MORE CORRECT LABELS CAN HURT: DELAYED FEEDBACK IN STREAMING ACTIVITY RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

In streaming activity recognition, a requested label may arrive after the activity has changed. Unsupervised test-time adaptation updates batch statistics (BN), minimizes entropy (Tent), or filters predictions (OATTA), while delayed replay (IWMS) learns from historical labels. These approaches leave open how a returned label should enter the current activity state and influence subsequent predictions. By analyzing errors at label arrival, we find that directly injecting a correct historical label can overwrite a correct current prediction. Local loss analysis further shows that even a useful correction can become harmful when applied too strongly. We therefore propose Gated Arrival, which separates historical learning, state injection, and output mixing. A calibrator learns from returned labels and their saved origin predictions. Two gates fitted offline use label age and current predictions to set the weight of state injection and its later contribution to the output, with the recognition backbone frozen. Experiments on five activity datasets under simulated response delays show that additional correct labels can reduce recognition and that delay can reverse feedback-rule rankings. Gated Arrival reduces errors caused by labels that no longer match the current activity. Component comparisons show how historical calibration and state correction affect recognition and probability estimates.

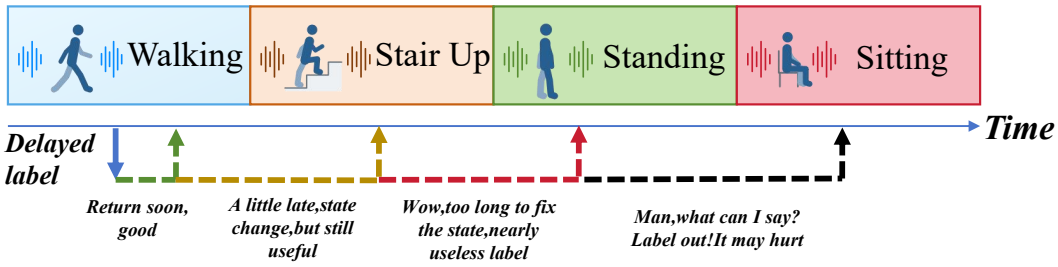


Figure 1: **A correct historical label can mislead the current decision.** A walking label arrives at different points in a changing activity stream. Its origin remains valid supervision; its relevance to the current state can diminish.

1 INTRODUCTION

Wearable activity recognition must follow a changing activity stream while adapting to differences between people. Test-time adaptation addresses such distribution shifts through normalization and entropy-based updates (Schneider et al., 2020; Wang et al., 2021; Niu et al., 2023; Yuan et al., 2023), or by maintaining supports and prototypes (Iwasawa & Matsuo, 2021; Jang et al., 2023; Wang et al., 2023). For sensor streams, temporal structure offers another source of information: STAD tracks evolving prototypes (Schirmer et al., 2025), OATTA combines filtering with source-predictor fallback (Kim et al., 2026), and SIGHT uses feature changes to regulate temporal inertia (Zhou et al., 2026). Reliability selection in EATA (Niu et al., 2022) and physical constraints in PITTA (Li et al., 2026) further support adaptation without labels. These approaches use observations available at prediction time. A system that can occasionally ask its user for an activity label has an additional source of supervision, but also a timing problem that unlabeled adaptation does not encounter.

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

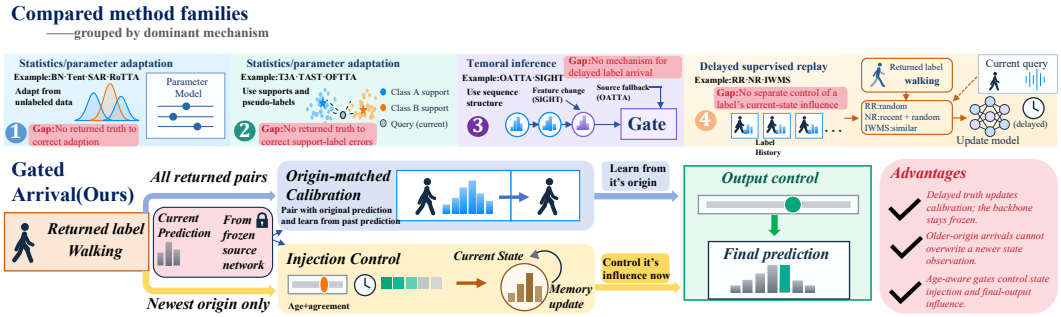


Figure 2: **Where delayed supervision enters the prediction.** The first three method families use no returned labels; delayed replay learns from historical examples. Gated Arrival separates historical learning, state injection and output mixing. Families indicate dominant mechanisms; the highlighted gaps concern annotation-return delay. Section 4 describes the supervision available to each method.

A returned answer describes the activity when the question was asked. Two minutes after a walking window, the user may already be sitting (Figure 1). The answer remains correct for its origin, yet using it as evidence of the present activity can overwrite a correct prediction. Active TTA methods such as SimATTA and HILTTA use annotations to adapt on selected samples (Gui et al., 2024; Li et al., 2024); EATTA reduces the effort required to obtain them while recognizing that responses can still be delayed (Wang & Ding, 2025). BiTTA explicitly evaluates delayed binary feedback over image batches (Lee et al., 2025). Delayed feedback is therefore already part of adaptation research. The unresolved issue we study is how a returned activity label should influence a prediction stream after the labeled event has passed.

The distinction matters because learning and state estimation ask different questions of the same label. Delayed online learning updates a model from historical examples (Joulani et al., 2013), and IWMS retrieves labeled examples similar to current inputs (Botos et al., 2024). Such updates can correct persistent prediction errors even after the activity changes. Injecting the returned label into a temporal state instead assumes that the past activity remains informative about the present. This connects to out-of-sequence observations in tracking (Bar-Shalom et al., 2004), but here the output is a class probability vector from a frozen recognition network. Neither the correctness of an annotation nor its age alone determines whether that correction improves the current prediction.

We investigate this distinction through Gated Arrival, which separates *historical calibration*, *state injection*, and *output mixing* (Figure 2). Every returned label updates a small calibrator using the saved prediction at its own origin. A separate gate uses the current source evidence to determine how strongly the newest returned observation resets the activity state, so a return does not automatically replace the current prediction. An output gate then combines the historical and state predictions as the stream proceeds. Both gates use causal features, including age, persistence and prediction agreement; their coefficients are fitted offline. A source compatibility check selects fallback when temporal filtering is unsuitable. The backbone remains frozen, and online learning updates only $2K$ calibration scalars for K classes. This construction makes the uses of supervision explicit and allows each contribution to be tested under shared feedback schedules.

Our analysis asks when a returned label improves the current prediction and how strongly it should influence that prediction. Two local loss identities characterize these decisions under accuracy and Brier loss. Complete-stream experiments then measure how historical learning and state correction affect subsequent predictions.

Our evaluation covers 42 participants from five activity datasets under simulated responses. We vary delay and request rate while sharing source models, windows and feedback schedules across methods. This complements latency-aware query selection (Pham et al., 2022; Castellani et al., 2022) by holding the query policy fixed within a comparison and studying the use of the returned information. Predictions are committed once, rather than continuously re-evaluated as labels arrive (Grzenda et al., 2020). The experiments show that doubling the request rate at a delay of 120 events reduces UCI HAR Macro-F1 by 5.38 points under hard state injection (Raw Arrival; 95% participant interval $[-6.18, -4.59]$), compared with 0.17 under Gated Arrival. Delay also reverses

feedback-rule rankings, and refitted smaller models show that state injection contributes much more on CAPTURE-24 than on the other datasets. Recognition and probability scores reveal different benefits of historical learning and temporal correction.

Our contributions are the following:

- We formulate the distinction between historical correctness and present relevance, and characterize immediate hard and soft corrections under accuracy and Brier loss (Section 2).
- We introduce Gated Arrival, an origin-aligned calibration and gated state-correction method that separates the learning and predictive uses of delayed supervision (Section 3).
- We establish how label budget, response delay and feedback use jointly affect streaming recognition through a 43-service sweep, paired budget comparisons and refitted component removals (Section 4). These comparisons identify both useful state corrections and conditions where simpler rules recover or exceed their benefit.

2 Historical Correctness and Present Relevance

We consider online activity recognition with delayed feedback. At decision t, a frozen source network maps sensor window x_t to class probabilities $p_t = f_\theta(x_t) \in \Delta^{K-1}$ for activity $y_t \in \{1, \dots, K\}$. Predictions use observations and labels already available and are never revised. A label y_o returned at $t > o$ remains valid supervision for its origin window x_o, even after the activity changes. Using it to correct the current prediction raises a different question: when does information about the past improve recognition of the present?

When does replacement help? To determine whether replacement helps, we compare the returned label's agreement with the current activity against the accuracy of the prediction it replaces. Let b_t be Base's top-1 prediction. On a fixed population $\mathcal{E}$ of eligible release decisions, we replace b_t with y_o, leaving other predictions unchanged. Using the same decisions and weights, define label–activity agreement $m_\mathcal{E} = \Pr(y_o = y_t \mid \mathcal{E})$ and Base accuracy $\alpha_\mathcal{E} = \Pr(b_t = y_t \mid \mathcal{E})$. The expected accuracy change on $\mathcal{E}$ is

$$\Delta_\mathcal{E} = \mathbb{E}[\mathbb{1}\{y_o = y_t\} - \mathbb{1}\{b_t = y_t\} \mid \mathcal{E}] = m_\mathcal{E} - \alpha_\mathcal{E}. \tag{1}$$

Thus hard replacement improves accuracy exactly when $m_\mathcal{E} > \alpha_\mathcal{E}$. A label that often matches the present activity can still reduce accuracy if the model is more accurate on those decisions. Historical correctness alone does not justify replacement (Appendix L).

How much correction helps? We next consider partial correction: mixing a current prediction h with a candidate based on feedback c. Let X contain information available at the current decision, including returned labels. Fix these two X-measurable probability vectors, and write $z_\eta = h + \eta v$ with $v = c - h$ and $\eta \in [0, 1]$. With one-hot e_y, let $\mu = \mathbb{E}[e_{y_t} \mid X]$ be the conditional current-class distribution. For multiclass Brier loss $\ell_B(z, y) = \|z - e_y\|_2^2$, expanding the square gives

$$\mathbb{E}[\ell_B(z_\eta, y_t) - \ell_B(h, y_t) \mid X] = 2\eta(h - \mu)^\top v + \eta^2 \|v\|_2^2. \tag{2}$$

A small positive correction improves this risk when $(h-\mu)^\top v < 0$; the cost grows quadratically with its strength. Thus a correction must counter the current prediction's error and have an appropriate weight. The distribution μ is unknown at deployment; Appendix L derives the optimum and NLL-plus-Brier expression.

For example, take $\mu = (0.6, 0.4)$, $h = (0.8, 0.2)$ and $c = (0.2, 0.8)$. The risk change is $-0.48\eta + 0.72\eta^2$: a weight of $1/3$ minimizes it, whereas full replacement increases risk by 0.24. Thus even a useful correction can increase error if its weight is too large.

From arrival to subsequent predictions. A label may match the activity at release, yet continue to influence filtering after the activity changes. We therefore separate learning from the labeled origin, updating the activity state, and weighting its contribution to later predictions. Section 3 uses historical and state predictions as h and c, with a mixing weight set by a gate. Table 22 examines immediate corrections; complete-stream experiments measure accumulated effects across budgets and delays.

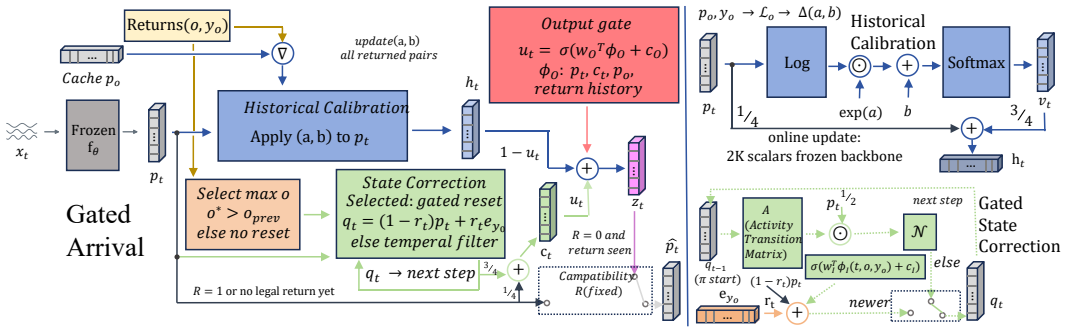


Figure 3: **Historical calibration and state injection.** Left: all returns calibrate cached p_o; only newly selected newest origins trigger state correction. Right: $2K$ calibration scalars update with a frozen backbone; gated resets alternate with temporal filtering. At selected returns, $r_t = 0$ resets to p_t. Gate coefficients and fallback flag R are fixed online. Output is Base before the first legal return or when $R = 1$.

3 Gated Arrival: Separating the Uses of Feedback

We design Gated Arrival to learn from returned labels while adjusting their influence on current recognition (Figure 3). Every return trains a calibrator on its saved origin prediction, preserving supervision even after the activity changes. A separate state path incorporates the newest returned observation through an injection gate. An output gate then reassesses the state's contribution at each prediction.

3.1 Request and Release Protocol

We process each decision in *release, predict, request* order. Each window x_t has timestamp τ_t and segment s_t; a label is requested with fixed Bernoulli probability ρ after committing its prediction. A request at origin o has due time $\tau_o + d_o$ and is released at the first decision t satisfying $\tau_o + d_o < \tau_t$ in the same segment. Thus even $d_o = 0$ provides no label for the prediction at o. Segment boundaries reset adaptation state and cancel outstanding requests.

Responses may arrive out of order. We use every released origin–label pair for calibration, but trigger injection only from the newest released origin when it exceeds the previously selected origin. Thus an older response still supplies supervision without displacing a newer state observation. We retain requested source probabilities until processing and keep the selected origin probability available to the gates.

3.2 Two Paths for a Returned Label

Historical calibration. We use historical feedback to correct classwise prediction errors while keeping the recognition network fixed. Pairing each label with the prediction made when it was requested preserves the target across activity changes. We adapt vector scaling (Guo et al., 2017) to log probabilities, with positive classwise scales $\exp(a)$ and biases $b \in \mathbb{R}^K$, both parameter vectors initialized to zero at each segment. We mix the result with p_t:

$$v_t = \mathrm{softmax}(\exp(a) \odot \log p_t + b), \qquad h_t = \tfrac{1}{4} p_t + \tfrac{3}{4} v_t. \tag{3}$$

On return of (o, y_o), one gradient step updates a, b using $-\log v_o(y_o) + \frac{\lambda}{2}(\|a\|_2^2 + \|b\|_2^2)$, where v_o is computed from saved p_o with current parameters. Learning therefore uses the origin's own error and updates only $2K$ scalars, with no backbone update (step size and constraints: Appendix B).

State injection and temporal filtering. To track activity between returns, we use a row-stochastic transition matrix A and class prior π fitted on the evaluated dataset's prescribed source/development participants, excluding the target participant (Appendix B). For $\mathcal{N}(z) = z / \sum_k z_k$, we combine

predicted transitions with tempered source scores:

$$\bar{q}_t = \begin{cases} \mathcal{N}\left(\pi \odot p_t^{1/2}\right), & \text{at a segment start,} \\ \mathcal{N}\left((q_{t-1}A) \odot p_t^{1/2}\right), & \text{otherwise.} \end{cases} \tag{4}$$

Here p_t supplies discriminative scores; Appendix F tests division by the source prior with fixed gates. At a newly selected return (o, y_o), we reset the state to a weighted average of the current source prediction and the returned class:

$$r_t = \text{sigmoid}(w_I^\top \phi_I(t, o, y_o) + c_I), \qquad q_t = (1 - r_t)p_t + r_t e_{y_o}; \tag{5}$$

otherwise $q_t = \bar{q}_t$. Here e_{y_o} is the one-hot returned label. The reset retains current evidence when injection is suppressed: $r_t = 0$ gives $q_t = p_t$, not $\bar{q}_t$. The state candidate is $c_t = \frac{1}{4}p_t + \frac{3}{4}q_t$ after the first legal return in the segment. Before that return the reported prediction is p_t.

We condition injection on 12 causal features: origin age in decision steps, source persistence, confidence, probability disagreement and variation, and return count (Appendix B.2). These features place label age alongside current evidence without observing the current true activity.

3.3 Output mixing and source compatibility

The activity can change between one label return and the next. We therefore reassess the state at every prediction, using an output gate to mix it with historical calibration:

$$u_t = \text{sigmoid}(w_O^\top \phi_O(t, o, c_t) + c_O), \qquad z_t = (1 - u_t)h_t + u_t c_t. \tag{6}$$

The injection gate sets the state used in later filtering; the output gate sets its weight in the current prediction. Its 14 features include label age, time since return, candidate/source agreement and entropy, and source persistence (Appendix B.2), allowing the weight to respond to new observations between returns. Both candidates retain a quarter-weight source anchor, so $z_t(k) \geq p_t(k)/4$ for every class. Source evidence therefore remains in the output at every mixing weight.

We fit the gates offline by minimizing NLL plus Brier (Gneiting & Raftery, 2007), with L_2 regularization, using the other four datasets and excluding the entire evaluated dataset. Injection fitting precedes output fitting; we freeze the $13 + 15 = 28$ coefficients (including biases) online while recomputing gate values from causal features. Appendix B specifies fitting; the supplementary `evidence/CONTROLLERS.json` records streams, coefficients and feature statistics.

Compatibility with source evidence. We assess the source temporal model before deployment by comparing the always-active unlabeled filter $\tilde{p}_t = \frac{1}{4}p_t + \frac{3}{4}\bar{q}_t$ with Base on source/development fitting participants. After the first legal return,

$$R = \mathbb{1}\{n_{\text{fit}} \geq 2,\ \Delta\text{NLL} > 0.02,\ \Delta\text{Brier} > 0.01\}, \qquad \hat{p}_t = \begin{cases} p_t, & R = 1, \\ z_t, & R = 0. \end{cases} \tag{7}$$

Deltas are participant-mean filter-minus-Base differences; n_{fit} counts fitting participants. Their labels set R, excluding the target; R remains fixed per checkpoint and fitting pool. This separates source compatibility from the gates' decisions about individual returns and predictions. $R = 1$ bypasses both paths; Appendix F also tests retaining history under fallback.

Relation to simpler feedback rules. *History* outputs h_t after the first return. *Raw Arrival* sets $q_t = e_{y_o}$ on every release, including an older origin that arrives later. *Direct* instead starts from e_{y_o} at the newest returned origin and re-filters the intervening source observations up to t. A *half mixture* averages historical calibration and hard newest-origin Arrival. These rules share source models and feedback schedules; matched comparisons also share fallback and development data for constant, age-based and single-gate rules.

Online footprint. Each decision requires one frozen-backbone inference, an $O(K^2)$ state step and $O(K)$ calibration/gating work, with $O(K)$ additional work per returned label. With B outstanding or retained requests, storage is $O(K^2 + BK)$ for the transition model and cached source probabilities. Appendix O reports state-only timings and full adapter measurements.

4 EXPERIMENTS

We evaluate delayed supervision across response delays, feedback budgets and component comparisons to distinguish historical learning from state correction.

4.1 EXPERIMENTAL SETUP

We evaluate 42 participants on CAPTURE-24 ($n = 12$, four classes) (Chan et al., 2024), HARTH ($n = 8$, 12 classes) (Logacjov et al., 2021), OPPORTUNITY ($n = 4$, 18 labels including NULL) (Chavarriaga et al., 2013), PAMAP2 ($n = 9$, 12 classes) (Reiss & Stricker, 2012), and UCI HAR ($n = 9$, six classes) (Anguita et al., 2013). Backbones are a three-block sensor CNN and, on CAPTURE-24, the authors' small residual model. Source training and within-dataset fitting exclude the evaluated participant; gates are fitted on the other four datasets (Appendices A–B).

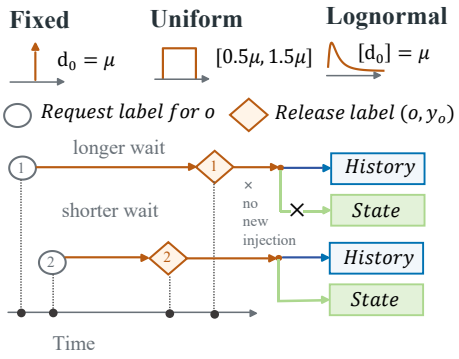


Figure 4: **Out-of-order returns.** Both labels calibrate history; the older return cannot overwrite newer state.

A *feedback service* specifies request rate, delay and segment-reset policy (Figure 4). The 43-service sweep covers 15 fixed delays from 0 to 300, plus uniform and lognormal delays at every positive mean. Delay is measured in seconds except on UCI, where released-window events specify sequence order. Matched comparisons pair fixed/LN delays at $d = 0, 30, 120, 300$. For budget interventions, we reuse per-segment random draws to form nested request sets, so increasing the rate adds labels to the original requests. We aggregate results within each participant and weight people equally; paired 95% intervals bootstrap participants. The supplementary records specify the run configurations for each comparison.

Comparators include unlabeled TAST, RoTTA, OATTA and SIGHT, delayed replay (RR, NR and IWMS), and our History, Direct and Raw Arrival rules. Replay shares our returned labels; unlabeled methods receive none, and our gates additionally use labeled offline streams. Macro-F1 (F1 below) measures recognition; NLL and Brier are proper probability scores (Gneiting & Raftery, 2007), and 15-bin ECE measures top-label calibration (Guo et al., 2017). Raw Arrival is unguarded in budget and delay comparisons; matched component comparisons share fallback. Full configurations, all metrics and evaluation settings are given in the appendices. Full denotes Gated Arrival in compact figure and table labels.

4.2 COMPARISON ACROSS RESPONSE DELAYS

Comparison with adaptation baselines. We evaluate Gated Arrival against Base across all 43 services, then compare it with unlabeled adaptation and RR, NR and IWMS under the same returned-label schedules at $d = 120$ (Table 23). Across the service sweep, Gated Arrival improves F1 over Base in 172 of 215 dataset/service conditions and matches it in the other 43. At $d = 120$, it leads the established methods on CAPTURE-24, HARTH and PAMAP2; TAST leads on OPPORTUNITY and RoTTA on UCI. We also form paired participant-level comparisons against OATTA: the CAPTURE-24 gain is 4.96 points $[3.60, 6.36]$ at $d = 120$ and 2.08 $[0.94, 3.25]$ at $d = 300$; the HARTH and PAMAP2 intervals at $d = 120$ include zero (Appendix E).

Varying response delay. To examine when each feedback rule is useful, we vary fixed delay from 0 to 300 and compare Full with Raw Arrival, Direct and History (Figure 5). On CAPTURE-24, Raw Arrival leads at all 13 measured delays through 120 seconds: at $d = 30$, its F1 is 81.00 versus 79.81 for Gated Arrival. Full leads by 0.97 points at $d = 240$ and 1.86 at $d = 300$; PAMAP2 shows a crossover between $d = 60$ and 80. We also observe differences across streams: UCI favors Direct at all measured delays from 10 to 120, while HARTH favors History at 10, 20, 30 and 60 seconds. These comparisons show how the preferred rule changes with delay and dataset. Table 21 gives post-hoc reversal anchors; Appendix H gives all short-delay results.

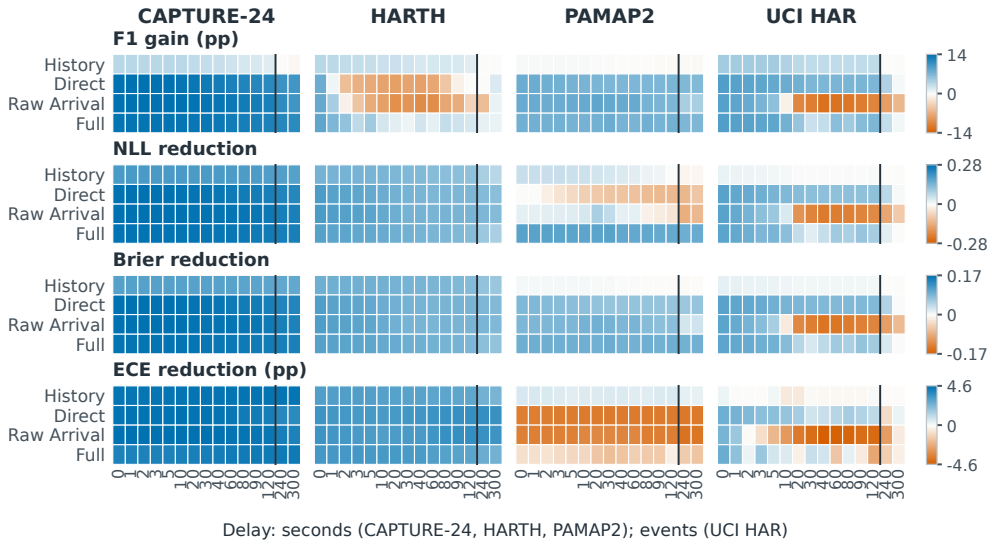


Figure 5: **Feedback paths and metrics respond differently to delay.** All 15 delays are labeled categorically; the divider follows $d = 120$. Blue denotes improvement over Base; orange denotes deterioration; white denotes equality. Symmetric-log scales are shared across datasets per metric. Equal participant means. Table 20 gives all short-delay F1 values; Figure 16 includes OPPORTUNITY and random delays.

Matching returned labels and evaluation windows. We construct a matched comparison to account for changes in label availability before resets. We intersect origins returned at $d = 0, 30, 120$ and evaluate shared windows after every service has activated. We first aggregate the matched comparisons within each participant. On UCI, Raw Arrival's gain over Base falls from $+2.00$ at zero delay to -11.41 at $d = 120$, while Full changes from $+2.61$ to $+1.53$. The shared windows cover 57.4% of valid UCI exposure (Table 17). Matching the final labels and scored windows therefore retains the loss from hard injection; return timing and state trajectories still differ across services.

4.3 Effect of the feedback budget

We double nested requests at fixed $d = 120$, keeping participants, checkpoints and per-segment random draws fixed. All three rules receive the same additional supervision; Table 1 reports full-stream F1 changes and paired intervals.

Table 1: **More correct labels can help or hurt.** Full-stream Macro-F1 change (pp) when nested requests double ($2 \times -1\times$), at $d = 120$ seconds (UCI: events). Entries are participant means [95% bootstrap CI], computed from within-person aggregate scores. Raw Arrival and Half mixture are unguarded. Zero denotes no budget effect, not Base.

Dataset	Raw Arrival	Half mixture	Gated Arrival
CAPTURE-24	$+3.10\,[2.44, 3.87]$	$+2.45\,[2.12, 2.83]$	$+3.05\,[2.50, 3.62]$
HARTH	$-1.08\,[-1.58, -0.67]$	$-0.14\,[-0.43, 0.08]$	$+0.05\,[-0.10, 0.24]$
OPPORTUNITY	$-2.75\,[-4.52, -1.05]$	$-1.84\,[-3.17, -0.59]$	$+0.00\,[0.00, 0.00]$
PAMAP2	$+0.10\,[-0.12, 0.31]$	$+0.16\,[0.05, 0.28]$	$+0.23\,[0.05, 0.39]$
UCI HAR	$-5.38\,[-6.18, -4.59]$	$-0.16\,[-0.35, 0.04]$	$-0.17\,[-0.39, -0.03]$

The budget effect depends on the feedback rule. All three rules benefit on CAPTURE-24; Raw Arrival and Gated Arrival gain 3.10 and 3.05 F1 points, respectively. Raw Arrival instead loses 1.08 points on HARTH, 2.75 on OPPORTUNITY, and 5.38 on UCI $[-6.18, -4.59]$. Gated Arrival limits the UCI loss to 0.17 points $[-0.39, -0.03]$; Half mixture has a similar mean change of -0.16. Both rules reduce the weight of returned labels and show smaller losses than Raw Arrival.

Table 2: **Feedback rules across response delays.** Macro-F1 (%), averaged within person, within dataset, then equally across datasets. Fixed/lognormal (LN) delays have mean d seconds (UCI: events) and coincide at $d = 0$. Rules share fallback; Source filter is always active. Blue: Full. Dataset scores and paired intervals: Appendix F.1, Tables 12–14.

Method	$d = 0$		$d = 30$		$d = 120$		$d = 300$	
	Fixed	LN	Fixed	LN	Fixed	LN	Fixed	LN
Base	60.51	60.51	60.51	60.51	60.51	60.51	60.51	60.51
Source filter	58.41	58.41	58.41	58.41	58.41	58.41	58.41	58.41
Raw Arrival + fallback	64.96	64.96	61.93	62.49	61.25	61.60	61.54	61.49
Half mixture + fallback	64.34	64.34	63.19	63.50	62.64	62.86	61.83	62.42
Fitted constant	62.99	62.99	61.96	62.23	61.79	61.87	61.34	61.67
Age/persistence	64.20	64.20	62.72	62.90	62.08	62.35	61.52	61.98
Output-only (refit)	64.99	64.99	62.71	63.66	62.04	62.86	61.76	62.34
No injection (refit)	61.98	61.98	61.88	61.88	61.78	61.85	61.55	61.80
No history (refit)	64.71	64.71	63.56	63.80	62.81	63.18	62.04	62.67
Gated Arrival	64.71	64.71	63.58	63.82	62.83	63.20	62.04	62.65

Examining errors at label return. To trace this loss, we examine additional release events where the returned label disagrees with the current activity. The fraction of these events that turn a correct Base prediction into an error is 73.20% on HARTH and 96.67% on UCI for Raw Arrival, versus 1.44% and zero for Full. We pool events within person and then average eligible participants; current labels are used only for evaluation (Appendix F).

Each extra return supplies a historical training pair and an opportunity to reset the state. When the activity has changed, hard injection starts subsequent filtering from the previous activity class. Gating reduces this immediate harm while retaining the returned pair for historical learning.

4.4 Analysis of the feedback paths

We compare smaller models to separate the contributions of historical learning and state correction. Requests, returns, development data, fitting objective and fallback are shared. We refit each reduced model's output gate, allowing it to recover performance after component removal (Figure 6; Table 2). Fixed-gate deletions are reported separately in Table 9.

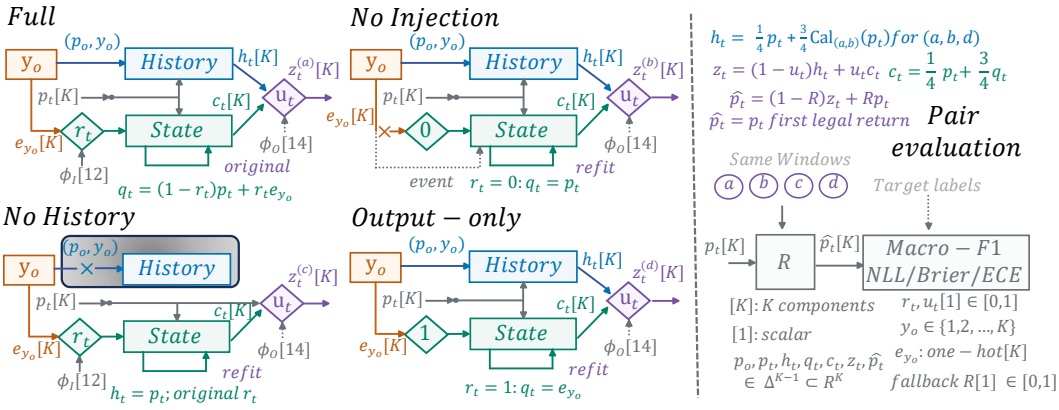


Figure 6: **Matched component comparisons.** Full retains both paths. No injection sets $r_t = 0$ but keeps return-event resets; No history sets $h_t = p_t$; Output-only sets $r_t = 1$. The three reduced rules refit the output gate on the same development pool. All share requests, returns and fallback R. Typed paths and fitting details: Appendix F.

Gating at injection and output. Fixing $r_t = 1$ and refitting the output gate tests output gating alone (Table 2). Across the six positive-delay services, Full gains 0.16–0.88 aggregate F1 points over Output-only and 0.19–0.39 over the half mixture. Output-only leads by 0.28 at zero delay. Paired intervals cross zero for Output-only at LN30 and for the half mixture at fixed $d = 120$ (Table 14). Full also exceeds the fitted constant and age/persistence rules in aggregate F1 across both delay laws, supporting the use of current prediction evidence to weight delayed feedback.

Component contributions across delays. Table 3 compares refitted removals at four delays. On CAPTURE-24, removing injection costs 10.63, 9.10, 5.46 and 2.59 F1 points as d increases from 0 to 300. UCI instead loses 1.15 points at zero delay but gains 0.73 at 30, 0.15 at 120 and approximately zero at 300. Each comparison retains return-event resets and refits the output gate.

Historical learning and metric choice. Removing history changes aggregate F1 by less than 0.03 points per service, partly because dataset effects offset, but raises mean NLL on all four active datasets under both delay laws. CAPTURE-24's fixed-delay NLL cost grows from 0.0085 to 0.0153 as injection's F1 benefit shrinks. NLL captures probability changes without class changes; ECE measures confidence–accuracy agreement and worsens in 60 of 215 Full conditions. Full scores and paired comparisons: Appendix F.1; aggregate NLL plus Brier comparisons: Table 15.

Table 3: **Component effects across delays.** Paired means (reduced minus Full), with refitted output gates and shared fallback. Cells: ΔF1 (pp) / $10^3\Delta$NLL; higher F1/lower NLL is better. Fixed d: seconds (UCI: events). OPPORTUNITY has zero changes under Base fallback.

Dataset	$d = 0$	$d = 30$	$d = 120$	$d = 300$
Remove state injection				
CAPTURE-24	-10.63 / $+124.2$	-9.10 / $+105.6$	-5.46 / $+73.2$	-2.59 / $+45.8$
HARTH	-1.30 / $+16.7$	-0.02 / $+2.9$	-0.02 / $+1.0$	-0.01 / $+0.4$
PAMAP2	-0.57 / $+19.6$	-0.11 / $+7.4$	$+0.08$ / $+3.7$	$+0.17$ / $+1.0$
UCI HAR	-1.15 / $+32.0$	$+0.73$ / -12.8	$+0.15$ / -2.7	0.00 / -0.1
Remove history				
CAPTURE-24	$+0.32$ / $+8.5$	$+0.29$ / $+9.3$	$+0.21$ / $+12.1$	$+0.07$ / $+15.3$
HARTH	-0.14 / $+6.9$	-0.13 / $+8.2$	-0.08 / $+6.5$	$+0.02$ / $+3.4$
PAMAP2	-0.03 / $+20.4$	-0.06 / $+19.2$	-0.07 / $+14.0$	-0.06 / $+10.5$
UCI HAR	-0.13 / $+2.0$	-0.22 / $+4.7$	-0.15 / $+1.9$	-0.02 / $+0.1$

A common explanation: present relevance. Historical labels remain valid for origin-paired learning, while state correction must improve on current evidence (Eq. 1). At $d = 120$, label–activity agreement versus Base accuracy is 94.62% versus 78.77% on CAPTURE-24 and 88.21% versus 94.14% on HARTH (Table 22). Feedback weights and subsequent filtering determine how corrections persist after activity changes (Eq. 2; Section 4.3). Appendix F.1 extends this explanation across delays.

5 Conclusion

In streaming activity recognition, a correct historical label can overwrite a better current prediction, and even a useful correction can become harmful when given too much weight. These observations motivate Gated Arrival: each returned label trains the historical calibrator, while label age and present evidence determine its weight in state injection and subsequent predictions. Across five datasets, increasing the label budget can reduce recognition, and delay can reverse method rankings. Component comparisons show why historical learning and state correction serve different purposes: the former mainly improves probability estimates, whereas the latter depends on whether past labels improve on current evidence. Historical learning can therefore benefit from labels that no longer help current-state correction. Evaluation should account for the timing and use of feedback alongside annotation budget.

AI USE STATEMENT

We used generative AI tools for three purposes: (i) writing assistance, including drafting, revising, and polishing the manuscript; (ii) retrieval and discovery, including finding related work and organizing the literature; and (iii) research ideation and execution, including developing and refining hypotheses and methods, assisting with mathematical derivations and proofs, designing and implementing experiments, and interpreting results. The authors reviewed all AI-assisted work, including the manuscript text, mathematical derivations, implementation code, and experimental results, and take full responsibility for the final content, claims, and artifacts reported in this work.

REPRODUCIBILITY STATEMENT

Appendices A and B specify the data splits, feedback services, model configurations, gate features and fitting procedure. Appendix L gives the loss derivations; the remaining appendices report complete comparisons, evaluation settings and computational costs. The supplementary records provide gate coefficients and participant-level numerical results.

REFERENCES

Motasem Alfarra, Hani Itani, Alejandro Pardo, Shyma Alhuwaider, Merey Ramazanova, Juan C. Pérez, Zhipeng Cai, Matthias Müller, and Bernard Ghanem. Evaluation of Test-Time Adaptation Under Computational Time Constraints. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 976–991. PMLR, 2024.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 437–442, 2013.

Y. Bar-Shalom, Huimin Chen, and M. Mallick. One-step solution for the multistep out-of-sequence-measurement problem in tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 40 (1):27–37, 2004. doi: 10.1109/taes.2004.1292140.

Csaba Botos, Wenxuan Zhang, Matthias Müller, Ser-Nam Lim, Mohamed Elhoseiny, Philip Torr, and Adel Bibi. Label Delay in Online Continual Learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 119976–120012, 2024. doi: 10.52202/079017-3813.

Andrea Castellani, Sebastian Schmitt, and Barbara Hammer. Stream-Based Active Learning with Verification Latency in Non-stationary Environments. In *Artificial Neural Networks and Machine Learning – ICANN 2022*, pp. 260–272, 2022. doi: 10.1007/978-3-031-15937-4_22.

Shing Chan, Yuan Hang, Catherine Tong, Aidan Acquah, Abram Schonfeldt, Jonathan Gershuny, and Aiden Doherty. CAPTURE-24: A large dataset of wrist-worn activity tracker data collected in the wild for human activity recognition. *Scientific Data*, 11(1):1135, 2024. doi: 10.1038/s41597-024-03960-3.

Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R. Millán, and Daniel Roggen. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15): 2033–2042, 2013. doi: 10.1016/j.patrec.2012.12.014.

Tilmann Gneiting and Adrian E. Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Maciej Grzenda, Heitor Murilo Gomes, and Albert Bifet. Delayed labelling evaluation for data streams. *Data Mining and Knowledge Discovery*, 34(5):1237–1266, 2020. doi: 10.1007/s10618-019-00654-y.

Shurui Gui, Xiner Li, and Shuiwang Ji. Active Test-Time Adaptation: Theoretical Analyses and An Algorithm. In *International Conference on Learning Representations*, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017.

Yusuke Iwasawa and Yutaka Matsuo. Test-Time Classifier Adjustment Module for Model-Agnostic Domain Generalization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2427–2440, 2021.

Minguk Jang, Sae-Young Chung, and Hye Won Chung. Test-Time Adaptation via Self-Training with Nearest Neighbor Information. In *International Conference on Learning Representations*, 2023.

Pooria Joulani, András György, and Csaba Szepesvári. Online Learning under Delayed Feedback. In *International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1453–1461. PMLR, 2013.

Young Kyung Kim, Oded Schlesinger, Qiangqiang Wu, J. Matías Di Martino, and Guillermo Sapiro. Order-Aware Test-Time Adaptation: Leveraging Temporal Dynamics for Robust Streaming Inference, 2026. arXiv preprint arXiv:2601.21012.

Taeckyung Lee, Sorn Chottananurak, Junsu Kim, Jinwoo Shin, Taesik Gong, and Sung-Ju Lee. Test-Time Adaptation with Binary Feedback. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 33005–33024. PMLR, 2025.

Changyu Li, Lu Wang, Ming Lei, Jiashen Liu, Yichen Zhang, Kaishun Wu, and Fei Luo. PI-TTA: Physics-Informed Source-Free Test-Time Adaptation for Robust Human Activity Recognition on Mobile Devices, 2026. arXiv preprint arXiv:2604.25435.

Yushu Li, Yongyi Su, Xulei Yang, Kui Jia, and Xun Xu. Exploring Human-in-the-Loop Test-Time Adaptation by Synergizing Active Learning and Model Selection. *Transactions on Machine Learning Research*, 2024.

Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. TTN: A Domain-Shift Aware Batch Normalization in Test-Time Adaptation. In *International Conference on Learning Representations*, 2023.

Aleksej Logacjov, Kerstin Bach, Atle Kongsvold, Hilde Bremseth Bårdstu, and Paul Jarle Mork. HARTH: A Human Activity Recognition Dataset for Machine Learning. *Sensors*, 21(23):7853, 2021. doi: 10.3390/s21237853.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient Test-Time Model Adaptation without Forgetting. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards Stable Test-Time Adaptation in Dynamic Wild World. In *International Conference on Learning Representations*, 2023.

Tuan Pham, Daniel Kottke, Georg Krempl, and Bernhard Sick. Stream-based active learning for sliding windows under the influence of verification latency. *Machine Learning*, 111(6):2011–2036, 2022. doi: 10.1007/s10994-021-06099-z.

Attila Reiss and Didier Stricker. Introducing a New Benchmarked Dataset for Activity Monitoring. In *2012 16th International Symposium on Wearable Computers*, pp. 108–109, 2012. doi: 10.1109/iswc.2012.13.

Mona Schirmer, Dan Zhang, and Eric Nalisnick. Temporal Test-Time Adaptation with State-Space Models. *Transactions on Machine Learning Research*, 2025.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11539–11551, 2020.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully Test-time Adaptation by Entropy Minimization. In *International Conference on Learning Representations*, 2021.

Guowei Wang and Changxing Ding. Effortless Active Labeling for Long-Term Test-Time Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25633–25642, 2025.

Shuoyuan Wang, Jindong Wang, Huajun Xi, Bob Zhang, Lei Zhang, and Hongxin Wei. Optimization-Free Test-Time Adaptation for Cross-Person Activity Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(4):1–27, 2023. doi: 10.1145/3631450.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust Test-Time Adaptation in Dynamic Scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15922–15932, 2023.

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On Pitfalls of Test-Time Adaptation. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42058–42080. PMLR, 2023.

Zishu Zhou, Zaipeng Xie, and Xuanyao Jie. Temporal Structure Matters for Efficient Test-Time Adaptation in Wearable Human Activity Recognition, 2026. arXiv preprint arXiv:2605.04617.

APPENDIX

CONTENTS

A Data, Splits, and Feedback Service **14**
A.1 Participant roles and recognition tasks 14
A.2 Service specification 15

B Implementation and Offline Fitting **16**
B.1 Transition matrix and calibration 16
B.2 Complete gate features 16
B.3 Gate fitting objective and provenance 17
B.4 Sequential algorithm 17
B.5 Backbones and baseline adapters 18

C Metrics and Participant-Level Results **19**

D Complete Principal Comparisons **20**
D.1 Complete mechanism comparison 20
D.2 Probability quality of the feedback paths 23

E Participant-Level Comparisons **23**
E.1 OPPORTUNITY foreground results 23
E.2 Comparisons with TAST and OATTA 25

F Matched Comparisons and Analysis of Feedback Effects **26**
F.1 Simple rules and refitted smaller models 26
F.2 Fallback, prior division, and out-of-order history 33
F.3 Where delayed state changes cause errors 33
F.4 Missing, noisy, and activity-dependent responses 35

G Context and Source-Statistic Sensitivity **37**

H Complete Short-Delay Comparison **37**

I Recognition and Probability Quality Across Delays **40**

J Random Delay and Request Budget **42**

K All Fixed and Random Delays **42**

L Local Loss Analysis and Matched Release Events **44**

M External Comparisons at $d = 120$ **46**

N Replay Error Analysis and Online Cost **46**

O Model Size and Measured Computational Cost **47**

P Development Alternatives: Update Direction and Feedback Utility **49**
P.1 Learning an update direction 49
P.2 Transferring a local feedback utility criterion 50

Q Evidence and Reproduction **51**

A DATA, SPLITS, AND FEEDBACK SERVICE

How to read the supplementary results. The appendix groups repeated comparisons into coordinated figures: the principal comparison matrix (Figure 7), participant profiles (Figure 8), matched-comparison metrics and uncertainty (Figures 10 and 11), robustness (Figure 13), and the full delay sweep (Figure 17). Exact original tables are retained in the separate `numeric_supplement.pdf`; the CSV and workbook evidence remains available for numerical lookup.

The appendix follows the evidence needed to assess the main argument. Appendices A–C specify the data, algorithm, and participant-level inference. Appendix F gives matched comparisons and analyses of feedback effects; Appendix E gives participant-level comparisons. The remaining sections provide complete method scores, context and delay sensitivity, computational costs, and reproduction details. Table 4 summarizes participant counts, class vocabularies, window/hop lengths, request probabilities, and the clock used to interpret feedback delay for each dataset.

Table 4: Fixed evaluation tasks. Window/hop lengths are in samples unless marked in seconds. Request probability ρ is per decision. OPPORTUNITY retains NULL; PAMAP2 retains unknown windows in the stream but excludes their labels from scoring. UCI delays use the released event order.

Dataset	People	Classes	Window/hop	ρ	Clock
HARTH	8	12	50/25	.05	seconds
CAPTURE-24	12	4	10 s/10 s	.05	seconds
OPPORTUNITY	4	18	30/15	.0025	seconds
PAMAP2	9	12	256/128	.0064	seconds
UCI HAR	9	6	official	.05	events

A.1 PARTICIPANT ROLES AND RECOGNITION TASKS

Model selection and online batch dependence are known evaluation pitfalls in TTA (Zhao et al., 2023). The source model, normalizer, class list, target window identifiers, segment identifiers, and timestamps are fixed within each comparison. "Development" denotes a role in fitting or selecting a rule. `evidence/ROLES.json` lists every source, development, rule-fitting, and target identity for the 42 evaluated people. The machine-readable data contract binds the underlying files by SHA-256.

HARTH. The source participants are S006, S010, S013, S015, S017, S021, S023, S025, S026, and S029. S008, S018, S024, and S028 are development participants and also form the transition/compatibility fitting pool. The eight target participants are S009, S012, S014, S016, S019, S020, S022, and S027. The task retains 12 activity classes, six sensor channels, 50-sample windows, and a 25-sample hop. Native recording boundaries and gaps define resets.

CAPTURE-24. The fixed split contains 81 source participants, 20 checkpoint-development participants, and 12 target participants. The target identities are P105, P112, P116, P117, P121, P122, P126, P128, P132, P141, P146, and P148. The four rule-fitting participants are P029, P050, P073, and P088, all within the development group. We use the fixed four-class Walmsley2020 mapping from the data preparation pipeline, wrist acceleration, and 10-second nonoverlapping windows. The original timestamp axis is retained.

OPPORTUNITY. Each of S1–S4 serves once as the outer target. The fold has two source people and one separate checkpoint-development person. Transition and compatibility fitting use the two source people. We retain 77 channels and use 30-sample windows with a 15-sample hop. The task includes all 18 classes, including NULL, and retains both ADL and Drill recordings. This differs from the 17-class task described by Wang et al. (2023); neither class removal nor their published score is imported into our comparison.

PAMAP2. Protocol participants 101–109 each serve as the target in an outer fold, with seven source people and one checkpoint-development person. Transitions and the compatibility flag are fitted on source people. The task uses 27 channels, 256-sample windows, a 128-sample hop, and 12 scored classes. Unknown activity 0 is mapped to -1. Such windows remain in the stream and retain their time, but their labels are unanswerable and are excluded from recognition metrics. Neither state propagation nor elapsed time skips these windows.

UCI HAR. The official test people are 02, 04, 09, 10, 12, 13, 18, 20, and 24. Among official training people, 03, 06, 07, and 14 are reserved for checkpoint selection; the remaining 17 form the source and rule-fitting pool. The task retains six classes and the released sensor windows and their order. Because these windows do not define the same continuous recording clock as the other datasets, the feedback clock advances by one event per decision.

All target groups have historical research exposure. Participants are excluded from their own fold's source training and selection; the evaluation is retrospective. On the small datasets, participants can have different roles across outer folds, so folds share training data. Our uncertainty calculation resamples target people, keeping each person's windows and evaluation results together.

A.2 Service specification

The fixed delays are $\{0, 1, 2, 3, 5, 10, 20, 30, 40, 60, 80, 90, 120, 240, 300\}$. For each $d > 0$, the uniform service draws $d_o \sim \mathrm{Uniform}(0.5d, 1.5d)$ and the lognormal service draws

$$d_o = d \exp\Big(-\tfrac{1}{2}\log 2 + \sqrt{\log 2}\,\epsilon_o\Big), \qquad \epsilon_o \sim \mathcal{N}(0,1). \tag{8}$$

The latter has mean d and coefficient of variation one. Random draws are assigned per participant and segment. A request does not inspect the unreleased ground-truth label. At a segment boundary, pending requests are cancelled and state, calibration, and adaptation memories reset. Each request is accounted for as returned, unanswerable, cancelled, or still pending at the end. All methods within a paired feedback condition use the same ledger, including unanswerable requests.

Figure 4 illustrates the distinct treatment of older returns by calibration and state correction.

Numbered circles in Figure 4 denote requests for already-predicted windows; matching diamonds release origin–label pairs. Both returns calibrate their saved predictions, but a late older origin cannot trigger new state injection. Release occurs at the first same-segment decision t_o with $\tau_{t_o} > \tau_o + d_o$, so the observed wait can exceed nominal delay d_o. Even $d_o = 0$ cannot revise the prediction at o. The main sweep has 43 services; matched comparisons use fixed and lognormal delays with means 0, 30, 120 and 300, coinciding at zero (seven distinct services). Units are seconds except UCI HAR, which uses released-window events.

Nominal delay and age at the next decision can differ because release requires a strict inequality. For example, the mean observed age at $d = 120$ is about 130 seconds on CAPTURE-24 and 121 events on UCI. Table 5 reports actual service exposure. Request probabilities are dataset-specific; hourly annotation rates also depend on window cadence.

Figure 2 in the main text compares the method families. RoTTA's sample timeliness does not model annotation-return delay. Delayed supervised replay (RR, NR, IWMS) learns from returned historical examples; IWMS retrieves examples by relevance to the current input. These replay rules do not separately gate a returned class's direct injection into activity state and its later output influence. Gated Arrival separates origin-matched calibration, newest-origin state correction and output mixing. Older-origin returns still calibrate history but cannot overwrite a newer state observation. The families summarize dominant mechanisms, with the comparison focused on supervision-return delay.

Table 5: Service measurements at $d = 120$, averaged across people. Requests/returns are counts per person, age uses the dataset's clock, and active is the fraction after a first legal return. Injection averages positive injection strengths at selected returns; output averages weights over all decisions, including zero before activation. Both are then averaged equally across people. These are pre-fallback traces; OPPORTUNITY does not apply them to its output.

Dataset	Requests	Returns	Age	Active	Injection	Output
CAPTURE-24	465.17	300.92	129.97	1.00	0.92	0.62
HARTH	598.38	394.00	120.49	0.67	0.28	0.12
OPPORTUNITY	34.00	29.75	120.50	0.71	0.01	0.67
PAMAP2	16.67	11.22	120.32	0.78	0.19	0.34
UCI HAR	15.56	9.56	121.00	0.56	0.06	0.24

B Implementation and Offline Fitting

B.1 Transition matrix and calibration

Transition counts use adjacent, validly labeled windows in the same native segment. Unknown windows are never bridged to create a transition. Each transition cell starts with pseudocount $1/K$, and each class count for the prior starts with one. After row normalization, off-diagonal mass is capped at 0.5, preserving relative off-diagonal probabilities, and the diagonal is set to the remaining mass. The prior is the normalized class count. HARTH and CAPTURE-24 use their rule-development pools; the three smaller datasets use the source people of the corresponding fold. These statistics remain fixed during target replay.

Historical calibration uses the analytic gradient of its regularized cross-entropy objective, step size 0.05, and $\lambda = 0.01$. The gradient is clipped to norm 1. After the step, a_k is clipped to $[-\log 4, \log 4]$ and b_k to $[-2, 2]$. Scale and bias initialize to zero. Every returned historical label is processed once, even if its origin is too old for state injection. The current prediction uses the updated calibration parameters, while the already committed origin prediction is left unchanged.

Both the history path and the state candidate retain a one-quarter source anchor. Consequently, for any active mixture and class k, $\hat{p}_t(k) \geq p_t(k)/4$. In particular, $-\log \hat{p}_t(y_t) \leq -\log p_t(y_t) + \log 4$. This pointwise floor bounds the NLL increase relative to Base by $\log 4$. Expected loss and F1 depend on the resulting predictions and are evaluated empirically. Fallback and pre-activation predictions equal Base exactly.

B.2 Complete gate features

Let o be the latest selected origin, $\ell_t = t - o$ its age in decision steps, $\nu_t = t - t_{\text{release}}$ the decision steps since its selection, and n_t the number of all returned labels in the segment. Define

$$L(v) = \log(1 + v)/8, \qquad H(p) = -\sum_k p_k \log p_k / \log K, \tag{9}$$

$$V(p, q) = \tfrac{1}{2}\|p - q\|_1, \qquad E_t = 0.9E_{t-1} + 0.1V(p_t, p_{t-1}), \tag{10}$$

with $E_t = 0$ at a segment start. For the newly selected label $y = y_o$, the injection feature vector, in implementation order, is

$$\begin{aligned}\phi_I = \big[&L(\ell_t),\ \min\{-\ell_t \log \max(A_{yy}, 10^{-12}), 20\}/20,\\ &p_t(y),\ p_o(y),\ \pi_y,\ A_{yy},\ H(p_t),\ \max_k p_t(k),\\ &V(p_t, p_o),\ E_t,\ L(n_t),\ p_t^\top p_o\big].\end{aligned} \tag{11}$$

The output feature vector is

$$\phi_O = \big[L(\ell_t),\ L(\nu_t),\ L(n_t),\ H(p_t),\ H(c_t),\ \max_k p_t(k),\ \max_k c_t(k),\ V(c_t, p_t),\ c_t^\top p_t,\ V(p_o, p_t),\ E_t,\ H(\pi),\ \sum_k p_o(k) A_{kk},\ p_t^\top \pi\big]. \tag{12}$$

Neither vector reads the current target label. Gate age is measured in decision steps even when the feedback service uses physical timestamps. We report this distinction explicitly because sampling rates differ.

B.3 Gate fitting objective and provenance

For each evaluated dataset, gate fitting excludes all streams from that dataset. The original fixed development pool comprises 25 streams across five datasets; each stream is replayed under fixed delays 0, 30, and 120, and lognormal mean delay 120. A held-dataset fit uses only the other four datasets. Ground truth is available for these offline development streams.

For injection fitting, each legally selected return provides a training row. Let $h = p_t$ and $q = \frac{1}{4}p_t + \frac{3}{4}e_{y_o}$. For output fitting, h is the history prediction and q the soft-injection state candidate; up to 2,048 valid, active rows per service are retained at evenly spaced indices. With $g = \mathrm{sigmoid}(w^\top \phi + c)$ and $m = (1-g)h + gq$, the objective is

$$\sum_i \omega_i \left[-\log m_i(y_i) + \|m_i - e_{y_i}\|_2^2\right] + 10^{-3}\|w\|_2^2. \tag{13}$$

Weights give equal total mass to each training dataset, then equal mass to its eligible stream/service cells, and then to rows within a cell. Weighted feature standard deviations are floored at 0.05. Both fits start at zero and use 300 Adam steps at learning rate 0.03 in float64. Affine feature standardization is folded into the final deployed weights. The injection fit precedes output fitting. The fitted parameters are frozen before the main panel and reused across all evaluation runs and delay services. `evidence/CONTROLLERS.json` contains all coefficients, feature statistics, and training-stream identifiers.

The source compatibility check is separately recomputed for each source checkpoint on its prescribed fitting pool. It compares the always-active source filter, without injected labels, to Base using mean participant NLL and Brier differences. All 12 OPPORTUNITY fold/checkpoint combinations trigger fallback; none of the other datasets' combinations do.

B.4 Sequential algorithm

Algorithm 1: Gated Arrival
Inputs: frozen f_θ, A, π, gate coefficients, compatibility flag R, and request probability ρ.
State: calibration (a, b), activity q, selected origin, return count, probability-variation EMA, and pending origin probabilities.

1. At a segment start, clear pending requests and initialize all state.
2. At decision t, compute the source probability p_t. Release only requests with due time strictly less than τ_t.
3. For each returned (o, y_o), update calibration with (p_o, y_o). Select the largest newly returned origin only if it exceeds the previously selected origin; count every return.
4. Propagate q by Equation 4. If an origin is newly selected, compute its injection features and reset q by Equation 5.
5. Before the first return, or if $R = 1$, output p_t. Otherwise form h_t, c_t and output their gated mixture in Equation 6. Commit this prediction.
6. Draw the label request for t; if requested, save p_t and enqueue the request with its origin and due time. Advance to the next window.

The frozen source probability depends on the current input alone and can be cached before processing releases. Adaptation and prediction retain the release–prediction–request information order.

B.5 Backbones and baseline adapters

HARTH uses three convolution blocks of widths 64, 128, 128, kernels 5, 5, 3, and strides 1, 2, 2, each with batch normalization and ReLU, followed by global average pooling and a linear classifier. The same structure, with dataset-specific input and output dimensions, is used for OPPORTUNITY, PAMAP2, and UCI HAR. The CAPTURE-24 model is the pinned Oxford small residual architecture: stage widths 32, 32, 64, 64, 128, 128 and block counts 0, 3, 3, 3, 3, 3. All methods within a dataset use the same source checkpoint and equivalent encoder/classifier interfaces.

HARTH source training uses Adam at learning rate 10^{-3}, batches of 256, at most 40 epochs, and development-person-mean fixed-class Macro-F1 for checkpoint selection (minimum 10 epochs, patience six). CAPTURE-24 uses AdamW at 3×10^{-4} with AMSGrad, weight decay 0.01, and batches of 128. Its training budget is 40 sampling rounds, each sampling 512 windows per source person, with validation every two rounds; selected checkpoints correspond to rounds 40, 32, and 32. The three smaller datasets use Adam at 10^{-3}, batches of 128, and 20 training epochs. Source checkpoints and their training records are fixed before adaptation.

Test-time batch statistics are sensitive to small, non-i.i.d. batches (Lim et al., 2023). BN uses past/current context statistics; T3A maintains classifier supports; OFTTA combines normalization and supports; SIGHT uses its temporal adapter. Tent and SAR update normalization affine parameters through their respective objectives, and RoTTA uses its teacher/memory adaptation. BN, Tent, and SAR concatenate the current sensor window with up to 15 preceding windows from the same native segment. They return only the current window's prediction, use the available shorter buffer at segment starts, and reset the buffer at segment boundaries. Thus one new window arrives per decision, while these three methods use a normalization batch of up to 16 windows. The value 16 was fixed in the adaptation protocol; context length was held fixed during hyperparameter selection. The subsequent sensitivity study varies it to 1, 4, and 64 while retaining the selected learning rate and other settings. Other methods retain their own support or memory mechanisms rather than this 16-window normalization buffer. Stochastic dropout is disabled during adaptation. SAR uses the shared BN backbone, whereas its original study emphasizes GN/LN models under small batches and class imbalance. OFTTA uses a single current window, layer-dependent source/target statistics, and its support mechanism; its source-weight sensitivity does not change the normalization context. The full selected method/fold settings are in `evidence/BASELINE_CONFIGS.csv`.

RR samples 32 returned windows with replacement from a memory of size 128. NR combines 16 samples from the most recent 16 memory entries with 16 uniformly sampled memory entries. IWMS combines newest/random samples with the authors' softmax similarity-weighted sampling routine (Botos et al., 2024). The sensor adaptation retains raw floating windows, uses returned labels only, and stores historical retrieval features at insertion time. It uses an explicit valid-memory fallback for invalid empty-class indices. Supervised adapters update the full network every eight decisions once memory is available, using SGD with momentum 0.9 and weight decay 10^{-5}; the selected learning rates and steps are in the configuration table. Copy uses $0.9e_y + 0.1\mathbf{1}/K$ for the most recently received label, with Base before the first return. Anchored Copy additionally retains a one-quarter Base mixture. All replay adapters share the same sensor stream and returned-label service.

TAST and OATTA. TAST uses the OFTTA authors' HAR implementation of the TAST algorithm, with ten projection ensemble members, projection width $D/4$ for feature width D, temperature 10, one neighbor, and one Adam step per decision. The backbone is frozen. Development selection chooses among learning rates $\{10^{-2}, 10^{-3}\}$ and per-class support limits $\{25, 5\}$. The output is the mean normalized ensemble probability before the current optimizer step. Supports, projections, and optimizer state reset at each segment; selection uses the designated development people.

OATTA is implemented from its published Algorithm 2. The transition-count matrix starts at the identity; initial probability states and priors are uniform. The current posterior uses the previous transition model, followed by the baseline/likelihood-ratio update and output mixture. Adjacent raw source probabilities then update transition counts. The main variant recursively retains the unmixed filtered posterior. We select on development people over $\gamma \in \{.01, .1\}$, likelihood-ratio window

$W \in \{32, 128\}$, $\tau \in \{1, .1\}$, and margin $m \in \{0, .05\}$, with $\eta = .01$. Appendix tables also include four internal variants: separately selected no-LLR, no-LLR with the main configuration, recursive mixed-output memory, and uniform initial transition counts. These variants make the state and initialization choices explicit. STAD is discussed as related work; it has no measured row in this comparison.

Definitions of component removals. Newest-origin Arrival prevents overwrites by older origins but uses hard injection. Injection-only uses the injection gate and omits output mixing. Output-only mixes history with hard newest-origin Arrival. No injection sets injection strength to zero at the same release events and retains historical calibration and output gating. No history replaces history by Base in the output mixture. No fallback uses both gates without the source compatibility check. All gates remain fixed in these deletions. The source filter is always active and unlabeled; Timing only applies the same unlabeled filter only after the first legal return. Prior-corrected Arrival replaces $p_t^{1/2}$ by $(p_t/\pi)^{1/2}$ in the raw filter. A half mixture combines history and hard newest-origin Arrival with equal weights.

C Metrics and Participant-Level Results

F1 denotes Macro-F1 unless weighted or classwise F1 is explicitly named. For each person and condition, classwise F1 is $2TP_k/(2TP_k + FP_k + FN_k)$. A class with zero denominator contributes zero. Macro-F1 averages the fixed K classes, including absent classes. Weighted F1 weights classes by their true-label support. Balanced accuracy averages recall over classes present in that person's true labels. Accuracy, NLL, and Brier average valid windows; Brier sums squared class errors per window. ECE uses 15 equal-width confidence bins. A true-class probability of zero yields infinite NLL. Percentage metrics are multiplied by 100 at reporting. Dataset means give every participant equal weight.

For a participant's N valid predictions, the probability scores are

$$\mathrm{NLL} = -\frac{1}{N}\sum_{t=1}^{N} \log \hat{p}_t(y_t), \tag{14}$$

$$\mathrm{Brier} = \frac{1}{N}\sum_{t=1}^{N}\sum_{k=1}^{K} (\hat{p}_t(k) - \mathbb{1}\{y_t = k\})^2, \tag{15}$$

$$\mathrm{ECE} = 100 \sum_{b=1}^{15} \frac{|B_b|}{N} \left|\mathrm{acc}(B_b) - \mathrm{conf}(B_b)\right|. \tag{16}$$

Here B_b bins the maximum predicted probability, $\mathrm{acc}(B_b)$ is top-label accuracy, and $\mathrm{conf}(B_b)$ is mean maximum probability. Empty bins contribute zero. ECE is computed separately for each participant before averaging, rather than by pooling their windows. NLL and Brier are proper scores (Gneiting & Raftery, 2007); ECE summarizes confidence calibration (Guo et al., 2017). These scores capture different properties, and none is inferred from F1.

Paired 95% bootstrap intervals resample participants 10,000 times. They describe participant variation conditional on the fixed checkpoints and request schedules. Delay services and evaluation runs are repeated conditions, not additional participants. Table 6 reports the principal-panel intervals; Figure 8 shows differences between participants after within-person aggregation.

The complete principal-panel participant scores are retained in the numerical companion. Figure 8 shows the same 42 people after within-person aggregation.

Table 6: Gated Arrival minus Base at $d = 120$. W/T/L counts participants with positive/zero/negative gain. Intervals are in Macro-F1 percentage points.

Dataset	Δ F1	95% interval	W/T/L
CAPTURE-24	+8.21	[6.49, 10.24]	12/0/0
HARTH	+0.36	[-0.09, 0.90]	5/0/3
OPPORTUNITY	+0.00	[0.00, 0.00]	0/4/0
PAMAP2	+1.75	[1.11, 2.32]	8/1/0
UCI HAR	+1.42	[0.45, 2.60]	7/2/0

D Complete Principal Comparisons

The principal panel contains 33 named methods and internal configurations; ten additional context configurations make 43 configurations in the numerical evidence. Figure 7 aligns all 33 rows at two delays, separating principal comparisons from internal variants. Every cell is an F1 difference from the same-dataset Base. Directional backgrounds encode the direction of the mean difference; printed values give its magnitude. All use the principal-panel evaluation settings. Strict release ordering also applies at $d = 0$.

The numerical companion retains the original absolute-score tables for $d = 0$ and $d = 120$, each dataset's F1/NLL/Brier/ECE table, and the accuracy, weighted-F1, and balanced-accuracy panels. Each metric is reported separately.

D.1 Complete mechanism comparison

Table 7 complements the main-text mechanism analysis. The internal-configuration panel above additionally retains the Injection-only, Output-only, prior-corrected, and newest-origin alternatives.

Table 7: Component comparisons at $d = 120$ seconds (UCI: events) (Macro-F1, %). No injection, no history, and no fallback are evaluated without the source compatibility check. Timing only activates the filter at the first return.

Method	CAPTURE	HARTH	OPPORT.	PAMAP2	UCI
Base	68.47	54.03	27.36	64.02	90.30
Source filter	71.49	51.73	11.96	65.54	92.32
Timing only	71.47	54.04	18.58	65.54	91.81
History	68.73	54.27	27.37	63.99	90.86
Direct	74.92	54.04	18.58	65.54	91.81
Raw Arrival	77.15	53.42	18.59	65.50	85.68
Half mixture	76.28	54.72	20.94	65.74	91.59
No injection	72.42	54.37	19.97	65.75	91.70
No history	76.64	54.19	19.98	65.80	91.56
No fallback	76.68	54.38	19.97	65.77	91.72
Gated Arrival	76.68	54.38	27.36	65.77	91.72

What does a learned gate add over a fixed mixture? Table 8 shows that the half mixture improves all four metrics over the complete method on HARTH. On PAMAP2 it has lower probability losses but slightly lower F1. The benefit of learned gating varies across datasets and evaluation metrics.

Macro-F1 gain over Base (pp)
Blue: gain Gray: tie Orange: loss

	(a) Delay 0					(b) Delay 120				
	CAP.	HARTH	OPPO.	PAMAP2	UCI	CAP.	HARTH	OPPO.	PAMAP2	UCI
Base	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0
BN	-43.3	-45.7	-17.8	-55.5	-45.0	-43.3	-45.7	-17.8	-55.5	-45.0
Tent	-44.2	-43.7	-20.3	-56.2	-45.5	-44.2	-43.7	-20.3	-56.2	-45.5
SAR	-43.3	-44.9	-18.5	-56.0	-45.1	-43.3	-44.9	-18.5	-56.0	-45.1
T3A	-0.7	-0.9	-0.3	-0.2	-0.9	-0.7	-0.9	-0.3	-0.2	-0.9
TAST	-0.9	-2.1	+0.5	+0.7	-0.8	-0.9	-2.1	+0.5	+0.7	-0.8
OFTTA	-49.1	-36.3	-22.7	-41.6	-19.3	-49.1	-36.3	-22.7	-41.6	-19.3
RoTTA	-20.5	-16.1	-10.9	-0.7	+2.0	-20.5	-16.1	-10.9	-0.7	+2.0
OATTA	+5.0	-0.2	-9.8	+0.9	+0.2	+5.0	-0.2	-9.8	+0.9	+0.2
SIGHT	+4.2	-3.3	-16.2	+1.5	-1.5	+4.2	-3.3	-16.2	+1.5	-1.5
RR	-7.6	-7.1	-5.4	-29.2	+1.0	-8.4	-3.9	-5.5	-39.4	+0.3
NR	-4.2	-7.0	-5.4	-32.2	+0.8	-6.7	-4.2	-5.4	-39.6	-4.2
IWMS	-2.8	-7.7	-6.0	-32.0	+0.1	-3.9	-3.9	-6.0	-39.8	-2.5
Copy	+13.1	-4.3	-15.0	-29.0	-48.1	+5.0	-8.3	-13.6	-47.3	-38.2
History	+0.6	+0.6	+0.0	+0.1	+0.8	+0.3	+0.2	+0.0	-0.0	+0.6
Direct	+13.5	+2.0	-9.7	+2.0	+3.2	+6.5	+0.0	-8.8	+1.5	+1.5
Raw Arrival	+13.9	+2.3	-9.6	+2.1	+3.0	+8.7	-0.6	-8.8	+1.5	-4.6
Gated Arrival	+12.9	+2.5	+0.0	+1.9	+3.2	+8.2	+0.4	+0.0	+1.8	+1.4
Source filter	+3.0	-2.3	-15.4	+1.5	+2.0	+3.0	-2.3	-15.4	+1.5	+2.0
Anchored Copy	+13.1	-4.3	-15.0	-29.0	-48.1	+5.0	-8.3	-13.6	-47.3	-38.2
Half mixture	+10.8	+3.1	-7.0	+1.9	+3.8	+7.8	+0.7	-6.4	+1.7	+1.3
Timing only	+3.0	-2.2	-10.0	+1.6	+2.0	+3.0	+0.0	-8.8	+1.5	+1.5
Newest-origin Arrival	+13.9	+2.3	-9.6	+2.1	+3.0	+8.7	-0.6	-8.8	+1.5	-4.6
Injection-only	+13.2	+2.0	-9.9	+1.8	+2.9	+8.3	-0.0	-8.6	+1.7	+1.0
Prior-corrected Arrival	-1.4	-2.9	-1.2	+0.7	+3.1	-6.7	-2.9	-1.5	+0.4	-4.6
Output-only	+13.7	+3.0	-8.3	+2.2	+3.3	+8.7	+0.4	-7.5	+1.6	-3.8
No injection	+4.0	+0.5	-8.6	+1.3	+1.5	+4.0	+0.3	-7.4	+1.7	+1.4
No history	+12.7	+2.3	-8.5	+1.9	+3.1	+8.2	+0.2	-7.4	+1.8	+1.3
No fallback	+12.9	+2.5	-8.6	+1.9	+3.2	+8.2	+0.4	-7.4	+1.8	+1.4
OATTA: mixed memory	+4.4	+0.3	-7.9	+0.9	+0.4	+4.4	+0.3	-7.9	+0.9	+0.4
OATTA: no LLR	+5.0	-0.6	-11.4	+1.7	-0.2	+5.0	-0.6	-11.4	+1.7	-0.2
OATTA: no LLR, matched	+5.0	-0.6	-11.4	+0.4	-0.2	+5.0	-0.6	-11.4	+0.4	-0.2
OATTA: uniform counts	+0.2	-0.4	-9.8	+0.6	-0.6	+0.2	-0.4	-9.8	+0.6	-0.6

Figure 7: **One aligned view of the principal comparison.** Macro-F1 gains over Base at delays 0 and 120. All 33 method and internal configuration rows are retained. Blue/gray/orange encodes positive/zero/negative gain, with values in percentage points; the outlined row is Gated Arrival. The lower block contains internal variants rather than additional independent baselines. Units are seconds except UCI events.

Table 8: Fixed half mixture and learned gates at $d = 120$. The half mixture omits fallback; neither dataset triggers fallback in the complete method. Each score is an equal-participant mean.

Method	F1 (%) ↑	ECE (%) ↓	Brier ↓	NLL ↓
HARTH				
Base	54.03	4.25	0.146	0.294
Half mixture	54.72	2.96	0.122	0.259
Gated Arrival	54.38	3.36	0.127	0.269
PAMAP2				
Base	64.02	9.89	0.324	1.017
Half mixture	65.74	9.30	0.299	0.963
Gated Arrival	65.77	9.96	0.300	0.971

Table 9: **History's contribution is visible beyond F1.** Component comparisons at $d = 120$. F1 and ECE are percentages. CAPTURE isolates the feedback paths; OPPORTUNITY isolates fallback. Timing only activates an unlabeled filter on the first return. Component removals omit fallback; only the last row applies the compatibility check.

	CAPTURE-24				OPPORTUNITY
Variant	F1 ↑	NLL ↓	Brier ↓	ECE ↓	F1 ↑
Base	68.47	0.579	0.314	8.35	27.36
Source filter	71.49	0.457	0.239	7.06	11.96
Timing only	71.47	0.458	0.240	7.03	18.58
Half mixture	76.28	0.378	0.196	7.03	20.94
No injection	72.42	0.439	0.234	4.87	19.97
No history	76.64	0.392	0.197	6.25	19.98
No fallback	76.68	0.370	0.190	4.77	19.97
Gated Arrival	76.68	0.370	0.190	4.77	27.36

D.2 Probability quality of the feedback paths

Historical calibration has little effect on CAPTURE-24 F1 in this deletion comparison, but improves NLL and ECE. The full method's OPPORTUNITY score depends on the offline compatibility check, which selects the source model.

Metric-specific absolute comparisons and all secondary recognition scores are grouped in the numerical companion, using the same frozen tables.

E Participant-Level Comparisons

This panel aggregates evaluation results within each participant, then weights participants equally. Unlabeled methods reuse their predictions across feedback services. The numerical companion reports 27 reference configurations and the TAST/OATTA extension; Figure 9 summarizes its paired effects. The data protocols and people match the principal panel. Context-size sensitivity is reported separately in Appendix G.

The numerical companion retains the complete 27-configuration participant-level comparison. Table 10 reports paired gains over Base with participant bootstrap intervals and corrected sign-flip tests; Figure 9 extends the paired comparisons to external methods.

Table 10: Gated Arrival minus Base at $d = 120$ seconds (UCI: events). Results are aggregated within person; F1 differences are percentage points. Intervals use 10,000 participant bootstrap draws. Exact one-sided sign-flip tests use Holm correction across five datasets.

Dataset	Δ F1	95% interval	W/T/L	Holm p
CAPTURE-24	+8.73	[7.58, 10.09]	12/0/0	0.0012
HARTH	+0.21	[-0.05, 0.53]	6/0/2	0.2188
OPPORTUNITY	+0.00	[0.00, 0.00]	0/4/0	1.0000
PAMAP2	+1.91	[1.02, 3.14]	8/1/0	0.0156
UCI HAR	+0.74	[0.36, 1.24]	8/1/0	0.0156

E.1 OPPORTUNITY foreground results

Table 11: OPPORTUNITY at $d = 120$ in the participant-level panel. FG averages the 17 non-NULL class F1s using full-stream confusion counts; BG FP is the fraction of true-NULL windows assigned a foreground class.

Method	F1	FG F1	BG FP (%)	NLL	Brier
Base	25.18	21.56	4.58	1.018	0.388
Source filter	10.89	6.55	1.52	1.233	0.452
Raw Arrival	16.45	12.40	2.19	1.174	0.432
No fallback	17.80	13.81	2.37	1.144	0.419
Gated Arrival	25.18	21.56	4.58	1.018	0.388

The source filter degrades both foreground recognition and proper scores (Table 11). Without fallback, the full model recovers part of that loss while reducing background false positives. This pattern is consistent with excessive persistence of background predictions. All 12 outer-fold/checkpoint fits trigger the offline check, so the final method retains Base. The 18-class full-stream task remains fixed throughout the comparison.

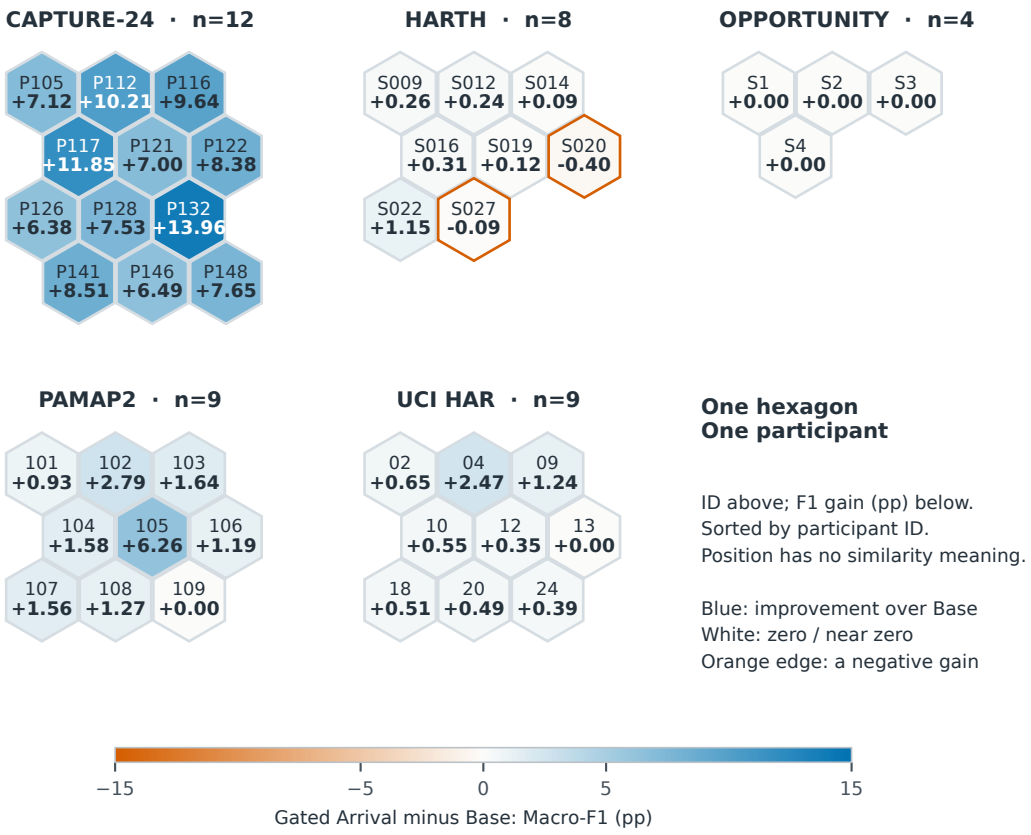


Figure 8: **Every participant remains visible.** Each hexagon is one person and prints the participant ID and Gated Arrival minus Base Macro-F1 (pp) at $d = 120$, aggregating results within person. All panels use the same color scale. Orange outlines mark negative gains; white cells include exact ties and very small gains. Positions follow participant IDs and imply no spatial similarity or density. The paired intervals and sign tests remain in Table 10.

E.2 Comparisons with TAST and OATTA

TAST and OATTA use their development-selected configurations on all 42 participants under shared source checkpoints. We average the scores from the corresponding checkpoints and feedback services within each participant. The offline gates are reused unchanged, without target-based selection. These comparisons retain the supervision distinction of the main text: TAST and OATTA use no returned target labels or cross-dataset gate-training labels. The numerical supplement reports scores by source checkpoint and request schedule. For Gated Arrival at $d = 120$, CAPTURE-24 F1 ranges from 75.48 to 76.99 across checkpoints and from 75.98 to 76.69 across request schedules. OPPORTUNITY ranges from 21.67 to 27.36 across checkpoints and is invariant to the request schedule because of fallback. All comparisons use the same participant panel.

At $d = 120$, Gated Arrival exceeds TAST by 8.70 F1 points on CAPTURE-24, with a paired interval of $[6.64, 11.01]$, and exceeds OATTA by 4.96 points $[3.60, 6.36]$. Its F1 differences from OATTA on HARTH and PAMAP2 are 0.40 and 0.56 points, with intervals spanning zero. On UCI, the OATTA comparison favors Gated Arrival in F1 by 1.31 points $[0.68, 2.11]$, but the NLL interval includes zero. OPPORTUNITY's full-minus-TAST F1 difference is -1.85 points $[-7.72, 1.95]$; its lower probability losses reflect preservation of Base through fallback. At $d = 300$, CAPTURE-24's gain over OATTA remains 2.08 points $[0.94, 3.25]$.

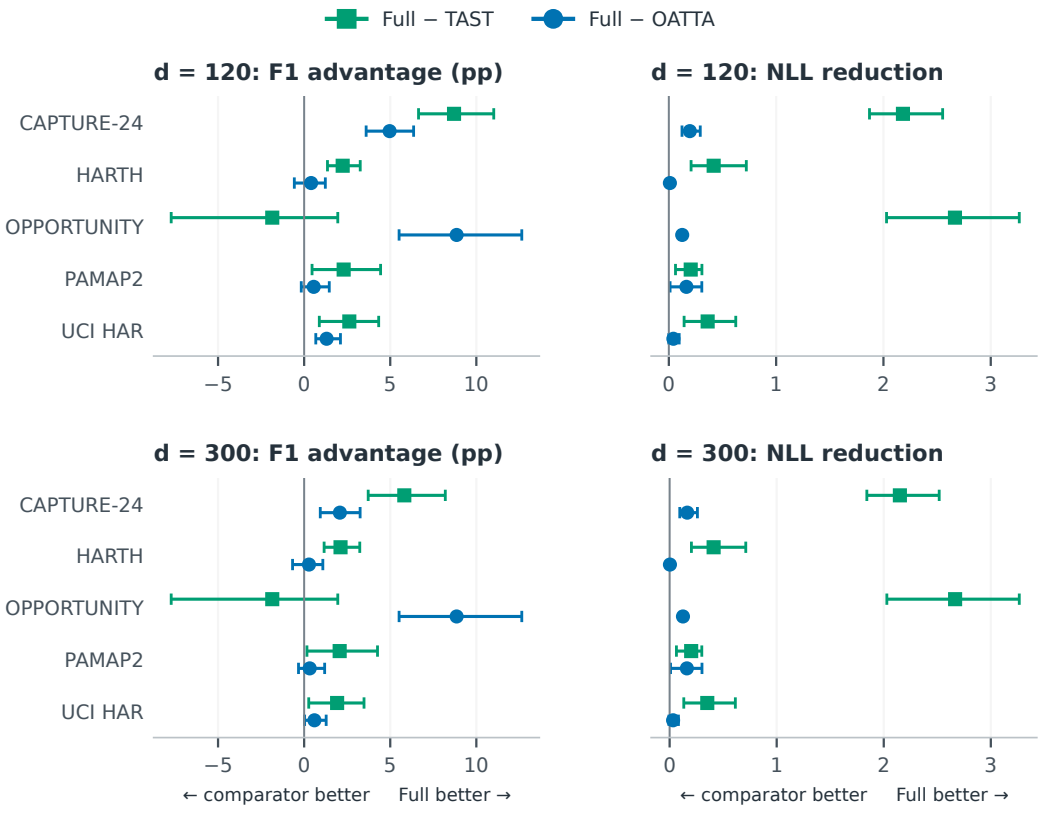


Figure 9: **External comparisons across response delays.** 95% participant-paired intervals for Gated Arrival versus TAST and OATTA at $d = 120$ and 300. F1 advantages and NLL reductions both point right; axes have metric-specific units. Results are aggregated within participant before forming paired comparisons. Unlabeled methods reuse their predictions across delays. Absolute values for all four metrics and exact paired tables remain in the numerical companion.

F MATCHED COMPARISONS AND ANALYSIS OF FEEDBACK EFFECTS

These experiments distinguish effects of returned label values, temporal filtering, and the contribution of learned gates. We evaluate all 42 participants at fixed and lognormal delays with means 0, 30, 120, and 300; both laws coincide at zero delay. Source checkpoints, windows, label requests, and segment boundaries match the main protocol. This extends the same previously exposed participant panel. Results are aggregated within participant before dataset means and paired intervals are computed. We use 10,000 bootstrap draws, resampling participants within each dataset and preserving each person's complete method/service vector.

F.1 SIMPLE RULES AND REFITTED SMALLER MODELS

Newly fitted rules use the original 25-stream development pool, excluding the entire evaluated dataset. They share the NLL-plus-Brier objective, dataset/cell/row weighting, active-row sampling, and fitting settings of Appendix B. The constant rule jointly selects $r, u \in \{0, 0.25, 0.5, 0.75, 1\}$. It chooses $(1, 0.25)$ when CAPTURE-24 is excluded and $(1, 0.5)$ for the other four excluded datasets. The half mixture remains a separate, fixed reference.

The age/persistence injection gate uses log age and age multiplied by source exit tendency, the first two injection features. Its output gate uses log origin age, log time since return, and origin-weighted source persistence, corresponding to output features 1, 2, and 13. Output-only gating fixes $r = 1$ and refits the output gate. Refitted no injection fixes $r = 0$; refitted no history replaces h_t by p_t, retaining the original injection gate and refitting the output gate. These refits differ from deleting components while keeping original coefficients fixed.

The matched gated and mixed rules receive the same compatibility flag as the full method. Raw Arrival and Direct also receive this flag. History and the always-active source filter are displayed as separate branches. Figure 10 puts all four metrics in the same layout.

Figure 6 in the main text shows these interventions. Shapes refer to one decision: $[K]$ denotes a K-component vector and $[1]$ a scalar. History pairs returned labels with cached p_o and evaluates its map at current p_t; State emits c_t, with internal state q_t. Reset equations apply at newly selected newest-origin returns; loops denote temporal filtering otherwise. Dashed edges carry auxiliary inputs, including causal gate features $\phi_I \in \mathbb{R}^{12}$ and $\phi_O \in \mathbb{R}^{14}$. Original/refit refers to coefficients, not constant online gate values. Each z_t stream passes separately through the shared compatibility flag R before scoring. Before the first legal return, the four rules in the diagram output p_t.

Comparison across delay distributions. Tables 12–13 expand the aggregate comparison in Table 2 by dataset. Every entry aggregates results within each participant, then weights participants equally. At each $d \in \{0, 30, 120, 300\}$, Fixed uses constant delay d and LN uses lognormal delay with mean d; the zero-delay columns coincide. Delay units are seconds except UCI events. The ten displayed rules match the main table. The LN30 and LN300 evaluations reuse the original frozen models and cached source predictions, with the same request schedules and lognormal sampling law as LN120. The full model's F1 advantage over the no-injection refit on CAPTURE-24 declines from 10.63 points at zero delay to 2.59 at 300, with paired participant intervals $[8.93, 12.42]$ and $[1.61, 3.63]$. Under lognormal delays, the corresponding injection gains are 9.59, 6.95 and 4.49 points at means 30, 120 and 300. On UCI, the no-injection refit has higher mean F1 at fixed 30 and 120 and all three positive lognormal means. Thus state injection contributes differently across datasets and feedback ages.

The full model has higher mean F1 than the Output-only refit at all six positive-delay services. Five paired intervals are above zero; the LN30 difference is 0.16 $[-0.03, 0.34]$. Output-only leads at zero delay. Gains over the half mixture range from 0.19 to 0.39 points, with the fixed-120 interval crossing zero and the other five positive-delay intervals above zero. Full-model F1 is 0.24, 0.37 and 0.61 points higher under lognormal than fixed delay at means 30, 120 and 300. Removing history has aggregate F1 effects below 0.03 points at every service, while increasing NLL on CAPTURE-24, HARTH, PAMAP2 and UCI under both delay laws. The mean tables and participant-level four-metric records are provided in `evidence/followup/MATCHED_DELAY_*.csv`.

Table 12: **Matched comparisons across delays: CAPTURE-24, HARTH and OPPORTUNITY.** Macro-F1 (%), within-person aggregate scores averaged over participants. All feedback rules share fallback; Source filter is always active. Blue identifies the full method. LN: lognormal delay with mean d; the two columns at $d = 0$ coincide.

(a) CAPTURE-24

Method	$d = 0$		$d = 30$		$d = 120$		$d = 300$	
	Fixed	LN	Fixed	LN	Fixed	LN	Fixed	LN
Base	67.70	67.70	67.70	67.70	67.70	67.70	67.70	67.70
Source filter	69.68	69.68	69.68	69.68	69.68	69.68	69.68	69.68
Raw Arrival + fallback	82.82	82.82	81.29	81.65	77.27	78.55	73.10	75.28
Half mixture + fallback	79.81	79.81	78.77	79.03	75.72	76.78	72.62	74.58
Fitted constant	73.06	73.06	72.58	72.72	71.48	71.79	70.20	70.84
Age/persistence	79.29	79.29	77.06	77.77	73.43	74.99	71.86	73.05
Output-only (refit)	82.54	82.54	80.99	81.40	76.94	78.47	72.81	75.54
No injection (refit)	70.99	70.99	70.95	70.92	70.97	70.96	70.95	70.97
No history (refit)	81.93	81.93	80.33	80.80	76.64	78.17	73.60	75.66
Gated Arrival	81.61	81.61	80.05	80.51	76.43	77.91	73.54	75.45

(b) HARTH

Method	$d = 0$		$d = 30$		$d = 120$		$d = 300$	
	Fixed	LN	Fixed	LN	Fixed	LN	Fixed	LN
Base	54.47	54.47	54.47	54.47	54.47	54.47	54.47	54.47
Source filter	53.26	53.26	53.26	53.26	53.26	53.26	53.26	53.26
Raw Arrival + fallback	56.26	56.26	51.81	52.06	53.43	52.50	54.45	53.56
Half mixture + fallback	56.65	56.65	54.43	54.66	54.66	54.37	54.63	54.60
Fitted constant	56.65	56.65	54.43	54.66	54.66	54.37	54.63	54.60
Age/persistence	56.52	56.52	54.38	54.49	54.55	54.30	54.49	54.58
Output-only (refit)	56.46	56.46	54.88	54.99	54.63	54.78	54.48	54.62
No injection (refit)	55.19	55.19	54.84	54.92	54.66	54.74	54.55	54.70
No history (refit)	56.35	56.35	54.73	54.91	54.61	54.72	54.58	54.69
Gated Arrival	56.49	56.49	54.86	55.07	54.68	54.80	54.56	54.72

(c) OPPORTUNITY

Method	$d = 0$		$d = 30$		$d = 120$		$d = 300$	
	Fixed	LN	Fixed	LN	Fixed	LN	Fixed	LN
Base	25.18	25.18	25.18	25.18	25.18	25.18	25.18	25.18
Source filter	10.89	10.89	10.89	10.89	10.89	10.89	10.89	10.89
Raw Arrival + fallback	25.18	25.18	25.18	25.18	25.18	25.18	25.18	25.18
Half mixture + fallback	25.18	25.18	25.18	25.18	25.18	25.18	25.18	25.18
Fitted constant	25.18	25.18	25.18	25.18	25.18	25.18	25.18	25.18
Age/persistence	25.18	25.18	25.18	25.18	25.18	25.18	25.18	25.18
Output-only (refit)	25.18	25.18	25.18	25.18	25.18	25.18	25.18	25.18
No injection (refit)	25.18	25.18	25.18	25.18	25.18	25.18	25.18	25.18
No history (refit)	25.18	25.18	25.18	25.18	25.18	25.18	25.18	25.18
Gated Arrival	25.18	25.18	25.18	25.18	25.18	25.18	25.18	25.18

Table 13: **Matched comparisons across delays: PAMAP2 and UCI.** Macro-F1 (%) under the same averaging and fallback protocol as Table 12; UCI delays use released-window events.

(a) PAMAP2

Method	$d = 0$		$d = 30$		$d = 120$		$d = 300$	
	Fixed	LN	Fixed	LN	Fixed	LN	Fixed	LN
Base	63.79	63.79	63.79	63.79	63.79	63.79	63.79	63.79
Source filter	65.80	65.80	65.80	65.80	65.80	65.80	65.80	65.80
Raw Arrival + fallback	66.62	66.62	66.34	66.31	65.70	65.69	64.77	65.05
Half mixture + fallback	65.99	65.99	65.90	65.85	65.61	65.60	65.26	65.29
Fitted constant	65.99	65.99	65.90	65.85	65.61	65.60	65.26	65.29
Age/persistence	65.87	65.87	65.23	65.31	64.96	64.95	64.61	64.64
Output-only (refit)	66.50	66.50	66.13	66.09	65.52	65.61	65.02	65.21
No injection (refit)	65.78	65.78	65.85	65.79	65.78	65.78	65.63	65.60
No history (refit)	66.31	66.31	65.90	65.88	65.63	65.69	65.41	65.38
Gated Arrival	66.35	66.35	65.96	65.92	65.70	65.77	65.47	65.45

(b) UCI

Method	$d = 0$		$d = 30$		$d = 120$		$d = 300$	
	Fixed	LN	Fixed	LN	Fixed	LN	Fixed	LN
Base	91.42	91.42	91.42	91.42	91.42	91.42	91.42	91.42
Source filter	92.44	92.44	92.44	92.44	92.44	92.44	92.44	92.44
Raw Arrival + fallback	93.89	93.89	85.05	87.23	84.68	86.07	90.19	88.39
Half mixture + fallback	94.06	94.06	91.69	92.75	92.04	92.39	91.44	92.43
Fitted constant	94.06	94.06	91.69	92.75	92.04	92.39	91.44	92.43
Age/persistence	94.13	94.13	91.77	91.76	92.26	92.30	91.46	92.43
Output-only (refit)	94.27	94.27	86.35	90.61	87.93	90.23	91.30	91.16
No injection (refit)	92.76	92.76	92.59	92.61	92.31	92.58	91.45	92.54
No history (refit)	93.78	93.78	91.64	92.22	92.01	92.15	91.43	92.45
Gated Arrival	93.91	93.91	91.86	92.41	92.16	92.34	91.45	92.46

Table 14: **Paired aggregate F1 differences across delays.** Gated Arrival minus each comparator, in percentage points; brackets give 95% participant-bootstrap intervals. Results are aggregated within person, then averaged over participants within dataset and over the five datasets equally. Each of 10,000 bootstrap draws resamples people within each dataset and retains their complete method/service vector. The Fixed 0 row also represents LN0, whose predictions are identical.

(a) Direct injection and output gating

Service	Raw Arrival	Half mixture	Output-only refit
Fixed 0	-0.25 [-0.53, +0.01]	+0.37 [+0.14, +0.63]	-0.28 [-0.45, -0.15]
Fixed 30	+1.65 [+1.27, +1.99]	+0.39 [+0.17, +0.62]	+0.88 [+0.66, +1.07]
LN 30	+1.33 [+0.94, +1.70]	+0.33 [+0.12, +0.55]	+0.16 [-0.03, +0.34]
Fixed 120	+1.58 [+1.20, +1.95]	+0.19 [-0.06, +0.40]	+0.79 [+0.51, +1.05]
LN 120	+1.60 [+1.23, +1.97]	+0.34 [+0.11, +0.55]	+0.34 [+0.14, +0.52]
Fixed 300	+0.50 [+0.18, +0.82]	+0.21 [+0.02, +0.38]	+0.28 [+0.08, +0.47]
LN 300	+1.16 [+0.78, +1.49]	+0.24 [+0.02, +0.41]	+0.31 [+0.10, +0.48]

(b) Constant mixing and refitted component removal

Service	Constant mixture	No-injection refit	No-history refit
Fixed 0	+1.72 [+1.40, +2.06]	+2.73 [+2.29, +3.18]	-0.00 [-0.09, +0.08]
Fixed 30	+1.63 [+1.35, +1.92]	+1.70 [+1.37, +2.02]	+0.02 [-0.06, +0.12]
LN 30	+1.59 [+1.30, +1.89]	+1.94 [+1.59, +2.28]	+0.02 [-0.05, +0.11]
Fixed 120	+1.04 [+0.84, +1.25]	+1.05 [+0.80, +1.31]	+0.02 [-0.04, +0.08]
LN 120	+1.33 [+1.10, +1.57]	+1.35 [+1.11, +1.58]	+0.01 [-0.05, +0.08]
Fixed 300	+0.70 [+0.51, +0.89]	+0.49 [+0.29, +0.70]	-0.00 [-0.05, +0.05]
LN 300	+0.98 [+0.80, +1.18]	+0.86 [+0.65, +1.07]	-0.02 [-0.07, +0.03]

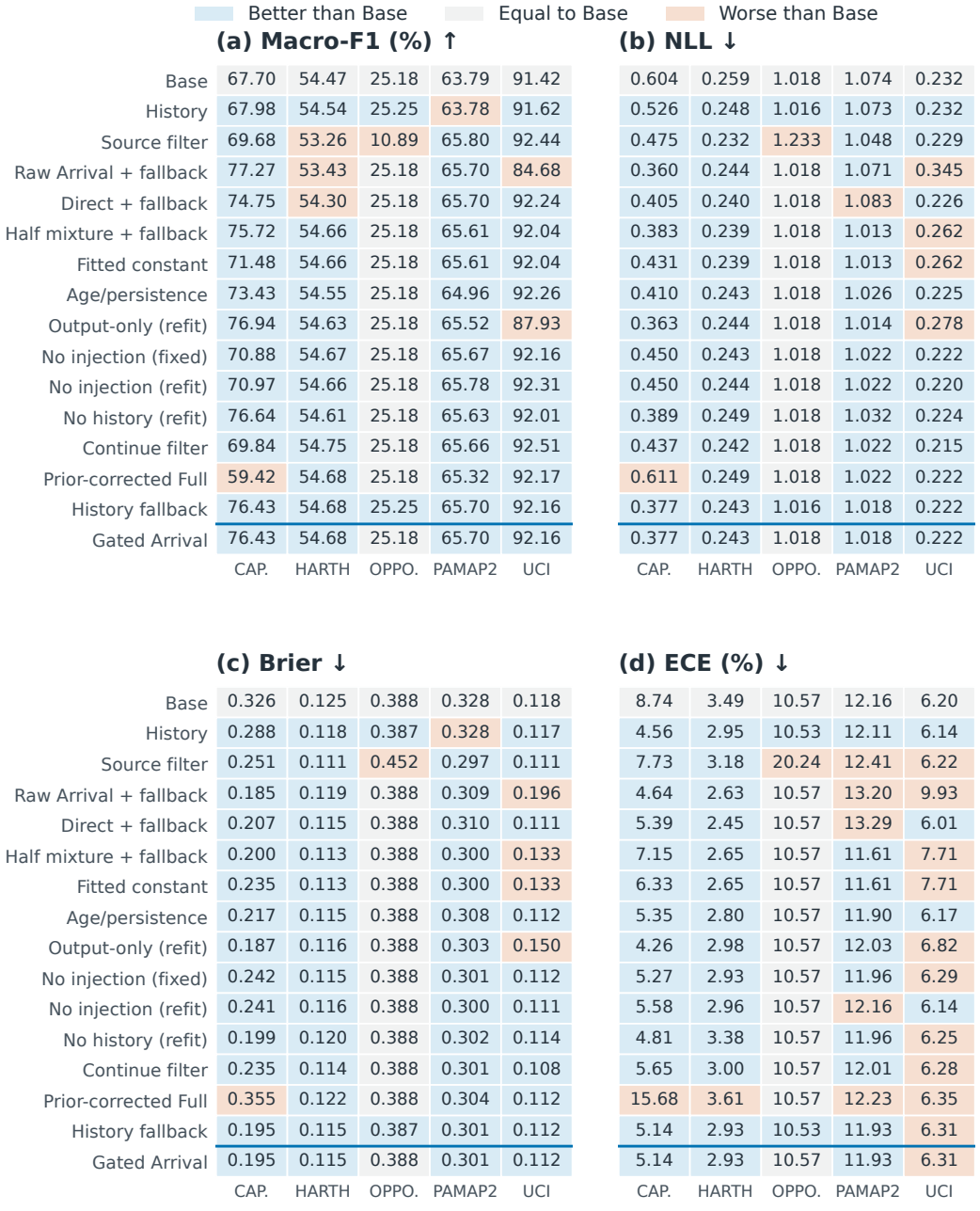
Table 15: **Comparison of learned gates and simpler rules.** (a) NLL plus Brier ↓, first averaging participants within each dataset and then weighting the five datasets equally; bold marks the lowest displayed value. All six simpler rules use the full model's fallback, including Raw Arrival and Direct. (b) Macro-F1 (%) at $d = 120$ against the empirically strongest simple rule of that set, using equal participant weights. Participant intervals condition on that selected rule; they do not account for comparator selection or test equivalence. OPPORTUNITY is exactly tied because its compatibility fallback fires in all 12 fold/checkpoint fits (Section 4.4). Blue: the full model.

(a) Aggregate proper score by delay

Method	d=0	d=30	d=120	d=300	LN 120
Gated Arrival	0.7490	**0.7830**	**0.7977**	**0.8142**	**0.7912**
Raw Arrival	0.7509	0.8257	0.8471	0.8373	0.8337
Half mixture	0.7572	0.7949	0.8095	0.8175	0.7974
Fitted constant	0.7777	0.8143	0.8262	0.8311	0.8150
Age/persistence	0.7576	0.8027	0.8125	0.8266	0.8062
Output-only	**0.7395**	0.7967	0.8119	0.8181	0.7958
Direct	0.7551	0.7948	0.8205	0.8373	0.8110

(b) Per-dataset recognition versus the strongest simple rule

Dataset	Best simple rule	Full	Rule	ΔF1 [95% CI]	Reading
CAPTURE-24	Raw Arrival	76.43	77.27	-0.84 [-2.01, 0.12]	CI includes 0
HARTH	Half mixture	54.68	54.66	0.02 [-0.19, 0.30]	CI includes 0
OPPORTUNITY	Raw Arrival	25.18	25.18	0.00 [0.00, 0.00]	CI includes 0
PAMAP2	Direct	65.70	65.70	0.00 [-0.24, 0.27]	CI includes 0
UCI HAR	Age/persistence	92.16	92.26	-0.10 [-0.25, 0.03]	CI includes 0



	(a) Macro-F1 (%) ↑					(b) NLL ↓				
	CAP.	HARTH	OPPO.	PAMAP2	UCI	CAP.	HARTH	OPPO.	PAMAP2	UCI
Base	67.70	54.47	25.18	63.79	91.42	0.604	0.259	1.018	1.074	0.232
History	67.98	54.54	25.25	63.78	91.62	0.526	0.248	1.016	1.073	0.232
Source filter	69.68	53.26	10.89	65.80	92.44	0.475	0.232	1.233	1.048	0.229
Raw Arrival + fallback	77.27	53.43	25.18	65.70	84.68	0.360	0.244	1.018	1.071	0.345
Direct + fallback	74.75	54.30	25.18	65.70	92.24	0.405	0.240	1.018	1.083	0.226
Half mixture + fallback	75.72	54.66	25.18	65.61	92.04	0.383	0.239	1.018	1.013	0.262
Fitted constant	71.48	54.66	25.18	65.61	92.04	0.431	0.239	1.018	1.013	0.262
Age/persistence	73.43	54.55	25.18	64.96	92.26	0.410	0.243	1.018	1.026	0.225
Output-only (refit)	76.94	54.63	25.18	65.52	87.93	0.363	0.244	1.018	1.014	0.278
No injection (fixed)	70.88	54.67	25.18	65.67	92.16	0.450	0.243	1.018	1.022	0.222
No injection (refit)	70.97	54.66	25.18	65.78	92.31	0.450	0.244	1.018	1.022	0.220
No history (refit)	76.64	54.61	25.18	65.63	92.01	0.389	0.249	1.018	1.032	0.224
Continue filter	69.84	54.75	25.18	65.66	92.51	0.437	0.242	1.018	1.022	0.215
Prior-corrected Full	59.42	54.68	25.18	65.32	92.17	0.611	0.249	1.018	1.022	0.222
History fallback	76.43	54.68	25.25	65.70	92.16	0.377	0.243	1.016	1.018	0.222
Gated Arrival	76.43	54.68	25.18	65.70	92.16	0.377	0.243	1.018	1.018	0.222

	(c) Brier ↓					(d) ECE (%) ↓				
	CAP.	HARTH	OPPO.	PAMAP2	UCI	CAP.	HARTH	OPPO.	PAMAP2	UCI
Base	0.326	0.125	0.388	0.328	0.118	8.74	3.49	10.57	12.16	6.20
History	0.288	0.118	0.387	0.328	0.117	4.56	2.95	10.53	12.11	6.14
Source filter	0.251	0.111	0.452	0.297	0.111	7.73	3.18	20.24	12.41	6.22
Raw Arrival + fallback	0.185	0.119	0.388	0.309	0.196	4.64	2.63	10.57	13.20	9.93
Direct + fallback	0.207	0.115	0.388	0.310	0.111	5.39	2.45	10.57	13.29	6.01
Half mixture + fallback	0.200	0.113	0.388	0.300	0.133	7.15	2.65	10.57	11.61	7.71
Fitted constant	0.235	0.113	0.388	0.300	0.133	6.33	2.65	10.57	11.61	7.71
Age/persistence	0.217	0.115	0.388	0.308	0.112	5.35	2.80	10.57	11.90	6.17
Output-only (refit)	0.187	0.116	0.388	0.303	0.150	4.26	2.98	10.57	12.03	6.82
No injection (fixed)	0.242	0.115	0.388	0.301	0.112	5.27	2.93	10.57	11.96	6.29
No injection (refit)	0.241	0.116	0.388	0.300	0.111	5.58	2.96	10.57	12.16	6.14
No history (refit)	0.199	0.120	0.388	0.302	0.114	4.81	3.38	10.57	11.96	6.25
Continue filter	0.235	0.114	0.388	0.301	0.108	5.65	3.00	10.57	12.01	6.28
Prior-corrected Full	0.355	0.122	0.388	0.304	0.112	15.68	3.61	10.57	12.23	6.35
History fallback	0.195	0.115	0.387	0.301	0.112	5.14	2.93	10.53	11.93	6.31
Gated Arrival	0.195	0.115	0.388	0.301	0.112	5.14	2.93	10.57	11.93	6.31

Figure 10: **Matched comparisons grouped by metric.** Absolute scores at $d = 120$, aggregated within participant before averaging over people. All four panels share the same method and dataset order. Backgrounds indicate better/equal/worse than Base using unrounded values; their saturation does not encode magnitude or significance. F1 and ECE are percentages. Fallback-matched mixtures, standalone History and Source filter retain their distinct definitions.

Why component value changes with delay. Table 3 separates two uses of the same returned label across fixed delays. The full fixed/LN comparison is reproducible from the participant records: numerical supplementary `evidence/matched_controls/component_effects.csv` reports all 160 F1/NLL mean differences, participant standard deviations and pointwise 95% bootstrap intervals. Repeated evaluations are averaged within each participant before pairing, and the bootstrap preserves each person's complete service/component/metric vector. On UCI, removing injection changes F1 by $+0.73$ $[0.48, 0.95]$ at fixed delay 30, $+0.15$ $[-0.09, 0.35]$ at 120 and approximately zero $[-0.03, 0.01]$ at 300.

Historical validity and present relevance are different. Historical learning uses (p_o, y_o), so an activity transition does not invalidate its training target. Its effect on current predictions still depends on whether the learned error correction transfers, how many labels have arrived, and when segment resets occur. Across fixed delays, CAPTURE-24's NLL penalty for removing history increases, whereas PAMAP2's decreases. In contrast, injecting y_o into the current state additionally relies on its present relevance. Equation 1 compares label–activity agreement with the accuracy of the displaced prediction. Table 22 shows why HARTH's high 88.21% agreement can still make raw replacement harmful: Base accuracy on the same release decisions is 94.14%. CAPTURE-24 has the opposite ordering; UCI's agreement is only 0.41% in its event-clock condition. These measurements describe Raw Arrival's immediate accuracy change at the same release events at $d = 120$.

Injection weight and later filtering shape the result. Equation 2 explains why correction direction and weight both matter. Injected state also affects predictions between returns, linking the delay comparison to the harmful flips and nested-budget intervention in Section 4.3. Additional returns can improve historical fitting while also resetting the state to an activity that has already ended. Gates reduce the label's weight in these resets. Gate fitting uses NLL plus Brier, whereas Macro-F1 depends on class decisions and ECE on confidence–accuracy agreement. After component removal, refitting allows the remaining path to recover performance. No injection still resets to p_t, and shared fallback makes OPPORTUNITY exactly zero regardless of delay.

Delay changes which activity a label meets on arrival, while the current prediction and gate weights determine its effect. An activity can also recur, making an older label relevant again (Appendix L). Fixed and lognormal services can therefore produce different effects at the same mean delay. The common-origin/window comparison in Table 17 shows that delay-related losses remain when the final returned labels and scored windows are matched.

Aggregate probability quality. For equal-dataset mean NLL plus Brier, the full model leads the post-hoc six-rule subset in Table 15 by 0.003–0.012 at fixed delays 30, 120 and 300 and lognormal mean 120. Output-only refitting leads by 0.0095 at zero delay, and retaining history under fallback gives aggregate loss 0.0005–0.0008 lower through OPPORTUNITY. Table 15 also reports per-dataset comparisons with the selected simple rules.

Label values and event resets. At fixed $d = 120$, CAPTURE-24 loses 5.46 F1 points when injection is removed and the output gate is refitted. HARTH changes by -0.02 points, whereas PAMAP2 and UCI gain 0.08 and 0.15 points under that removal. Source-filter F1 also exceeds full-model F1 on PAMAP2 and UCI. These comparisons retain historical label supervision while removing label-based state injection and refitting the output gate. Their changes measure how much performance the reduced model recovers after refitting, rather than the effect of removal alone. The no-injection rule still resets q_t to p_t at selected return events. The "continue filter" row removes that reset while retaining the original output gate.

Historical calibration after refitting. At fixed $d = 120$, refitting without history raises CAPTURE-24 F1 from 76.43 to 76.64 and lowers ECE from 5.14 to 4.81, but increases NLL from 0.377 to 0.389. Refitting changes the tradeoff between calibration and recognition relative to the fixed-gate deletion. Raw Arrival and refitted Output-only gating exceed the full model on all four CAPTURE-24 means at $d = 120$. At $d = 300$, full-model F1 is 73.54 versus 73.10 for Raw Arrival and 72.81 for Output-only gating. NLL is 0.408 versus 0.409 and 0.403, respectively, so the recognition ranking reverses while Output-only gating retains lower NLL.

Figure 11 shows a positive mean CAPTURE-24 F1 difference from the half mixture, with a participant interval spanning zero. Table 16 summarizes the proper-score objective: the full model has

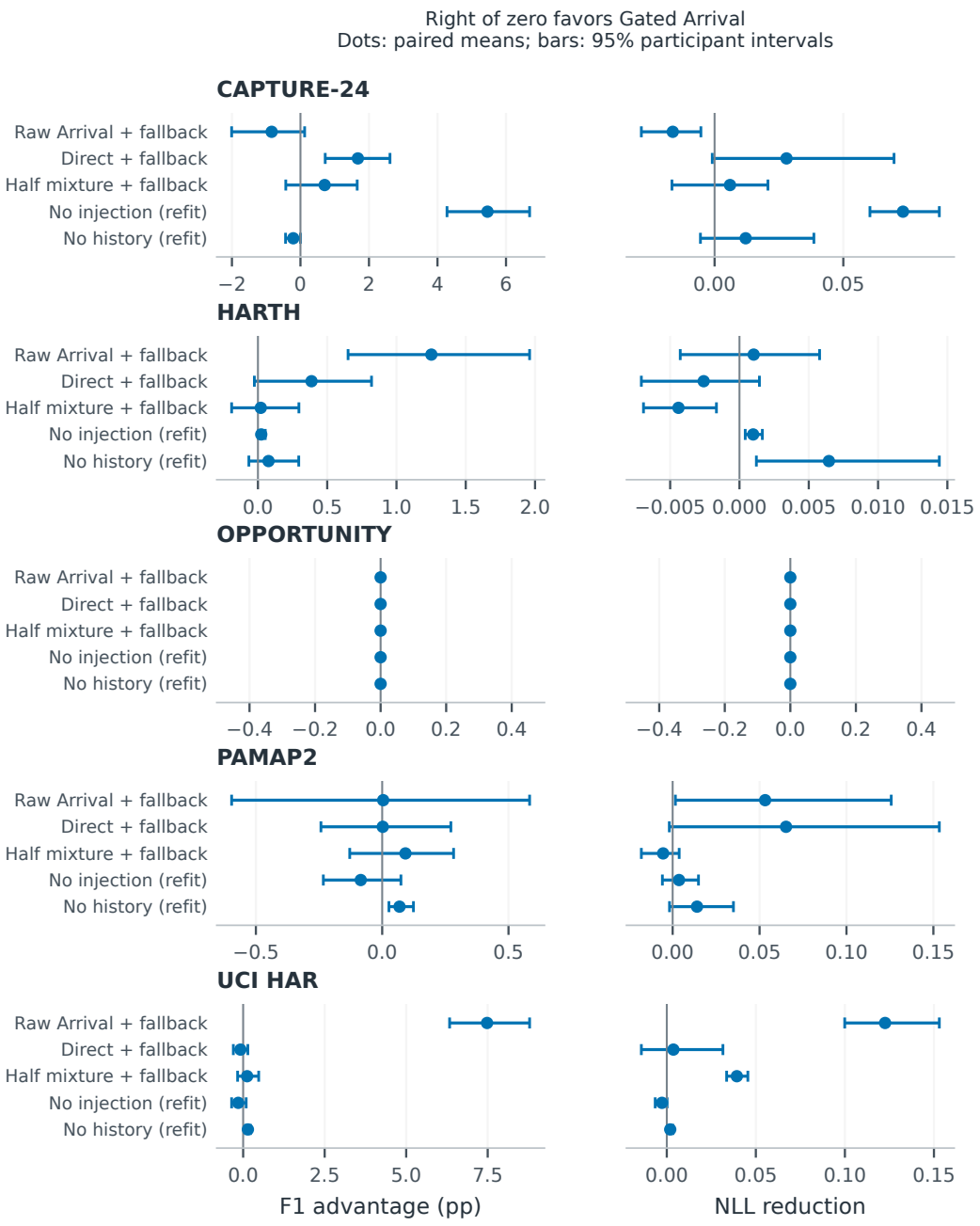


Figure 11: **Paired effects distinguish component reliance from uncertainty.** Gated Arrival versus five matched comparators at $d = 120$. Dots and bars are participant-paired means and 95% bootstrap intervals after within-person aggregation. Positive F1 differences and NLL reductions favor Gated Arrival; NLL signs are reversed for consistent rightward improvement. Each dataset/metric panel has its own labeled scale. These exploratory intervals are not simultaneous.

Table 16: Mean NLL plus Brier ↓, first averaging participants within dataset and then weighting the five datasets equally. All rows use the same compatibility flag; results are aggregated within participant. Delays are seconds except UCI events.

Method	Fixed 0	Fixed 30	Fixed 120	Fixed 300	LN 120
Gated Arrival	0.7490	0.7830	0.7977	0.8142	0.7912
Raw Arrival + fallback	0.7509	0.8257	0.8471	0.8373	0.8337
Half mixture + fallback	0.7572	0.7949	0.8095	0.8175	0.7974
Fitted constant	0.7777	0.8143	0.8262	0.8311	0.8150
Age/persistence	0.7576	0.8027	0.8125	0.8266	0.8062
Output-only (refit)	0.7395	0.7967	0.8119	0.8181	0.7958
No injection (refit)	0.8103	0.8167	0.8218	0.8290	0.8192
No history (refit)	0.7585	0.7938	0.8070	0.8222	0.8008

lower aggregate loss than the displayed simpler rules at positive tested delays and lognormal mean 120, while the Output-only refit has lower loss at zero delay. This is a cross-dataset tradeoff, with different components useful in different conditions.

F.2 Fallback, prior division, and out-of-order history

All 12 OPPORTUNITY fold/checkpoint combinations trigger fallback. Applying that flag to Raw Arrival, Direct, and simple mixtures makes them identical to Base and the full method. History-only fallback reaches 25.248 F1 and 1.0158 NLL versus Base's 25.179 and 1.0181. Its paired F1 advantage is 0.069 points with an interval of $[0.020, 0.151]$, based on four people. Retaining historical calibration during fallback yields a small gain on this panel.

Across the 12 OPPORTUNITY fits, fitting-pool mean NLL differences range from 0.094 to 0.164 and Brier differences from 0.073 to 0.116. Doubling and quadrupling both compatibility thresholds preserve every original flag. Halving them additionally rejects three PAMAP2 source/target streams, reducing its mean F1 from 65.70 to 65.53. The original thresholds are retained for the reported full model.

Removing one participant from the compatibility-check pool, with all models and temporal statistics fixed, preserves flags on the other four datasets. OPPORTUNITY has only two fitting participants per fold: either removal leaves one, disabling the rule's $n_{\text{fit}} \geq 2$ requirement and reducing mean F1 to the unguarded 17.80. Its protection therefore depends on having the specified fitting pool, even though the score thresholds are stable over the tested range.

Replacing $p_t^{1/2}$ by $(p_t/\pi)^{1/2}$ in the full state filter, with both gates and fallback fixed, reduces CAPTURE-24 F1 from 76.43 to 59.42 and raises NLL from 0.377 to 0.611. Changes on HARTH, PAMAP2, and UCI are smaller; OPPORTUNITY still falls back. The effect of source-prior division therefore varies across datasets under the fixed gates and fallback.

Under lognormal mean delay 120, we keep newest-origin state selection unchanged and vary history updates. One variant rejects an origin no newer than any already processed origin; another uses only the newest eligible origin at each release event. Retaining all returned labels improves CAPTURE-24 NLL by 0.00260 over the first variant, with an interval of $[0.00161, 0.00414]$, and improves F1 by 0.027 points. HARTH's NLL improves by 0.00320; effects on PAMAP2 and UCI are smaller, and OPPORTUNITY's fallback masks the change. Retaining out-of-order history primarily improves probability quality, with a smaller change in recognition.

F.3 Where delayed state changes cause errors

We group events by ground truth only after predictions are saved. We measure label–activity agreement and count changes between adjacent valid labels in the same segment; unknown windows are not bridged. Outcomes are measured at release and 1, 5, and 20 decisions later. Event means are pooled within participant across evaluation runs, then averaged equally over participants with eligible events; events are not treated as independent people. A harmful flip changes a correct Base

prediction into an error; a helpful flip changes an incorrect Base prediction into a correct one. Each rate uses all eligible events in the stated stratum as its denominator.

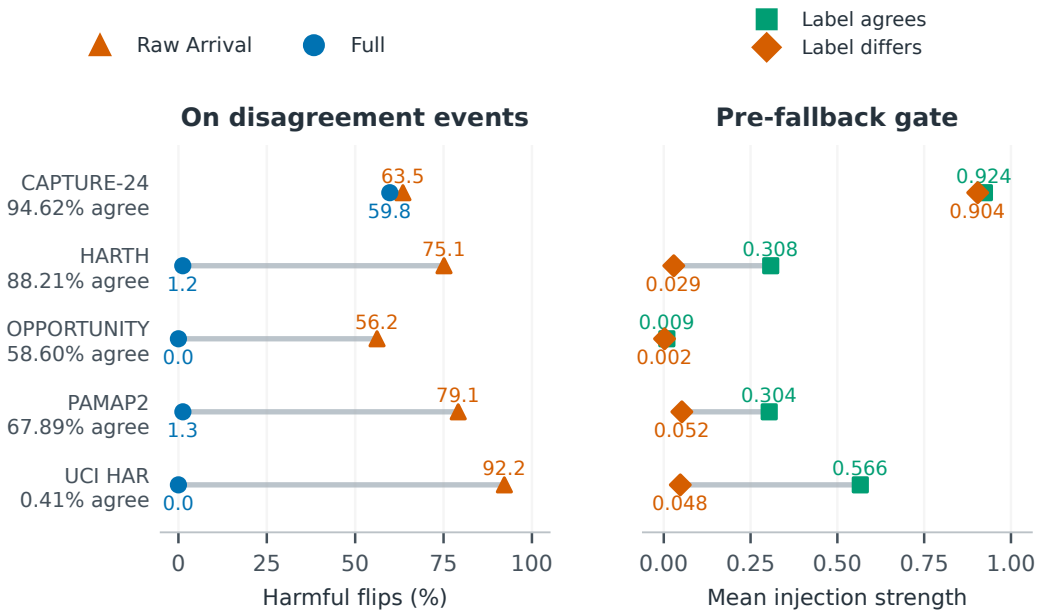


Figure 12: **Harmful corrections and gate response at the same release events.** Left: percentage of disagreement events that change a correct Base prediction into an error. Right: mean injection strength when the returned label agrees or disagrees with the current activity. Each connector joins two means on the same dataset, not a confidence interval. Ground-truth agreement is an evaluator-only stratum. Gate values are pre-fallback; OPPORTUNITY does not apply them. Fixed $d = 120$, equal-participant means of the pooled event measurements.

Figure 12 shows harmful flips alongside injection strengths at the same release events. When the returned label differs from the current activity, the injection gate assigns much less weight to it on HARTH and PAMAP2 than on CAPTURE-24. On CAPTURE-24, mean injection remains 0.904 when the returned label disagrees with the current activity, and 59.79% of these events produce harmful flips. These errors occur when a returned label no longer matches the activity. On UCI, agreement is only 0.41% in this event-clock condition, explaining why hard injection can be particularly damaging on that released sequence.

The nested-budget analysis uses the recorded predictions at all five request rates. Among the additional origins queried at $2\times$, label–activity agreement is 87.64% on HARTH and 2.22% on UCI. At their disagreement events, Raw Arrival's harmful-flip rates are 73.20% and 96.67%; the full method's are 1.44% and zero. These conditional counts locate added errors, while full-stream budget curves measure their combined effect with activation, filtering, and historical updates.

Table 1 summarizes the $2\times$ versus $1\times$ paired effects. The numerical companion retains the full five-budget participant-level absolute scores. The paired budget study preserves the $1\times$ reference predictions and all per-segment random draws. Increasing $1\times$ to $2\times$ reduces Raw Arrival F1 by 1.08 points on HARTH, 2.75 on OPPORTUNITY, and 5.38 on UCI. The respective paired intervals are $[-1.58, -0.67]$, $[-4.52, -1.05]$, and $[-6.18, -4.59]$. Our method's UCI F1 also falls by 0.17 points, with interval $[-0.39, -0.03]$, while remaining above Base. Additional feedback can therefore hurt even our method.

Matching return identities and evaluation exposure. For each participant and request schedule, we intersect the origins that legally return under all compared delays. Each replay receives exactly that set at its own service's release times. All services are scored on the same valid windows after each has activated in the segment. This retrospective comparison matches final returned-label counts

and scored window identities. Intermediate return counts, calibration trajectories, and reset timing still vary across delays.

Table 17: Common-origin and common-window comparison for delays 0, 30, and 120. Entries are within-panel F1 gains over Base. Coverage is the fraction of valid window/run exposures retained; people counts exclude participants with no common legal exposure. Results are aggregated within participant; delays are seconds except UCI events. Raw Arrival is unguarded.

Dataset	People	Coverage (%)	Raw, $d = 0$	Full, $d = 0$	Raw, $d = 120$	Full, $d = 120$
CAPTURE	12	99.4	15.15	13.94	9.60	8.76
HARTH	8	65.0	-0.66	0.89	-3.92	0.15
OPPO.	4	71.1	-13.50	0.00	-13.90	0.00
PAMAP2	8	87.7	3.25	2.99	2.55	2.27
UCI	9	57.4	2.00	2.61	-11.41	1.53

Table 18: Common-origin and common-window comparison for delays 0, 30, 120 and 300 (seconds; UCI: events), aggregated within participant and averaged over people. Raw Arrival is unguarded. Restricted activity support changes absolute Macro-F1, so only within-panel gains are compared. UCI retains very little common exposure at delay 300.

Dataset	People	Coverage (%)	Raw, $d = 0$	Full, $d = 0$	Raw, $d = 300$	Full, $d = 300$
CAPTURE	12	99.1	15.16	13.95	5.46	5.88
HARTH	8	45.0	-0.35	0.22	-1.87	0.01
OPPO.	4	56.2	-13.45	0.00	-13.80	0.00
PAMAP2	8	81.9	2.89	2.64	1.28	1.95
UCI	6	6.1	0.62	0.61	-4.45	0.22

At maximum delay 120, UCI's Raw Arrival gain changes from $+2.00$ to -11.41 points on identical scored windows and returned origins; our method's gain changes from $+2.61$ to $+1.53$. HARTH's corresponding Raw Arrival gains are -0.66 and -3.92. Delay losses therefore persist after matching final return counts and evaluation exposure. Table 18 extends the matched comparison to delay 300. This panel retains about 6% of UCI's valid exposure, motivating the separate maximum-120 panel for all datasets. Neither panel selects gate parameters.

F.4 Missing, noisy, and activity-dependent responses

At fixed delay 120, the service independently drops 25% of otherwise returned labels, or replaces 5% or 10% of their values uniformly with another class. Missing and noise masks are deterministic and shared across source checkpoints and methods for each participant/request schedule. Models and compatibility flags remain fixed, and all streams are retained.

On CAPTURE-24, 10% noise reduces full-model F1 by 1.93 points versus 3.12 for Raw Arrival. Missing returns reduce their F1 by 1.09 and 1.13 points. On UCI, dropping returns improves Raw Arrival by 1.79 points, and label noise can also improve it slightly: a correct old label is usually incorrect for the current activity in this condition. This response occurs because Raw Arrival injects the returned class even after the activity changes. OPPORTUNITY's unchanged output follows mechanically from fallback.

For activity-dependent response times, the simulator assigns delay weight 2 to classes with source exit rate at or above the classwise median and weight 0.5 to the remaining classes. We normalize the weights using the source prior to a mean of 120 and freeze the policy from the designated source statistics for every evaluated checkpoint. Only the simulator uses the true origin class to schedule responses; learners see legally returned labels. Target frequencies and segment censoring can change the observed mean.

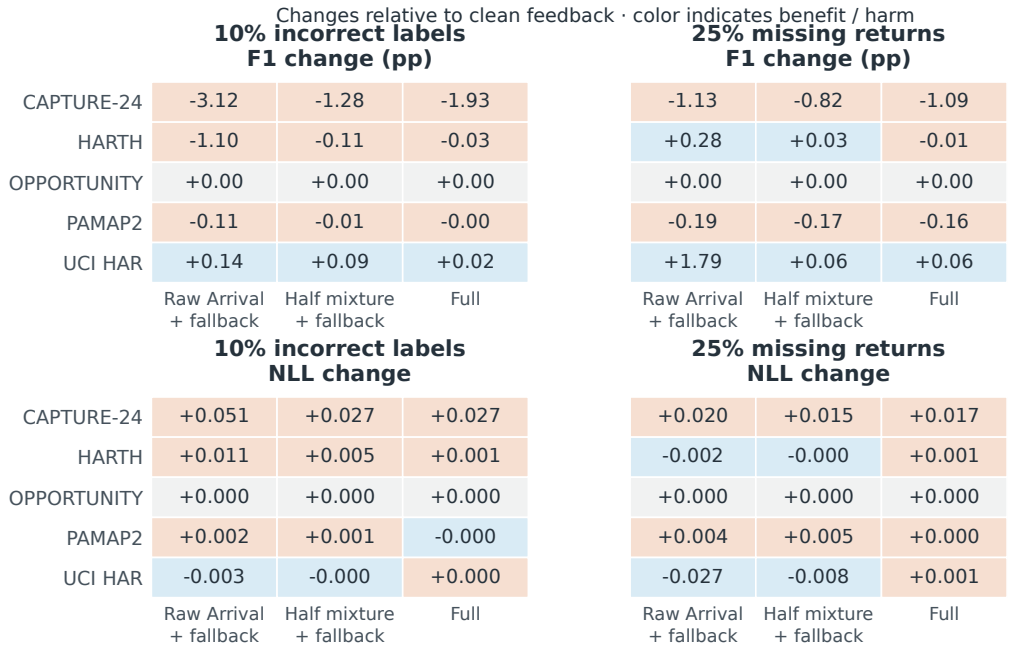


Figure 13: **Missing and noisy labels affect feedback rules differently.** Changes from clean feedback at fixed $d = 120$, equal participant weights. All three rules share the compatibility fallback. Cell labels retain the original F1 and NLL change signs; blue means benefit and orange degradation. The 5% noise condition, all metrics, and remaining rules are retained in the numerical evidence.

Table 19: Synthetic activity-dependent delays. Mean returned age uses seconds except on UCI (events); it is not constrained to 120 on targets. Raw Arrival and the full model share the compatibility fallback. Results are aggregated within participant before averaging over people.

Dataset	Mean age	Raw F1	Full F1	Raw NLL	Full NLL
CAPTURE	153.1	74.24	74.31	0.387	0.389
HARTH	104.9	53.52	54.74	0.240	0.242
OPPO.	113.5	25.18	25.18	1.018	1.018
PAMAP2	120.0	65.64	65.65	1.071	1.019
UCI	77.5	85.48	92.44	0.327	0.219

Table 19 reports F1, NLL, and realized return ages under this activity-dependent service. Our method's F1 remains above Base on the four active datasets and equals Base on OPPORTUNITY. CAPTURE-24's full and Raw Arrival F1 are close (74.31 and 74.24); their NLL ranking favors Raw Arrival. This simulation tests the specified dependence between queried activity and response delay.

The supplement retains per-participant scores, return ledgers, gate measurements, the paired fixed/lognormal delay grid, and the 5% noise condition. `build_review_followup.py` generates the tables from these outputs.

G Context and Source-Statistic Sensitivity

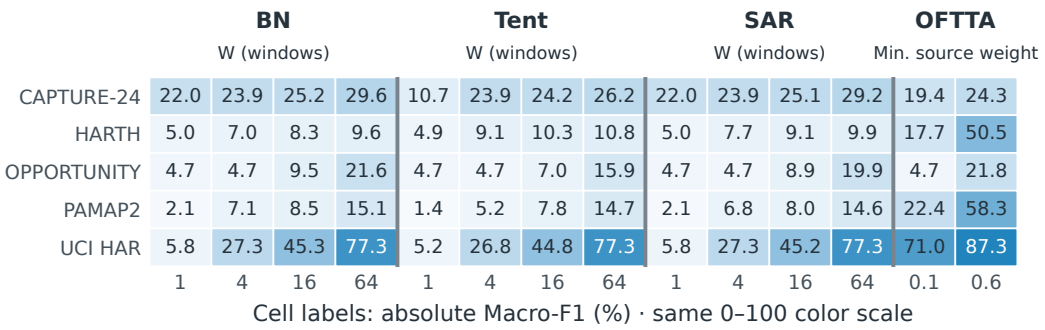


Figure 14: **Context and source-statistic sensitivity in one matrix.** All 70 dataset/configuration cells retain absolute Macro-F1 (%). BN, Tent, and SAR vary the number of past/current windows W; OFTTA keeps singleton input and varies minimum source-statistic weight. These are distinct interventions, separated into column blocks. Colors share a 0–100 scale; source checkpoints and all other selected settings remain fixed.

Figure 14 compares Macro-F1 across context lengths and source-statistic weights. BN, Tent, and SAR concatenate the current sensor window with at most $W - 1$ preceding windows in the same segment. Only the current prediction is scored; shorter buffers are used at segment starts. Increasing W changes normalization context, not sensor-window duration, future access, or the feedback clock. The study covers 70 dataset/configuration groups over the same 42 people, including the four reference configurations.

OFTTA's intervention increases the minimum source-statistic weight from .1 to .6 while preserving the maximum .99 and layer assignment rule. Its input remains one current window. Both studies keep source checkpoints, class mappings, normalization inputs, selected learning rates, support settings, and segment resets fixed. They measure sensitivity to the named variable rather than joint hyperparameter retuning. All seven metrics appear in `CONTEXT_METRICS.csv` and `RESULTS.xlsx`.

H Complete Short-Delay Comparison

The fixed-delay grid contains 13 measured points in $[0, 120]$: $\{0, 1, 2, 3, 5, 10, 20, 30, 40, 60, 80, 90, 120\}$. Figure 15 and Table 20 retain every one of these points for Base and the four feedback rules in the main delay atlas, including OPPORTUNITY. Values are equal-participant means recomputed from all 42 participants. The matched comparison at $d = 0, 30, 120$ in Table 17 uses a different, common-window evaluation population and is reported separately.

At short delays, learned gates improve F1 on some datasets and reduce it on others. On CAPTURE-24, Raw Arrival has higher F1 at all 13 points, by 0.47–1.21 points. On PAMAP2 it leads at all ten points through $d = 60$; Gated Arrival leads at 80, 90 and 120. On UCI, Direct leads our method at $d = 0$ and at all eight measured points from 10 through 120, while our method leads at 1, 2, 3 and 5. On HARTH, History exceeds our method at 10, 20, 30 and 60. OPPORTUNITY remains an exact source fallback under our method (27.36); History is slightly higher at every short delay. These comparisons use unrounded scores for the five displayed methods. The relative performance

Table 20: **Complete short-delay comparison.** Macro-F1 (%) for all five datasets, all 13 short delays, and the four feedback rules plus Base. Equal participant means. Blue identifies Gated Arrival. Unrounded values and all seven metrics for all 43 method configurations are in `SHORT_DELAY_ALL_METHODS.csv`; the 325 cells here are also provided in `SHORT_DELAY_CORE.csv`.

Method / delay d	0	1	2	3	5	10	20	30	40	60	80	90	120
CAPTURE-24 (seconds)													
Base	68.47	68.47	68.47	68.47	68.47	68.47	68.47	68.47	68.47	68.47	68.47	68.47	68.47
History	69.09	69.09	69.09	69.09	69.09	69.07	69.02	68.99	68.94	68.89	68.82	68.80	68.73
Direct	81.92	81.92	81.92	81.92	81.92	80.92	79.92	79.07	78.24	76.86	75.91	75.59	74.92
Raw Arrival	82.39	82.39	82.39	82.39	82.39	81.94	81.47	81.00	80.48	79.45	78.55	78.17	77.15
Gated Arrival	81.34	81.34	81.34	81.34	81.34	80.73	80.34	79.81	79.36	78.45	77.68	77.37	76.68
HARTH (seconds)													
Base	54.03	54.03	54.03	54.03	54.03	54.03	54.03	54.03	54.03	54.03	54.03	54.03	54.03
History	54.66	54.61	54.57	54.55	54.54	54.49	54.49	54.43	54.36	54.39	54.38	54.35	54.27
Direct	56.06	54.14	53.13	52.75	52.41	52.12	51.92	52.16	52.00	52.24	53.24	53.85	54.04
Raw Arrival	56.33	54.82	53.87	53.22	52.49	51.90	50.68	51.11	50.34	50.55	52.33	52.57	53.42
Gated Arrival	56.57	55.51	55.06	54.61	54.88	54.40	54.26	54.16	54.42	54.23	54.54	54.46	54.38
OPPORTUNITY (seconds)													
Base	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36
History	27.39	27.39	27.39	27.39	27.39	27.39	27.39	27.39	27.39	27.39	27.38	27.37	27.37
Direct	17.62	17.37	17.35	17.34	17.37	17.51	17.79	17.84	18.02	18.20	18.25	18.39	18.58
Raw Arrival	17.71	17.42	17.41	17.37	17.32	17.48	17.76	17.74	17.97	18.13	18.25	18.32	18.59
Gated Arrival	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36	27.36
PAMAP2 (seconds)													
Base	64.02	64.02	64.02	64.02	64.02	64.02	64.02	64.02	64.02	64.02	64.02	64.02	64.02
History	64.07	64.07	64.07	64.07	64.06	64.06	64.04	64.04	64.03	64.02	64.00	63.99	63.99
Direct	66.03	66.03	65.91	65.82	65.76	65.61	65.54	65.50	65.50	65.49	65.52	65.50	65.54
Raw Arrival	66.11	66.11	66.05	66.08	66.14	66.24	66.56	66.40	66.16	66.18	65.65	65.56	65.50
Gated Arrival	65.88	65.88	65.81	65.78	65.74	65.72	65.77	66.04	65.70	65.61	65.86	65.70	65.77
UCI HAR (events)													
Base	90.30	90.30	90.30	90.30	90.30	90.30	90.30	90.30	90.30	90.30	90.30	90.30	90.30
History	91.11	91.11	91.11	91.11	91.11	91.01	90.89	90.86	90.86	90.86	90.86	90.86	90.86
Direct	93.49	93.25	93.05	92.85	92.61	92.32	92.08	91.76	91.76	91.84	91.87	91.87	91.81
Raw Arrival	93.31	92.95	92.41	91.91	91.38	90.13	85.96	84.91	82.13	82.77	83.59	84.57	85.68
Gated Arrival	93.49	93.36	93.07	92.90	92.80	92.09	91.23	90.76	90.82	91.46	91.15	91.54	91.72

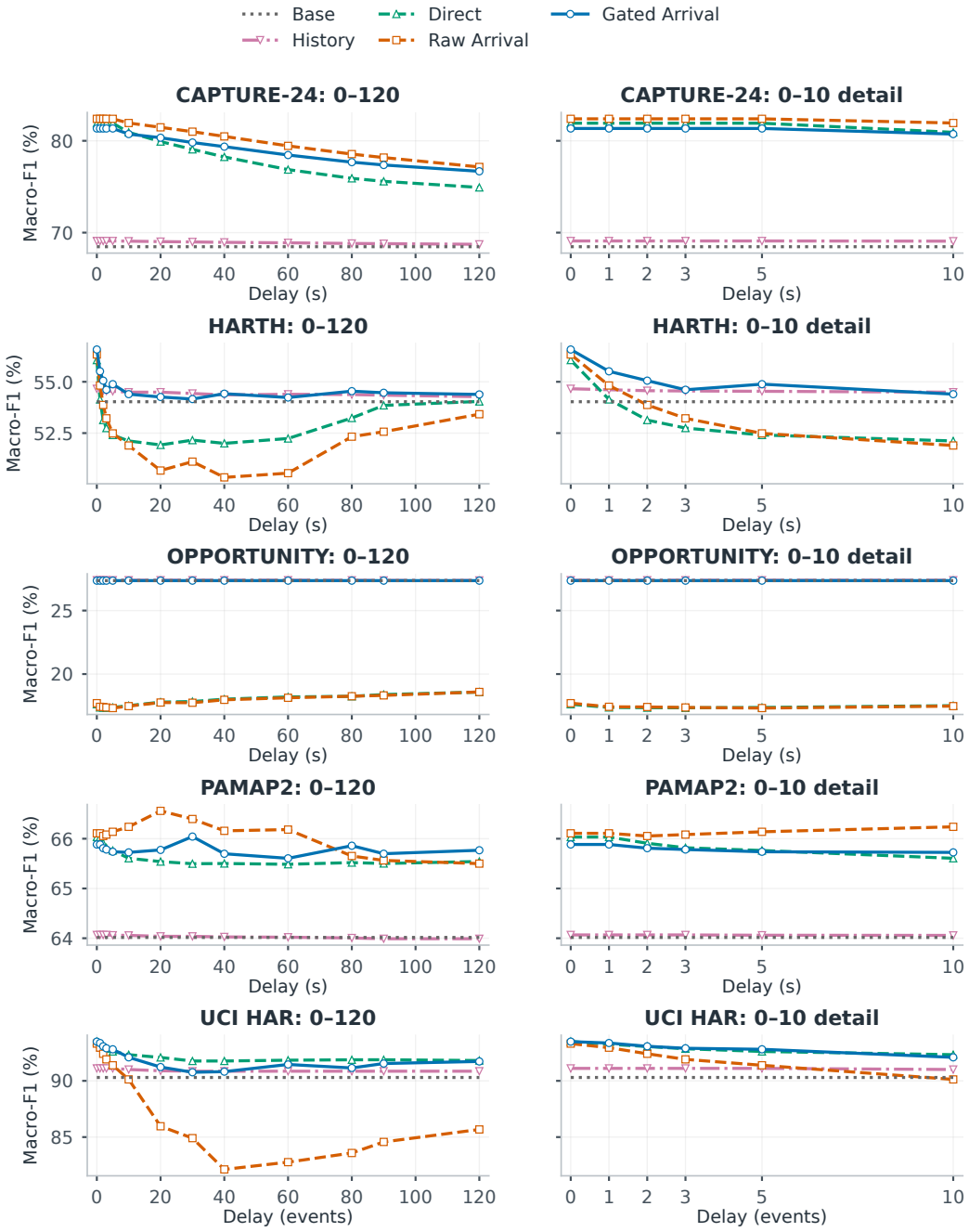


Figure 15: **Short-delay performance trajectories.** Left: all 13 measurements between 0 and 120. Right: the same observations between 0 and 10, enlarged along the horizontal axis. Both axes are linear; panels in each row share a vertical scale. Markers identify measurements, and connecting segments are visual guides, not additional results. Base is dotted; on OPPORTUNITY it coincides exactly with Gated Arrival. Delays use seconds except UCI released-window events. All methods obey causal release, including $d = 0$.

of our method varies with dataset, metric and delay. The long-delay reversal anchors in Table 21 were selected after inspecting the sweep to illustrate that dependence.

The reproducible export in `evidence/short_delay_20260925/` also includes paired F1 differences, the contributing participant values, and a verification record. Recomputing the 325 means from 2,730 participant/method/delay records agrees with the frozen exports to floating-point precision. All 2,795 short-delay rows across the 43 method configurations are retained in the all-method CSV.

I Recognition and Probability Quality Across Delays

Table 21: **Delay can reverse the ranking of feedback rules.** Macro-F1 (%) ↑, with equal participant weights. Base is delay-independent and shown once per dataset. Bold/underlining rank each feedback value against Base and the three rules at that delay (best/second distinct value, including ties). Blue identifies our method independently of rank. Delays use seconds, except UCI events; $d = 0$ still obeys causal release. OPPORTUNITY is omitted here because Gated Arrival equals Base (27.36) at every delay; its complete comparisons remain in the appendix.

Method / source reference	Dataset-specific delay d			
CAPTURE-24 (Base: 68.47)	$d = 0$	$d = 120$	$d = 240$	$d = 300$
Raw Arrival	**82.39**	**77.15**	<u>74.41</u>	<u>72.89</u>
Direct	<u>81.92</u>	74.92	73.21	72.69
Gated Arrival	81.34	<u>76.68</u>	**75.38**	**74.75**
HARTH (Base: 54.03)	$d = 0$	$d = 2$	$d = 40$	$d = 300$
Raw Arrival	<u>56.33</u>	53.87	50.34	<u>54.21</u>
Direct	56.06	53.13	52.00	**54.33**
Gated Arrival	**56.57**	**55.06**	**54.42**	54.20
PAMAP2 (Base: 64.02)	$d = 0$	$d = 60$	$d = 80$	$d = 300$
Raw Arrival	**66.11**	**66.18**	<u>65.65</u>	64.79
Direct	<u>66.03</u>	65.49	65.52	<u>65.38</u>
Gated Arrival	65.88	<u>65.61</u>	**65.86**	**65.48**
UCI HAR (Base: 90.30)	$d = 0$	$d = 10$	$d = 40$	$d = 300$
Raw Arrival	<u>93.31</u>	90.13	82.13	89.20
Direct	**93.49**	**92.32**	**91.76**	<u>90.37</u>
Gated Arrival	**93.49**	<u>92.09</u>	<u>90.82</u>	**90.41**

Figure 16 displays recognition and probability changes over the full 43-service grid of 215 dataset/service comparisons. Gated Arrival has lower NLL and Brier than Base in 171 conditions, identical values in all 43 OPPORTUNITY conditions, and slightly higher values in one UCI condition (uniform mean 300: NLL $+0.000069$, Brier $+0.000059$). Its ECE decreases in 112 conditions, ties in 43, and increases in 60. These counts summarize repeated conditions on the same participants. Recognition and probability metrics show different responses to delay. The underlying values are in `ALL_D_MEANS.csv`; the plotted values are in `visual_revision_20260923/plot_data/` (the probability atlas).

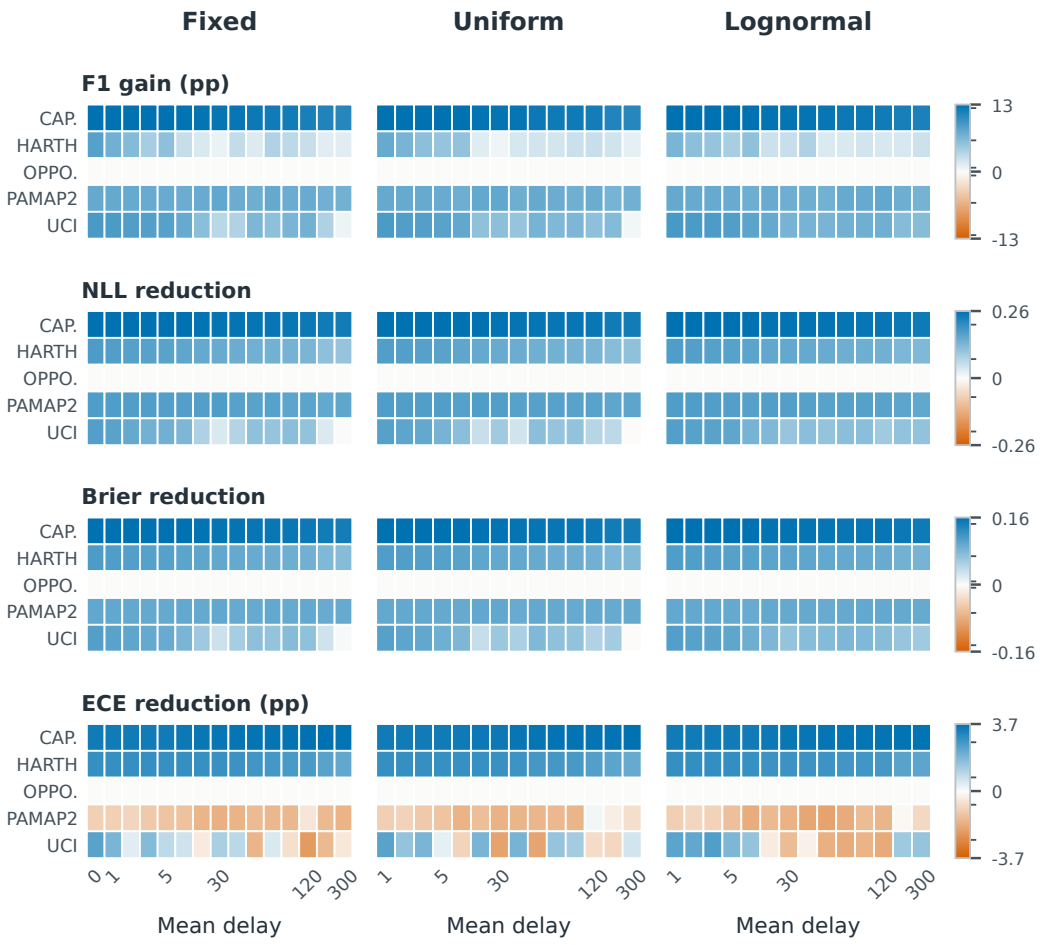


Figure 16: **The complete recognition–probability profile of Gated Arrival.** All 215 dataset/service cells are shown for each metric. Blue means improvement over Base, orange deterioration; white includes equality and very small effects. Each metric has a shared symmetric-log color scale in its original units. Columns enumerate measured delays and are not equally spaced time samples. OPPORTUNITY is exactly unchanged. Delays are in seconds except UCI events. The text quantifies the small UCI uniform-300 loss increases.

J Random Delay and Request Budget

RR, NR, IWMS, and Copy cover all 43 services, together with the probability-space feedback methods. The complete grid is in Appendix K and `ALL_D_MEANS.csv`. Delay-independent rows reuse their single prediction stream. Fixed, uniform, and lognormal services share the aligned layout of Figure 17, showing changes under variable age and out-of-order feedback.

For the budget study, per-segment draws are shared across request-rate multipliers $\{0, 0.25, 0.5, 1, 2\}$, producing nested request sets. All 42 participants complete all five budgets for the 16 probability-space configurations. The $1\times$ predictions, requests, returns, and metrics match the principal condition. At zero requests, every feedback variant equals Base; the always-active source filter remains a separate unlabeled comparator. The full budget trajectories remain in the original editable `figures/budget.svg`; Table 1 reports the participant-paired doubling effect. Budget metrics appear in the budget worksheet in `RESULTS.xlsx`.

K All Fixed and Random Delays

Figure 17 replaces the repeated fixed/uniform/lognormal tables with a common layout. Every one of the 43 measured services is retained for each dataset and displayed feedback rule. Base is the shared zero reference. The original full numeric grid remains in `numeric_supplement.pdf`, and all seven metrics and remaining variants remain in `ALL_D_MEANS.csv`.

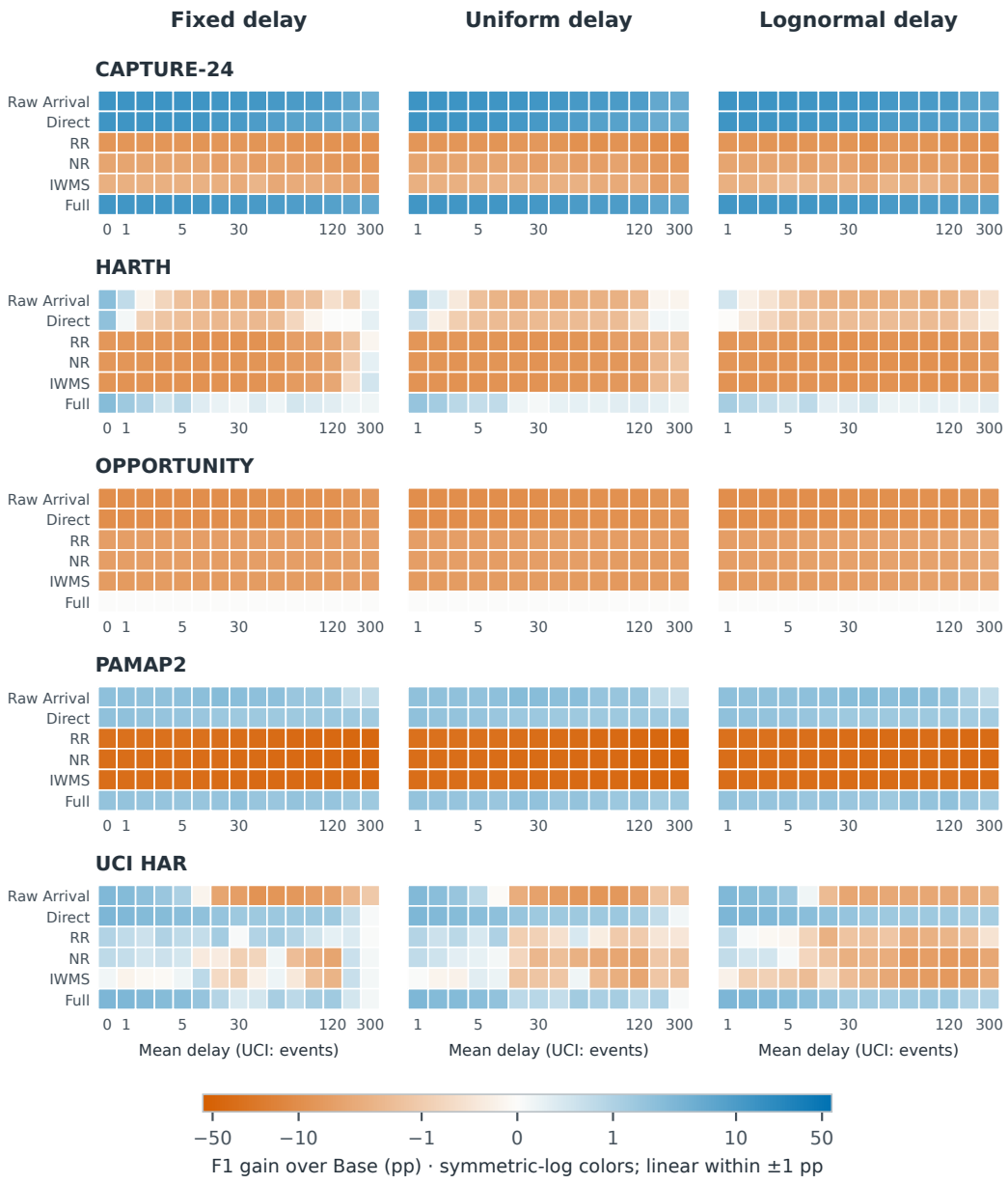


Figure 17: **Feedback rules across fixed, uniform and lognormal delays.** F1 gain over Base for five datasets, six feedback rules and all 43 services (1,290 values). Symmetric-log colors use the same range across every panel and are linear within ±1 pp; no values are clipped. Columns enumerate tested delays, without interpolation or a continuous-time spacing assumption. Delay and mean-delay units are seconds except UCI events. Exact values and all other metrics remain in the numerical companion.

L Local Loss Analysis and Matched Release Events

Exact decomposition. Condition throughout on the same eligible release decisions $\mathcal{E}$. Write $M = \mathbb{1}\{y_o = y_t\}$ and $B = \mathbb{1}\{b_t = y_t\}$. Hard replacement changes accuracy by $M - B$ on each decision, so

$$\mathbb{E}[M - B \mid \mathcal{E}] = \Pr(M = 1, B = 0 \mid \mathcal{E}) - \Pr(M = 0, B = 1 \mid \mathcal{E}) \tag{17}$$

$$= m_{\mathcal{E}} - \alpha_{\mathcal{E}}. \tag{18}$$

The two joint probabilities count helpful and harmful flips and allow arbitrary dependence between agreement and Base correctness. When $M = B$, the accuracy difference is zero.

Conditional Brier risk of a soft correction. Condition on the available information X, which fixes h, c, and $v = c - h$. Expanding the squared loss pointwise yields

$$\|h + \eta v - e_y\|_2^2 - \|h - e_y\|_2^2 = 2\eta(h - e_y)^\top v + \eta^2 \|v\|_2^2.$$

Taking the conditional expectation and substituting $\mathbb{E}[e_{y_t} \mid X] = \mu$ gives Equation 2. For $v \neq 0$, differentiation and restriction to $[0, 1]$ give the conditional minimizer

$$\eta_B^*(X) = \left[\frac{(\mu - h)^\top v}{\|v\|_2^2} \right]_{[0,1]}, \tag{19}$$

where $[a]_{[0,1]} = \min(1, \max(0, a))$. If $v = 0$, every mixing weight produces the same prediction. For $v \neq 0$ and $\eta > 0$, the correction reduces Brier risk exactly when $2(h - \mu)^\top v + \eta \|v\|_2^2 < 0$. A direction that helps at small strength can therefore become harmful at a larger strength.

For example, take two classes with current distribution $\mu = (0.6, 0.4)$, reference $h = (0.8, 0.2)$ and candidate $c = (0.2, 0.8)$. The Brier-risk difference is $-0.48\eta + 0.72\eta^2$: its minimum occurs at $\eta = 1/3$, whereas full replacement increases risk by 0.24.

Relation to the gate-fitting objective. For strictly positive h, c, the conditional expected sum of NLL and Brier is, up to a constant independent of η,

$$\mathcal{L}(\eta \mid X) = -\sum_k \mu_k \log(h_k + \eta v_k) + \|h + \eta v - \mu\|_2^2 + \text{const}. \tag{20}$$

Its first and second derivatives are

$$\mathcal{L}'(\eta \mid X) = -\sum_k \frac{\mu_k v_k}{h_k + \eta v_k} + 2(h + \eta v - \mu)^\top v, \tag{21}$$

$$\mathcal{L}''(\eta \mid X) = \sum_k \frac{\mu_k v_k^2}{(h_k + \eta v_k)^2} + 2\|v\|_2^2. \tag{22}$$

Thus the conditional loss is strictly convex when $v \neq 0$, with a unique minimum on $[0, 1]$. Equation 19 minimizes Brier alone; the combined objective has its own minimizer. In deployment, μ is unknown. Our regularized logistic gate fits this combined probability objective across development decisions using causal features. The calculus explains the local role of output mixing; finite data, restricted gate features and a changed deployment distribution determine how closely the fitted gate approaches that conditional minimum.

Local changes and accumulated state. At a given decision, the loss decomposition fixes both candidates. An injection also changes the filter state from which later candidates are computed. A pulse comparison changes one injection from a common state; a complete-stream comparison repeatedly applies each rule and evaluates the states that rule produces. The two comparisons consequently answer different questions. The budget comparison and refitted models measure cumulative recognition and probability quality; release-event counts locate immediate helpful and harmful changes. Appendix P.2 reports a development example in which local feedback utility and full-stream loss rankings differ.

Causal order and counterfactual scope. Release at t precedes prediction at t; the request was made after the already committed prediction at its earlier origin o. The identity compares a hard replacement and Base at t, not a revision of the prediction at o. Raw Arrival's source/one-hot mixture has the returned label as its argmax at release. The identity isolates this immediate accuracy effect; subsequent filtering and probability quality are evaluated over the stream.

Activity recurrence and label agreement. Activity recurrence can increase agreement at a longer delay. In a stationary, irreducible, aperiodic Markov model, agreement is $m(d) = \sum_k \nu_k [A^d]_{kk}$. For example,

$$A = \begin{pmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{pmatrix}, \qquad \nu = (1/2, 1/2), \qquad m(1) = 0.1, \quad m(2) = 0.82.$$

With constant Base accuracy 0.7, the hard-replacement effect changes from negative to positive between these delays because the activity often returns to its earlier class after two steps. In the evaluated services, segment resets and cancellations also change which labels reach a release event.

Budget scaling needs additional assumptions. For J distinct, fixed release decisions without any effect on subsequent predictions, total additional correct predictions equal $\sum_{j=1}^{J}(m_j - \alpha_j)$ in expectation. Only if those conditional probabilities remain constant does this simplify to $J(m - \alpha)$; accuracy over N windows divides this count by N. Increasing request rate in our implementation changes returns and future state, so linear scaling and sign preservation are not implied. Macro-F1 is also not an additive per-window loss.

Table 22: **Matched release-event comparison at $d = 120$.** All columns use the same selected Raw Arrival release decisions, pooling recorded events within participant and then weighting participants equally. $m_{\mathcal{E}}$: label–activity agreement; $\alpha_{\mathcal{E}}$: Base accuracy at those decisions; $\Delta_{\mathcal{E}}$: hard-replacement accuracy difference, in percentage points. n counts participants with eligible events; one PAMAP2 participant contributes none. The table reconstructs the immediate hard-replacement effect on recorded release events.

Dataset	$m_{\mathcal{E}}$ (%)	$\alpha_{\mathcal{E}}$ (%)	$\Delta_{\mathcal{E}}$ (pp)	n
CAPTURE-24	94.62	78.77	+15.85	12
HARTH	88.21	94.14	-5.93	8
OPPORTUNITY	58.60	77.72	-19.12	4
PAMAP2	67.89	81.61	-13.72	8
UCI HAR	0.41	92.28	-91.87	9

The comparison uses ground truth only for evaluation and stored helpful/harmful flip counts. Same-activity and different-activity strata are weighted by their respective event counts within each participant. The weighted helpful-minus-harmful flip rate recovers $\Delta_{\mathcal{E}}$, and $\alpha_{\mathcal{E}} = m_{\mathcal{E}} - \Delta_{\mathcal{E}}$. Table 22 is an arithmetic reconstruction of the matched events. The separate $2\times$ versus $1\times$ intervention in Section 4.3 measures the complete-stream effect of added requests.

M External Comparisons at $d = 120$

Table 23: **External comparisons: recognition and probability quality.** $d = 120$ seconds (UCI: events), shared checkpoints. (a) Macro-F1 (%) ↑, grouped by label access. (b) Probability scores against each dataset's F1-leading external method, selected after evaluation. Bold: best displayed; underline: second in (a). Our method additionally uses labeled gate-fitting data from other datasets; unlabeled comparators receive neither these nor returned target labels. Blue: our method.

(a) Recognition across five datasets

Method	CAPTURE	HARTH	OPPO.	PAMAP2	UCI
Base	68.47	<u>54.03</u>	<u>27.36</u>	64.02	90.30
Unlabeled adaptation					
TAST	67.55	51.89	**27.83**	64.75	89.46
RoTTA	47.96	37.96	16.43	63.29	**92.28**
OATTA	<u>73.45</u>	53.81	17.55	64.89	90.51
SIGHT	72.71	50.72	11.18	<u>65.49</u>	88.81
Same delayed-label service					
RR	60.11	50.16	21.89	24.62	90.59
NR	61.74	49.83	21.99	24.37	86.12
IWMS	64.58	50.15	21.41	24.21	87.84
Gated Arrival	**76.68**	**54.38**	<u>27.36</u>	**65.77**	<u>91.72</u>

(b) Probability quality against the F1-leading external comparator

Dataset	Method	NLL ↓	Brier ↓	ECE (%) ↓
CAPTURE-24	OATTA	0.538	0.258	7.48
	Gated Arrival	**0.370**	**0.190**	**4.77**
HARTH	OATTA	0.284	0.140	4.44
	Gated Arrival	**0.269**	**0.127**	**3.36**
OPPORTUNITY	TAST	3.649	0.837	37.86
	Gated Arrival	**0.926**	**0.359**	**8.60**
PAMAP2	SIGHT	2.426	0.326	15.37
	Gated Arrival	**0.971**	**0.300**	**9.96**
UCI HAR	RoTTA	**0.219**	**0.105**	**5.43**
	Gated Arrival	0.232	0.114	6.29

N Replay Error Analysis and Online Cost

We rerun all nine PAMAP2 participants for RR, NR, and IWMS at delay 120. Saving float64 log-softmax from the same logits leaves every original float32 prediction exactly unchanged in all 27 reruns. Original probabilities contain 6, 144, and 138 true-class zeros for RR, NR, and IWMS. Their infinite stored-probability NLL includes numerical underflow, but stable mean NLL remains 6.41, 7.25, and 7.16. Recognition loss and overconfident errors persist after this numerical distinction is made.

Figure 18 relates recognition and stable NLL to replay update counts. Each participant receives an average of 11.22 usable labels, but periodic replay performs roughly 803–920 gradient steps. Freezing source BN running statistics, with full-network gradients preserved, raises F1 to 49.86–50.54. Keeping the original eight-decision update slots but requiring new returns since the last update reduces mean updates to 35–39 and raises F1 to 44.43–45.23. These separate interventions

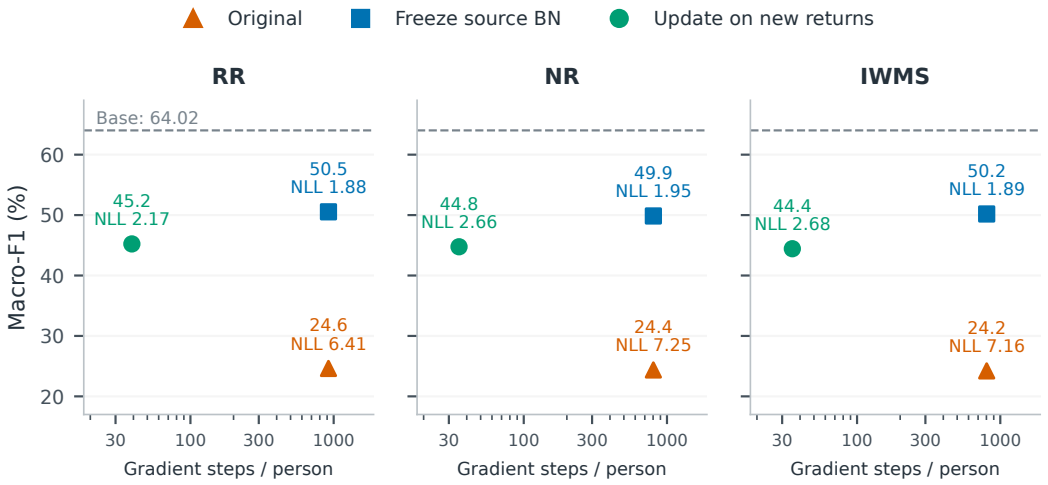


Figure 18: **Replay failures separated into numerical and update-load effects.** PAMAP2 means over nine participants, $d = 120$. Horizontal position is mean gradient steps per person on a log scale; vertical position is Macro-F1. Labels give stable log-softmax NLL, while the original stored-probability NLL remains infinite. Fixing source BN or requiring new returns improves recognition but remains below Base. Each variant keeps the selected learning rate and changes only the stated part of replay.

point to normalization drift and repeated updating from sparse feedback. Both interventions retain a recognition gap to Base's principal-panel F1 of 64.02.

CPU state processing averages 40.7–53.5 μs per active decision, excluding the encoder forward pass and the feedback service. The common measurement protocol and computational-load summary follow in Section O.

O Model Size and Measured Computational Cost

Computation-aware TTA evaluation accounts for adaptation speed under a fixed-rate input stream (Alfarra et al., 2024). Our feedback delay is a separate service variable; these timings do not simulate compute-induced sample skipping.

Online work. Each decision uses one backbone inference, an $O(K^2)$ state step, and $O(K)$ calibration/gating work, plus $O(K)$ per returned label. With B outstanding/retained requests, online storage is $O(K^2 + BK)$. The fixed gates are never trained online.

Table 24: Backbone and additional parameter counts. Gates and calibration belong to Gated Arrival; projection parameters belong to TAST. The last column counts backbone Conv/Linear forward MMAC per window.

Dataset	Backbone	Gates	Calibration	TAST projection	Conv/Linear MMAC
CAPTURE-24	890,912	28	8	20,224	51.50
HARTH	94,220	28	24	6,016	1.76
OPPORTUNITY	117,714	28	36	6,016	1.75
PAMAP2	100,940	28	24	6,016	10.60
UCI HAR	94,406	28	12	6,016	4.56

Table 24 separates backbone and adaptation costs. The 28 gate coefficients remain fixed online; $2K$ calibration scalars update on returns. The transition matrix and prior store $K^2 + K$ statistics. TAST

updates its projection ensemble. Tent, SAR, and RoTTA update normalization affine parameters in this shared backbone; RR, NR, and IWMS update the full network.

Figure 19 reports measured GPU adapter latency and forward-operation subtotals for the compared methods under the following protocol. Measurements use one representative person per dataset: CAPTURE-24 P105, HARTH S012, OPPORTUNITY S1, PAMAP2 101, and UCI 02. GPU timing covers at most 512 consecutive events and excludes 32 warm-up events. Each measurement is repeated three times; these are timing repetitions. All adapters see the same source checkpoint and prefix. The adapter-call timer includes the source forward and GPU-to-CPU output transfer; it excludes current-input transfer, feedback-service processing, and evidence hashing. Measurements were collected on shared NVIDIA RTX 4090 GPUs with PyTorch 2.10.0 and CUDA 12.8, so they describe the observed deployment load.

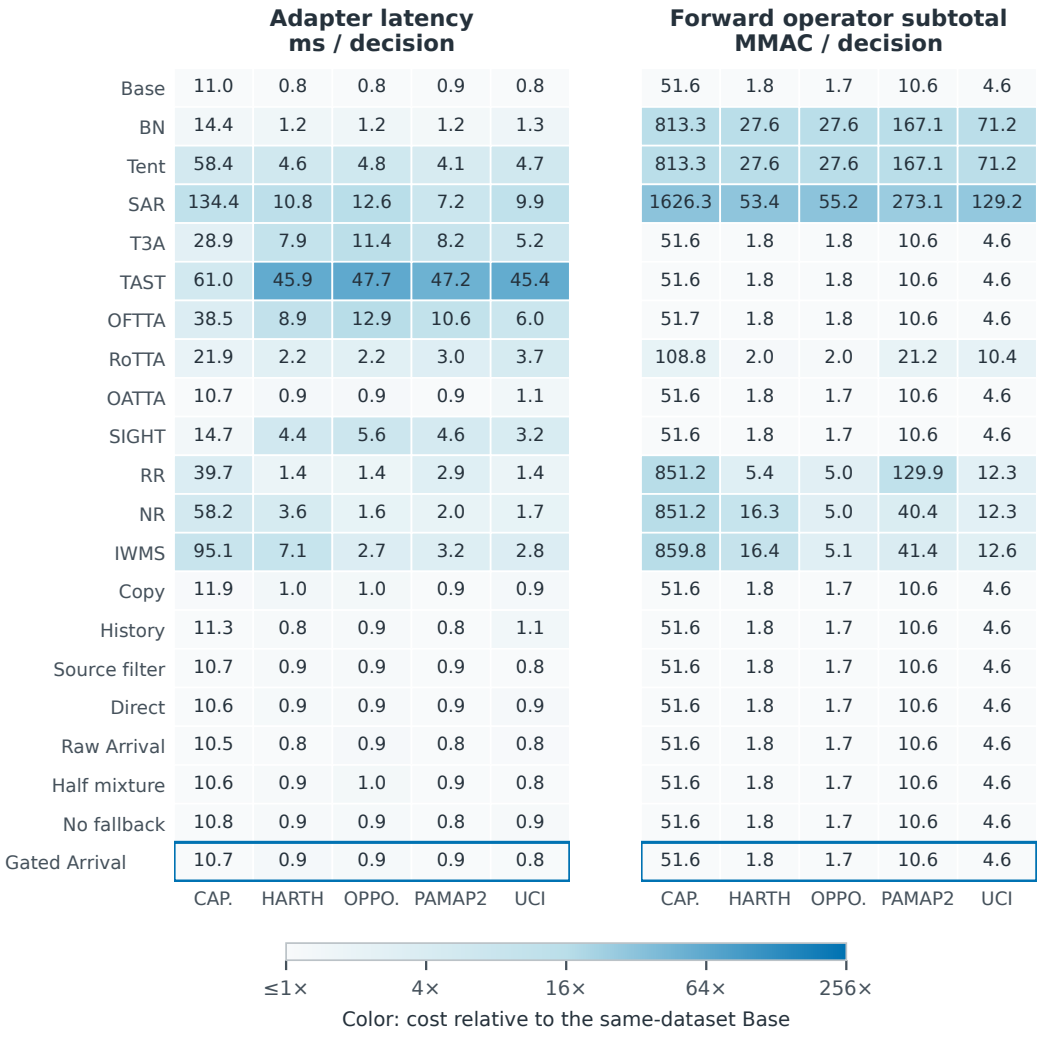


Figure 19: **Computational load alongside absolute measurement values.** Cells print mean GPU adapter latency and the measured forward Conv/Linear/Downsample subtotal per decision; colors encode cost relative to the same-dataset Base, on a log scale, with ratios at or below one sharing the lightest color. One representative participant per dataset, three timing repetitions. Backward passes, normalization, state processing and TAST projection are excluded from this forward subtotal. Exact memory reservations, online neural parameter counts, and projection work remain in the numerical companion.

Table 25: Gated Arrival CPU state time at $d = 120$ (μs/event), excluding encoder, feedback service, and I/O. Return/no-return columns condition on release events. One representative participant per dataset. OPPORTUNITY takes the source-fallback path.

Dataset	All events (μs)	Return (μs)	No return (μs)
CAPTURE-24	46.66	116.79	44.10
HARTH	48.60	132.24	45.42
OPPORTUNITY	1.06	1.06	1.06
PAMAP2	53.48	170.89	52.85
UCI HAR	40.66	139.31	37.40

The CPU benchmark processes full representative streams (Table 25). The active state costs 40.7–53.5 μs per decision on average; return events cost 116.8–170.9 μs. OPPORTUNITY bypasses correction and allocates no adaptation state. The supplemental workbook retains the measured percentiles, operator records, and state-memory summaries. Forward-operation subtotals and wall time answer different questions; no complete forward-plus-backward FLOP total is inferred from them.

P Development Alternatives: Update Direction and Feedback Utility

Two development studies examine alternatives to the fixed feature-based gates: learning an update direction and learning a local feedback utility criterion. Their comparisons isolate the contribution of the new objective or criterion using existing participant streams.

P.1 Learning an update direction

Design. The candidate replaces the fixed injection rule with a low-rank adapter whose update direction is trained to predict *forward*: given the last eight returned feedback items and the current source probability, predict the probability the model should have produced for windows after the latest returned origin. This cross-block objective is intended to reduce the damage caused by using historical labels after an activity change. The capacity-matched *origin-fitting* variant trains the same adapter with cross-entropy on the true returned labels at the support origins. Other comparisons include a *random-direction* basis with matched rank and initialisation scale, a fixed *linear* update, and a *future-pseudo* variant that uses Base pseudo-labels instead of returned labels.

Training uses four development participants per dataset, disjoint from the four historically exposed participants used for scoring; the source backbone is frozen throughout. We evaluate the adapter and request-schedule configurations at delays 0 and 120. Complete streams are scored, giving 84,523 stream windows across both datasets and 48 paired services overall. Six baselines that do not depend on adapter initialization are computed once. The backbone remains frozen, hyperparameter selection excludes scored participants, and the expansion gate was fixed before results were observed.

Result. Table 26 compares the learned update direction with Base, random-direction, origin-fitting, and fixed linear updates across recognition and probability metrics. On CAPTURE-24 the learned target gains 5.45 F1 points over Base $[3.74, 7.02]$ and 4.66 over the future-pseudo variant $[2.24, 8.28]$, supporting a contribution from the returned label content. Its point estimate is 0.61 points below the equal-capacity origin target $[-2.36, 1.13]$ and 6.26 below the existing two-gate model $[-8.86, -3.66]$. On HARTH it gains 0.10 $[-0.25, 0.57]$ over Base and loses to a fixed linear update by 0.39 $[-0.93, -0.01]$. Both datasets fall short of the predefined expansion criteria. On CAPTURE-24, the unmet criteria concern the increment over the origin target and cross-block NLL and accuracy.

Correction after an activity change. At $d = 120$ seconds, the CAPTURE-24 evaluator-defined cross-block windows, including returns to the origin activity after a known intervening transition,

Table 26: **Learned update direction, development-scale panel.** Macro-F1, accuracy and ECE are percentages. Values aggregate fixed delays 0 and 120 seconds (D0/D120) and the evaluated configurations within participant, then average over participants. This panel uses four scored participants per dataset, with its own Base reference and paired comparisons.

Dataset	Update target	F1 ↑	Acc. ↑	NLL ↓	Brier ↓	ECE ↓
CAPTURE-24	Base (no update)	56.12	72.15	0.746	0.404	8.56
CAPTURE-24	Random direction	56.12	72.16	0.746	0.404	8.56
CAPTURE-24	Origin-fitting target	62.18	82.24	0.491	0.252	3.42
CAPTURE-24	Fixed linear update	56.02	74.84	0.638	0.351	5.28
CAPTURE-24	Learned cross-block target	61.57	79.74	0.560	0.299	5.31
HARTH	Base (no update)	51.07	88.27	0.381	0.185	6.45
HARTH	Random direction	51.07	88.28	0.381	0.185	6.44
HARTH	Origin-fitting target	51.24	90.30	0.355	0.166	7.16
HARTH	Fixed linear update	51.56	90.49	0.316	0.145	4.29
HARTH	Learned cross-block target	51.18	88.67	0.368	0.180	6.22

show a tradeoff: the origin target attains 46.12% accuracy and 1.356 NLL, while the learned target reaches 56.57% and 1.175. Base attains 58.11% and 1.072 on the same windows. The new objective reduces the cross-block damage of origin fitting, while its overall F1 is lower. A post-hoc analysis of predictions shows a related tradeoff: the learned direction changes 14.11% of valid window decisions versus 20.06% for the origin target, and its helpful/harmful flip rates are $9.32\%/1.73\%$ versus $13.24\%/3.16\%$—fewer harmful corrections, but also fewer beneficial ones.

The measured correction frequencies show that reducing harmful changes also reduces beneficial changes. This study uses four development participants per dataset, an eight-item training context and rank four, holding these factors fixed.

P.2 Transferring a local feedback utility criterion

A second development study fits a 12-feature ridge criterion to *local feedback utility*: mean NLL-plus-Brier loss without a single injection minus loss with that injection, over up to five predictions starting at release and ending at the segment boundary if earlier. Both branches start from the same state and share subsequent injections; positive utility means that the injection reduces loss. This is a loss reduction, distinct from label–activity agreement and from the absolute NLL-plus-Brier loss reported in Table 16. It uses four previously exposed development participants in each of CAPTURE-24 and HARTH, fitting the criterion in one dataset and evaluating it in the other. The prototype uses a fixed historical/state mixture. We use a frozen backbone and evaluate the request schedules at fixed delays 30, 120 and 300 seconds, giving 72 service replays. Development scores select among always-inject, never-inject, age-based and confidence-based rules; both directions select always-inject. The candidate set and source-only selection function were frozen before replay; selected names were recorded after scoring.

Relative to always-inject, the learned local feedback utility rule increases complete-stream NLL plus Brier by 0.00202 on HARTH and 0.00803 on CAPTURE-24. Always-inject-minus-utility-rule loss differences and exploratory participant intervals are -0.00202 $[-0.01263, 0.00615]$ and -0.00803 $[-0.02432, 0.00045]$, respectively. F1 changes by $+0.119$ and -0.448 points. Both directions fall short of the expansion criterion on these development streams.

The pulse and complete-stream comparisons illustrate the distinction in Appendix L. On HARTH, always-inject has mean five-step local feedback utility -0.01278, yet its complete-stream loss is 0.36352, below never-inject's 0.36637. The pulse changes one decision from a common state; the deployed rules generate different state histories. In addition, five prediction points span 40 seconds on CAPTURE-24 and 2 seconds on HARTH. The transfer therefore uses a common decision horizon with different physical durations. These observations specify the temporal and state conditions of the comparison; they leave the contribution of each factor to the criterion's performance open.

Q EVIDENCE AND REPRODUCTION

The principal-panel supplement reports all seven metrics for 33 method/variant configurations across 43 services, plus ten context-study configurations. Delay-independent methods reuse the same prediction stream across service comparisons. Per-participant scores accompany dataset means, and four context reference configurations are checked against their principal scores. Delay values are measured individually without interpolation.

`INPUT_HASHES.json` binds source inputs, and `NUMERIC_LEDGER.csv` records table lookups. `PLOT_DATA.csv` records the quantitative panels. `RESULTS.xlsx` separates the principal panel, context study, request budgets, costs, and participant-level comparisons into named sheets. `ROLES.json`, `CONTROLLERS.json`, and the data contract record participant roles, fixed gate coefficients, and input identities. The script `build_completed_results.py` rebuilds the principal-panel tables and quantitative figures from these reports. The participant-level panel uses the same participants, source checkpoints, and causal service definition. Matched-comparison tables are generated by `build_review_followup.py`; their numerical inputs and table lookups appear in `evidence/followup`.