-versus-old evidence and recent-versus-complete history as distinct accuracy contrasts
Shifted dueling bandits (Suk & Agarwal, 2023), NeurIPS	Tracking preference shifts with conditional dynamic-regret guarantees	A fixed-predictor memory diagnostic, not a new regret guarantee
CORE (Zhang et al., 2026b), preprint	Controlled commitment, deferral, or rejection of persistent persona updates	Predictive memory and its response to recorded alarms, rather than persistent persona revision

Table 27: Evidence needed for the distinctions tested here. The comparisons apply to the stated study protocols and do not claim that all prior work lacks their constituent ideas. See Appendix O for the connections to selection bias, change detection, and reward-model evaluation.

Inference	Insufficient Evidence By Itself	Comparison And Its Boundary
Individual actionability	Population temporal signal or a per-user winner selected on the scored outcomes	Earlier-window choice evaluated later. The result concerns the three-expert bank, not every selector.
Predictive staleness	Recent evidence outperforming old evidence	Complete history versus recent-only history on matched later outcomes under the fixed-prefix diagnostic
Memory response	An alarm, a temporal boundary, or success of one end-to-end adaptive system	Reset versus temporary fast discount under shared alarms. Response scales remain separately calibrated.
Temporal attribution	Different user states or improved average personalized prediction	Stable-user cohorts and history-length ablation. These are exploratory rather than identical-history causal interventions.
Downstream benefit	Reward accuracy alone	Downstream reward-model evaluations address a complementary question (Frick et al., 2025; Ma et al., 2026). No downstream policy gain is claimed here.

Reward Accuracy And Downstream Personalization. PPE tests whether reward-model measurements predict downstream alignment performance (Frick et al., 2025). Personalized RewardBench tests personalized reward accuracy and its relation to Best-of-N selection and PPO, including controls intended to separate user preference from general response quality (Ma et al., 2026). Those

controls address meaningful aspects of personalization and should not be described as no attribution analysis. Our specific attribution question is temporal. Does a changing assigned memory for a stable user establish that the model inferred an individual preference change? Cohort contrasts and the history-length ablation provide exploratory evidence about that interpretation, not a replacement for downstream reward-model evaluation.

The scientific distinction lies in the inference supported by a comparison, not in renaming familiar operators. The separate tests concern population signal versus individual actionability, recency versus predictive staleness, event selection versus memory response, and individualized state versus evidence of individual temporal change. LETHE supplies one multiscale memory construction for those tests. The experiments do not establish an unprecedented combination of all adaptive-learning ingredients or a universal failure of the cited systems.